Distribution-based Active Learning

by

Xiaofeng Cao

Australian Artificial Intelligence Institute
University of Technology Sydney

Supervisor: Prof. Ivor W. Tsang
A thesis submitted for the degree of

Doctor of Philosophy

Sydney, Australia

2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as part of the collaborative doctoral degree and/or fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This research is supported by the Australian Government Research Training Program under Grant DP180100106 and DP200101328.

Production Note:
Signature: Signature removed prior to publication.
Date: 30-Apr-2021

Acknowledgements

In a sense, Prof. Ivor W. Tsang brought hope for my research career. The early journey of pursuing the Ph.D. degree in Australia was tortuous and arduous. As the Hong Kong film propagated in my childhood, *"professional"* is the most important impression Ivor gave me. He gradually reversed my dilemma on the academic path using his values. He is a humble scholar who maintains earnest and prudent attitudes, leading me to be a righteous person. I can never forget what he often told me: *"what kind of person you want to be, what you will do"*.

My research went smoothly when I was a master's degree candidate in China. This leads me to ambitiously select some challenging research topics at the beginning of my Ph.D. study. Active learning theory, a fundamental topic in PAC learning, started my Ph.D. journal. Also because of this, it is difficult to understand those theorems and proofs at the beginning, since my master research background was totally about applications. However, Ivor never gave up on me. Continuous encouragement and enthusiasm inspired me to explore the deeper theoretical mysteries behind those complicated formulas and theorems. My theoretical ability thus has been greatly improved. This is something I never thought of. No matter how difficult the time, I always insisted on one truth: *"if you cannot please others, you can please yourself "*. This made me gradually understand why Ph.D. was termed philosophy in early western culture. The harvested values will accompany me all my life.

I met my wife in Australia. Lin is my biggest support in this learning journey. Love and marriage can be selfless. We married at 160 Marsden St, BDM Parramatta, Sydney. Thanks to the wedding witness officer Ms. Aila and my friend Mr. Shuai Cheng. Thanks my co-supervisor Prof. Ling Chen for her help in my study. I want to thank my father-in-law and mother-in-law. They provided strong support in my study whether in funding or life. This allows us to establish a small family at 174/11 Potter Street, Waterloo, Sydney, where provides us with very high-quality study and life. I know they do not care about these expenses. However, it greatly relieved my stress and anxiety during my second half of pursuing the PhD degree. My father-in-law is a farsighted person in Hong Kong. I saw the elite spirits of the Cantonese and Chaoshan. His extraordinary achievements inspired me to work harder and go further. My mother-in-law is an elegant lady who requires Lin to unconditionally support my study in daily life. I want to thank my parents, brothers, and sister in law for their consistent support in my study. Specially thanks to my brother, he supported our family and relieved my stress during this period.

"Be humble, be reasonable, and be respected."

—It is the advice from Ivor before I am leaving UTS and going to start my personal academic career. I will spread this motto to the younger students.

Xiaofeng Cao
Sydney, Australia, 2021.

Abstract

Active learning aims to maximize the learning performance of the current hypothesis by drawing as few labels as possible from an input distribution. To build a near-optimal hypothesis, halfspace learning improved the generalization of a perceptron vector over a unit sphere, presenting model guarantees for the reliable (practical) active learning, in which the error disagreement coefficient controls the hypothesis update via pruning the hypothesis class. However, this update process critically depends on the initial hypothesis and the coefficient. Their improper settings may improve the bounds on the label complexity, which estimates the label demands before achieving a desired error for the hypothesis. One question thus arises: how to reduce the label complexity bounds? In a worse situation, estimating updates of hypothesis using error lacks feasible guarantees, if the initial hypothesis is a null (insignificant) hypothesis. Another question also arises: how to control the hypothesis update without errors, when estimating the error disagreement is infeasible? For error disagreement, most of its generalizations regarding to hypothesis update, either make strong distribution assumptions such as halfspace learning, or else they are computationally prohibitive. How to improve the performance of deep active learning based on the theoretical results of active learning of halfspace?

This thesis tries to answer the three questions from shattering, disagreeing, and matching over distributions. With halfspace learning, the first work presents a novel perspective of shattering the input distribution that, guaranteeing from a lower bound on Vapnik-Chervonenkis (VC) dimension, further reduces the label complexity of active learning. When estimating errors is infeasible, the second work proposes a distribution disagreement graph coefficient, which estimates hypothesis from distribution, yielding a tighter bound on typical label complexity. The constructed hyperbolic model, generalizing distribution disagreement by focal representation, shows effective improvements compared to generalization algorithms of error disagreement. On deep learning settings for active learning, the Bayesian neural network shows expressive distribution matching on the massive training parameters, which allows estimating error disagreement can work effectively. We thus integrate the error and distribution disagreements to establish a uniform framework, which matches the geometric core-set expression of the distribution, interacting with a deep learning model.

Contents

1 Introduction 1

1.1 Motivation . . . 1

1.2 Background . . . 3

1.2.1 Active Learning of Halfspace . . . 3

1.2.2 Error Disagreement . . . 5

1.3 Related Work . . . 7

1.3.1 Halfspace Learning . . . 7

1.3.2 Hyperbolic Geometry . . . 8

1.3.3 Deep Active Learning . . . 8

1.4 Contributions . . . 9

1.4.1 Tighter Label Complexity Bounds . . . 9

1.4.2 Distribution Disagreement . . . 10

1.4.3 A Unified Framework . . . 11

1.5 Thesis Organizations . . . 12

1.6 Publications . . . 13

2 Distribution Shattering 15

2.1 Main Theoretical Insights . . . 15

2.1.1 Error Disagreement of Hypothesis-pruning . . . 16

2.1.2 Sparse Hypothesis Class . . . 17

2.1.3 Performance Disagreement . . . 18

2.2 Hypothesis and Distribution . . . 20

2.2.1 Monotonic Property of the Active Query Set . . . 20

2.2.2 Error Rate Change by Shattering Number Density . . . 20

2.2.3 Bottleneck of Hypothesis-pruning . . . 21

2.3 Distribution-shattering for Active Learning . . . 22

2.3.1 Shattering: From Halving to Splitting . . . 22

2.3.2 Halving Number Density for Shattered Distribution . . . 22

2.3.3 Advantages of Shattered Distribution . . . 23

2.3.3.1 Bounds of Error Disagreement in Shattered Distribution 23

2.3.3.2 Fall-back Analysis in Shattered Distribution . . . 24

2.3.3.3 Bounded Noise Analysis of Shattered Distribution . . . 24

2.3.4 Distribution-shattering for Active Learning Tasks . . . 25

2.3.4.1 Active Scoring for Halving . . . 25

2.3.4.2 Splitting by Distribution Density . . . 26

2.3.4.3 Querying by SDAL . . . 26

2.4 Experiments . . . 27

2.4.1 Experimental Setup 27
2.4.2 Effectiveness of Halving 29
2.4.3 Optimal Error of Querying 29
2.4.4 Average Error of Querying 31
2.4.5 Querying with Adversarial Examples 32
2.4.6 Querying with Noisy Labels 35
2.4.7 Calculation Complexity 36
2.5 Discussions 36
2.5.1 Distribution-shattering and Experimental Design 36
2.5.2 Learning Curves of Sequential Sampling in Shattered Space . . . 39
2.6 Summary of This Chapter 39

3 Distribution Disagreeing 41
3.1 Distribution Disagreement Graph Coefficient 41
3.1.1 Graph Coefficient 41
3.1.2 Tightness of Approximation 45
3.1.3 Why Hyperbolic Geometry? 47
3.2 Hyperbolic Focal Representation 50
3.2.1 Squared Lorentzian Distance 50
3.2.2 Geometric Centroids 51
3.2.3 Lorentzian Focal Point Approximation 52
3.2.4 Lorentzian Focal Representation 53
3.2.5 Tree-likeness Splitting 54
3.3 Experiments 56
3.3.1 Experimental Setup 56
3.3.2 Centroid vs. Focal Representations 57
3.3.3 Deep Active Learning with ResNet20 58
3.3.4 Learning Curves of ResNet20 59
3.3.5 Visualization of Lorentzian Focal Representations 60
3.3.6 Batch Performance of Deep Active Learning 62
3.4 Discussions 62
3.4.1 Deep Active Learning with Weak-supervision 62
3.4.2 Limitations of Our Approach 63
3.4.3 Computational Complexities 63
3.5 Summary of This Chapter 64

4 Distribution Matching 65
4.1 Bayesian Active Learning by Disagreements 65
4.1.1 Interpretations 66
4.1.1.1 Model Uncertainty Estimation 67
4.1.1.2 Core-set Construction 67
4.2 Framework 68
4.2.1 Geometric Interpretation of Core-set 68
4.2.2 Initial Acquisitions based on Core-set Construction 69
4.2.3 Model Uncertainty Estimation with Core-set 71
4.2.4 Two-stage GBALD Algorithm 72

4.3 Experiments . . . 72
    4.3.1 Experimental Setup . . . 72
    4.3.2 Uninformative Priors . . . 75
    4.3.3 Improved Informative Acquisitions . . . 75
    4.3.4 Active Acquisitions . . . 76
    4.3.5 Active Acquisitions with Repeated Samples . . . 79
    4.3.6 Active Acquisitions with Noisy Samples . . . 79
    4.3.7 BatchBALD vs. GBALD . . . 80
4.4 Discussions . . . 81
    4.4.1 Acceleration of Accuracy . . . 81
    4.4.2 Hyperparameter Settings . . . 82
    4.4.3 Two-sided t-test . . . 84
4.5 Summary of This Chapter . . . 85

5 Conclusion **86**
5.1 Thesis Summarization . . . 86
5.2 Future Work . . . 86

References **88**

Appendix **97**
Proofs of Chapter 2 . . . 97
Proofs of Chapter 3 . . . 98

List of Figures

1.1 Active learning queries unlabeled data selected by the machine learning model. The querying loops stop until the model achieved a desired generalization performance. 2
1.2 Halfspace learning over a unit sphere with a radius of R, where +, − denote different class labels. The error disagreement-based active learning strategy prunes the hypothesis set (reduce the number of candidate hypotheses, i.e., the diameters across the colored regions) via querying data distributed in the colored pool. 4
1.3 Halfspace learning presents guarantees for the theoretical analysis and generalization performance of distribution-based active learning. Theoretical analysis includes contributions of tighter label complexity bounds of Chapter 1.4.1 and distribution disagreement graph coefficient of Chapter 1.4.2. Generalization performance includes hyperbolic focal representation which generalizes distribution disagreement in hyperbolic geometry (related work in Chapter 1.3.2), and deep active learning framework of Chapter 1.4.3 (related work in Chapter 1.3.3). 7
1.4 A uniform framework consists generalizations of distribution and error disagreements. 11
1.5 The skeleton of the thesis. 12

2.1 Error disagreement of halfspace learning over a unit sphere with a radius of R, where + and − symbols denote the positive and negative class labels, respectively. The error disagreement-based active learning strategy prunes the hypothesis set via updating a subsequent hypothesis with large ϑ; the initial hypothesis has an error of $1/2$, and the subsequent hypothesis has an error of ϑ/π, thereby the error disagreement is $1/2 - \vartheta/\pi$. 16
2.2 Shattering distribution of a unit sphere with a radius of R, where +, − denote the class labels. Distribution-shattering halves the number density of an input distribution w.r.t. $\frac{12}{\pi R^2}$ into a shattered distribution w.r.t. $\frac{6}{\pi R^2}$. Any hypothesis generated from the original distribution is charactered with a lower bound on VC dimension. Thereby, we can find a representation structure that induces a tighter label complexity without estimating the hypothesis. 18

2.3 Number density models the relationship of hypothesis and distribution, i.e. VC dimension increases proportionally as the number density. If number density=$\frac{4}{a^3}$ and the hypothesis is generated from i samples, the VC bound ≤ $\sum_{i=1}^{4}\binom{4}{i} = 16$; given the number density=$\frac{6}{a^3}$, the VC bound ≤ $\sum_{i=1}^{6}\binom{6}{i} = 64$. Note the version space of number density=$\frac{6}{a^3}$ only presents 32 hypotheses, where the red nodes are the newly added ones from the version space of number density=$\frac{4}{a^3}$, and each blue node denotes one feasible hypothesis, respectively. 19
2.4 The assumption of distribution-shattering with a sparse hypothesis class. Each node denotes one realizable hypothesis, and the lengths of the red lines denote the hypothesis disagreement. Hypothesis-pruning updates the initial hypothesis w.r.t. h_0 into h_ϵ with a desired error ϵ in the original hypothesis class (Fig. 2.4(a)). Distribution-shattering optimizes a group of local hypothesis spheres (Fig. 2.4(b)). Shattering by those sphere centers, the original hypotheses are transformed into a sparse hypothesis class (Fig. 2.4(c), thereby finding h_ϵ can achieve a lower label complexity than in its original hypotheses. 21
2.5 Error rate changes of undertaking passive sampling in input and shattered spaces on different datasets. 29
2.6 The error rate performances of the seven active learning approaches on the active learning test. (a)-(d) are four UCI datasets; (e)-(k) are that on the selected sub datasets of *letter*, where the class number of them are 12, 16, 20, and 26. 30
2.7 Illustration of the produced adversarial examples by FGSM with different perturbation parameters, where the marked examples are clean data without feature perturbations. 33
2.8 The performance of error rate on active learning querying with adversarial examples, where the adversarial examples are produced by FSGM with different perturbation parameter settings. 34
2.9 The performance of error rate on active learning querying with noisy labels. 36
2.10 Representative samples of MAED using different λ. 37
2.11 Representative samples of SDAL using different initializations ($r = 0.25$, $\varepsilon = 0.1$). 38
2.12 Error rate changes of undertaking sequential sampling in the shattered space and original input space. 39

3.1 Motivation of distribution disagreement graph coefficient over halfsapce learning. As the decrease of number density, the graphs covering all feasible hypotheses keep consistent structures, where each vertex denotes one hypothesis and each edge denotes the hypothesis disagreement (distance) of the two connected vertices. Learning a halfspace with graph coefficient can completely replace the error disagreement. 42

3.2 Generalization test over MNIST dataset using θ and θ_G. 46

3.3 Euclidean, Gaussian kernelized, Poincaré and Lorentzian centroids on a noisy spherical Gaussian dataset. The minimization or maximization on the centroids are performed with 30 iterations against k-medoids algorithm. 48

3.4 Test accuracies of Euclidean, squared Euclidean, Gaussian kernelized, Poincaré and Lorentzian centroids with varying parameters. 49

3.5 The Lorentzian version space ($\triangledown A'BC$, $\triangledown D'EF$) vs. Euclidean version space ($\triangledown ABC$, $\triangledown DEF$). Lorentzian norm shrinks the volume of Euclidean version space by shifting Euclidean centroids (A,D) into Lorentzian focal points (A′, D′) that are close to the boundary region. As such, sampling from regions $ABA'C$ and $DED'F$ is ineffective, which means any active learning model in the Lorentzian version space has a tighter bound on label complexity. 50

3.6 Geometric centroids of Euclidean, Poincaré, and focal points of squared Lorentzian distances in a given set of points. Lorentzian norm updates the focal points of the embedded half-sphere toward the surface as the parameter $\mathcal{B}$ decreases w.r.t. Eq. (3.14). 53

3.7 Illustration of tree-likeness splitting. The first layer nodes are k global focal points. A binary tree splitting strategy begins from the second layer of the tree. 55

3.8 Test accuracies of Euclidean, squared Euclidean, Gaussian kernelized, Poincaré centroids and our Lorentzian focal points with varying parameters on MNIST. 58

3.9 Test accuracies of training ResNet20 with different active learning outputs on CIFAR-10, CIFAR-100, and SVHN. 58

3.10 Test accuracies of each epoch of training ResNet20 with different active learning outputs on CIFAR-10, CIFAR-100, and SVHN. (Before 80 epochs, the learning rate is 0.001, and afterwards it is 0.1.) 60

3.11 Training losses of each epoch of training ResNet20 with different active learning outputs on CIFAR-10, CIFAR-100, and SVHN. (Before 80 epochs, the learning rate is 0.001, and afterwards it is 0.1.) 60

3.12 Projection of Lorentzian focal points in the first two dimensional feature space of CIFAR-10. 61

3.13 Lorentzian focal vs. batch performance of error entropy and error variation on CIFAR-10. 62

4.1 Illustration of two-stage GBALD framework. BALD has two types of expression: model uncertainty estimation and core-set construction where the deeper the color of the core-set elements, the higher the representation; GBALD integrates them to an uniform framework. Stage ①: core-set construction is with an ellipsoid, not typical sphere, representing the original distribution to indicate the input features of DNN. Stage ②: model uncertainty estimation of DNN then derives the subsequent highly informative and representative samples. 66

4.2 Optimizing BALD with sphere and ellipsoid geodesics. Ellipsoid geodesic rescales the sphere geodesic to prevent the updates of the core-set towards the boundary regions of the sphere where the characteristics of the distribution cannot be captured. Black points denote the feasible updates of the red points. Dash lines denote the geodesics. . . 68
4.3 Acquisitions with uninformative priors from digit ‘0’ and ‘1’. 74
4.4 GBALD outperforms BALD using ranked informative acquisitions which cooperate with representation constraints. 76
4.5 Active acquisitions on MNIST, SVHN, and CIFAR10 datasets. 77
4.6 Active acquisitions on SVHN with 5,000 and 10,000 repeated samples. . 78
4.7 Active noisy acquisitions on SVHN with 5,000 and 10,000 noisy labels. . 80
4.8 Comparisons of BALD, BatchBALD, and GBALD of active acquisitions on MNIST with bath settings. 81
4.9 Accelerations of accuracy of different baselines on MNIST, SVHN, and CIFAR10 datasets. 82
4.10 Accelerations of accuracy of active acquisitions on SVHN with 5,000 and 10,000 repeated samples. 82
4.11 Accelerations of accuracy of active noisy acquisitions on SVHN with 5,000 and 10,000 noisy labels. 83
4.12 A pairwise penalty matrix over active acquisitions on MNIST, SVHN, and CIFAR10. Column-wise values at the bottom of each matrix show the overall performance of the compared baselines (larger value has more significant superior performance). 85

List of Tables

2.1 The statistical results (mean±std in %) of error of different active learning baselines on six real-world datasets 32
2.2 The statistical results (mean±std in %) of active learning with adversarial examples of different algorithms 35
2.3 The statistical results (mean±std in %) of error of experimental design active learning baselines and SDAL on three real-world datasets 37

3.1 Mean±standard deviation of the breakpoints of the generalization test over MNIST using θ and θ_G 46
3.2 Label complexities of the generalization test over MNIST using θ and θ_G (Radius denotes "radius of hypothesis class") 47
3.3 Summary of distance metrics and centroid expressions in the Euclidean space, RKHS, and hyperbolic space. 48
3.4 Mean±standard deviation of the test accuracies on CIFAR-10, CIFAR-100 and SVHN after 80 epochs over 3 runs. 61
3.5 Mean±standard deviation of the training losses on CIFAR-10, CIFAR-100 and SVHN after 80 epochs over 3 runs. 61
3.6 Mean±standard deviation of the test accuracies of different baselines with batch settings on CIFAR-10. 63
3.7 Time computations of selecting 10,000 examples from CIFAR-10 using different baselines. 63

4.1 Mean±std of the test accuracies of the breakpoints of the learning curves on MNIST, SVHN, and CIFAR-10. 77
4.2 Number of acquisitions on MNIST, SVHN and CIFAR10 until 70%, 80%, and 90% accuracies are reached. 78
4.3 Mean±std of active acquisitions on SVHN with 5,000 and 10,000 repeated samples. 78
4.4 Mean±std of active noisy acquisitions on SVHN with 5,000 and 10,000 noises. 80
4.5 Mean±std of BALD, BatchBALD, and GBALD of active acquisitions on MNIST with batch settings. 81
4.6 Relationship of accuracies and sizes of core-set on SVHN. 83

Chapter 1

Introduction

The motivation of this thesis is firstly depicted, following the background and related work. The three research questions throughout the entire structure of the thesis are naturally proposed. Corresponding solutions covering the contributions are highlighted. Organizations describe the framework of the thesis.

1.1 Motivation

Active learning [Settles, 2009], leveraging abundant unlabeled data to improve the generalization performance of a classifier, has been widely adopted in various machine learning tasks, such as regression analysis [Wu, 2018], label-scarce classification [Qiu et al., 2016], dynamic data stream processing [Mohamad et al., 2016], multi-task learning [Harpale and Yang, 2010; Fang et al., 2017], curriculum learning [Matiisen et al., 2019], etc. By employing an active learning algorithm, human experts (annotators) strategically query some "highly informative" data [McCallumzy and Nigamy, 1998] to reduce the error rate of the current learning model in different classification tasks. Figure 1.1 describes this learning loop. However, a natural question that arises is the following: if we continuously increase the size of the active query set, does the error rate of prediction keep decreasing? Furthermore, can we finally find a hypothesis whose error rate is close to what we desire?

Active learning of halfspace [Gonen et al., 2013] also called halfspace learning[*] studied these questions from a set of hypothesis class. Over a unit sphere, active learning explored the theoretical guarantees on error rate and label complexity [†] [Gonen et al., 2013]. To enhance the generalization of the halfspaces learning, Hanneke et al. [2014] developed an error disagreement coefficient that measures how well (or not well) the informative samples satisfy the hypothesis update via a greedy search [Dasgupta et al., 2005]. This is a fundamental metric that can be generalized into various different types of error, such as the best-in-class-error [Cortes et al., 2019c], all-in-class error [Beygelzimer et al., 2009a], etc. A common policy of determining which samples to send for human annotation is to select the data that maximize the error disagreement between the current hypothesis and its subsequent update [Zhang and Chaudhuri, 2014]. Therefore, with active learning, a complete sampling process effectively equates to optimizing the minimum cut of the graph [Blum and Chawla, 2001] that covers all feasible hypotheses

---

[*] Halfspace learning has more broad concepts in learning theory as regression [Kalai et al., 2008], Fourier-transform based algorithms [Linial et al., 1993], kernel SVM and kernel ridge regression [Shalev-Shwartz et al., 2011], etc. In this thesis, it specifically refers to generalizing halfspace with active learning.

[†] The number of labels requested before achieving a desired error.

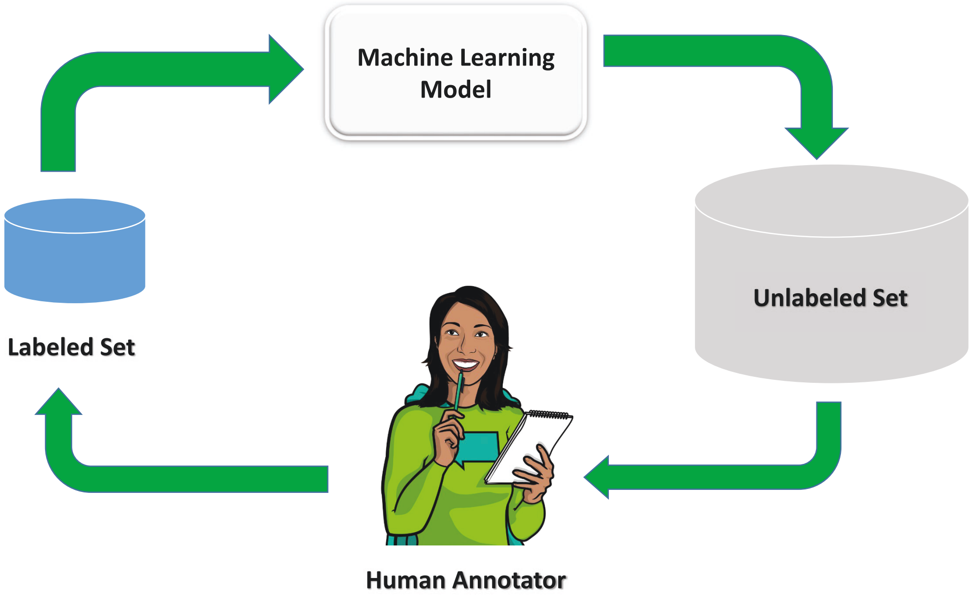


Figure 1.1 : Active learning queries unlabeled data selected by the machine learning model. The querying loops stop until the model achieved a desired generalization performance.

over a version space [Cortes et al., 2019c].

Estimating the disagreement of errors advises the selection of those informative samples, which give the highest rewards to an active learning algorithm, such that these largely update the current learning model. However, only using informative samples produces highly-skewed results [Gao et al., 2020] in the absence of sufficient human supervision, e.g. few labels, improper classifier parameters, etc. The first research question thus arises: how to reduce the bounds of label complexity?

Moreover, many traditional active learning strategies based on error disagreement are not efficient in deep learning settings, because the computational cost of performing a greedy search in an unlabeled data pool, requiring expensive deep network training, is intolerable. Caught between an infeasible computation overhead and an infeasible manual annotation overhead, representative sampling has become a key alternative strategy. Ensuring the model is trained on a set of samples that fairly reflects the distribution of the data minimizes the annotation budget and improves the network architecture without the supervision of any hypothesis class, as core-set selection [Sener and Savarese, 2018a] and sparse subset approximation of batch active learning [Pinsler et al., 2019]. The second research question arises: how to estimate the hypothesis update without the errors?

In deep active learning, Deep neural networks (DNNs) lack the ability of learning from limited (insufficient) labels, which degenerates its generalizations to new tasks. Recently, leveraging the abundance of unlabeled data has become a potential solution to relieve this bottleneck whereby the expert knowledge is involved to annotate those unlabeled data. In such setting, the deep learning researchers introduced the active learning [Gal et al., 2017], which solicit experts' annotations from the informative or representative unlabeled data by maximizing the model uncertainty [Ashukha et al., 2019; Lakshminarayanan et al., 2017] of a learning model. During this active learning process, the learning model tries to achieve a desired accuracy performance using the minimal data labeling. The recent shift of the model uncertainty in many fields shows that the deep Bayesian active learning [Pinsler et al., 2019; Kirsch et al., 2019] derives more and more new scenarios, e.g. Bayesian neural networks [Blundell et al., 2015], Monte-Carlo (MC) dropout [Gal and Ghahramani, 2016], and Bayesian core-set construction [Sener and Savarese, 2018b], etc.

Bayesian active learning [Golovin et al., 2010; Jedoui et al., 2019] presents an expressive probabilistic interpretation on model uncertainty [Gal and Ghahramani, 2016]. Theoretically, for a simple regression model such as linear, logistic, and probit, active learning can derive their closed-forms on updating one sparse subset that maximally reduces the uncertainty on the posteriors over the regression parameters [Pinsler et al., 2019]. However, for a DNN model, optimizing massive training parameters is not easily tractable. Moreover, the similarity or consistency of those acquisitions to the previously acquired samples, brings redundant information to the model and decelerates its training. The third question arises: how to improve the performance of deep active learning based on the theoretical results of active learning of halfspace?

In summarize, this thesis aims to solve the following three research questions:

- How to reduce the typical theoretical bounds of label complexity?
- How to control hypothesis update without errors when estimating the error disagreement is infeasible?
- How to improve deep active learning based on the theoretical results of halfspace learning?

The remainder of this chapter is organized as follows. Section 1.2 introduces the background. Section 1.3 presents the related work. Section 1.4 summarizes the contributions. The thesis organizations and publications are presented in Sections 1.5 and 1.6, respectively.

1.2 Background

1.2.1 Active Learning of Halfspace

Active learning [Cohn et al., 1994] can be traced back to the early probability support vector machine (SVM) that acquires data with minimum margin to effectively update the support vectors. To find a hypothesis whose error rate is close to what we desire, agnostic active learning [Balcan et al., 2006] presents a series of algorithmic paradigms with a fixed or bounded version space [Cohn et al., 1994] covering a possible hypotheses class [Vapnik and Chervonenkis, 2015]. Candidates from this class is assigned with a goal of minimizing the queries from the unlabeled pool, where the desired one is with the optimal querying budget. To build a near-optimal querying algorithm in real world, agnostic active learning [Dasgupta et al., 2008; Balcan et al., 2006] improved the generalization of a realizable-theoretical model with prior labels selected from various distributions and

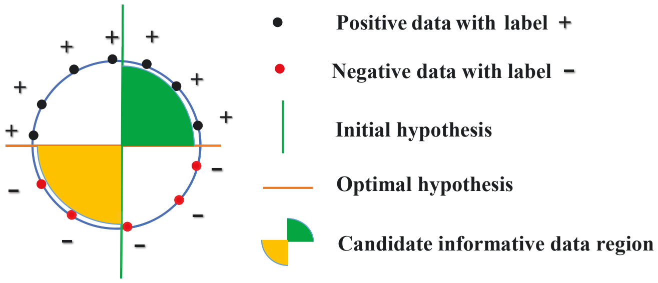

Positive data with label +
Negative data with label −
Initial hypothesis
Optimal hypothesis
Candidate informative data region
+
−

Figure 1.2 : Halfspace learning over a unit sphere with a radius of R, where $+, -$ denote different class labels. The error disagreement-based active learning strategy prunes the hypothesis set (reduce the number of candidate hypotheses, i.e., the diameters across the colored regions) via querying data distributed in the colored pool.

diverse noise conditions [Yan and Zhang, 2017]. Those generalized active learning algorithms involved with pruning the hypothesis set of the version space can be regarded as a hypothesis-pruning strategy [‡]. For example, halfspace learning [Cohn et al., 1994] is one active learning problem over a unit sphere to explore the theoretical guarantees on error rate and label complexity [Gonen et al., 2013]. Its goal is to learn a halfspace which accurately classifies binary classes. We here use halfspace learning to visualize the querying process of active learning.

Figure 1.2 describes halfspace learning. Over a binary classification task in a two-dimensional sphere (circle) with a uniform distribution, an arbitrary halfspace can generate a linear classifier. To reduce the error rate of the initial hypothesis, an active learning algorithm usually samples a number of informative points from the colored candidate pools that can largely update the current classifier.

From the perspective of version space in SVM, the querying process of active learning is equivalent to searching a subspace that characterizes the same hypothesis with a lower bound on the Vapnik-Chervonenkis (VC) dimension [Cortes et al., 2019b; Dasgupta, 2011]. With each query, the disagreement between the initial and desired hypotheses is expected to shrink. Thus, the disagreement between the initial and optimal hypotheses can be used as a measure to determine the distribution of the candidate hypothesis class in a version space. However, the label complexity of querying unseen samples is sensitive to this measure. That is, a poor initial hypothesis, which is far from the desired hypothesis, results in an increase of their generalized disagreement. The label complexity of querying increases rapidly as well. Therefore, the query samples heavily depend on the initial hypothesis.

Most previous work regarding pruning the hypothesis class either makes strong distribution assumptions such as halfspace learning [Gonen et al., 2013], or else it is computationally prohibitive [Brightwell and Winkler, 1991]. For any data distribution, Cortes et al. [2019b] remove the hypotheses whose connected edges are labeled with any dis-

---

[‡] Hypothesis-pruning is a high level description of active learning from the view of hypothesis class, where error disagreement-based active learning can be generalized as one hypothesis-pruning strategy. Therefore, hypothesis-pruning is a more broad expression of conceptual learning.

agreements larger than a given threshold. Their goal is to decrease the dependency of the initial hypothesis by a group of representative hypotheses. In their work, the version space [Cohn et al., 1994] which includes all feasible hypotheses, is embedded as a graph in a high-dimensional space. After pruning with this graph, any hypothesis in the original version space would be characterized with a lower bound on VC dimension. Then, the upper bound of the label complexity is reduced. However, hypothesis-pruning strategy has the following limitations:

1. performing hypothesis-pruning in the candidate data pool could reduce the influence of the initial hypothesis but it does not completely eradicate its dependence;
2. hypothesis-pruning strategies with the hypothesis class need a special distribution assumption, but it cannot be applied in arbitrary input distributions, though this theoretical description has attracted a lot of attention from researchers.

Therefore, it is desirable to develop a novel shattering strategy which achieves the same goal as the hypothesis-pruning strategy and deals with the input distribution in real-world tasks. To this end, we attempt to bridge the connection between the version space and input distribution.

1.2.2 Error Disagreement

Error disagreement [Hanneke, 2014] is a class of coefficients used to guide the informative sampling in active learning. Generally, an error disagreement coefficient estimates a feasible update to the current hypothesis and selects the best data for a human to annotate so as to maximize this update.

Theoretically, the assumption behind these coefficients is that the hypothesis class $\mathcal{H}$ is covered by an embedded graph G [Cortes et al., 2019c] with finite vertices, where each vertex denotes one hypothesis, and the geodesic distance between any pair of vertices reflects the level of disagreement with the hypothesis. The annotation process of active learning then becomes one of shrinking the candidate hypothesis set over the version space, by estimating the edges between the current and next hypotheses. The final goal is to minimize the cuts of an initial vertex to a desired one over this graph.

Usually, a generalized error disagreement is used to define the length of any pair of hypothesis edges. A classic generalized type of error disagreement is presented below. This one comes from the "importance weighted active learning (IWAL)" algorithm [Beygelzimer et al., 2009a]. Let $\mathcal{X}$ be an input dataset, $\mathcal{Y}$ be its output label set, and $\mathcal{H}$ be a hypothesis class over the marginal distribution $\mathcal{D}$ of $\mathcal{X}$, for any hypothesis pair $\{h, h'\}$, $\mathcal{L}(h(x), h'(x))$ denotes the hypothesis disagreement (distance), where $x \in \mathcal{X}$. Correspondingly, in graph G, $\mathcal{L}(h(x), h'(x))$ denotes the length between the vertices of h and h'. We here define this hypothesis disagreement as:

$$\mathcal{L}(h(x), h'(x)) = |\max_{y \in \mathcal{Y}} \ell(h(x), y) - \ell(h'(x), y)|, \tag{1.1}$$

where ℓ denotes the loss function of mapping $\mathcal{X}$ to $\mathcal{Y}$ and $y \in \mathcal{Y}$. When maximizing the disagreement of the losses on a single class in Eq. (1.1), we know $\mathcal{L}(h(x), h'(x))$ is a generalization of the best-in-class error on $\mathcal{Y}$.

To find a plausible hypothesis candidate, the learner uses an error disagreement coefficient to prune $\mathcal{H}$. Given $r > 0$, let $B(h^*, r)$ denote a ball centered in $h^* \in \mathcal{H}$ with a radius r: $B(h^*, r) = \{h' \in \mathcal{H} : \mathcal{L}(h^*, h') \leq r\}$, where h^* is the best hypothesis of $\mathcal{H}$. The error disagreement coefficient is defined as the minimum value of θ for any $r > 0$:

$$\theta \geq \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \Big[\max_{h \in B(h^*, r)} \frac{\mathcal{L}(h^*(x), h(x))}{r} \Big]. \tag{1.2}$$

Given the current hypothesis h, within a given hypothesis radius r which already bounds the loss disagreement of h to its posterity, if h is far away from the optimal hypothesis h^*, θ will be a large value. It means the algorithm will make large number of queries to approximate its desired. Also, the sampling is a rough process. A simple expression for the sampling policy is as follows.

- Any subsequent hypothesis with an error disagreement smaller than θ would be a null hypothesis, i.e. insignificant update.
- Any subsequent hypothesis with an error disagreement larger than θ would be a significant update.

Therefore, the value of θ in Eq. (1.2) decides the lower bound of the label complexity for pruning the current hypothesis set into its best posterity.

However, in multi-class settings, using this error type leads to class biases, which then means the pruning process roughly shrinks the volume of the version space. It then results in a sub-optimal solution [Hoang et al., 2014] on minimizing the cuts of active learning over graph G.

Recently, Cortes et al. [2019c] provided a tighter and more provable coefficient θ', where the length of the hypothesis edge w.r.t. Eq. (1.1) is re-expressed as the average loss of all-in-class errors, i.e. $\rho(h, h')$,

$$\rho(h, h') = \mathop{\mathbb{E}}_{(x,y) \sim \mathcal{D}} \Big[|\ell(h(x), y) - \ell(h'(x), y)| \Big]. \tag{1.3}$$

Concisely, Cortes et al. [2019c] controlled the hypothesis disagreement (distance) to reduce θ by replacing the loss function $\mathcal{L}(\cdot, \cdot)$ with $\rho(\cdot, \cdot)$. By applying Eq. (1.3) in the graph pruning, $B(h^*, r)$ is re-expressed as: $B'(h^*, r) = \{h \in \mathcal{H} : \rho(h, h^*) \leq r, r \geq 0\}$. Then, θ is updated to θ':

$$\theta' \geq \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \Big[\max_{h \in B'(h^*, r)} \frac{\mathcal{L}(h^*(x), h(x))}{r} \Big]. \tag{1.4}$$

Those types of error disagreement always assume the $\mathcal{X}$ is with a uniform distribution over the unit sphere or a log-concave distribution [Balcan and Feldman, 2013]. When the learner who has no knowledge to access its hypothesis class, the above coefficients would be not applicable e.g. teaching a black-box learner [Dasgupta et al., 2019]. However, θ' successfully shrinks θ via reducing the volume of $B(h^*, r)$ with provable bounds.

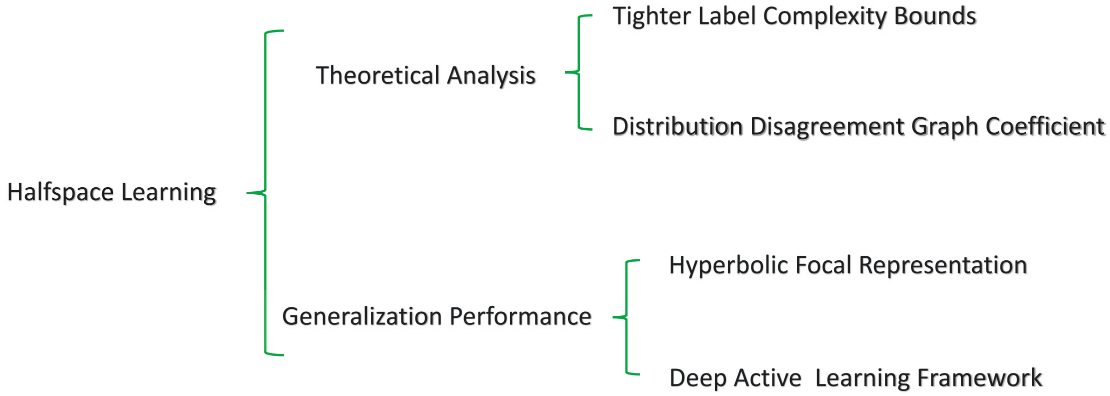

Halfspace Learning
Theoretical Analysis
Tighter Label Complexity Bounds
Distribution Disagreement Graph Coefficient
Generalization Performance
Hyperbolic Focal Representation
Deep Active Learning Framework

Figure 1.3 : Halfspace learning presents guarantees for the theoretical analysis and generalization performance of distribution-based active learning. Theoretical analysis includes contributions of tighter label complexity bounds of Chapter 1.4.1 and distribution disagreement graph coefficient of Chapter 1.4.2. Generalization performance includes hyperbolic focal representation which generalizes distribution disagreement in hyperbolic geometry (related work in Chapter 1.3.2), and deep active learning framework of Chapter 1.4.3 (related work in Chapter 1.3.3).

1.3 Related Work

Figure 1.3 firstly presents the connections of the related work and subsequent contributions.

1.3.1 Halfspace Learning

To reduce the dependence of the labeled set, active learning tries to find a near-optimal [Chen et al., 2017; Golovin et al., 2010] hypothesis from the hypothesis class in the version space. In this theoretical learning task, learners are given access to a stream of unlabeled data drawn i.i.d. from a fixed distribution. The proposed algorithm paradigms, which have already achieved a dramatic reduction in label complexity, are loosely termed hypothesis-pruning.

Substantial hypothesis-pruning frameworks under various assumptions of classifiers and labeled sets were proposed in past decades. For example, the query by committee algorithm [Freund et al., 1997] assumes that a correct Bayesian prior exists on the hypothesis class. To find a desired hypothesis, the committee members vote to eliminate the updated hypothesis with maximal disagreement between them. For any hypothesis class, Dasgupta [2006] presents the sufficient and necessary conditions for active learning such as classifier setting and initial labeled set, etc. When there exists a perfect separator in classification tasks, any hypothesis-pruning algorithm could directly improve the current classifier in a rapid fashion such as uncertainty evaluation [Yang et al., 2015], expected error rate change [Roy and McCallum, 2001], etc. Over this assumption, the learning algorithms do not need to consider the distribution it induces. Any inconsistent hypothesis such as a subsequent hypothesis with higher error rate can then be pruned by a single or group of querying samples. With the increase in the number of queries, the VC bound of any hypothesis in the candidate hypothesis class would be shrunk continuously, regardless from which distribution this query comes.

Since the optimal hypothesis is in respect to the input distribution, some learners generate the distribution under a fixed case, such as [Balcan et al., 2006], [Beygelzimer et al., 2010], etc. Of these, halfspace learning [Gonen et al., 2013] becomes a special setting over a unit sphere with uniform distribution. This problem takes a binary classification issue as an example to study label complexity and error rate change after sampling, where a halfspace is either of the two convex sets into which a hyperplane divides the sphere. The goal is to find the optimal halfspace over a unit sphere. For example, in Figure 1.2, researchers try to reduce the vector angle θ between the initial and optimal hypotheses as rapidly as possible, in which θ decides the VC dimension of the current hypothesis. Under this training assumption, two methods are presented to reduce the label complexity: (1) halving [Hanneke, 2007b] the volume of the candidate pool to obtain a sparse space, and (2) binary search for halving. By halving, the learner can rapidly reduce the hypothesis capacity of the version space to decrease the label complexity of querying since a part of the hypotheses would be removed. Therefore, the hypothesis-pruning strategy is an effective solution in active learning theory. However, most of these active learning algorithms either make strong distribution assumptions such as separability, uniform input distribution or are generally computationally prohibitive [Dasgupta et al., 2008], thus they cannot effectively be applied in active learning tasks with input distribution.

1.3.2 Hyperbolic Geometry

Hyperbolic space is a construct of non-Euclidean geometry that is d-dimensional Riemannian manifold with a constant negative curvature [Nickel and Kiela, 2018]. Due to its property of preserving tree-likeness orders of element anatomy, hyperbolic space is an effective way of embedding a representative structure in hierarchical data. In this form of geometry, it simply convinces any Euclidean algorithm with a vector structure and a closed-form inner product metric. Hence, the key theorems of Euclidean space still hold. Applications such as hyperbolic neural networks [Ganea et al., 2018], clustering [Monath et al., 2019], and graph embeddings [De Sa et al., 2018] further demonstrate the advantages of this type of approach for embedding the representativeness of any geometric structure. To date, the most common model of hyperbolic space has been a Poincaré ball. However, recently, Nickel and Kiela [2018] studied alternative types of models and found that the Lorentzian (hyperboloid) model is substantially more efficient than the Poincaré ball for learning embeddings. Following this conclusion, Law et al. [2019] further proved that formulating the centroid with respect to the squared Lorentzian distance can be written in closed-form solution.

1.3.3 Deep Active Learning

In deep learning community, active learning was introduced to improve the training of a DNN model by annotating unlabeled data, where the data which maximize the model uncertainty [Ashukha et al., 2019; Lakshminarayanan et al., 2017] are the primary acquisitions. For example, in ensemble deep learning, out-of-domain uncertainty estimation [Malinin and Gales, 2018] selects those data which do not follow the same distribution as the input training data; in-domain uncertainty [Ashukha et al., 2019] draws the data from the original input distribution, producing reliable probability estimates. Gal et al. [2017] use Monte Carlo dropout (MC-dropout) to estimate predictive uncertainty for approximating a Bayesian convolutional neural network. Lakshminarayanan et al. [2017] estimate predictive uncertainty using a proper scoring rule as the training criteria to fed a DNN.

Taking a Bayesian perspective [Golovin et al., 2010], active learning can be deemed as minimizing the Bayesian posterior risk with multiple label acquisitions over the input unlabeled data. A potential informative approach is to reduce the uncertainty about the parameters using Shannon's entropy [Tang et al., 2002]. This can be interpreted as seeking the acquisitions for which the Bayesian parameters under the posterior disagree about the outcome the most, so this acquisition algorithm is referred to as Bayesian active learning by disagreement (BALD) [Houlsby et al., 2011].

Since Bayesian DNN presents effective matching on the distribution of the training parameters, active learning using error disagreement, generalizing as model uncertainty in deep learning, is feasible and highly effective. Therefore, Gal et al. [2017] proposed to cooperate BALD with a Bayesian DNN to improve the training. The unlabeled data which maximizes the model uncertainty provides positive feedback. However, it needs to repeatedly update the model until the acquisition budget is exhausted. To improve the acquisition efficiency, batch sampling with BALD is applied [Kirsch et al., 2019; Pinsler et al., 2019]. In BatchBALD, Kirsch et al. [2019] developed a tractable approximation to the mutual information of one batch of unlabeled data and current model parameters. However, those uncertainty evaluations of Bayesian active learning whether in single or batch acquisitions all take greedy strategies, which lead to computationally infeasible, or excursive parameter estimations. Pinsler et al. [2019] thus approximated the posterior over the model parameters by a sparse subset, i.e. core-set construction. Applying Frank-Wolfe optimization [Vavasis, 1992], batch acquisitions of large-scale dataset can be efficiently derived, thereby interpreting closed-form solutions for core-set construction on linear and probit regression functions. As a consequence, non-deep models obtained theoretical guarantees from this optimization solver due to tractable model parameters. For deep Bayesian active learning, lacking of interaction to DNNs may not maximally drive their model performance. In applications, BALD was introduced into natural language processing [Siddhant and Lipton, 2018], text classification [Burkhardt et al., 2018], decision making [Javdani et al., 2014], data augmentation [Tran et al., 2019], etc.

1.4 Contributions

1.4.1 Tighter Label Complexity Bounds

As discussed in [Balcan et al., 2010], the VC dimension with respect to the optimal hypothesis in the version space affects the number of querying candidate hypotheses, and plays an important role in its distribution description. We propose a fresh proposition that the version space could be shattered by the number density [§] of the input distribution. Then, any hypothesis can be characterized with a lower bound on VC dimension. Especially for any input distribution with a bounded space, the more data located in the input space, the more hypotheses the version space would have. Moreover, *the input dis-*

[§] https://en.wikipedia.org/wiki/Number_density *tribution induces a natural topology on the version space, and a local hypothesis would easily capture its relevant local distribution [Dasgupta et al., 2008].* Hereafter, we would perform the shattering on the number density of the input distribution with the following advantages: (1) it provides theoretical guarantees in relation to reducing the generalized bounds of label complexity and error disagreement as hypothesis-pruning; (2) it breaks the curse of the initial hypothesis; and (3) it provides model guidance for distribution-shattering algorithms in real-world active learning tasks.

Based on the above insights, the first work generalizes the distribution-shattering strategy in an input distribution. Firstly, we halve the number density of the input distribution to obtain a shattered distribution. We then compare the generalization bounds between the shattered distribution and input distribution on *error disagreement* and *label complexity* for any hypothesis class under arbitrary data distributions. Our theoretical results show that the shattered distribution has lower generalization bounds in terms of the above two properties. Thus, we continuously split the shattered distribution to find a representation structure. This process is guided by a derived algorithm termed Shattered Distribution-based Active Learning (SDAL), which optimizes a group of local sphere centers as representative samples. Based on the analysis of the performance disagreement over hypothesis-pruning and distribution-shattering, we explore a series of scenarios including active querying with a limited labeled set, adversarial examples and noisy labels, where the first scenario is in regard to the poor initial hypothesis, and the last two scenarios are involved with the hypothesis update. The contribution of the first work is summarized as follows.

- We model the version space and input distribution by number density, which characterizes the generalized capacity of any hypothesis in a natural and direct way. We present a theoretical guarantee of the improvement on error disagreement and label complexity for shattering the number density of the input distribution. A derived algorithm named SDAL, which is independent of the initial labeled set and classifier, achieves lower error performance than the hypothesis-pruning algorithms when querying with limited labels, adversarial examples and noisy labels.

1.4.2 Distribution Disagreement

Typically, representations of data structure can relieve the dependency on error disagreement such as Euclidean centroids of an enclosed geometric space over the data. Core vector machine [Tsang et al., 2005b], geometric enclosing networks [Le et al., 2018], and adversarial training [Cranko et al., 2019] are examples of this type of geometric approaches. Based on this observation, we propose a Distribution Disagreement Graph Coefficient (DDGC) [¶] (written as θ_G) to replace the classic error disagreement coefficient used in non-deep active learning. We theoretically prove that distribution disagreement coefficient can produce a tighter bound on label complexity than that of error disagreement coefficient.

Naturally, generalizing DDGC by geometric centroids is a straightforward approach. However, centroid formulation requires the data space to have a strong spherical distribution; in the real world, the boundary surface over the data space is usually aspherical and

---

[¶] DDGC is a generalization of distribution disagreement over graph.

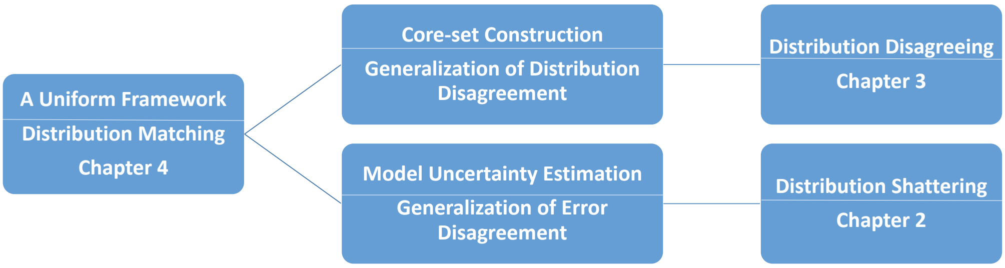


Figure 1.4 : A uniform framework consists generalizations of distribution and error disagreements.

most of the data resides densely on one side of this fitted surface. Thus, the focal points of enclosed aspherical space are more characteristic of the representation boundaries than Euclidean centroids. We then embed active learning onto a non-Euclidean hyperbolic geometry, in which the centroid representation is shifted towards the boundary by replacing the Euclidean norm with the Lorentzian norm [Nickel and Kiela, 2018; Law et al., 2019]. With this change, the version space of Euclidean is shrunk into tighter Lorentzian space that, deriving the focal point representation, further can be a generalization of DDGC. The used squared Lorentzian distance [Law et al., 2019], which yields a closed-form update formula on focal points. The last element to our strategy is a splitting approach for the Lorentzian representation based on tree-likeness [Hamann, 2018] that significantly speeds up the learning process. Concretely, contribution of the second work is that:

- We derive a more general distribution disagreement expression to replace the typical error disagreement for active learning, yielding tighter bound on label complexity. For the generalizations of distribution disagreement, we find that Lorentzian focal [‖] points of hyperbolic space present more effective representations than Euclidean, Gaussian kernelized, and Poincaré centroids on aspherical distributions.

1.4.3 A Unified Framework

In deep active learning, we propose an improved Geometric Bayesian Active Learning by Disagreement (GBALD) framework over the geometric interpretation of BALD that, interpreting BALD with core-set construction on an ellipsoid, match an effective distribution representation to drive a Bayesian model. The goal is to seek for significant accuracy improvements against an uninformative prior and redundant information. In the first stage of GBALD, geometric core-set construction on an ellipsoid initializes effective distribution matching to start a DNN model regardless of the uninformative prior. Taking the core-set as the input features, the next stage of GBALD ranks the batch acquisitions of model uncertainty according to their geometric representativeness, and then solicits some highly-representative examples from the batch. With the representation constraints, the ranked acquisitions reduce the probability of sampling nearby samples of the previous acquisitions, preventing redundant acquisitions. Contribution of the third work is as follows.

[‖] Hyperbolic focal adopts Lorentzian norm, it is therefore also called Lorentzian focal.

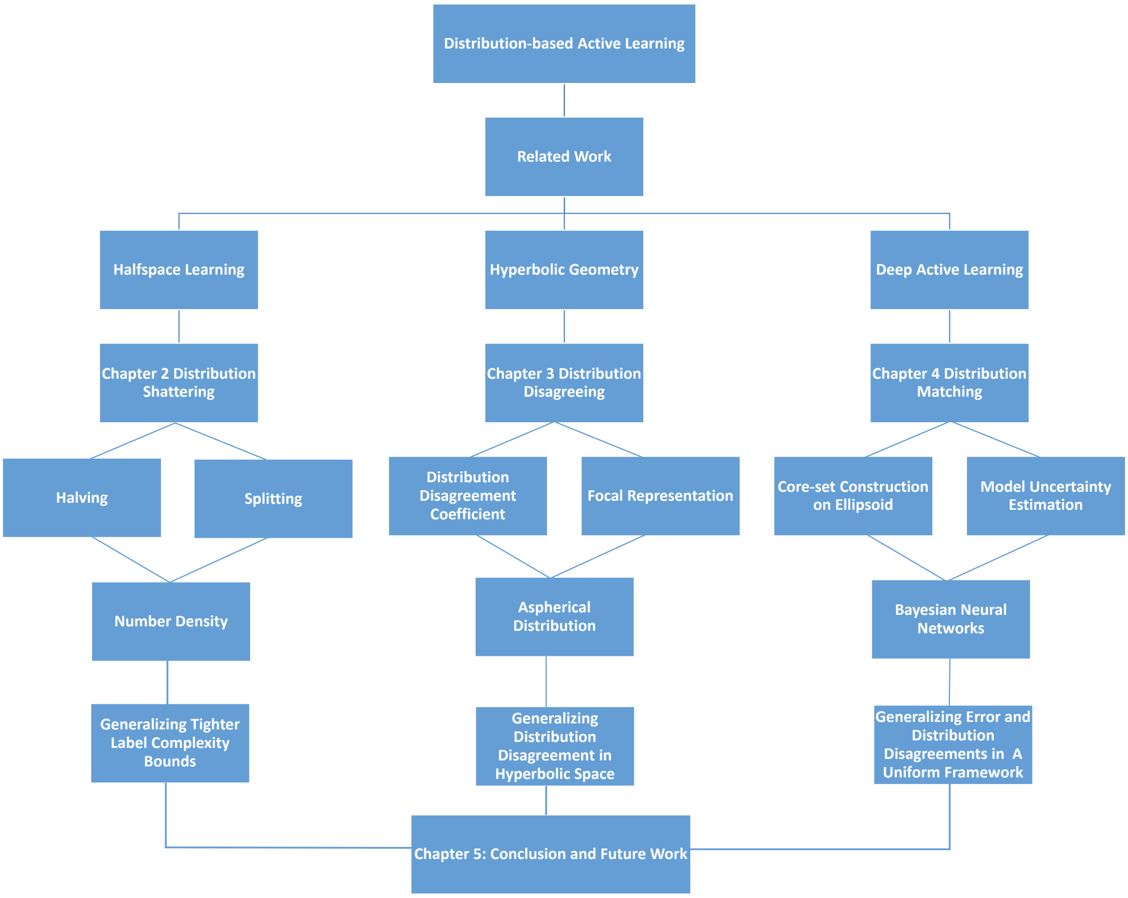

Distribution-based Active Learning
Related Work
Halfspace Learning
Hyperbolic Geometry
Deep Active Learning
Chapter 2 Distribution Shattering
Chapter 3 Distribution Disagreeing
Chapter 4 Distribution Matching
Halving
Splitting
Distribution Disagreement Coefficient
Focal Representation
Core-set Construction on Ellipsoid
Model Uncertainty Estimation
Number Density
Aspherical Distribution
Bayesian Neural Networks
Generalizing Tighter Label Complexity Bounds
Generalizing Distribution Disagreement in Hyperbolic Space
Generalizing Error and Distribution Disagreements in A Uniform Framework
Chapter 5: Conclusion and Future Work

Figure 1.5 : The skeleton of the thesis.

- GBALD, a geometry-driven Bayesian active learning framework that matches the distribution on ellipsoid, deriving the deep learning model into highly informative and representative acquisitions.

Figure 1.4 presents the structure of GBALD framework, which consists of core-set construction and model uncertainty estimation. In a generalization view, the core-set is generalized from distribution disagreement of Chapter 3 and model uncertainty is generalized from error disagreement of Chapter 2.

1.5 Thesis Organizations

This thesis studies the distribution-based active learning from halfspace theory to deep learning applications. Figure 1.5 presents the skeleton of the thesis. The structure of the remaining chapters is organization as follows.

1. Chapter 2 introduces the active learning by shattering the number density of the input distribution, deriving tighter theoretical bounds for label complexity, where shattering is implemented from halving to splitting.

2. Chapter 3 introduces the Lorentzian focal representation to generalize the distribution disagreement in hyperbolic geometry, which presents a novel alternative for infeasible estimations on error disagreement.

3. Chapter 4 introduces a geometrical Bayesian deep active learning framework which matches the core-set expression of the distribution, interacting with model uncertainty estimation of a Bayesian neural network model.

4. Chapters 5 concludes this thesis.

1.6 Publications

1. **Xiaofeng Cao**, Baozhi Qiu, Xiangli Li, et al. Multidimensional Balance-Based Cluster Boundary Detection for High-Dimensional Data, IEEE transactions on neural networks and learning systems, 30(6): 1867-1880, 2019.

2. **Xiaofeng Cao**, Ivor W. Tsang. Shattering distribution for active learning, IEEE transactions on neural networks and learning systems, 2020.

3. **Xiaofeng Cao**, Ivor W. Tsang, Jianliang Xu. Cold-start Active Sampling via γ-Tube, IEEE Transactions on Cybernetics, 2021.

4. **Xiaofeng Cao**, Ivor W. Tsang. Distribution Disagreement via Lorentzian Focal Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, revision.

5. **Xiaofeng Cao**, Ivor W. Tsang. Distribution-based Machine Teaching for a Black-box, Artificial Intelligent, under review.

6. **Xiaofeng Cao**, Ivor W. Tsang. Bayesian Active Learning by Disagreements: A Geometric Perspective, IEEE Transactions on Pattern Analysis and Machine Intelligence, under review.

7. **Xiaofeng Cao**, Ivor W. Tsang. Poincaré Fréchet Mean. IEEE transactions on neural networks and learning systems, under review.

8. **Xiaofeng Cao**, Ivor W. Tsang. Minimizing Hyperspherical Energy in Decision Boundary for Active Learning. NeurIPS 2021, under review.

9. **Xiaofeng Cao**, Baozhi Qiu, et al. BorderShift: toward optimal MeanShift vector for cluster boundary detection in high-dimensional data, Pattern Analysis and Applications 22(3): 1015-1027, 2019.

10. **Xiaofeng Cao**. A structured perspective of volumes on active learning. Neurocomputing, 377: 200-212, 2020.

11. **Xiaofeng Cao**. A divide-and-conquer approach to geometric sampling for active learning, Expert Systems with Applications, 140, 2020.

12. **Xiaofeng Cao**. High-dimensional cluster boundary detection using directed Markov tree, Pattern Analysis and Applications, 2020.

13. Zenglin Shi, Le Zhang, Yun Liu, **Xiaofeng Cao**, et al. Crowd counting with deep negative correlation learning. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 5382-5390) (CVPR 2018).
14. Xiaofeng Xu, Ivor W. Tsang, **Xiaofeng Cao**, et al. Learning image-specific attributes by hyperbolic neighborhood graph propagation. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 3989-3995) (IJCAI 2019).

Chapter 2

Distribution Shattering

This chapter discusses the first question of the thesis: "how to reduce the typical theoretical bounds of label complexity?". From a perspective of hypothesis class, we summarize the current active learning approaches involved with hypothesis updating as a hypothesis-pruning strategy, where error disagreement is a typical coefficient to control the feasible updates. However, those updates heavily depend on the initial hypothesis regard to classifier and labeled set. An improper initialization inevitably degenerates the pruning process of the hypothesis, which then rapidly increases the label complexity bounds.

To reduce the typical theoretical bounds of label complexity, we present a distribution-shattering strategy that shatters the number density of the input distribution without estimations on hypotheses. For any hypothesis class, we halve the number density of the input distribution to obtain a shattered distribution, which characterizes any hypothesis with a lower bound on VC dimension. Our analysis shows that sampling in a shattered distribution reduces label complexity and error disagreement. With this paradigm guarantee, in an input distribution, a Shattered Distribution-based Active Learning (SDAL) algorithm is derived to continuously split the shattered distribution into a number of representative samples. An empirical evaluation on benchmark datasets further verifies the effectiveness of the halving and querying abilities of SDAL in real-world active learning tasks with limited labels. Experiments on active querying with adversarial examples and noisy labels further verify our theoretical insights on the performance disagreement of the hypothesis-pruning and distribution-shattering strategies

The remainder of this chapter is organized as follows. Section 2.1 introduces the main theoretical insights. Section 2.2 models hypothesis and distribution by number density. Section 2.3 proposes the distribution-shattering strategy. Section 2.4 presents the experiments. Section 2.5 presents the discussions. Conclusion of this chapter is drawn in Section 2.6.

2.1 Main Theoretical Insights

Section 2.1.1 presents the preliminaries for one fundamental learning policy of the hypothesis-pruning strategy, which uses a disagreement coefficient to control the sampling of active learning. Specifically, the concept of the sparse hypothesis class which provides the foundation for the distribution-shattering is introduced in Section 2.1.2. Then, Section 2.1.3 analyzes the hypothesis-pruning and distribution-shattering active learning by halfspace learning and discusses their performance disagreements.

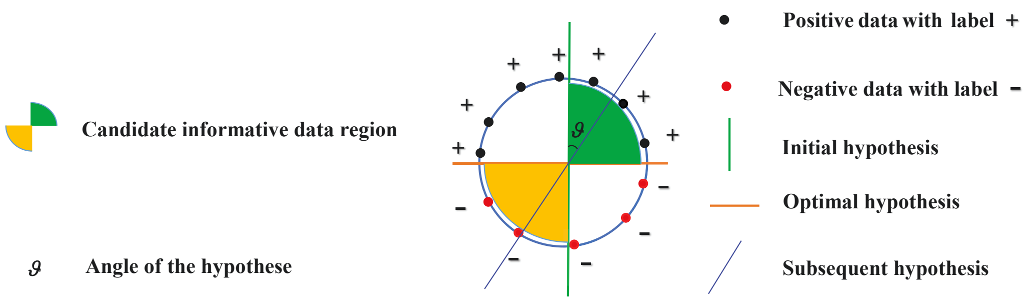

Positive data with label +
Negative data with label –
Candidate informative data region
Initial hypothesis
Optimal hypothesis
ϑ Angle of the hypothese
Subsequent hypothesis
+
–
ϑ

Figure 2.1 : Error disagreement of halfspace learning over a unit sphere with a radius of R, where + and − symbols denote the positive and negative class labels, respectively. The error disagreement-based active learning strategy prunes the hypothesis set via updating a subsequent hypothesis with large ϑ; the initial hypothesis has an error of $1/2$, and the subsequent hypothesis has an error of ϑ/π, thereby the error disagreement is $1/2 - \vartheta/\pi$.

2.1.1 Error Disagreement of Hypothesis-pruning

The hypothesis-pruning active learning algorithm queries the label of one example based on the empirical rule of error rate difference after assigning a positive or negative label. To describe the basic model of hypothesis-pruning, we present some preliminaries in this section.

Given a data set $\mathcal{X}$ with binary class labels, and $\mathcal{D}$ is the distribution over $\mathcal{X} \times \{\pm 1\}$, we divide $\mathcal{X}$ into two groups: $\mathcal{L}$ and $\mathcal{U}$, in which $\mathcal{L}$ contains the labeled set of $\mathcal{X}$, and $\mathcal{U}$ contains the unlabeled set. Let $\mathrm{err}(\mathcal{X}, \mathcal{L})$ denote the error rate of predicting $\mathcal{X}$ by training the labeled set $\mathcal{L}$, $\{\hat{x}, -1\}$ and $\{\hat{x}, +1\}$ denote the queried data with negative or positive labels, $\hat{x} \in \mathcal{X}$, t denote the tth query, $\mathcal{L}_t$ denote the labeled set in the tth query, and k denote the total number of queries. We present the policy for querying by the error disagreement Δ_{t} [Dasgupta et al., 2008],

$$\begin{aligned} &|\mathrm{err}(h_{-1}, \mathcal{L}_t \cup \{\hat{x}, -1\}) - \mathrm{err}(h_{+1}, \mathcal{L}_t \cup \{\hat{x}, +1\})| > \Delta_{\mathrm{t}} \\ &\text{s.t.} \quad \{\hat{x}, \pm 1\} \subset \mathcal{U}, t = 1, 2, 3, \cdots, k, \end{aligned} \tag{2.1}$$

where h_{-1} and h_{+1} denote the classification hypotheses after assuming $\hat{x}$ with a negative and positive label, respectively. By employing this policy, the active learners pick up those data whose error disagreements of $|\mathrm{err}(h_{-1}, \mathcal{L}_t \cup \{\hat{x}, -1\}) - \mathrm{err}(h_{+1}, \mathcal{L}_t \cup \{\hat{x}, +1\})|$ are larger than the given coefficient Δ_{t}. If the error disagreement of one data is far greater than the coefficient, it updates the current classification hypothesis significantly. Otherwise, the influence on the current hypothesis of adding the data to $\mathcal{L}$ is insignificant. Figure 2.1 explains the error disagreement over halfspace learning.

The theoretical guarantees for this policy can be expressed in terms of the generalized disagreement coefficient [Hanneke, 2007a] over a fixed assumption. Given a hypothesis class $\mathcal{H}$ over $\mathcal{X}$, let h^* be the optimal hypothesis which satisfies $h^* = \mathrm{arginf}_{\mathrm{h} \in \mathcal{H}}\, \mathrm{err}_{\mathcal{D}}(h)$, $\nu = \mathrm{err}_{\mathcal{D}}(h^*)$, and $h(x) \neq h^*(x)$, where $\mathrm{err}_{\mathcal{D}}(h)$ denotes the error of hypothesis h with respect to distribution $\mathcal{D}$. Let $B(h^*, r)$ [Dasgupta, 2011] be a ball centered with h^*, given a radius r limits the volume of the candidate hypotheses around h^*, we define

$B(h^*, r) = \{h' \in \mathcal{H} : \ell(h^*, h') < r\}$, where $\ell(\cdot,\cdot)$ denotes the metrical distance between the two hypotheses. Generally, $\ell(\cdot,\cdot)$ can be generalized as the error disagreement of Eq. (2.1). Assume there exists a descried error rate ϵ, the generalized disagreement coefficient [Cortes et al., 2019b] is defined as the minimum value of θ such that for any r:

$$\theta = \sup\left\{\frac{\Pr_{x\sim\mathcal{D}}[\exists h \in B(h^*, r)]}{r} : r \leq \epsilon + \nu\right\}, \tag{2.2}$$

where $\Pr$ denotes the probability mass in $B(h^*, r)$ such as the candidate hypothesis disagreement or misclassified data amount.

The generalized types of $\Pr$ are typically used in various hypothesis-pruning active learning: Dasgupta [2006] presents an upper bound of label complexity using maximum disagreement between any hypothesis in $\mathcal{H}$; Cortes et al. [2019b] tight this bound by using the best-in-class error disagreement, etc.

2.1.2 Sparse Hypothesis Class

To find the instance with the highest informativeness, the hypothesis-pruning active learning algorithms using error disagreement select the data which maximally split $\mathcal{H}$, and then shrink the number of candidate hypothesis. However, the general error disagreements need a linear classifier or fixed distribution and it is only a special metric over hypothesis disagreement. In this section, we study the hypothesis distribution which is independent of the structural assumption of fixed distribution.

Without a given distribution, we assume the hypothesis class is distributed in an unseen graph structure G and each node denotes a hypothesis. Then, $B(h^*, r)$ denotes a ball centered with h^* and radius r in G. Afterwards, finding a sparse hypothesis class is the most important splitting factor.

Let h_t be the current hypothesis, x_i and x_j be two candidate sampling points in $\mathcal{X}$. Assume h_{t,x_i} and h_{t,x_j} are the updated hypotheses after sampling x_i and x_j, respectively, the disagreement coefficient can be the infimum value of θ'

$$\max_{\mathrm{h}\in\mathrm{B}(\mathrm{h}^*,\mathrm{r})} \ell(h^*, h_{t,x_i}) + \ell(h^*, h_{t,x_j}) \leq 2\theta' r, \forall r > 0, \tag{2.3}$$

where h_{t,x_i} is assumed to be the hypothesis with the maximum disagreement to h_{t,x_j} in a given radius setting r. In $B(h^*, r)$ of $\mathcal{G}$, h_{t,x_j} denotes the node which is the farthest from the node of h_{t,x_i}.

Let m be the number of unlabeled data in the candidate pool, the constrained hypothesis relationship set is descried as

$$\mathcal{H}' = \{(h_{t,x_1}, h'_{t,x_1}), (h_{t,x_2}, h'_{t,x_2}), ..., (h_{t,x_m}, h'_{t,x_m})\}, \tag{2.4}$$

where h'_{t,x_i} denotes the hypothesis that is the furthest from h_{t,x_i}. By employing the hypothesis disagreement function $\ell(\cdot,\cdot)$ of Eq. (2.3), learners can remove a part of the hypotheses by a margin distance θ'. Then, we obtain a sparse hypothesis class $\mathcal{H}^*$. With this splitting strategy, characterizing any hypothesis in $\mathcal{H}$ with a lower VC bound may be possible. Therefore, the key study of this paper is to prune the original hypothesis class

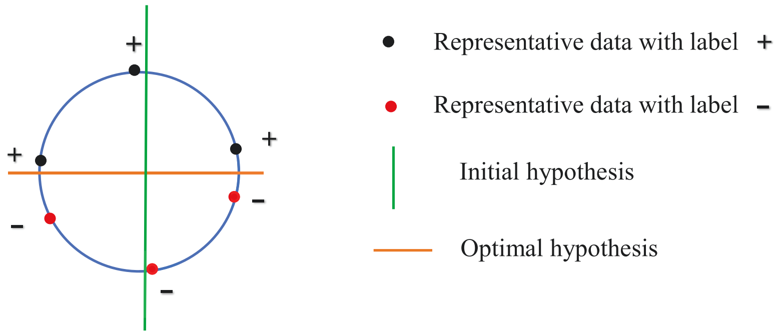

+
+
+
–
–
–
Representative data with label +
Representative data with label –
Initial hypothesis
Optimal hypothesis

Figure 2.2 : Shattering distribution of a unit sphere with a radius of R, where $+, -$ denote the class labels. Distribution-shattering halves the number density of an input distribution w.r.t. $\frac{12}{\pi R^2}$ into a shattered distribution w.r.t. $\frac{6}{\pi R^2}$. Any hypothesis generated from the original distribution is charactered with a lower bound on VC dimension. Thereby, we can find a representation structure that induces a tighter label complexity without estimating the hypothesis.

into a sparse structure from the distribution view. Figure 2.2 describes this process.

2.1.3 Performance Disagreement

Halfspace learning provides a clear visualization to describe the hypothesis relationship. Based on this advantage, in this section, we describe the performance disagreement of the hypothesis-pruning and distribution-shattering active learning by halfspace learning. We firstly describe different cases of learning a halfspace over a unit sphere.

Case 1. Halfspace learning. Learning a halfspace c^* [Alabdulmohsin et al., 2015; Chen et al., 2017] in a united sphere is to estimate an unknown vector μ that takes the sphere center as the start point,

$$c^* = \{x \in \mathbb{R} | \langle \mu, x \rangle \geq 0\}, \text{s.t. sign}(\langle x_i, \mu \rangle) \in \{+1, -1\}\,. \tag{2.5}$$

In this case study, the goal of halfspace learning is to estimate the optimal c^* using the lowest number of queries as possible. However, the label complexity of the unseen sampling process heavily depends on the initial hypothesis. Suppose that the points which could maximize the hypothesis or distribution update are the primary sampling data, we utilize label complexity to observe the difference of hypothesis-pruning and distribution-shattering active learning of the halfspace.

To explain the notion of label complexity, we take the label complexity of the passive (random) learning of halfspace as prior knowledge.

Case 1.1. Passive learning of halfspace. Let $\mathcal{D}$ be the distribution over a unit sphere with $1/\epsilon$ data, then the label complexity of passive sampling is $\mathcal{O}(\frac{1}{\epsilon})$.

Let v_t be the vector classifier on the tth query, and θ_t be the angle between v_t and μ, we give the following case studies.

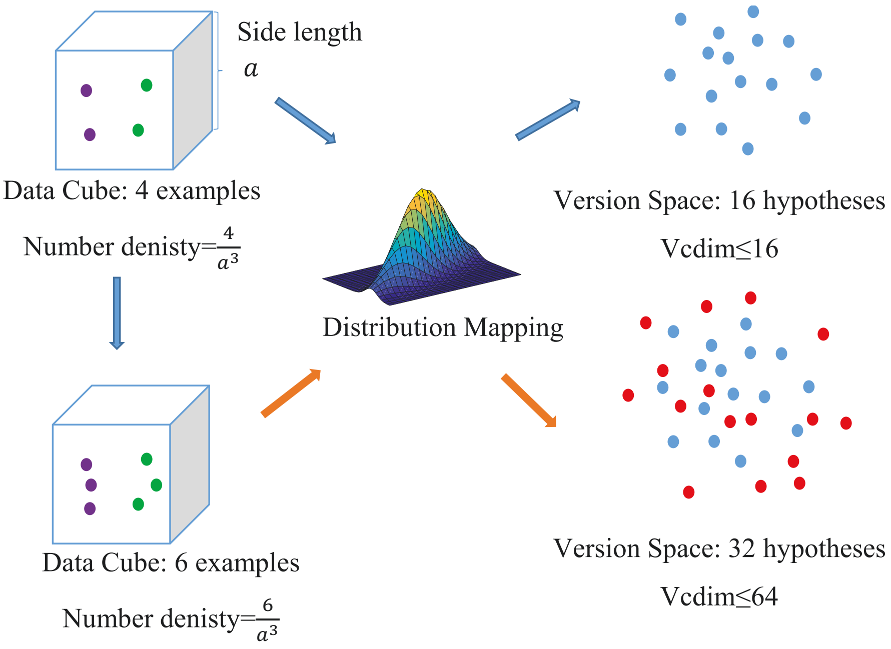

Side length
a
Data Cube: 4 examples
Number denisty=$\frac{4}{a^3}$
Distribution Mapping
Version Space: 16 hypotheses
Vcdim≤16
Data Cube: 6 examples
Number denisty=$\frac{6}{a^3}$
Version Space: 32 hypotheses
Vcdim≤64

Figure 2.3 : Number density models the relationship of hypothesis and distribution, i.e. VC dimension increases proportionally as the number density. If number density=$\frac{4}{a^3}$ and the hypothesis is generated from i samples, the VC bound $\leq \sum_{i=1}^{4}\binom{4}{i} = 16$; given the number density=$\frac{6}{a^3}$, the VC bound $\leq \sum_{i=1}^{6}\binom{6}{i} = 64$. Note the version space of number density=$\frac{6}{a^3}$ only presents 32 hypotheses, where the red nodes are the newly added ones from the version space of number density=$\frac{4}{a^3}$, and each blue node denotes one feasible hypothesis, respectively.

Case 1.2. Hypothesis-pruning active learning of halfspace. Let $\mathcal{D}$ be the distribution over a unit sphere with $1/\epsilon$ data, the label complexity of obtaining a lower error rate compared to the initial hypothesis is $\mathcal{O}(\frac{\theta_t}{\pi\epsilon})$. Even using the halving algorithm, the label complexity is $\mathcal{O}(\log\frac{\theta_t}{\pi\epsilon})$.

To reduce the error of the initial hypothesis, we need to query the labels of the data distributed between v_t and μ (colored area in Figure 2.1). Over a unit sphere with $1/\epsilon$ data, the candidate pool which can reduce the error of the initial hypothesis has $\frac{\theta_t}{\pi\epsilon}$ data. If we use the halving algorithm such as binary search in the candidate pool, the label complexity would be $\mathcal{O}(\log\frac{\theta_t}{\pi\epsilon})$. Different from the hypothesis-pruning active learning, the distribution-shattering active learning that requires the unseen sampled data is independent of the initial hypothesis.

Case 1.3. Distribution-shattering active learning of halfspace. Let $\mathcal{D}$ be the distribution over a unit sphere with $1/\epsilon$ data, the label complexity of obtaining a lower error rate compared to the initial hypothesis is $\mathcal{O}(1)$.

The above cases compare the sampling policies of the hypothesis-pruning and distribution-shattering active learning strategies over the unit sphere. The performance of the hypothesis-pruning active learning strategy heavily depends on the initial hypothesis. In real-world active learning tasks, the querying results of active learning depend on the input labeled set and updating of the training model. For example, an limited labeled set and misguided model update will degenerate the performance of the subsequent sampling. However, the final estimation on error rate of the proposed distribution-shattering strategy depends on the representation structure of the input distribution. In simple terms, learning the representation structure of the distribution could help to address the limitation of hypothesis-pruning with a certain sampling selection. In a real active learning task, the queried samples of any generalized distribution-shattering algorithm will be independent of the input training set.

2.2 Hypothesis and Distribution

In Section 2.2.1, we firstly present the monotonic property of the active query set to show the uncertain error rate change after querying. Then, we discuss the bottleneck of informative active learning and describe our splitting rule by representation sampling in Section 2.2.2. Finally, we discuss the relationship between error rate and number density of input distribution in Section 2.2.3.

Based on these theoretical analysis, we are motivated to undertake the splitting in input distribution. The goal is to eliminate the hypothesis supervision by learning the structure of the input distribution. Proofs are presented in Appendix A. Figure 2.3 presents the motivation of number density.

2.2.1 Monotonic Property of the Active Query Set

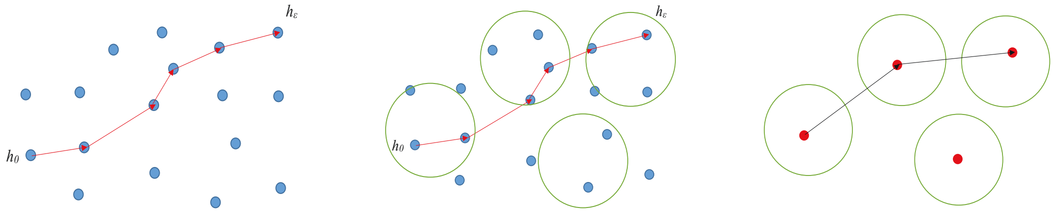
To observe the error rate change after increasing the size of the active query set, we follow the perceptron training (see Figure 2.1) to analyze the hypothesis relationship. In our perspective, training the updated hypothesis will result in two uncertain situations: (1) the error rate declines after querying, and (2) the error rate shows negative (or slow) improvement when querying a lot of unlabeled data. Therefore, the monotonic property of the active query set size and error rate are unknown. The following proposition provides a mathematical description for this discovery.

Proposition 1. *The monotonic property of active query set and error rate is unsatisfied or negative. Suppose ϵ_t and ϵ_{t+1} respectively are the error rates of training the active query sets $\mathcal{D}_{\epsilon_t}$ and $\mathcal{D}_{\epsilon_{t+1}}$. There must hold an uncertain probability relationship which satisfies* $\Pr(\epsilon_{t+1} \leq \epsilon_t | \mathcal{D}_{\epsilon_t} \subset \mathcal{D}_{\epsilon_{t+1}}) < 1$.

Proposition 1 describes the first perspective of this paper about the relationship between the performance of the hypothesis and the active query set size. It shows that the probability of reducing the current error rate by increasing the size of the active query set is unpredictable and answers the question that we proposed in the beginning of this paper. In the following, we observe the error rate change by shattering the number density of the candidate pool.

2.2.2 Error Rate Change by Shattering Number Density

Following the perceptron training in the unit circle with uniform distribution, we find the error rate grows with the number density of the input distribution. This study also



(a) Hypothesis-pruning (b) Local hypothesis spheres (c) Sparse hypothesis class

Figure 2.4 : The assumption of distribution-shattering with a sparse hypothesis class. Each node denotes one realizable hypothesis, and the lengths of the red lines denote the hypothesis disagreement. Hypothesis-pruning updates the initial hypothesis w.r.t. h_0 into h_ϵ with a desired error ϵ in the original hypothesis class (Fig. 2.4(a)). Distribution-shattering optimizes a group of local hypothesis spheres (Fig. 2.4(b)). Shattering by those sphere centers, the original hypotheses are transformed into a sparse hypothesis class (Fig. 2.4(c), thereby finding h_ϵ can achieve a lower label complexity than in its original hypotheses.

appears in active learning of halfspace.

Proposition 2. *Assume* $\theta_{t+1} > \theta_t$*, we know* $\mathrm{err}(\mathcal{D}_{\epsilon_t}) - \mathrm{err}(\mathcal{D}_{\epsilon_{t+1}}) = (\theta_{t+1} - \theta_t)\frac{\mathrm{Den}(B)}{n}$ *(w.r.t. the volume of the circle is* π*), where* $\mathrm{Den}(\cdot)$ *denotes the number density of the distribution.*

Error rate disagreement denotes the distance between two arbitrary hypotheses. By observing the above propositions, we find that number density affects the hypothesis disagreements. Furthermore, we know the number density roughly decides the VC dimension bound of the optimal hypothesis since $\mathrm{Vcdim}(B) = \sum_{k=1}^{n} \binom{n}{k} = 2^n = 2^{\pi \mathrm{Den}(B)}$. For these two reasons, number density is a direct way to describe the hypothesis distribution in version space. Therefore, we are motivated to shatter the number density of the input distribution to both reduce the VC bound and find a lower label complexity. In addition, we define $\mathrm{Den}(B)$ for the real active learning tasks in Section 2.3. In the following, we discuss the bottleneck of querying informative samples and present our solution to this issue.

2.2.3 Bottleneck of Hypothesis-pruning

In hypothesis-pruning, the generalized algorithm updates the initial hypothesis w.r.t. h_0 into h_ϵ with a desired error ϵ in the original hypothesis class over version space (Fig. 2.4(a)). The informative samples are the primary querying targets. However, estimating the hypothesis disagreement is challenging. In particular, when the initial hypothesis is set improperly (far from the optimal hypothesis in version space), the path of finding the optimal hypothesis might be difficult. Thus, there exists a bottleneck for the active learning sampling by querying informative samples, *i.e., the hypothesis disagreement from the initial hypothesis to the descried hypothesis is uncertain.*

Since the VC dimension greatly affects the path finding process for the optimal hypothesis, splitting the hypothesis class of version space into a sparse structure can alleviate the bottleneck of querying the informative samples. In our assumption, we use distribution-shattering to optimize a group of hypothesis spheres (Figure 2.4(b)). Shattering by those sphere centers, the original hypotheses are transformed into a sparse hypothesis class, thereby finding h_ϵ can achieve a lower label complexity than in that original hypotheses (Fig. 2.4(c)). To implement this proposal, we perform the splitting idea on the input distribution by finding k local balls constrained by the following rules.

Solution. Given $B_{\mathcal{D}}$ is a ball which tightly encloses $\mathcal{D}$, and $\{B_1, B_2, ..., B_k\}$ are the k local split balls with the condition of $\forall i, B_i \subset \mathcal{D}$. Let $\mathrm{Vol}(\cdot), r(\cdot)$ define the volume and radius of the input hypothesis object, respectively, the splitting must satisfy the following conditions: (1) the volume of arbitrary split ball B_i is smaller than that of $B_{\mathcal{D}}$, i.e., $\forall B_i$, $\mathrm{Vol}(B_i) < \mathrm{Vol}(B_{\mathcal{D}})$, (2) the sum of the volumes of all the split balls B_i is smaller than that of $B_{\mathcal{D}}$, i.e., $\sum_{i=1}^{k} \mathrm{Vol}(B_i) < \mathrm{Vol}(B_{\mathcal{D}})$, (3) the radius of an arbitrary ball is smaller than the radius of $B_{\mathcal{D}}$, i.e., $\forall B_i, r(B_i) < r(B_{\mathcal{D}})$, and (4) the distance between any two local hypothesis balls is bigger than the sum of their radii, i.e., $\ell(c_i, c_j) > r(B_i) + r(B_j)$, where $\ell(\cdot,\cdot)$ denotes the distance between the two inputs, and c_i denotes the center of the ith split ball.

The above splitting rules provide an algorithmic paradigm for distribution-shattering strategy. A generalized algorithm termed SDAL is then presented in Section 2.3.

Remark 1. *The policy of $\ell(c_i, c_j) > r(B_i) + r(B_j)$ is the key of the theoretical solution that avoids overlapping in representations of local hypothesis spheres. It is generalized in the convergence condition w.r.t. Line 15 of SDAL algorithm.*

2.3 Distribution-shattering for Active Learning

Section 2.3.1 explains how to shatter the input distribution from halving to splitting. Using a heuristic greedy selection, we halve the number density of the input distribution to obtain a shattered distribution in Section 2.3.2. Then, we discuss its theoretical advantages in Section 2.3.3. With these guarantees, Section 2.3.4 splits the shattered distribution of the input distribution into a certain number of local balls to find a representation structure. Proofs are presented in Appendix.

2.3.1 Shattering: From Halving to Splitting

Shattering the input distribution is proposed to eliminate the dependence of the hypothesis. In the last section, halving the number density of the colored candidate pool yields exponential reduce on the label complexity of halfspace learning. To prove the positive help of shattering, we propose to implement the halving algorithm against the input distribution. The theoretical estimations on the generalized label complexity and error rate difference reveal the effectiveness of shattering. If all feasible change can converge uniformly with the shattering percentages, we split the shattered distribution into several representation regions and use their central points as the query samples of active learning.

2.3.2 Halving Number Density for Shattered Distribution

By sorting the hypothesis disagreement of each pair in $\mathcal{H}'$ of Eq. (2.4), we use a splitting threshold θ' to halve the number density of the input space under arbitrary data distributions. The cutting rule is: let h_t is centered with its update h_{t,x_i} on x_i, for any

$x_j \in \mathcal{X}$, if $\ell(h_{t,x_i}, h_{t,x_j}) \geq \theta'$, we remove x_j from $\mathcal{X}$. After the cutting, $\mathcal{H}'$ will be reduced to $\mathcal{H}^*$ over a shattered distribution.

In hypothesis class $\mathcal{H}$, the VC dimensions of $\mathcal{H}$ and $\mathcal{H}^*$ can be written as $\mathrm{Vcdim}(\mathcal{H}) := d = \sum_{i=1}^{m} \binom{m}{i} = 2^m$ and $\mathrm{Vcdim}(\mathcal{H}^*) := d' = \sum_{i=1}^{m/2} \binom{m/2}{i} = \sqrt{2}^m$ [Cao et al., 2018]. Based on these assumptions, let us discuss the advantages of shattered distribution on label complexity and the upper bound of the querying.

Lemma 1. *Label complexity. Let each hypothesis hold for a probability at least* $1 - \delta$*, the label complexity* $m(\epsilon, \delta, \mathcal{H}^*)$ *is*

$$m(\epsilon, \delta, \mathcal{H}^*) = \frac{64}{\epsilon^2} \left(\frac{1}{\sqrt{2}^{m-2}} \mathrm{In} \frac{12}{\epsilon} \right) + \mathrm{In} \left(\frac{4}{\delta} \right) < m(\epsilon, \delta, \mathcal{H}). \tag{2.6}$$

Lemma 2. *Upper bound of queries. Following [Balcan et al., 2006], let us assume* $0 < \epsilon < 1/2$, $< 0 < \delta < 1/2$*, then the active learning will make at most* $2m(\epsilon, \delta'_{\mathcal{H}^*}, \mathcal{H}^*) < 2m(\epsilon, \delta'_{\mathcal{H}}, \mathcal{H})$ *queries, where* δ'_H *is denoted as* $\delta'_H = \frac{\delta}{N(\epsilon, \delta, H)^2 + 1}$.

Based on the above discussion, we can easily observe that the values of the two properties of the hypothesis class of the shattered distribution are lower than that of the original hypothesis class since it characterizes any hypothesis with a lower bound on VC dimension.

2.3.3 Advantages of Shattered Distribution

To observe the advantages of the shattered distribution, we 1) analyze the bounds of error disagreements between the hypotheses with positive or negative labeling assumptions, 2) discuss the upper bound of the error rate by fall-back analysis which requires a change in different assumptions that can hold for the same algorithm, and 3) present the label complexities in η-bounded and υ-adversarial noise conditions.

2.3.3.1 Bounds of Error Disagreement in Shattered Distribution

In this learning process, we continue to use the greedy strategy of halving to split the local unit ball $B(h^*, r)$. Before splitting, here we present the halving guarantees of error rate difference on the shattered distribution.

Theorem 1. *Let* $\mathcal{D}'$ *be the distribution over* $\mathcal{H}^*$*,* $\{h_i, h'_i\} \in \mathcal{H}$*,* h'_i *be furthest from* h_i *in* $\mathcal{H}$*,* $\mathcal{F}$ *be a family of functions* $f : \mathcal{Z} \to \{0, 1\}$*,* $\mathcal{S}(\mathcal{H}, n)$ *be the nth shatter coefficient with infinite VC dimension,* $\alpha_t = \sqrt{(4/t)\mathrm{In}(8\mathcal{S}(\mathcal{H}, 2t)^2)/\delta}$*,* $\mathbb{E}_Z f$ *be the empirical average of* f *over a subset* $Z \subset \mathcal{Z} \subset \mathcal{X}$ *with probability at least* $1 - \delta$*. Then, we have* $\Delta' = (\mathrm{err}(h_i, \mathcal{D}') - \mathrm{err}(h_i, \mathcal{D})) - (\mathrm{err}(h'_i, \mathcal{D}') - \mathrm{err}(h'_i, \mathcal{D})) \leq 0$.

Using this lemma, the error rate of the shattered distribution guarantees the decrease. However, it has a relationship with the size of $\mathcal{D}$. To obtain the structure of the version space, we continue to use the halving approach to split $\mathcal{H}$ into k local balls with a fall-back and bounded noises-tolerant guarantees.

2.3.3.2 Fall-back Analysis in Shattered Distribution

Fall-back analysis [Dasgupta et al., 2008] helps us to observe the upper bound of error rate in the shattered distribution. Before analyzing the fall-back of querying, we need some technical lemmas.

Lemma 3. *With an assumption of normalized uniform,* Δ_t *of Eq. (2.1) could be defined as:* $\Delta_t := \beta_t^2 + \beta_t(\sqrt{\mathrm{err}_t(h_{+1})} + \sqrt{\mathrm{err}_t(h_{-1})})$ *[Dasgupta et al., 2008], where* β_t *follows a PAC slack of* $\beta_t = \sqrt{(4/n)\mathrm{In}(8(n^2+n)\mathcal{S}(\mathcal{H}^*, 2n)^2\delta)}$.

Lemma 4. *With the assumptions of* $\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h_{-1}) > \Delta_t$, $\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h_{-1}) > \frac{2\beta_t^2}{1-\beta_t}$ *and it is consistent with the labeled set* $\mathcal{L}_t$ *for all* $t \geq 0$.

With Lemma 4, we then produce the upper bound of error of sampling in a shattered distribution.

Theorem 2. *Assume there exists a hypothesis* h_f *which satisfies* $\mathrm{err}_{D'}(h_f) \leq \mathrm{err}_D(h^*)$. *If the active learning algorithm is given by* k *queries with probability of* $1 - \delta$, *let* $\nu = \mathrm{err}_{D'}(h^*)$, *the error rate of shattered distribution is at most* $(\sqrt{\nu} + \beta_k)^2$.

From the above analysis, sampling in a shattered distribution can still converge safely. The upper bound of error of sampling in a shattered distribution is further proven to be tighter than sampling in the input distribution without halving. It shows sampling in a shattered distribution may save sampling consumption and a continuous splitting algorithm may further reduce this bound uniformly. Next, let us analyze the bounds of the label complexity in the noise settings.

2.3.3.3 Bounded Noise Analysis of Shattered Distribution

Under the uniform assumption, noises affect the unseen queries. Here we discuss the label complexities of the shattered distribution in η*-bounded* and v*-adversarial* noise settings [Yan and Zhang, 2017].

Theorem 3. *For some* $\eta \in [0, 1/2]$ *with respect to* μ *(w.r.t. Case 1), if for any* $x_i \in D'$, $\Pr[Y \neq \mathrm{sign}(\mu{\cdot}x_i)|X = x_i] \leq \eta$, *we say the distribution of* D' *is* η*-bounded [Massart et al., 2006]. Under this assumption, (1) there are at most* $\widetilde{\mathcal{O}}\left(\frac{d'}{(1-2\eta)^3\epsilon}\right)$ *unlabeled data, and (2) the number of queries is at most* $\widetilde{\mathcal{O}}\left(\frac{d'}{(1-2\eta)^2}\mathrm{In}\frac{1}{2\epsilon}\right)$, *where* $\widetilde{\mathcal{O}}(f(\cdot)) := \mathcal{O}(f(\cdot)\mathrm{ln}f(\cdot))$.

Theorem 4. *For some* $v \in [0, 1]$ *with respect to* μ, *if for any* $x_i \in D'$, $\Pr[Y \neq \mathrm{sign}(\mu \cdot x_i)|X = x_i] \leq v$, *we say the distribution of* D' *is* v*-adversarial noise condition [Awasthi et al., 2014]. Under this assumption, (1) there are at most* $\widetilde{\mathcal{O}}(\frac{d'}{2\epsilon})$ *unlabeled data, and (2) the number of queries is at most* $\widetilde{\mathcal{O}}\left(d'\mathrm{In}\frac{1}{2\epsilon}\right)$.

Compared to the original input distribution, the shattered distribution has lower label complexity since the VC bound of any hypothesis is shattered into a shaper value.

2.3.4 Distribution-shattering for Active Learning Tasks

Shattered distribution provides theoretical advantages without special distribution assumptions since number density is independent of arbitrary distribution situation. Therefore, in real-world active learning tasks, we firstly halve the number density of the input distribution to learn a shattered distribution via an active scoring strategy. After obtaining the shattered distribution, we split the shattered distribution into k balls via the distribution density. Then, we propose the SDAL algorithm for active learning querying.

2.3.4.1 Active Scoring for Halving

Active scoring is used to measure the local representativeness of arbitrary data, in which the score value monotonically grows with the representativeness. By removing some data with the lowest representativeness (i.e., halving the number density of the input distribution), we try to shatter the unlabeled data pool. This reduces the label complexity of the subsequent active learning sampling. Here we use the experimental design [Yu et al., 2006] to finish the operation of halving.

Considering a linear function $f(x) = \mathbf{w}^T x$ from measurements $y_i = \mathbf{w}^T x_i + \xi_i$, where $w \in \mathbb{R}$, and $\xi_i \sim \mathcal{N}(0, \sigma^2)$. The halving algorithm is to optimize a set $\mathbf{V} = \{(v_1, y_1), (v_2, y_2)$, ..., $(v_m, y_m)\}$ to represent x, where $m = \lfloor n/2 \rfloor$. Therefore, the maximum likelihood estimate of $\mathbf{w}$ is obtained by

$$\underset{\mathbf{w}^*}{\operatorname{argmin}} \left\{ \mathcal{J}(\mathbf{w}) = \sum_{i=1}^{n} (\mathbf{w}^T v_i - y_i) \right\} \tag{2.7}$$

and the error rate is $e = w - \mathbf{w}^*, \text{s.t. } \mu(e) = 0, D(e) = \sigma^2 \mathbf{C}_w$, where $\mu(\cdot)$ denotes the mean value of the input variable, $D(\cdot)$ denotes the covariance matrix of the input object, and

$$\mathbf{C}_w = \left(\frac{\partial^2 \mathcal{J}}{\partial \mathbf{w}\mathbf{w}^T} \right)^{-1} = (\mathbf{V}\mathbf{V}^T)^{-1}. \tag{2.8}$$

Then the average expected square predictive error over $\mathcal{X}$ can be written as

$$\mathrm{E}(y_i - w^* T x_i) = \sigma^2 + \sigma^2 \mathbf{Tr}(\mathcal{X}^T \mathbf{V}\mathbf{V}^T \mathcal{X}). \tag{2.9}$$

In order to minimize the average expected square predictive error, we need to minimize $\mathbf{Tr}(\mathcal{X}^T \mathbf{V}\mathbf{V}^T \mathcal{X})$. With mathematical derivations, the minimization issue changes into:

$$\begin{gathered} \underset{\mathbf{V},\mathbf{A}}{\operatorname{argmin}} \sum_{i=1}^{n} \|x_i - \mathbf{V}^T \alpha_i\| + \mu \|\alpha_i\|, \\ \mathbf{V} \subset \mathcal{X}, \mathbf{A} = [\alpha_1, \alpha_2, ..., \alpha_n], \end{gathered} \tag{2.10}$$

where u is the penalty factor of the global optimization.

After mapping the original input space into a non-linear kernel space, we iteratively project the top-($\lfloor n/2 \rfloor$) data with the highest confidence scores to a shattered space[*]. To

---

[*] Shattered space is a generalization from shattered distribution in real-world.

solve this equation, Yu et al. [2006] use sequential optimization to iteratively select the data with high representativeness in kernel space. In this paper, we follow their results and use the confidence score function to define the representativeness of one data:

$$\begin{aligned} &\text{Score}(x_i) = \frac{\|K(\kappa,:)K(:,\kappa)\|^2}{K(\kappa,\kappa)+u}, \forall i, \\ &\text{s.t. } K = K - \frac{K(:,\kappa')K(\kappa',:)}{K(\kappa',\kappa')+u}, \end{aligned} \tag{2.11}$$

where K denotes the kernel matrix of $\mathcal{X}$, κ denotes the sequence position of x_i in $\mathcal{X}$, and κ' denotes the sequence position of the data with the current highest confidence score in $\mathcal{X}$. Generally, sequential optimization costs a time calculation of $O(n^2)$ with a greedy strategy. For a large-scale data set, we can adopt the kernel relevant component analysis trick [Tsang et al., 2005a] to reduce the calculation complexity.

2.3.4.2 Splitting by Distribution Density

Implementing splitting in the input distribution by number density has already been proved effective in agnostic distributions (unknown assumptions). However, in $\hat{d}$-dimensional space, calculating the number density of a high dimensional-bounded space is challenging. To approximately generalize number density, we propose to use the exponential value of the distribution density to quickly split the input distribution due to their positive proportional relationship. Here we nearly generalize the number density as

$$\text{Den}(B_i) = \frac{1}{m_i} \sum_{x_j,x_l \in B_i} f^{\hat{d}}(x_j, x_l, h), \tag{2.12}$$

where $f^{\hat{d}}(\cdot)$ denotes the exponential value of the distribution density, $f(\cdot)$ can be generalized by arbitrary kernel function $\mathcal{K}(\cdot)$ with a bandwidth setting of $\mathcal{K}(\frac{x_j - x_l}{h})$, h denotes the kernel bandwidth, and m_i denotes the data number in B_i. Then, we propose the splitting rule:

$$\min \sum_{B_1,B_2,\ldots,B_k} \sum_{x_j,x_l \in B_i} \frac{1}{m_i} f^{\hat{d}}(x_j, x_l, h). \tag{2.13}$$

To solve the above minimum optimization problem, we use the (1+ε)-approximation [Tsang et al., 2005b] approach to increase the ball radius to make it converge, where ε is set by the empirical threshold.

2.3.4.3 Querying by SDAL

How to query unlabeled data is an important step for active learning tasks. In this section, we propose a Shattered Distribution-based Active Learning algorithm (SDAL) to implement the proposed distribution-shattering strategy by following the splitting rule in Section IV.B. The algorithm has two steps. Step 1 (Lines 2 to 10) is to find a shattered distribution which contains the optimal data sequences by the active scoring using Eq. (2.11). Step 2 (Line 11 to 25) is to solve the optimization of Eq. (2.13). Finally, the output data of the algorithm are used as the active learning queries.

Algorithm 1: SDAL algorithm

```
1  Input: dataset $\mathcal{X}$, radius $r$, approximation ratio $\varepsilon$, number of epochs $T$.
2  while $l = 1 < \lceil n/2 \rceil$ do
3      for $i=1,2,3...,n$ do
4          Calculate the score of $x_i$: $\Omega(i) = \frac{\|K(\kappa,:)K(:,\kappa)\|^2}{K(\kappa,\kappa)+u}$.
5      end
6      Find the sequence $\kappa'$ with the maximum value in $\Omega$: $\kappa' = \underset{i}{\mathrm{argmax}}\,\Omega(i)$.
7      Add $x_i$ to $\mathcal{X}^*$.
8      Update matrix $K = K - \frac{K(:,\kappa')K(\kappa',:)}{K(\kappa',\kappa')+u}$.
9      $l = l + 1$.
10 end
11 Initialize $k$ data points as the ball centers from $\mathcal{X}^*$ using $k$-means.
12 $f_0 = \sum_{B_1,B_2,...,B_k} \sum_{x_j,x_l \in B_i} \frac{1}{m_i} f^{\hat{d}}(x_j, x_l, h)$.
13 while $t = 1 \le T$ do
14     $f_t = \sum_{B_1,B_2,...,B_k} \sum_{x_j,x_l \in B_i} \frac{1}{m_i} f^{\hat{d}}(x_j, x_l, h)$
15     if $f_t - f_{t-1} \to 0 \,\Big|\Big|\, \|c_i - c_j\|_2 \le 2r, \exists i, j$ then
16         break;
17         else
18             Update ball centers $\{c_1, c_2, c_3, ..., c_k\}$, where $c_i = \frac{1}{m_i}\sum_{x_j \in B_i} x_j$.
19             Update ball radius $r = r(1 + \varepsilon)$.
20             Update $\{B_1, B_2, ..., B_k\}$ by new radius setting.
21         end
22     end
23     $t = t + 1$.
24 end
25 Update $c_i$ by their nearest neighbor in $B_i, \forall i < k$.
26 Output: $\{c_1, c_2, c_3, ..., c_k\}$.
```

The detailed process is as follows. Lines 2 to 10 iteratively halve the number density of input data set $\mathcal{X}$ by removing a half of the data. The remaining data $\mathcal{X}^*$ with high representativeness denote the data of shattered distribution of $\mathcal{X}$. It reduces the label complexity for the subsequent sampling. In the $(1+\varepsilon)$-approximation, Lines 11 and 12 firstly initialize k balls with the input radius setting. The approximation converges when the balls overlap or the splitting function stops updating (see Line 15). Otherwise, Lines 18 to 20 iteratively update the centers, balls, and radius. The code is released at GitHub [†].

2.4 Experiments

2.4.1 Experimental Setup

In this section, we investigate the halving and querying performance of the SDAL algorithm on three groups of experiments:

1. comparing the error rates of passive sampling in input and shattered spaces;
2. comparing the optimal error rates of different baselines;
3. comparing the average error rates of different baselines on six real-world datasets, where the datasets used in the querying tests have limited labels.

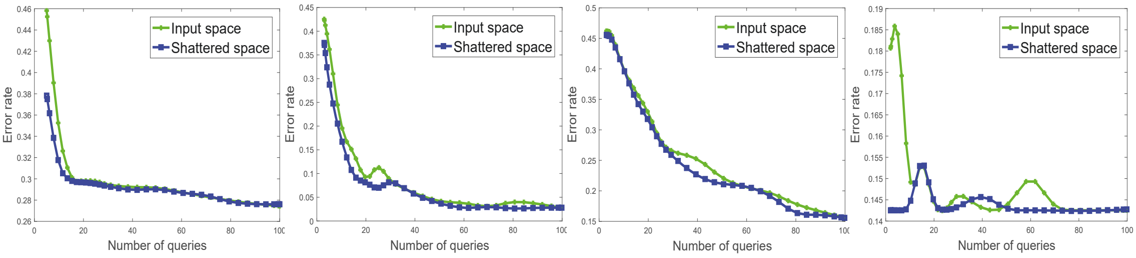
[†] https://github.com/XiaofengCao-MachineLearning/Shattering-Distribution-for-Active-Learning

To defend our theoretical insights on the performance disagreement of hypothesis-pruning and distribution-shattering strategies, we compare their error performance on querying with adversarial examples and noisy labels. In these experiments, the LIBSVM(3.22 version) [Chang and Lin, 2011] and convolutional neural network (CNN) are set as the default classification tools. The error rate and mean±std are used as evaluation standards, where error rate is over the entire input set.

There exists two main steps in SDAL algorithm: halving and splitting. In step 1, halving introduces the sequential optimization to score the representativeness of the data, which relates transductive experimental design (TED). In step 2, splitting uses (1+ε)-approximation to find a group of representative spheres, which is related to Hierarchical clustering-based active learning algorithm. We thus select these two approaches as our baselines. GEN is a comprehensive approach that introduces the representative measure in the process of estimating the hypothesis update. It is different with traditional estimation methods. Self-paced active learning is a generalization of hypothesis-pruning that estimates the hypothesis update with error loss and representativeness. Besides this, we present two generalizations of the k-means clustering approaches with different label estimation schemes. The details of these algorithms are described as follows.

- Hiera(Hierarchical clustering-based Active Learning): Dasgupta and Hsu [2008] utilize the prior knowledge of hierarchical clustering to actively annotate more unlabeled data by an established probability evaluation model, but it is sensitive to cluster structure.
- TED(Transductive Experimental Design): Yu et al. [2006] prefer data points that are not only hard to predict but also representative for the rest of the unlabeled pool. It is also called T-optimization.
- GEN(a GENeral active learning framework): Du et al. [2017] pay attention to the data which minimizes the difference between the distribution of the labeled and unlabeled sets.
- k-meansN: update the final k-means cluster centers into their Nearest neighbors and then queries the labels.
- k-meansA: estimate the label of each final k-means cluster center by rounding the Average label value of its cluster members.
- Self-Paced(Self-Paced active learning): Tang and Huang [2019] optimize the least squared loss and maximum mean discrepancy for finding an instance with informativeness and representativeness.
- SDAL(Shattered Distribution-based Active Learning algorithm): the proposed algorithm in this paper.

Note all features of the input data are rescaled into [0,1] before the experiments.



(a) german (b) iris (c) monk1 (d) vote

Figure 2.5 : Error rate changes of undertaking passive sampling in input and shattered spaces on different datasets.

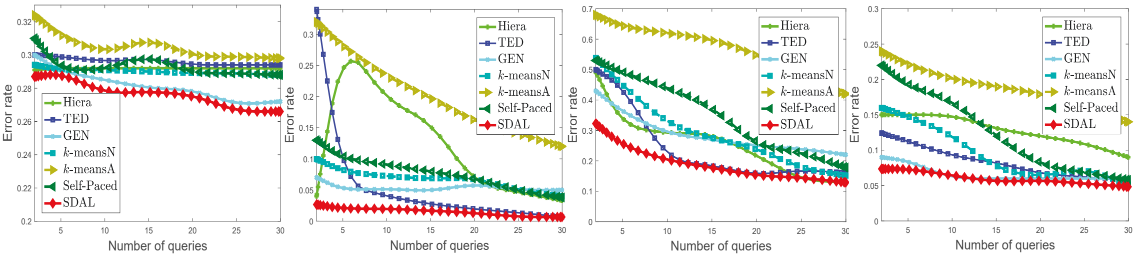
2.4.2 Effectiveness of Halving

To verify the halving ability of SDAL, we undertake passive sampling in input and shattered spaces to compare their prediction abilities over the input data. The tested datasets are four UCI real datasets: german (1,000 examples), iris (150 examples), monk1 (124 examples), and vote (435 examples). In the experimental process, we undertake passive sampling 10 times to obtain the mean error rate under different querying numbers in the two different spaces. Figure 2.5 presents the test results, where LIBSVM follows a parameter setting of [-c 1].

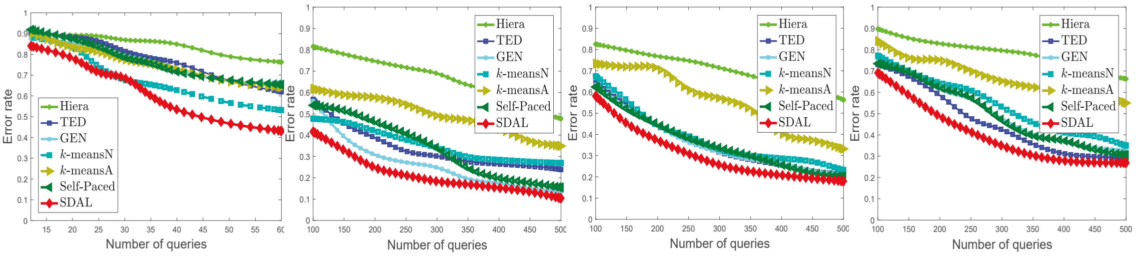
Shattering removes some "low informative points" deriving small influence to training model, thereby querying in a shattered space always has lower error rates than that of the original input space as learning curves in Figure 2.5. Assume that there exists p "highly informative points" that determine the final learning model in the input space, with a limited sampling budget k, do not consider the influences of the classifiers and parameter settings, the probabilities of obtaining a descried hypothesis in the two spaces are $\Pr(\mathcal{D}) = \frac{\binom{k}{p}}{\binom{k}{n}}$ and $\Pr(\mathcal{D}') = \frac{\binom{k}{p'}}{\binom{k}{n/2}}$, respectively, where p' denotes the number of the highly-informative points in the shattered space. If $p - p'$ is small enough, $\Pr(\mathcal{D}) < \Pr(\mathcal{D}')$ must hold.

2.4.3 Optimal Error of Querying

The experiments on halving have shown that the shattered space could have a better passive sampling performance compared to the original input space. It provides a guarantee for performing active learning querying by the distribution-shattering strategy in a shattered space. However, most of the active learning work require the supervision from a labeled set. To run these hypothesis-pruning algorithms in a warm start, we set the size of the initial training set as the class category via randomly selecting one datum from each class of the input datasets. Because these active learning algorithms always show negative performance when the start labeled set is insufficient, we minimize the influence of the labeled set by tunning their best parameters (related tunning is described in Section 2.4.4). Under different settings on the querying numbers, we collected the their optimal prediction results by initializing the start labeled set 100 times.



(a) german (b) iris (c) monk1 (d) vote



(e) A-T (f) A-L (g) A-P (h) A-Z

Figure 2.6 : The error rate performances of the seven active learning approaches on the active learning test. (a)-(d) are four UCI datasets; (e)-(k) are that on the selected sub datasets of *letter*, where the class number of them are 12, 16, 20, and 26.

Figure 2.6 presents the error rate curves of the seven active learning approaches on different tested datasets these being german, iris, monk1, vote, and four subsets of the *letter* data set. Note that A-T denotes the instances of letter A to T. The classifier toolbox is LIBSVM that follows a parameter setting of [-c 1]. Although we have maximized the model performance of the hypothesis-pruning active learning algorithms, the SDAL algorithm is still better than others in terms of optimal error.

To analyze the paradigm differences of these algorithms, we begin the discussions: (1) The idea of Hieral is active annotation. It depends on the cluster assumption from version space. Classification ability of it in unstructured datasets such as the subsets of *letter* thus is unstable. This makes the recorded error rates of Hieral be higher than that of other approaches, although we have increased the test number. Moreover, active annotation has a negative influence on the subsequent querying once the clustering result is not correct as its error rate curves in Figure 2.6(a). In other words, actively annotating the labels of a given budget have to undertake the positive or negative influences of pre-clustering. (2) TED tends to select those points with large norms, which might be hard to predict, but they do not best represent the whole data set. Also, the noises or low informative data are sampled in its querying process. So the reported classification results are good but not the best. (3) GEN always presents disappointing results at the beginning of training in all the tested datasets. Its error rate declines rapidly with the increase of the number of queries. The reason is that the established objective function prefers the data located at the center area of classes, which does not reflect the whole class structure well. (4) The performance of k-meansN is at middle level amongst all compared baselines because of the intuitive cluster structures of the tested datasets. While the error rate cannot decrease rapidly as other baselines. Besides this, the performance of k-meansA presents the worst performance of this group of experiments since averaging the labels of the cluster members cannot provide a correct estimation. (5) The performance of Self-Paced active learning optimizes the hypothesis update with a constrain on distribution representation. When the initial hypothesis is set improperly, the update will lead to an biased selection or random index. Thus, the performance of it is similar with GEN. (6) Compared to the above algorithms, the SDAL algorithm halves the number density of the data distribution into a shattered distribution, which removes most of the redundant points. The remaining points, which represent the local data distributions, help the learner to obtain the structure of the original data distribution. In the reported error rate curves, this represented structure shows effective sampling guidance when the number of queries is insufficient.

2.4.4 Average Error of Querying

The optima error of querying reflects the best sampling performance of different active learning algorithms. To tightly analyze their performance dependency on the initialized hypothesis (labeled set), this section presents their average error rates on three UCI datasets namely Phishing (11,055 examples), Satimage (4,435 examples), and one handwritten digit dataset MNIST (60,000 examples). [‡] Parameter settings are:1) vary the pruning budget of Hieral from 100 to 1000 with a step of 100; 2) kernel bandwidth parameter of TED is set as σ=1.8, then vary the kernel ridge regression λ from 0.01 to 1 with a step of 0.01; 3) vary the trade-off parameter of Self-Paced active learning from 1 to 1000 with a step of 10; 4) vary the paced learning parameter from 0.01 to 1 with a step of 0.01; 4) number of queries are set as the clustering number of k-meansA and k-meansN; 5) for SDAL, the used kernel in the sequential optimization is RBF, where the hyper parameter h is set as 1.8, and the hyper parameter μ is set as $10e$-4, we then vary the ball radii from 0.01 to 0.51 with a step of 0.05 and ε from 0.01 to 0.51 with a step of 0.05. To run Hieral, TED, GEN and Self-Paced algorithms, we respectively select one datum with label from each class of the six datasets as their initialized labeled sets. The classifier toolbox is LIBSVM with following parameter settings: 1) [-c 1 -g 25] for $N \leq 600$, [-c 1 -g 20] for $N > 600$ on Phishing, 2) [-c 1] on Satimage, 3) [-c 2] for $N \leq 300$ and [-c 4 -g 0.0015 -r 91.1 0.001] for $N > 300$ on MNIST, where N denotes the number of queries. The diver settings make the derived errors of active learning process decrease slowly but finally achieve the optimal; better observation on learning changes by adding perturbations of classifier. The mean and standard deviation (std) errors of the that algorithms on these datasets are reported in Table I with the results showing that SDAL significantly outperforms the others indicated in bold.

As shown in Table 2.1, (1) on all settings of the querying numbers, the SDAL algorithm achieves the lowest error rates over other baselines; (2) with the experience setting on parameters, all baselines achieve an average error below 0.5 after querying 300 data from the unlabeled data; (3) the SDAL algorithm produces significantly less errors when the numbers of querying are less than 600, benefiting from the representative structure of the input space; (4) for Hieral, TED, GEN and Self-Paced, the initial selection of the labeled set greatly affects their subsequent sampling; (5) on all settings, all algorithms obtain an average error below 0.3 after querying 600 data from the unlabeled data; (6) with

---

[‡] http://yann.lecun.com/exdb/mnist/

Table 2.1 : The statistical results (mean±std in %) of error of different active learning baselines on six real-world datasets

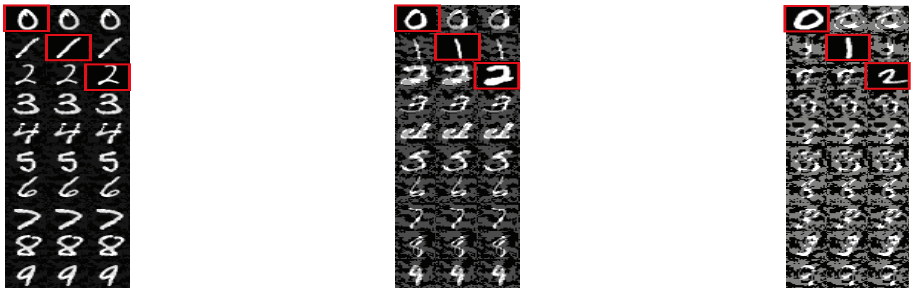
Datasets	Algorithms	Number of queries								
		100	200	300	400	500	600	700	800	900
Phishing	Hiera	49.6±3.0	45.0±7.2	42.5±2.3	38.5±1.7	33.2±2.1	22.6±1.4	19.3±1.2	18.0±1.1	14.6±0.7
	TED	39.0±1.9	39.1±0.9	34.9±0.3	34.1±0.6	31.2±0.7	28.5±0.5	27.9±0.5	18.6±0.5	13.8±0.8
	GEN	47.4±3.6	45.4±2.2	38.6±3.8	32.8±2.9	31.7±2.1	22.6±3.2	19.8±3.6	16.8±2.1	14.5±2.2
	k-meansN	37.0±0.1	**36.1±1.2**	34.9±0.1	32.1±0.1	30.2±0.5	27.5±0.0	25.9±0.4	16.6±0.3	14.8±0.7
	k-meansA	58.0±0.6	56.3±0.8	52.2±0.1	52.1±1.1	50.7±1.0	47.6±1.2	45.4±1.5	42.1±0.8	40.2±0.4
	Self-Paced	47.4±2.7	44.4±2.8	41.2±2.8	36.8±3.4	35.7±2.8	35.6±2.9	22.8±2.8	17.8±3.2	15.5±2.8
	SDAL	**36.5±1.7**	36.5±1.2	**30.4±2.8**	**30.0±2.8**	**27.4±2.2**	**19.1±1.7**	**16.5±1.7**	**12.6±1.8**	**11.2±2.1**
Satimage	Hiera	22.1±0.9	19.9±0.6	18.9±1.1	18.5±1.0	18.4±1.1	18.4±2.3	17.3±1.2	16.7±1.0	15.9±0.6
	TED	20.4±0.7	19.6±0.1	18.4±0.4	17.8±0.6	17.6±0.2	17.4±0.2	17.2±0.1	16.8±0.3	16.0±0.1
	GEN	21.9±4.0	20.1±2.7	18.5±0.5	18.4±0.3	18.1±1.2	18.0±1.5	17.8±0.7	16.5±0.9	16.4±2.0
	k-meansN	22.2±1.1	22.0±0.1	19.8±0.7	18.7±1.2	18.2±0.9	17.5±0.8	17.0±0.7	17.1±0.0	17.2±0.1
	k-meansA	34.2±1.0	32.0±1.1	30.8±1.2	28.5±1.2	26.3±1.1	25.4±0.0	22.8±0.8	19.6±1.0	19.2±0.9
	Self-Paced	24.7±2.8	23.1±2.7	20.5±0.5	18.2±0.3	17.3±1.2	17.8±1.5	17.1±0.7	16.4±0.9	16.2±2.0
	SDAL	**18.4±1.5**	**17.5±1.2**	**17.4±2.3**	**16.8±0.1**	**16.4±1.3**	**15.1±2.2**	**14.9±1.2**	**14.1±1.3**	**14.1±1.9**
MNIST	Hiera	51.2±2.7	46.0±1.7	37.3±2.4	21.3±2.8	20.1±2.3	11.9±2.2	9.6±1.5	9.3±1.3	9.0±1.0
	TED	63.3±1.2	40.7±2.3	21.5±3.2	21.7±0.8	**8.9±0.5**	8.3±0.9	8.3±0.2	8.2±0.5	7.8±0.6
	GEN	57.3±5.7	50.7±1.9	30.1±1.6	20.7±1.3	14.9±1.6	11.0±0.6	9.2±0.6	8.1±1.4	8.0±0.1
	k-meansN	65.7±0.7	52.7±1.3	32.4±1.2	29.8±0.2	16.4±0.3	12.5±0.2	11.6±0.1	10.7±0.1	7.8±0.4
	k-meansA	82.6±1.2	75.4±1.1	64.4±0.8	57.8±0.6	46.7±0.6	42.2±0.4	34.7±0.2	28.9±0.4	26.3±0.5
	Self-Paced	78.6±4.3	54.8±2.7	42.2±3.2	35.5±2.1	26.4±2.3	22.6±3.7	10.6±1.9	9.3±1.7	9.4±0.9
	SDAL	**44.2±2.4**	**37.0±2.8**	**19.8±3.8**	**11.5±1.9**	9.0±1.2	**8.1±0.9**	**7.9±0.8**	**7.7±0.6**	**7.6±0.3**

an increase of the querying percentages, the differences between each algorithm begin to narrow since the number of their overlapped data increases. Therefore, we conclude that our proposed SDAL algorithm, an approach derived from distribution-shattering strategy, breaks the curse of the initial hypothesis.

2.4.5 Querying with Adversarial Examples

In the machine learning community, the training models may misclassify the adversarial examples [Goodfellow et al., 2014] generated from the distribution of the correctly classified examples. The degradation of the performance in supervision training, caused by adversarial examples, is already not a mystery: the adversarial perturbation affects the precision of the features. In particular, the linear models are vulnerable to adversarial perturbation, such as regression and SVM models. In our study, the general hypothesis-pruning active learning strategies which need the support of the classifiers preferably pick up the adversarial examples. The underlying reason is that the adversarial examples make disagreement between the current and subsequent models more obvious than the examples without perturbation (clear data). Therefore, active querying with adversarial examples significantly describes the performance disagreement of hypothesis-pruning and distribution-shattering active learning strategies, and further defends our theoretical insights.

The experiments are tested on the MNIST dataset and we respectively generate 9,000 adversarial samples by the Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2014] attack method under different perturbation parameter ε: 0.1, 0.3, 0.5. For each parameter, such as ε= 0.1, we randomly choose 1,000 legitimate images from the MNIST test



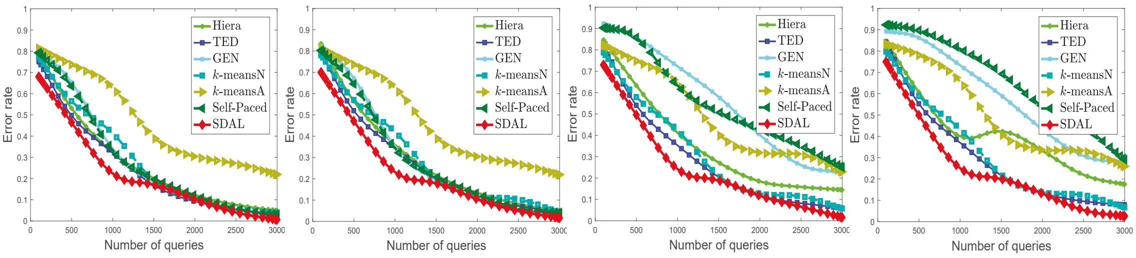
(a) ε=0.1 (b) ε=0.3 (c) ε=0.5

Figure 2.7 : Illustration of the produced adversarial examples by FGSM with different perturbation parameters, where the marked examples are clean data without feature perturbations.

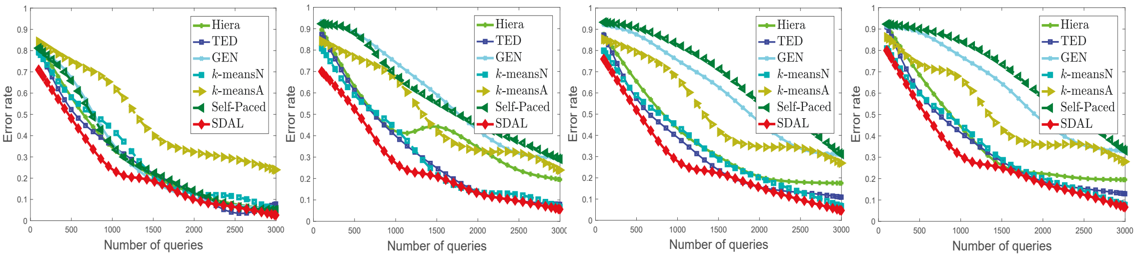
dataset, and 100 images for each class. For each image, we generate 9 adversarial samples with different labels. For example, for an image with label 0, we generate 9 adversarial samples with labels 1 to 9. Figure 2.7 presents a group of illustrations of adversarial examples, where each illustration marks three clean examples. To intuitively observe the influence of the adversarial examples in active learning querying, we use the 9,000 examples with ground truth labels as the unlabeled set of active learning and the 9,000 data with misclassified labels as the adversarial set. The features are extracted by the LeNet model and the classification model is CNN. To accelerate the experiments, we adjust the umber of epochs: 1) epoch=1 for N≤2000, 2) epoch=5 for 2000≤N<2500, 3) epoch=20 for N>2500, where N denotes the number of queries. This way defers the decrease of error rate that benefits the observation on the influences of subsequent perturbations from adversarial examples. Parameters of baselines follow their best tunning in Section 2.4.4.

In a dynamic view, we add a different number of adversarial examples to see the error change of different algorithms in the querying process. Before the querying test, we randomly select 20 data from the training set as the initial (start) labeled set for the hypothesis-pruning active learning algorithms including Hiera, TED, GEN and Self-Paced. Figure 2.8 draws the error rate change of predicting the labels of the entire training set under different settings on the perturbation parameter, number of added adversarial examples (N_{adv}), and the number of queries.

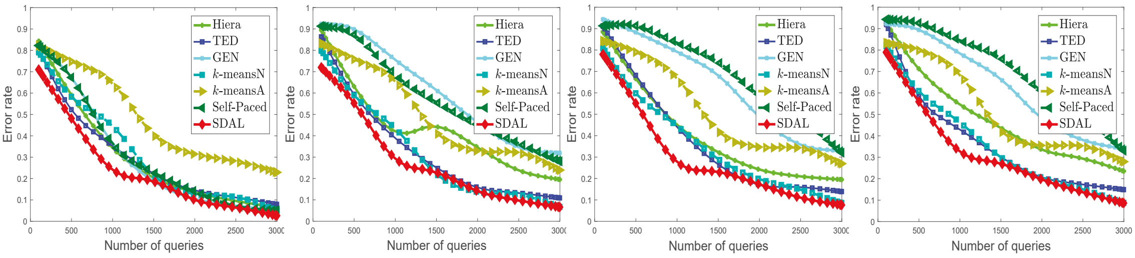
With the dynamic views on Figures 2.8(a) to 2.8(d), 2.8(e) to 2.8(h), and 2.8(i) to 2.8 (l), we find that the three hypothesis-pruning active learning algorithms significantly degenerate their error rates. Because the added adversarial examples misclassify the classifier using fraudulent labels, they significantly affect the update of the training model largely. By mixing more adversarial examples into the training set, the current training model has a greater chance to select the adversarial examples. However, our proposed SDAL algorithm which utilizes the distribution-shattering strategy is not sensitive to the classifier. Thus, its error rates only slightly reduce when querying the same number of unlabeled data, even adding more adversarial examples. In another view of setting different perturbation parameters, i.e., from the comparison of Figures $\{$2.8(b), 2.8(e), 2.8(i)$\}$,



(a) No-adversarial-examples (b) ε=0.1, N_{adv}=100 (c) ε=0.1, N_{adv}=500 (d) ε=0.1, N_{adv}=1000



(e) ε=0.3, N_{adv}=100 (f) ε=0.3, N_{adv}=500 (g) ε=0.3, N_{adv}=1000 (h) ε=0.3, N_{adv}=2000



(i) ε=0.5, N_{adv}=100 (j) ε=0.5, N_{adv}=500 (k) ε=0.5, N_{adv}=1000 (l) ε=0.5, N_{adv}=2000

Figure 2.8 : The performance of error rate on active learning querying with adversarial examples, where the adversarial examples are produced by FSGM with different perturbation parameter settings.

{2.8(c), 2.8(f), 2.8(j)}, {2.8(d), 2.8(g), 2.8(k)}, and {2.8(h), 2.8(l)}, we find the error rates of these hypothesis-pruning active learning algorithms also reduce significantly with an increase of ε.

To tighten the above analysis, Table 2.2 calculates the mean and std values of querying 1,000 legitimates when varying the number of adversarial examples with different ε. By observing the statistical results, we can clearly find that SDAL presents a slight change on error rate even when adding a different number of adversarial examples or setting different perturbation parameters. However, the estimation of hypothesis update on an adversarial example is highly-skewed than a clear example. It then leads to sensitive perturbations for GEN and Self-Paced algorithms. Moreover, the approaches involved with representative examples such as Hiera, TED, k-meansN, and k-meansA also present small perturbations.

Table 2.2 : The statistical results (mean±std in %) of active learning with adversarial examples of different algorithms

Parameter ε	Algorithm	Number of added adversarial examples (N_{adv})				
		0	100	500	1,000	2,000
0.1	Hiera	29.3±27.3	31.4±28.7	37.4±7.3	42.4±5.5	39.0±7.1
	TED	26.8±26.7	29.0±7.5	30.5±7.9	32.0 ±7.9	35.7±7.0
	GEN	30.2±31.6	31.3±10.3	56.6±8.1	58.5±6.7	60.1±6.0
	k-meansN	30.0±1.0	31.3±0.8	32.5±0.5	33.5±0.9	34.1±1.2
	k-meansA	47.9±1.4	31.3±0.7	49.7±0.6	51.6±1.0	53.1±1.3
	Self-Paced	30.6±28.5	31.3±8.7	56.3±9.3	66.0±8.3	70.1±7.2
	SDAL	23.6±24.6	24.5±6.2	26.0±6.6	26.7±6.9	28.9±5.9
0.3	Hiera	29.3±27.3	31.2±8.4	44.7±5.9	39.8±7.3	41.2±8.0
	TED	26.8±26.7	29.9 ±7.6	34.3 ±8.9	34.6±7.9	37.6±8.1
	GEN	30.2±31.6	32.0±9.8	59.8 ±7.1	61.2±6.5	62.3±6.2
	k-meansN	30.0±1.00	30.2±1.3	31.3±0.9	35.7±0.6	36.3±0.8
	k-meansA	47.9±1.4	49.2±1.2	31.3±0.6	52.7±0.7	52.5±1.1
	Self-Paced	30.6±28.5	59.5±12.5	67.6±9.3	58.5±8.9	68.7±11.4
	SDAL	23.6±24.6	25.4 ±6.2	28.0±5.9	29.2±6.5	31.9±6.7
0.5	Hiera	29.3±27.3	31.9±8.4	44.7±5.9	41.9±6.9	49.4±6.1
	TED	26.8±26.7	31.3±6.8	35.0±7.8	39.0±9.0	38.8±7.6
	GEN	30.2±31.6	32.0±9.8	61.6±6.5	63.8±6.4	63.8±6.0
	k-meansN	30.0±1.0	33.3±1.3	33.5±0.9	36.4±0.5	37.2±0.8
	k-meansA	47.9±1.4	49.9±1.2	50.7±0.7	52.6±0.6	53.2±1.1
	Self-Paced	30.6±28.5	33.2±12.5	59.0±9.3	68.8±8.9	70.1±9.7
	SDAL	23.6±24.6	25.4 ±6.2	29.6 ±7.2	30.9±6.7	33.3±6.3

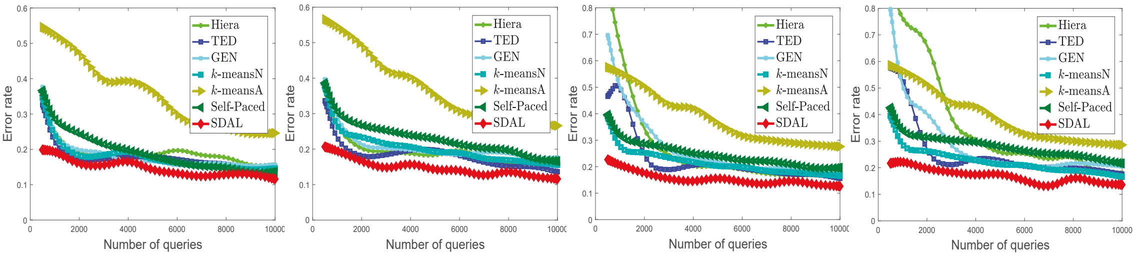
2.4.6 Querying with Noisy Labels

In many learning issues, the cost of obtaining the ground truth labels is expensive. A group of good annotation results on the unlabeled set is difficult to obtain due to manual error or simply a lack of precision of the original data [Natarajan et al., 2013]. This also makes the queried labels in active learning noisy. When hypothesis-pruning querying meets the noisy labels, these examples will generate an unprepared perturbation for the estimation of model change of a hypothesis-pruning active learning. Further, querying with noisy labels zooms the performance disagreement of the hypothesis-pruning and distribution-shattering active learning. Therefore, the experiment results can be a group of evidence to defend our theoretical insights.

We firstly collect the Fashion-Mnist dataset [§]. With a similar experiment setting, we respectively revise the original labels of the first 10, 500, 1000 data with noisy labels such as revising the label ‘0’ to ‘1’. Figure 2.9 describes the error rate change of adding a different number of noisy labels (N_{noi}), where the classifier also is a CNN model following Section 2.4.5, and parameters of baselines follow their best tunning in Section 2.4.4. In the drawn curves, the noisy examples have a negative influence on active learning querying since they may misclassify a lot of unlabeled data after adding them in to the labeled set. Thus, they are also picked up as the primary sampling objects in the estimation of the model chance policy of hypothesis-pruning active learning methods. However, the distribution-shattering approach avoids the perturbations. Only if the percentage of the noisy labels are large, the influence on the SDAL algorithm is obvious. Besides this, GEN shows a biased selection with the noisy setting. The noise perturbation to it is the

---

[§] https://github.com/zalandoresearch/fashion-mnist



(a) N_{noi}=0 (b) N_{noi}=100 (c) N_{noi}=500 (d) N_{noi}=1000

Figure 2.9 : The performance of error rate on active learning querying with noisy labels.

most sensitive among the compared baselines. The others keep clear perturbation but not so series as GEN. The inherent reason follows the analysis of Section 2.4.5.

2.4.7 Calculation Complexity

The proposed SDAL algorithm (Algorithm 1 on Page 26) has two steps: halving and splitting, where Lines 2-10 describe the halving process using Eq. (2.11), and Lines 11-24 split the shattered distribution into k geometrical balls using Eq. (2.13). Generally, the halving step costs a calculation complexity of $\mathcal{O}(n^3)$ and the splitting step costs a time complexity of $\mathcal{O}(nk)$. Therefore, the total calculation complexity of SDAL algorithm is $\mathcal{O}(n^3)$. For any generalized hypothesis-pruning algorithm, estimating the hypothesis update needs to retrain the classification models, which results an uncertain calculation complexity. For example, GEN and SPAL algorithms repeatedly train a SVM model to select the samples which can maximize the error update, in the experiments. Generally, sampling k data will retrain and repredict the classifier kn' times, where n' denotes the unlabeled data number that is usually close to n. Then, the calculation complexity is almost $\mathcal{O}(kn^3)$ to $\mathcal{O}(kn^4)$ since SVM costs a calculation complexity of $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$. In addition the two generalized k-means algorithms approximately cost $\mathcal{O}(kn)$. The TED approach costs $\mathcal{O}(n^2)$ due to a greedy selection. The Hierarchical clustering-based active learning costs $\mathcal{O}(n^3)$ due to the pre-clustering.

2.5 Discussions

Experiments of Section 2.4 have demonstrated that, the derived SDAL algorithm from distribution-shattering, achieved lower errors than the generalized hypothesis-pruning algorithms. SDAL also yields a shattered distribution, which is highly related to experimental design optimization [Yu et al., 2006]. We thereby begin to discuss their relationships.

2.5.1 Distribution-shattering and Experimental Design

Two experimental design active learning algorithms are compared to SDAL on three tested datasets of Section 2.4.3, i.e. Phishing, Satimage and MNIST.

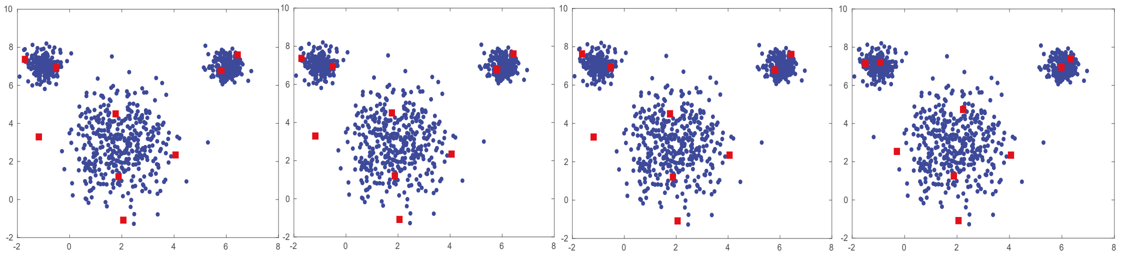
- MAED(Manifold Adaptive Experimental Design)[¶] [Cai and He, 2012]: perform

---

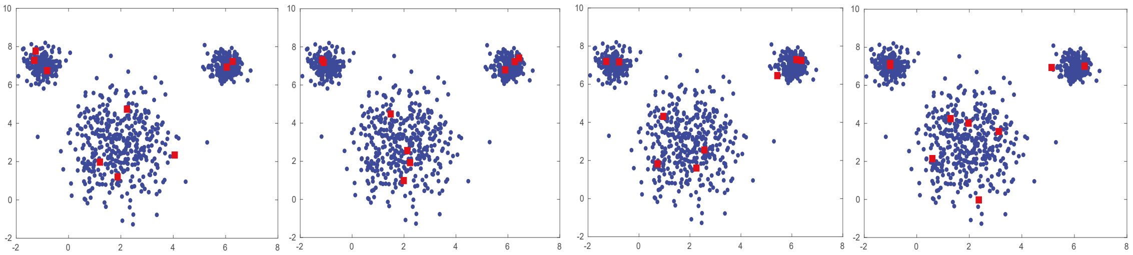
[¶] Code: http://www.cad.zju.edu.cn/home/dengcai/Data/data.html

Table 2.3 : The statistical results (mean±std in %) of error of experimental design active learning baselines and SDAL on three real-world datasets

Datasets	Algorithms	Number of queries								
		100	200	300	400	500	600	700	800	900
Phishing	MAED	44.3±0.0	44.3±0.1	38.2±0.2	32.3±0.0	**27.3±0.3**	**18.7±0.7**	17.0±0.6	12.9±0.2	11.8±0.1
	ALNR	42.8±2.7	37.5±1.9	33.9±1.6	33.1±1.5	30.2±1.2	27.4±1.1	22.9±1.4	14.3±1.1	12.1±1.3
	SDAL	**36.5±1.7**	**36.5±1.2**	**30.4±2.8**	**30.0±2.8**	27.4±2.2	19.1±1.7	**16.5±1.7**	**12.6±1.8**	**11.2±2.1**
Satimage	MAED	43.3±0.8	36.1±0.3	28.7±0.5	22.5±0.1	17.1±0.3	15.8±0.4	15.4±0.1	15.2±0.1	15.0±0.2
	ALNR	26.7±1.8	24.1±2.1	22.9±1.8	19.1±1.6	18.1±1.7	17.4±1.1	16.9±0.8	15.6±0.7	14.5±0.6
	SDAL	**18.4±1.5**	**17.5±1.2**	**17.4±2.3**	**16.8±0.1**	**16.4±1.3**	**15.1±2.2**	**14.9±1.2**	**14.1±1.3**	**14.1±1.9**
MNIST	MAED	48.4±2.4	41.9±1.3	28.5±4.7	18.5±2.7	16.1±1.8	15.4±1.6	14.5±1.3	14.2±1.1	14.0±0.8
	ALNR	50.1±3.2	42.1±2.9	31.9±0.3	21.1±2.6	12.2±2.7	11.5±1.5	10.8±2.1	10.0±2.2	9.6±2.8
	SDAL	**44.2±2.4**	**37.0±2.8**	**19.8±3.8**	**11.5±1.9**	**9.0±1.2**	**8.1±0.9**	**7.9±0.8**	**7.7±0.6**	**7.6±0.3**



(a) λ=0.001 (b) λ=0.01 (c) λ=0.1 (d) λ=1



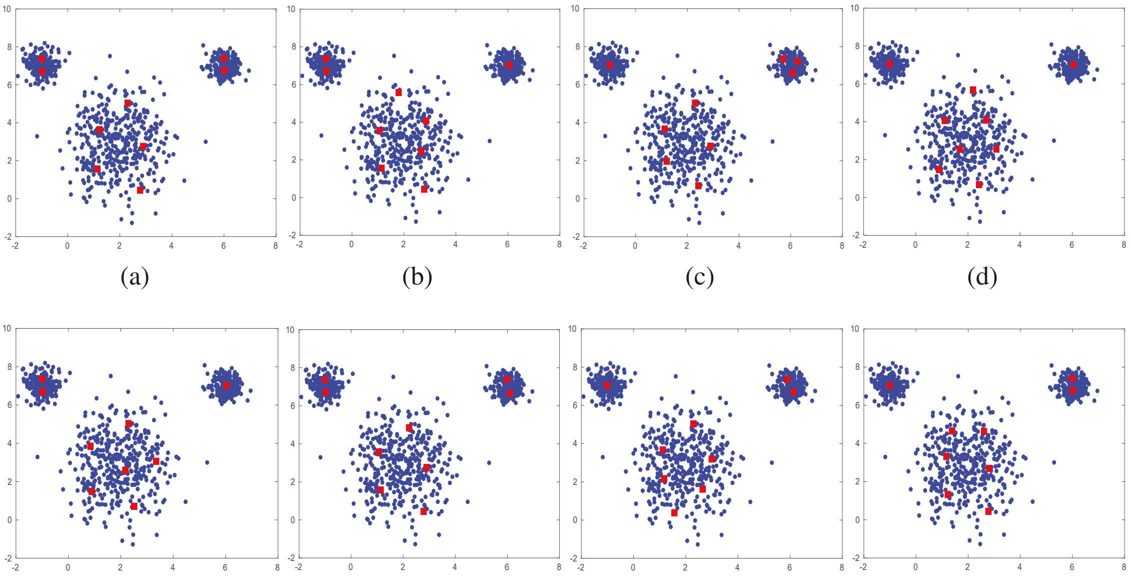
(e) λ=10 (f) λ=100 (g) λ=1000 (h) λ=10000

Figure 2.10 : Representative samples of MAED using different λ.

experimental design in the data manifold adaptive kernel space. MAED boils down to TED if the regularization parameter $\lambda = 0.1$. We vary λ from 0.001 to 0.101 with a step of 0.001 following the suggestion of [Cai and He, 2012].

- ALNR(Active Learning via Neighborhood Reconstruction) [Hu et al., 2013]: reconstruct the target (representative) data with its nearer neighbors optimized in experimental design process, not the linear combination of all the selected points as TED. Two trade-off parameters are induced: μ controls the locality and λ controls the sparsity. We vary μ from 0.4 to 0.8 with a step of 0.04, and λ from 0.1 to 1 with a step of 0.09 following the suggestion of [Hu et al., 2013].

Table 2.3 presents the statistical results of MAED, ALNR, and SDAL following the experimental settings of Section 2.4.4, where parameters of MAED and ALNR follow the



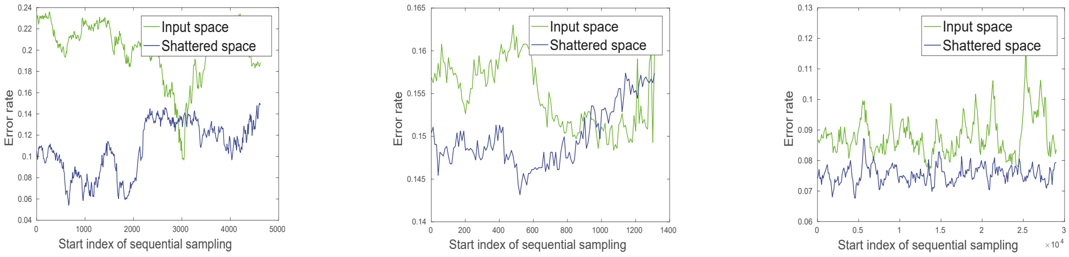
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2.11 : Representative samples of SDAL using different initializations ($r = 0.25$, $\varepsilon = 0.1$).

above settings expect the number of queris are 500, 600 on Phishing. The results show that SDAL still achieves lower average errors than that of MAED and ALNR. The main reason is that finding the representative data over the original input space may have a large probability to target low-informative/under-representative subregions, which makes the output representations degrade into a local optimum such as MAED on MNIST. Figure 2.10 presents the representative samples of MAED using different λ on a 2-D dataset. It intuitively shows that MEAD may degrade into local representations, which may lead to highly-nearby samples. ALNR also has similar local convergence conditions. Technically, MAED and ALNR combined the local representativeness into experimental design optimization. This improves the optimal solution of experimental designs, but also may increase the risk of falling back into local optima, where the individual drawn samples may be redundant. Moreover, MAED may need more experienced tunning on parameters.

In Figure 2.11, the representative samples of SDAL are spread over each subregion, covering the original distribution without any redundant samples, due to global shattering and local $(1+\varepsilon)$-approximation. Therefore, shattering the distribution into a shattered space can remove a part of low-informative/under-representative samples, reducing the chance to optimize representations in those regions. Theoretically, shattering can present a tighter upper bound on error as shown in the statements of Theorems 3 and 4 in label complexity analysis.



(a) Phishing (b) Satimage (c) MNIST

Figure 2.12 : Error rate changes of undertaking sequential sampling in the shattered space and original input space.

2.5.2 Learning Curves of Sequential Sampling in Shattered Space

We reveal why our shattering strategy can achieve significant performance in the above experiments. Figure 2.12 presents a group of learning curves of undertaking sequential sampling in shattered space and input space over the three datasets, where the sampling budget is 900, i.e. given a start index $\boldsymbol{p}$, sequential sampling continuously selects the data with an index collection of $\{\boldsymbol{p}, \boldsymbol{p}+1, ..., \boldsymbol{p}+899\}$. The shattering step adopts the halving step (lines 2 to 10) of SDAL algorithm.

As the figure shown, sampling in a shattered space has lower errors than sampling in its original input space. An inherent reason is that shattering has already removed a half of low-representative or redundant samples from the input space. The probability of achieving a desired error in shattered space is improved. This explains the effectiveness of our distribution-shattering strategy from the practical perspective.

2.6 Summary of This Chapter

Active learning algorithms provide strong theoretical guarantees on supervision sampling under fixed distribution and noise conditions. However, the label complexity bounds of the general hypothesis-pruning methods heavily depend on the initial hypothesis. This generates a challenging gap between the theoretical guarantee and application performance of active learning algorithms.

To bridge this gap, this chapter proposes a distribution-shattering strategy from a theoretical perspective of number density. With lower generalization error and label complexity in the shattered distribution, we implement the proposed theoretical strategy against an arbitrary distribution by the SDAL algorithm in real-world querying tasks. The empirical results demonstrate the effectiveness of the halving and querying abilities of SDAL algorithm. Moreover, the active querying with adversarial examples and noisy labels further demonstrate the performance disagreement of the hypothesis-pruning and distribution-shattering strategies. Based on these theoretical analyses, empirical evaluation, and experiment results, we conclude that the hypothesis-pruning active learning strategies degenerate their performance when querying with limited labels, adversarial examples, and noisy labels since they heavily depend on the initial labeled set and classifier. However, the proposed distribution-shattering strategy only presents slight perturbations in these querying scenarios.

Chapter 3

Distribution Disagreeing

This chapter discusses the second question of the thesis: "how to control hypothesis update when estimating the error disagreement is infeasible?". Usually, error disagreement-based active learning selects the data that maximally update the error of a classification hypothesis. However, poor human supervision (e.g. few labels, improper classifier parameters) may weaken or clutter this update; moreover, the computational cost of performing a greedy search to estimate the errors using a deep neural network is intolerable.

In this chapter, a novel disagreement coefficient (DDGC) based on distribution, not error, provides a tighter bound on label complexity, which further guarantees its generalization in hyperbolic space. The focal points derived from the squared Lorentzian distance, present more effective hyperbolic representations on aspherical distribution from geometry, replacing the typical Euclidean, kernelized, and Poincaré centroids. Experiments on different deep active learning tasks show that, the focal representation adopted in a tree-likeness splitting, significantly perform better than typical generalization baselines of centroid representations and error disagreement, dramatically accelerating the learning process. Our motivation on DDGC is presented at Figure 3.1.

The rest of this chapter is organized as follows. In Section 3.1, we establish a graph channel with DDGC for active learning. Section 3.2 begins by generalizing DDGC as Lorentzian focal representation and presents the focal approximation algorithm adopted in tree-likeness splitting. Experiments are presented in Section 3.3 and discussions are presented in Section 3.4. We conclude this chapter in Section 3.5.

3.1 Distribution Disagreement Graph Coefficient

3.1.1 Graph Coefficient

In this section, we generalize the disagreement of error on distribution via a graph channel and we call the new coefficient "distribution disagreement graph coefficient" (DDGC).

Our DDGC is developed based on an alternative assumption over a disagreement based on distribution, which is that the optimal subgraph of the source distribution tightly approximates the optimal hypothesis of $\mathcal{H}$. We then present some theoretical analysis and proofs for the label complexity based on this assumption. This section concludes with the establishment of a graph channel to define the distribution disagreement.

Suppose G is defined as $G = \{\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_{VC(\mathcal{H})}\}$, where $\mathcal{G}_i$ denotes one subgraph of G and $VC(\mathcal{H})$ denotes an upper bound on the Vapnik-Chervonenkis (VC) dimension

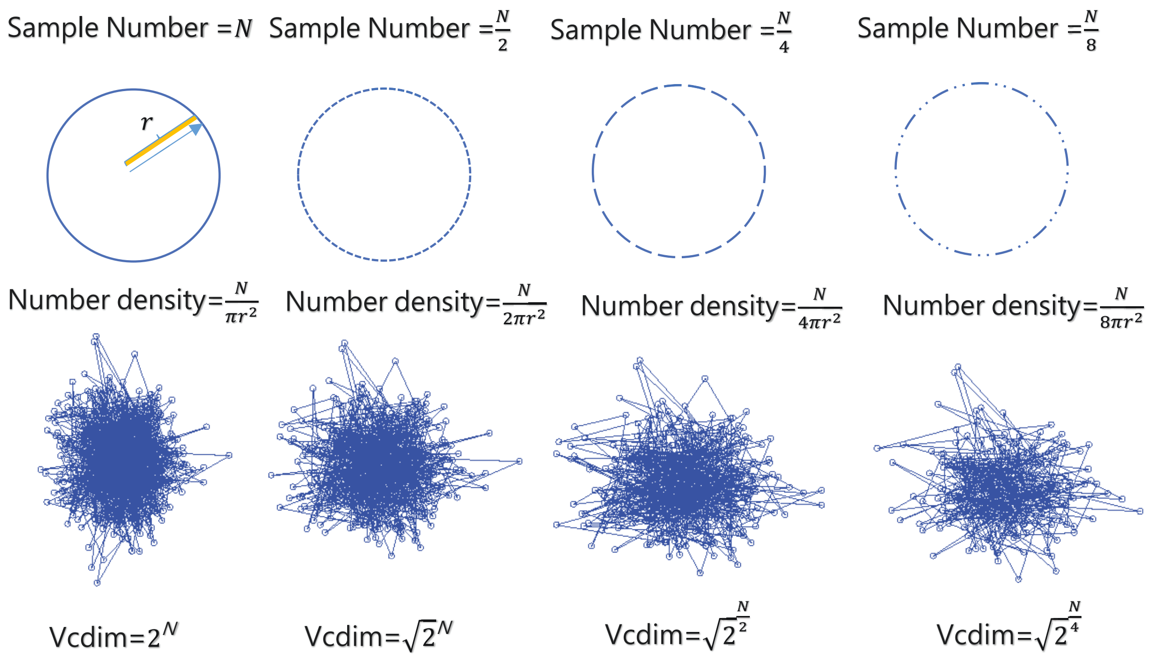


Figure 3.1 : Motivation of distribution disagreement graph coefficient over halfsapce learning. As the decrease of number density, the graphs covering all feasible hypotheses keep consistent structures, where each vertex denotes one hypothesis and each edge denotes the hypothesis disagreement (distance) of the two connected vertices. Learning a halfspace with graph coefficient can completely replace the error disagreement.

[Blumer et al., 1989] of $\mathcal{H}$. Our alternative assumption is to replace the error disagreement coefficient θ' with a measure of distribution disagreement.

Assumption 1. *Let h^* be uniquely associated with the optimal training set $\mathcal{X}^*$, $\mathcal{G}^*$ be the subgraph of G over $\mathcal{X}^*$'s hypothesis set, and $\mathfrak{f}(\cdot,\cdot)$ be a distance metric function against a node level in G. Assume that the expected error disagreement using the estimation of the loss $\ell(\cdot,\cdot)$ is uniquely approximated with the distribution graph metric $\mathfrak{f}(\cdot,\cdot)$, we hold*

$$\mathop{\mathbb{E}}_{h^*\in\mathcal{H}} \ell(h^*(x),y) := \mathop{\mathbb{E}}_{x\sim G, x'\sim\mathcal{G}^*}\Big[\mathfrak{f}(x,x')\Big], \tag{3.1}$$

where any graph $\mathcal{G}_i \in G$ over the hypothesis class $\mathcal{H}$ must have an inherent topology [Dasgupta, 2005; Cao and Tsang, 2020] on $\mathcal{X}$, thereby analyzing $\mathcal{G}_i$ can be observed from the data level of $\mathcal{X}$.

Specifically, the expected error disagreement $\mathop{\mathbb{E}}_{h^*\in\mathcal{H}} \ell(h^*(x),y)$ is a population risk [Jin et al., 2018] but derivable function that satisfies $\ell(h^*(x),y) \propto \mathfrak{f}(x,x')$, that is, there exists a dualistic function $\mathfrak{R}(h^*,\mathcal{G}^*) = \frac{\ell(h^*(x),y)}{\mathfrak{f}(x,x')}$, indicating there must exist an optimal subgraph $\mathcal{G}^* \in G$ deriving h^* from $\mathcal{H}$. To generalize Assumption 1, Section 3.1.2 then presents the empirical case study with regard to tightening the approximation of Theorem 5.

Besides Assumption 1, we still need an intuitive distribution metric over the graph level, not the node level as $\mathfrak{f}(\cdot,\cdot)$.

Definition 1. *For any hypothesis* $h' \in \mathcal{H}$ *that is uniquely associated with the graph* $\mathcal{G}' \in G$*, we define the distribution metric* $\mathfrak{L}(\cdot,\cdot)$ *for any* $\mathcal{G}$ *and* $\mathcal{G}'$*:*

$$\mathfrak{L}(\mathcal{G},\mathcal{G}') = \mathop{\mathbb{E}}_{x\sim\mathcal{G},x'\sim\mathcal{G}'}\Big[\mathfrak{f}(x,x')\Big], \tag{3.2}$$

where $\mathfrak{f}(x,x')$ *denotes a distribution metric over* x *and* x'*, i.e. a node level, and* $\mathfrak{L}(\mathcal{G},\mathcal{G}')$ *denotes a distribution metric over* G *and* G'*, i.e. a graph level.*

To specify our assumption, we connect the distribution metric w.r.t. Eq. (3.2) to the average loss of all-in-class errors w.r.t. $\rho(h,h')$ of Eq. (1.3), there coming with the following proposition.

Proposition 3. *For any hypothesis* $h' \in \mathcal{H}$ *that is uniquely associated with the graph* $\mathcal{G}' \in G$*, let* $\mathfrak{L}(\cdot,\cdot)$ *be the distribution metric for any* $\mathcal{G}$ *and* $\mathcal{G}'$*, we hold*

$$\mathfrak{L}(\mathcal{G},\mathcal{G}') = \mathop{\mathbb{E}}_{x\sim\mathcal{G},x'\sim\mathcal{G}'}\Big[\mathfrak{f}(x,x')\Big] := \rho(h,h'), \tag{3.3}$$

where $\rho(h,h')$ *is defined in Eq. (1.3), deriving* θ' *of Eq. (1.4).*

We denote $B(\mathcal{G}^*,r_G)$ as the ball with a radius of $r_G \geq 0$: $B(\mathcal{G}^*,r_G) = \{\mathcal{G}' \in G : \mathfrak{L}(\mathcal{G}^*,\mathcal{G}') \leq r_G, r_G \geq 0\}$. The new distribution disagreement can then be generalized as the minimum value of θ_G such that for any $r_G > 0$

$$\theta_G \geq \mathop{\mathbb{E}}_{\mathcal{G}'\in B(\mathcal{G}^*,r_G)}\Big[\max_{h\in B(\mathcal{G}^*,r_G)} \frac{\mathfrak{L}(\mathcal{G}^*,\mathcal{G}')}{r_G}\Big]. \tag{3.4}$$

This assumption provides a solution to supervise a learner who does not disclose any clues about its hypothesis class. Then, we come with the following IWAL scenario.

Considering that there exists such an IWAL [Beygelzimer et al., 2009a] scenario: we define a set of observations $\mathcal{F}$ on the sampling process, $\mathcal{F}_t = \{(x_1,y_1,p_1,Q_1),(x_2,y_2,p_2,Q_2)$, ..., $(x_t,y_t,p_t,Q_t)\}$ be the observations on the t-th sampling, where x_t be the sampled data in the t-th sampling with a probability of p_t such that

$$p_t = \max_{f,g\in\mathcal{H}_t} \mathcal{L}(f(x_t),g(x_t)), \tag{3.5}$$

$\mathcal{H}_t$ denotes the t-time hypothesis class $\mathcal{H}_t$, y_t be the label of x_t, and $Q_t = \{0,1\}$ be the parameters of Bernoulli distribution. Note g denotes another loss function in $\mathcal{H}_t$ that maps $\mathcal{X}$ into $\mathcal{Y}$. This means that, any selected sample x_t with $Q_t = 1$, will be assigned a weight $\frac{1}{p_t}$. Then, the weighted loss at t-time is defined as $\frac{1}{p_t}\ell(h(x_t),y_t)$. With this importance weighting skeleton, estimating the error disagreement θ is expressed as an online type, that is, for any t-time updated hypotheses $\{h_t,h_t'\}$, their weighted hypothesis disagreement of Eq. (1.1) is expressed as

$$\mathcal{L}(h_t(x),h_t'(x)) = \sum_{s=1}^{t} \frac{Q_t}{p_t}\Big|\max_{y_s\in\mathcal{Y}} \ell(h(x_s),y_s) - \ell(h'(x_s),y_t)\Big|. \tag{3.6}$$

The weighted hypothesis disagreement of Eq. (1.3) also follows this expression.

Proposition 4. *With Eq. (3.6), the expected label requesting probability at* t*-time, written as* $\underset{x\sim\mathcal{D}}{\mathbb{E}}[p_t|\mathcal{F}_{t-1}]$*, then can be inferred as* $\underset{x\sim\mathcal{D}}{\mathbb{E}}[\max_{f,g\in\mathcal{H}_t}\mathcal{L}(f(x_t),g(x_t))]$ *that satisfies*

$$\max_{f,g\in\mathcal{H}_t}\mathcal{L}(f(x_t),g(x_t))] \leq 2[\max_{h\in\mathcal{H}_t}\mathcal{L}(h(x),h^*(x))], \tag{3.7}$$

that is, the maximum hypothesis disagreement in $\mathcal{H}_t$ *yields within its diameter of the hypothesis class, where* $[\max_{h\in\mathcal{H}_t}\mathcal{L}(h(x),h^*(x))]$ *denotes the radius of* $\mathcal{H}_t$*, generalized from the hypothesis disagreement of the optimal hypothesis* h^* *and any* $h\in\mathcal{H}_t$*.*

IWAL assumes $Q_t = 1$ if p_t exists. It is thus $\underset{x\sim\mathcal{D}}{\mathbb{E}}[p_t|\mathcal{F}_{t-1}]$ reflects the label complexity in terms of hypothesis disagreement against an agnostic active learning sampling progress. Following this conclusion, we next use the distribution graph metric function to replace the error loss in the sampling process. With this alternative generalized loss function, we show that it can still converge with a more favorable label complexity, in terms of $\theta_G \leq \theta$.

Before presenting Theorem 5, we need a technical lemma about the importance-weighted empirical risk minimization on $\rho(h_t,h^*)$. The involved techniques refer to Corollary 4.2 of Langford et al. in Langford [2005], or Theorem 1 in of Sahyoun, C., et al Beygelzimer et al. [2009a].

Lemma 5. *Let* $R(h)$ *be the generalization expected loss (also called learning risk) that stipulates* $R(h) = \mathbb{E}_{x\sim D}[\ell(h(x),y)]$*, and* $R^* = R(h^*)$ *be its minimizer.* $\rho(h_t,h^*)$ *then can be bounded by* $\rho(h_t,h^*) \leq R(h_t) - R(h^*)$ *that stipulates* $\mathcal{H}_t := \{h\in\mathcal{H}_{t-1} : R(h_t) \leq R(h^*) + 2\Delta_{t-1}\}$*, where* Δ_{t-1} *adopts a form [Cortes et al., 2019a] of*

$$\frac{1}{t-1}\left[\sqrt{\left[\sum_{s=1}^{t-1}p_s\right]\log\left[\frac{(\mathrm{t}-1)|\mathcal{H}|}{\delta}\right]}+\log\left[\frac{(\mathrm{t}-1)|\mathcal{H}|}{\delta}\right]\right],$$

where $|\mathcal{H}|$ *denotes the number of hypothesis in* $\mathcal{H}$*, and* δ *denotes a probability threshold requiring* $\delta > 0$*. Since* $\sum_{s=1}^{t-1}p_s \leq t-1$*,* Δ_{t-1} *can then be bounded by*

$$\Delta_{t-1} = \sqrt{(\frac{2}{t-1})\log(2t(t-1)|)\frac{|\mathcal{H}|^2}{\delta})},$$

which denotes the loss disagreement bound to approximate a desired target hypothesis such that $R(h_t) - R(h^*) \leq 2\Delta_{t-1}$*.*

Theorem 5. *Let* $R(h)$ *be the generalization expected loss (also called learning risk) that stipulates* $R(h) = \mathbb{E}_{x\sim D}[\ell(h(x),y)]$*, and* $R^* = R(h^*)$ *be its minimizer. For any* $\delta > 0$*, with a probability of at least* $1-\delta$*, we have the following generalized distribution disagreement graph coefficient* θ_G *for all* t *that approximately satisfies:*

$$\theta \geq \theta_G \geq \frac{\underset{x\sim\mathcal{D}}{\mathbb{E}}[p_t|\mathcal{F}_{t-1}]}{4\Delta_{t-1}}, \tag{3.8}$$

where $\Delta_{t-1} = \sqrt{(\frac{2}{t-1})\log(2t(t-1)|)\frac{|\mathcal{H}|^2}{\delta})}$ *denotes the loss disagreement bound to approximate a desired target hypothesis such that* $R(h_t) - R(h^*) \leq 2\Delta_t$*, and* $|\mathcal{H}|$ *denotes the* *number of hypothesis in $\mathcal{H}$. Note that Δ_t comes from the sample complexity bound [Balcan et al., 2010] adopted in a PAC-style* [*]*. Related descriptions refer to Corollary 4.2 in Langford [2005], or Theorem 1 in Beygelzimer et al. [2009a].*

3.1.2 Tightness of Approximation

We study the tightness of the approximate inequality of Eq. (3.8) by generalizing the coefficients of error disagreement θ and distribution disagreement θ_G over practical active learning. Our analysis techniques follow the generalizations of $\mathcal{A}$-distance and $\mathcal{H}$-divergence in domain adaption theory [Ben-David et al., 2007], that is, specifying realizable variables and functions.

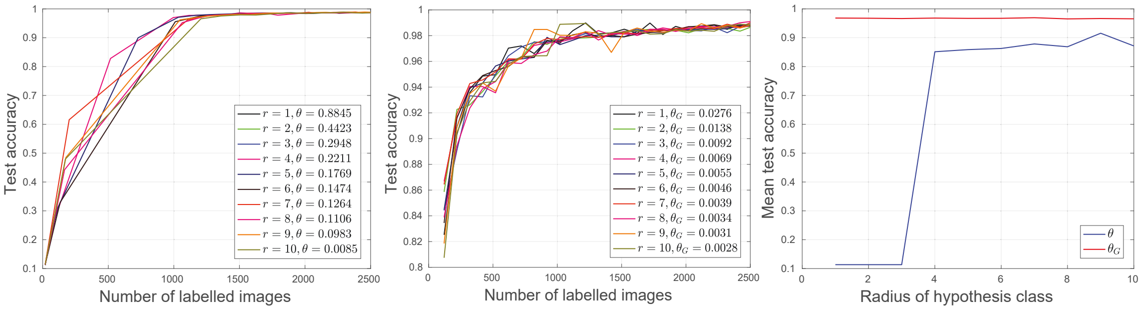
Datset Selection of Case Study. The empirical case study is used to observe the performance disagreement of the error disagreement (θ)-based active learning and our proposed distribution disagreement (θ_G)-based active learning. The policy of the dataset selection requires a nearly-zero learning risk, that is $R(h^*) \approx 0$. We consider three benchmark datasets usually used in deep active learning: MNIST, CIFAR-10, and CIFAR-100. Amongst them, MNIST is actually the simplest dataset but can derive a nearly zero learning risk: the best-in-class (w.r.t. Eq. (1.1)) classification accuracy trained by a convolutional neural network (CNN) is 0.9980 that stipulates the learning risk (w.r.t. error) of h^*: $R(h^*) = 1 - 0.9980 \approx 0$, where the optimal hypothesis h^* is consistent with the full training over the 60,000 training data. For CIFAR-10 and CIFAR-100, their learning risks are far away from a zero risk, which cannot be properly used in the specification of θ and θ_G. Therefore, soliciting MNIST as the case study is an optimal selection amongst the three benchmark datasets.

Case Study of Generalization. Here, we use MNIST as the dataset $\mathcal{X}$ for the case study of generalization, which has 60,000 training data and 10,000 test data. We use a convolutional neural network with one block of [convolution, dropout, max-pooling, relu], with 32, 3x3 convolution filters, 5x5 max pooling, and 0.5 dropout rate, as the classifier. To generalize the proposition of Eq. (3.3) on G and $\mathcal{G}^*$, the distribution disagreement $\mathfrak{L}(G, \mathcal{G}^*)$ is defined as the hypothesis disagreement $\rho(h, h^*)$, which is assumed to be tighter than the another disagreement metric of $\mathcal{L}(h, h^*)$. That is, $\mathfrak{L}(G, \mathcal{G}^*) := \rho(h, h^*) < \mathcal{L}(h, h^*)$. Here, we know the test accuracy $\alpha = 1 - \rho(h, h^*)$. To satisfy the above inequality, we have: $\rho(h, h^*) = 1 - \alpha < \mathcal{L}(h, h^*)$. Here, we set $\alpha = 0.9900$ in this case study.

Protocol of Case Study. After that, we begin the active learning on MNIST using θ and θ_G. For error disagreement-based active learning, the algorithm continuously selects those samples which can generate an error disagreement larger than θ. Specifically, the algorithm begins the sampling from x_1 to x_p until the error (risk) disagreement of $R(h) - R(h') \geq \theta$, where h' denotes the updated hypothesis after adding those p unlabeled data. The next sampling iteration begins from x_{p+1}. Note, the larger the value of θ, the smaller iteration steps the algorithm costs, the more coarse-grained the sampling process will be. When θ is large enough, the active learning process will degenerate into passive

---

[*] Probably approximately correct (PAC) learning [Haussler, 1990] requires the learner to receive samples and must select a generalization hypothesis from its hypothesis class. The goal is that, with high probability, the selected hypothesis will be approximately correct with low generalization error. In computational learning theory, IWAL is a typical framework of PAC learning.



(a) θ (b) θ_G (c) Mean accuracy

Figure 3.2 : Generalization test over MNIST dataset using θ and θ_G.

Table 3.1 : Mean±standard deviation of the breakpoints of the generalization test over MNIST using θ and θ_G

Radius of hypothesis class	Mean±standard accuracy	
	θ	θ_G
$r = 1$	0.1135±0.0000	0.9682±0.0360
$r = 2$	0.1135±0.0000	0.9676±0.0295
$r = 3$	0.1135±0.0000	0.9665±0.0341
$r = 4$	0.8514±0.3015	0.9681±0.0286
$r = 5$	0.8588±0.2955	0.9671±0.0317
$r = 6$	0.8628±0.2942	0.9673±0.0330
$r = 7$	0.8785±0.2733	0.9694±0.0279
$r = 8$	0.8687±0.2843	0.9653±0.0352
$r = 9$	0.9156±0.2181	0.9666±0.0373
$r = 10$	0.8716±0.2819	0.9656±0.0394

(random) sampling. For distribution disagreement-based active learning, sampling by θ_G is independent of the training model.

Generalization of Assumption 1. Based on Assumption 1, the expected loss over h^* satisfies $\mathbb{E}_{h^* \in \mathcal{H}} \ell(h^*(x), y) := \mathbb{E}_{x \sim G, x' \sim \mathcal{G}^*} \Big[\mathfrak{f}(x, x')\Big]$, and we then know $\rho(h, h^*) = 1 - \alpha \propto \mathfrak{L}(G, \mathcal{G}^*)$. With the specification of Assumption 1, $\mathfrak{R}(h^*, \mathcal{G}^*) = \frac{\ell(h^*(x), y)}{\mathfrak{f}(x, x')} \propto \frac{1 - \rho(h, h^*)}{1} \propto \frac{\mathbb{E}_{x \sim \mathcal{X}, x' \sim \mathcal{X}'_t} (\|x - x'\|_2)}{\mathfrak{L}(G, \mathcal{G}^*))}$, that is, the ratio of the accuracy '$1 - \alpha$' to the optimal accuracy '1' approximates the ratio of the distribution disagreement of $\mathcal{X}'_t$ i.e. $\mathbb{E}_{x \sim \mathcal{X}, x' \sim \mathcal{X}'_t} (\|x - x'\|_2)$ to the optimal distribution disagreement $\mathfrak{L}(G, \mathcal{G}^*)$. Given a budget $B = 100$, each iterative sampling seeks the best subset $\mathcal{X}'_t$ with B data to minimize the distribution disagreement: $\min_{\mathcal{X}'_t \subset \mathcal{X}} \Big| \mathbb{E}_{x \sim \mathcal{X}, x' \sim \mathcal{X}'_t} \Big[(\|x - x'\|_2 / \mathfrak{L}(G, \mathcal{G}^*)) * \alpha - \theta_G \Big] \Big|$. However, there is randomness dur-

Table 3.2 : Label complexities of the generalization test over MNIST using θ and θ_G (Radius denotes “radius of hypothesis class”)

Radius	Accuracy threshold			
	80%	85%	90%	95%
	θ, θ_G	θ, θ_G	θ, θ_G	θ, θ_G
$r = 1$	infeasible, [20,120]	infeasible, [120,220]	infeasible, 220	infeasible, [520,620]
$r = 2$	infeasible, [20,120]	infeasible,120	infeasible, [120,220]	infeasible, [520,620]
$r = 3$	infeasible, [20,120]	infeasible,120	infeasible, [220,320]	infeasible, 520
$r = 4$	[73,517], [20,120]	[517,995],[20,120]	[517,995],[20,120]	[517,995], [520,620]
$r = 5$	[112,727], [20,120]	[112,727],[20,120]	[727,1037],220	[727,1037],520
$r = 6$	[133,1011], [20,120]	[133,1011],[120,220]	[133,1011],[120,220]	1011,520
$r = 7$	[202,1104], [20,120]	[112,727], [20,120]	[112,727], [120,220]	1104,520
$r = 8$	[167,1064], [20,120]	[167,1064],[120,220]	[167,1064], [220,320]	1064,620
$r = 9$	[177,1046], [20,120]	[177,1046],[120,220]	[177,1046], 220	1046, 620
$r = 10$	[173,1206], 120	[173,1206],[120,220]	[173,1206], 220	1206, 620

ing the sampling process. A brute-force method is employed to search the minimizer by randomly sampling 100,000 times. The experiment process stops until 2,500 data are imported to the neural network (2500 is the sampling budget). Therefore, the number of the AL loops of error disagreement is decided by the value of θ; distribution disagreement has a fixed learning loop of $2,500/B = 25$. The details of specification of θ and θ_G can be found in Appendix C.

Generalization Results. As the reported experiment results in Figure 3.2, active learning using θ_G achieves higher test accuracies than that of θ on any of the same number of labelled images. Table 3.1 reports the mean±standard deviation of the breakpoints of the learning curves over the generalization test. Together the analysis with Figure 3.2(c), sampling using θ_G achieves more stable and higher accuracies than that of θ. Table 3.2 reports the label complexities of achieving a desired accuracy of 80%, 85%, 90%, and 95%, respectively. The results show θ_G costs much lower than θ. It consistently verifies Theorem 5 here. Moreover, the error disagreement determines how many active learning loops will perform against a given sampling budget. As a result, the error disagreement-based active learning becomes a multiple-step iterative algorithm; while the distribution disagreement drives the active learning to converge at one-step.

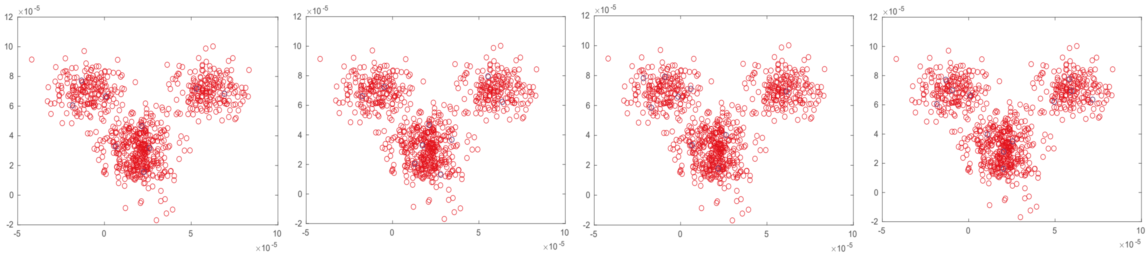
Remark on Case Study. Based on above generalization test results, the approximate inequality of our main theoretical result, i.e. Eq. (3.8), is tight. We thus begin to generalize θ_G into a proper function that induces a highly-representative subgraph over the distribution.

3.1.3 Why Hyperbolic Geometry?

Guaranteed from the distribution disagreement-based active learning, our implementation of active learning sampling with Lorentzian representation that starts by relaxing the hypothesis class in hyperbolic space. Then, the focal expression based on Lorentzian

Table 3.3 : Summary of distance metrics and centroid expressions in the Euclidean space, RKHS, and hyperbolic space.

	Euclidean Space	**Reproducing Kernel Hilbert Space**	**Hyperbolic Space**
Metrics	$d_{\mathcal{R}}(x_i, x_j) = \lVert x_i - x_j \rVert_2$ $d^2_{\mathcal{R}}(x_i, x_j) = \lVert x_i - x_j \rVert_2^2$	$d_H(x_i, x_j) = \exp\Big(- \frac{\lVert x_i - x_j \rVert^2}{2\sigma^2} \Big)$	$d_{\mathcal{P}}(x_i, x_j) = \mathrm{arccosh}\Big(1 + 2\frac{\lVert x_i - x_j \rVert^2}{(1-\lVert x_i \rVert^2)(1-\lVert x_j \rVert^2)}\Big)$ $d^2_{\mathcal{L}}(x_i, x_j) = -2\mathcal{B} - 2\langle x_i, x_j \rangle_{\mathcal{L}}$
Centroid	$\mathrm{argmin}_{\mu \in \mathbb{R}^d} \sum_{i=1}^n w_i \lVert x_i - \mu \rVert_2$ $\mathrm{argmin}_{\mu \in \mathbb{R}^d} \sum_{i=1}^n w_i \lVert x_i - \mu \rVert_2^2$	$\mathrm{argmax}_{\mu \in \mathbb{R}^d} \sum_{i=1}^n w_i \exp\Big(- \frac{\lVert x_i - \mu \rVert^2}{2\sigma^2} \Big)$	$\mathrm{argmin}_{\mu \in \mathbb{R}^d} \sum_{i=1}^n w_i \mathrm{arccosh}\Big(1 + 2\frac{\lVert x_i - \mu \rVert^2}{(1-\lVert x_i \rVert^2)(1-\lVert \mu \rVert^2)}\Big)$ $\mathrm{argmin}_{\mu \in \mathbb{R}^d} \sum_{i=1}^n w_i \Big(- 2\mathcal{B} - 2\langle x_i, \mu \rangle_{\mathcal{L}} \Big)$



(a) Euclidean centroid; Test accuracy=0.9175

(b) Gaussian Kernelized centroids; Test accuracy=0.9225

(c) Poincaré centroids; Test accuracy=0.8275

(d) Lorentzian centroids, Test accuracy=0.98125

Figure 3.3 : Euclidean, Gaussian kernelized, Poincaré and Lorentzian centroids on a noisy spherical Gaussian dataset. The minimization or maximization on the centroids are performed with 30 iterations against k-medoids algorithm.

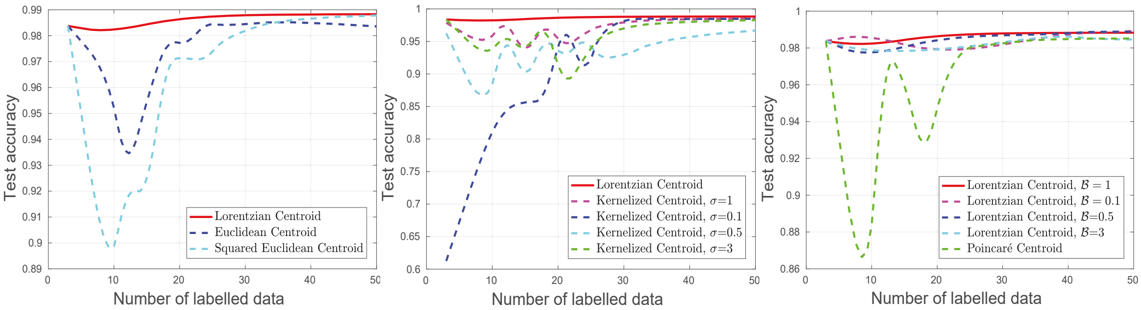
centroid is introduced for aspherical distributions.

We consider three typical geometric structures: Euclidean space, reproducing kernel Hilbert space (RKHS), and hyperbolic space, where Hilbert space is a generalization of Euclidean space with any finite or infinite dimensions [Young, 1988; Quang et al., 2014]. By synthesizing a Gaussian-mixture dataset with varying densities [Bratieres et al., 2014; Ertöz et al., 2003], we present a case study to explore the characteristics of these representation spaces on spherical distributions.

Figure 3.3 shows a dataset with three Gaussian clusters, where the noises around the boundary connect them and each dimension of the points is synthesized with a unit of 10^{-5}. The selected metrics are Euclidean, squared Euclidean, Gaussian kernel, and Poincaré distances, which further derive their centroid expressions based on Proposition 5.

Proposition 5. *Given a point set* $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ *and* $x_i \in \mathbb{R}^d, \forall i$. *A centroid of* $\mu \in \mathbb{R}^d$ *that minimizes the following problem* $\min_{\mu \in \mathbb{R}^d} \sum_{i=1}^n w_i d_{\mathcal{X}}(x_i, \mu)$, *where* $d_{\mathcal{X}}$ *denotes the metric over* $\mathcal{X}$, w_i *denotes the weight coefficient of* x_i *and* $w_i > 0$.

Table 3.3 presents the detailed expressions of the centroids using different distance metrics derived from Proposition 5, where $d_{\mathcal{R}}(\cdot,\cdot)$ denotes the Euclidean distance, $d^2_{\mathcal{R}}(\cdot,\cdot)$



(a) Lorentzian Centroids vs. Euclidean Centroids (b) Lorentzian Centroids vs. Gaussian Kernelized Centroids (c) Lorentzian Centroids vs. Poincaré Centroids

Figure 3.4 : Test accuracies of Euclidean, squared Euclidean, Gaussian kernelized, Poincaré and Lorentzian centroids with varying parameters.

denotes squared Euclidean distance, $d_H(\cdot,\cdot)$ denotes the Gaussian kernel distance, $d_{\mathcal{P}}(\cdot,\cdot)$ denotes the Poincaré distance, and $d_{\mathcal{L}}^2(\cdot,\cdot)$ denotes the squared Lorentzian distance w.r.t. Eq. (3.10). On their centroid expressions, we set $w_i = 1$ for any i to do unbiased estimation (see the third row of Table 3.3). The minimization or maximization of the centroids on those distance metrics need to adopt a gradient solver, which cannot guarantee consistent convergence conditions for them. To fairly compare these metrics, we revise the constraint of $\mu \in \mathbb{R}$ into $\mu \in \mathcal{X}$, yielding alternative expressions over the data. With this approximation method, the k-medoids algorithm is employed to estimate the centroids of different distance metrics.

Figures 3.3(a), 3.3(b), and 3.3(c) estimate nine approximate centroids using k-medoids algorithm with different geometric distance metrics. The Lorentzian centroids shown in Figure 3.3(d) are derived by the squared Lorentzian distance and optimized in a hierarchical way, that is, iteratively performing k-medoids in the clusters of the last clustering. The predictive accuracy on those centroids are made by a SVM [Chang and Lin, 2011] classifier[†]. It is interesting that the Lorentzian centroids are first identified on three points distributed in the central regions of the three spherical clusters. These three points further enforce the subsequent centroids to update towards the cluster boundaries, and so better classification results are manifested. While other centroids derived from Euclidean, squared Euclidean, and Gaussian kernel distances, are distributed spread across each subregion but not uniformly, in which some of them are a bit far away from the decision boundaries between the clusters.

Figure 3.4 presents the test accuracies of training different numbers of centroids by the SVM model, where the Gaussian kernel and Lorentzian distances are performed with different parameters. The results show that the squared Lorentzian distance has much slighter parameter perturbations than other metrics in terms of the test accuracies. However, the Gaussian kernel function shows sensitive parameter perturbations to the accuracies. The Euclidean, squared Euclidean, and Poincaré distances keep consistent results, performing in an unsupervised way, that show a bit lower accuracies than the Lorentzian

---

[†] LIBSVM toolbox with its default hyperparameters is used for the SVM.

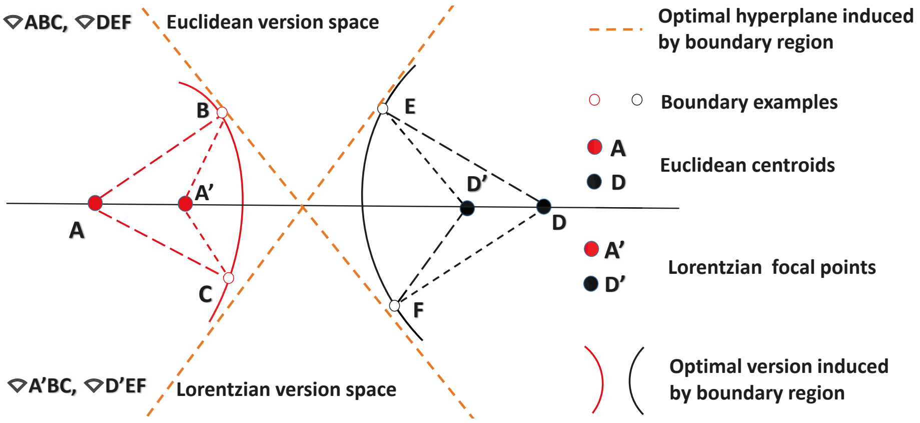


Figure 3.5 : The Lorentzian version space ($\triangledown A'BC$, $\triangledown D'EF$) vs. Euclidean version space ($\triangledown ABC$, $\triangledown DEF$). Lorentzian norm shrinks the volume of Euclidean version space by shifting Euclidean centroids (A,D) into Lorentzian focal points (A′, D′) that are close to the boundary region. As such, sampling from regions $ABA'C$ and $DED'F$ is ineffective, which means any active learning model in the Lorentzian version space has a tighter bound on label complexity.

centroids derived from a hierarchical way. Therefore, we are motivated to generalize the distribution disagreement in hyperbolic geometry and hierarchical splitting is considered to optimize those representation points.

3.2 Hyperbolic Focal Representation

With the guarantee from the distribution disagreement coefficient, our implementation of active learning sampling with the Lorentzian focal representation that starts by relaxing the hypothesis class in hyperbolic space. Then, the focal expression based on Lorentzian centroid is introduced for aspherical distributions. Our motivation is presented in Figure 3.5.

3.2.1 Squared Lorentzian Distance

The Lorentzian distance function $d_{\mathcal{L}}$ [Nickel and Kiela, 2018] has been proved to be an expressive tool for embedding the hyperbolic representation. Law et al. [2019] further proved that the squared Lorentzian distance $d_{\mathcal{L}}^2$, coupled with an upper bound $\mathcal{B}$ on the Lorentzian inner product, can effectively shift the centroid position of a given set of points, i.e. adjusting $\mathcal{B}$ can offset the centers of the samples in an enclosed data space. Therefore, following [Law et al., 2019], we generalize DDGC as the squared Lorentzian distance.

Beginning with the definition of $d_{\mathcal{L}}$ as background, let $\mathbf{u}, \mathbf{v}$ be any two vectors in a d-dimensional hyperboloid model $\mathcal{H}^{d,\mathcal{B}} \subseteq \mathbb{R}^{d+1}$ that contains embedded Lorentzian norms, the Lorentzian distance metric is defined as follows:

$$d_{\mathcal{L}}(\mathbf{u},\mathbf{v}) := \langle \mathbf{u},\mathbf{v}\rangle_{\mathcal{L}} = -u_0 v_0 + \sum_{i=1}^{d} u_i v_i \le -\mathcal{B}, \tag{3.9}$$

where $\mathbf{u} = [u_0, u_1, ..., u_d]$ and $\mathbf{v} = [v_0, v_1, ..., v_d]$, $\mathcal{H}^{d,\mathcal{B}} := \{\mathbf{u} = (u_0, u_1, ..., u_d) \in \mathbb{R}^{d+1}, \text{ s.t. } \langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{L}} = -\mathcal{B}, u_0 > 0, \mathcal{B} > 0\}$. With this definition, we know $\|\mathbf{u}\|^2_{\mathcal{L}} = -\mathcal{B}$ and $u_0 = \sqrt{\mathcal{B} + \sum_{i=1}^{d} u_i^2}$. The squared type of Lorentzian distance then can be written as:

$$d^2_{\mathcal{L}}(\mathbf{u}, \mathbf{v}) := \|\mathbf{u} - \mathbf{v}\|^2_{\mathcal{L}} = -2\mathcal{B} - 2 \langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{L}} . \tag{3.10}$$

Compared to the negative characteristics of the Lorentzian distance, its squared type produces a more nature metric over any Euclidean vectors due to $d^2_{\mathcal{L}}(\mathbf{u}, \mathbf{v}) \geq 0, \forall \mathbf{u}, \mathbf{v}$. In a high dimensional hyperboloid model, the squared Lorentzian distance further avoids the numerical instabilities[‡] and exploding gradients [§] [Law et al., 2019; Kanai et al., 2017].

3.2.2 Geometric Centroids

Geometric centroid [Tsang et al., 2005b] is an important concept in feature representation. We next compare the geometric centroid formulations of $d_{\mathcal{R}}$, $d_{\mathcal{P}}$ in a given set of points, where $d_{\mathcal{R}}$ denotes the Euclidean distance and $d_{\mathcal{P}}$ denotes the Poincaré distance [Nickel and Kiela, 2017], another common function in hyperbolic space.

Theorem 6 firstly presents the centroid formulation of $d_{\mathcal{R}}$ in Euclidean space.

Theorem 6. *Given a set of points* $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ *and* $x_i \in \mathbb{R}^d, \forall i$. *A centroid of* $\mu \in \mathbb{R}^d$ *maximizes the following problem* $\max_{\mu \in \mathbb{R}^d} \sum_{i=1}^{n} w_i \langle x_i, \mu \rangle_{\mathcal{R}}$, *where* w_i *is the weight coefficient of* x_i *and* $w_i > 0$. *With the inner product constraint of* $\langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\mathcal{R}} \leq \mathcal{B}$ *in* $\mathbb{R}^d$, *the centroid* μ *is formulated as:*

$$\mu = \sqrt{\mathcal{B}} \frac{\sum_{i=1}^{n} w_i x_i}{\| \sum_{i=1}^{n} w_i x_i \|_{\mathcal{R}}}. \tag{3.11}$$

We next discuss the geometric centroid of a Poincaré ball $\mathcal{P}^d$ in hyperbolic space.

Theorem 7. *Given a point set* $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ *and* $x_i \in \mathcal{P}^d, \forall i$. *A centroid of* $\mu \in \mathcal{P}^d$ *that minimizes the following problem* $\min_{\mu \in \mathcal{P}^d} \sum_{i=1}^{n} w_i d_{\mathcal{P}}(x_i, \mu)$, *where* w_i *is the weight coefficient of* x_i *and* $w_i > 0$. *The Poincaré centroid* μ *has no closed-form solution.*

Because the centroid formulation based on a Poincaré norm cannot be written in closed-form, a further claim is needed as follows:

Claim 1. *To produce an alternative expression for Poincaré centroid over the data,* $\min_{\mu \in \mathcal{X}} \sum_{i=1}^{n} w_i d_{\mathcal{P}}(x_i, \mu)$ *is a feasible scheme via replacing* $\mu \in \mathcal{P}^d$ *with* $\mu \in \mathcal{X}$.

In our investigation, there exists manifold gradient solvers to obtain an approximate centroid under finite learning loops e.g. exponential and logarithmic mapping [Lou et al., 2020b]. We do not introduce these optimization tricks since different distance metrics cannot keep consistent convergence conditions in seeking their centroids.

[‡] In a numerically instable algorithm, errors from irrational inputs cause a considerably more significant mistakes in the final outputs.

[§] Large error gradients accelerate the updates of the model weights during the training, possibly resulting larger errors.

3.2.3 Lorentzian Focal Point Approximation

Lorentzian centroids are not well representative for the aspherical distributions. We thus introduce its focal expression.

Our formulation of an approximate focal point of a hyperboloid model $\mathcal{H}^{d,\mathcal{B}}$ based on Lorentzian centroid is outlined below.

Proposition 6. *Given a set of points* $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ *and* $x_i \in \mathcal{H}^{d,\mathcal{B}}, \forall i$. *A center of* $\mu \in \mathcal{H}^{d,\mathcal{B}}$ *that maximizes the following problem* $\max_{u \in \mathcal{H}^{d,\mathcal{B}}} \sum_{i=1}^{n} w_i \langle x_i, \mu \rangle_{\mathcal{L}}$, *where* w_i *is the weight coefficient of* x_i *and* $w_i > 0$. *With an inner product constraint of* $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \leq -\mathcal{B}$ *in* $\mathcal{H}^{d,\mathcal{B}}$, *a closed-form expression of the Lorentzian center* μ *can be written as:*

$$\mu = \sqrt{\mathcal{B}} \frac{\sum_{i=1}^{n} w_i x_i}{\| \sum_{i=1}^{n} w_i x_i \|_{\mathcal{L}}}, \tag{3.12}$$

where w_i *controls the positions of the center.*

The formulation of Lorentzian center can be used in hard clustering if $w_i = 1/n, \forall i$. Here Theorem 3.3 of [Law et al., 2019] gives rise to an additional claim on Lorentzian centroid based on Proposition 6.

Claim 2. *The center* μ *of Eq. (3.12) can also minimize* $\min_{\mu \in \mathcal{H}^{d,\mathcal{B}}} \sum_{i=1}^{n} w_i d_{\mathcal{L}}^2(x_i, \mu)$, *that is,* $\mu = \sqrt{\mathcal{B}} \frac{\sum_{i=1}^{n} w_i x_i}{\| \sum_{i=1}^{n} w_i x_i \|_{\mathcal{L}}}$ *can also be one feasible generalized centroid of the squared Lorentzian distance.*

The Euclidean norm of Lorentzian centroid decreases as $\mathcal{B}$ decreases, which yields an effective approximation to the focal point. However, the approximation cannot only depend on $\mathcal{B}$ due to uncertain parameter perturbations. That is to say, approximating focal expression by adjusting $\mathcal{B}$ may be ideally better but possibly worse. We thus control the another parameter w_i to implement the focal approximation

$$w_i = \frac{d_{\mathcal{L}}^2(x_i, \mu)}{\sum_{i=1}^{n} d_{\mathcal{L}}^2(x_i, \mu)}. \tag{3.13}$$

With the constraints of w_i, the Lorentzian centroid will be shifted into a more natural position over the data, not only the geometry, to capture the aspherical distribution. Then, we present a specified description on the focal point.

Proposition 7. *Approximation of Lorentzian focal. Given a set of points* $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ *and* $x_i \in \mathcal{H}^{d,\mathcal{B}}, \forall i$. *A focal point of* $\mu \in \mathcal{H}^{d,\mathcal{B}}$ *that minimizes the following problem* $\min_{\mu \in \mathcal{H}^{d,\mathcal{B}}} \sum_{i=1}^{n} w_i \langle x_i, \mu \rangle_{\mathcal{L}}$, *where* w_i *is the weight coefficient of* x_i *and* $w_i > 0$. *With an inner product constraint of* $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \leq -\mathcal{B}$ *in* $\mathcal{H}^{d,\mathcal{B}}$, *a Lorentzian focal point* μ *can be approximately formulated as*

$$\mu = \sqrt{\mathcal{B}} \frac{\sum_{i=1}^{n} w_i x_i}{\left| \| \sum_{i=1}^{n} w_i x_i \|_{\mathcal{L}} \right|}, \tag{3.14}$$

where w_i *follows Eq. (3.13).*

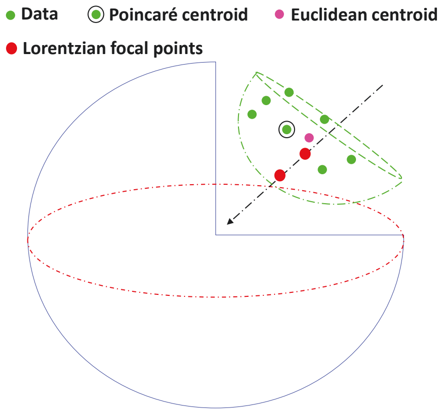


Figure 3.6 : Geometric centroids of Euclidean, Poincaré, and focal points of squared Lorentzian distances in a given set of points. Lorentzian norm updates the focal points of the embedded half-sphere toward the surface as the parameter $\mathcal{B}$ decreases w.r.t. Eq. (3.14).

The formulation of Lorentzian focal point can be used in soft or fuzzy clustering of hyperbolic space. To more easily compare the Euclidean centroid, Poincaré centroid, and Lorentzian focal point, Figure 3.6 depicts a sphere with the positions of each centroid/focal point in a set of synthetic data embedded in a half-sphere. There are seven green data points distributed within this half-sphere. As shown, the centroid based on the Euclidean norm (pink) is centered among all seven points, while the Poincaré formulation has simply selected one of the data points as its centroid based on Claim 1 (the circled one). However, the squared Lorentzian focal point(s) (red) has moved away from the center region of the embedded half-sphere according to the decrease of $\mathcal{B}$.

3.2.4 Lorentzian Focal Representation

Our focal representation is calculated from Lorentzian norms following [Nickel and Kiela, 2018] and [Law et al., 2019]. The parameter $\mathcal{B}$ shifts the position of the original Euclidean centroid toward the geometric boundary and, ideally, to (or close to) the focal point. The key step of optimizing the Lorentzian focal representation is to generalize DDGC w.r.t. Eq. (3.4) in hyperbolic space, where this process is performed in the hyperboloid geometry with Lorentzian norm. Specifically, we optimize the input dataset $\mathcal{X}$ by producing a group of subgraphs $G_i (i = 1, ..., K)$ in hyperbolic space. The disagreements between the nodes within one graph is measured by $\mathfrak{f}(\mathbf{u}, \mathbf{v})$. The goal is to minimize the distribution disagreement between any pair of subgraphs, i.e. $\min\limits_{G_1,..,G_K} \mathop{\mathbb{E}}\limits_{\mu_k \sim G_k, x_i \sim G_k} \mathfrak{f}(\mu_k, x_i)$ which can further be defined as

$$\min_{G_1,..,G_K} \mathop{\mathbb{E}}_{\mu_k \sim G_k, x_i \sim G_k} \mathfrak{f}(\mu_k, x_i) := \min_{G_1,..,G_K} \frac{1}{n} \sum_{k=1}^{K} \sum_{x_i \sim G_k} \mathfrak{f}(\mu_k, x_i), \tag{3.15}$$

where μ_i denotes the focal point of graph G_i, U denotes the focal point of a collection of the K subgraphs, i.e. $U = \{\mu_1, \mu_2, ..., \mu_K\}$. Only the data x_i achieves the minimum distance to the focal point of G_k, can it be divided into this subgraph, i.e. $x_i \in$

Algorithm 2: Lorentzian Focal Approximation

```
Input: Data set $\mathcal{X}$, number of subgraphs $K$, maximum number of iteration $T$.
Initialization: randomly select $K$ focal points from $\mathcal{X}$ to initialize
    $U^0 = \{\mu_1^0, \mu_2^0, ..., \mu_K^0\}$, $t = 1$, $F = \varnothing$.
while $t \leq T$ do
    Split $\mathcal{X}$ into $K$ subgraphs $\{G_1^{t-1}, G_2^{t-1}, ..., G_K^{t-1}\}$ based on the condition of
    $x_i \in G_k^{t-1} \ iff \ d_{\mathcal{L}}^2(x_i, \mu_k^{t-1}) < d_{\mathcal{L}}^2(x_i, \mu_{k'}^{t-1}), \forall k', 0 < k, k' \leq K, k' \neq k$.
    Update $F(t-1) = \sum_{k=1}^{K} \sum_{x_i \in G_k} d_{\mathcal{L}}^2(\mu_k^{t-1}, x_i)$.
    Update $U^t = \{\mu_1, \mu_2, ..., \mu_K\}$ by $\mu_k^t = \sum_{x_i \in G_k^{t-1}} \frac{d_{\mathcal{L}}^2(x_i,\mu)}{\sum_{i=1}^{\|G_k^{t-1}\|_0} d_{\mathcal{L}}^2(x_i,\mu)} x_i$, $0 < k \leq K$.
    Update $F(t) = \sum_{k=1}^{K} \sum_{x_i \in G_k} d_{\mathcal{L}}^2(\mu_k^t, x_i)$.
    if $F(t) - F(t-1) = 0$ then
        break.
    end
    $t = t + 1$.
end
Output: final update on $U$.
```

$G_k \ iff \ \mathfrak{f}(x_i, \mu_k) < \mathfrak{f}(x_i, \mu_{k'}), \forall k', 0 < k, k' \leq K, k' \neq k$. The node distribution disagreement of any two nodes within one subgraph is defined as $\mathfrak{f}(\cdot, \cdot)$, i.e. $\mathfrak{f}(\mathbf{u}, \mathbf{v}) := d_{\mathcal{L}}^2(\mathbf{u}, \mathbf{v})$. With Proposition 6 and Claim 2, we know the focal points can solve the minimization of Eq. (3.13).

Updating Lorentzian focal points. To fast solve Eq. (3.15), the parameters in Eq. (3.14) for our Lorentzian focal representation was set to $\mathcal{B} = 1$. Following the update policy in Eq. (3.14), the focal point μ_t is updated at the t-th iteration of G_k (also write as G_k^t) with

$$\mu_k^t = \sum_{x_i \in G_k^{t-1}} \frac{d_{\mathcal{L}}^2(x_i, \mu)}{\sum_{i=1}^{\|G_k^{t-1}\|_0} d_{\mathcal{L}}^2(x_i, \mu)} x_i, \tag{3.16}$$

where $\|G_k^{t-1}\|_0$ denotes data number of G_k^{t-1}. Then, with the t-th update on U^t w.r.t. $\mu_k^t, 1 \leq k \leq K$, G_i^t, $\forall i$ is updated, following the constraints of Eq. (3.15).

Algorithm 2 presents an unsupervised solver to the minimization of Eq. (3.15). The while loop stops if the focal point collection U has no further updating. The computational complexity of the algorithm is $O(nd)$. To strength the hierarchical characteristics of squared Lorentzian distance, the active learning strategy is splitting the Lorentzian focal points with a tree structure.

3.2.5 Tree-likeness Splitting

Given an annotation budget of K, our deep active learning method will pick up K Lorentzian focal points from the input features as its output representations. To select those data, we optimize the focal points following a tree paradigm that hierarchically splits data set $\mathcal{X}$. Details of the splitting steps are as follows.

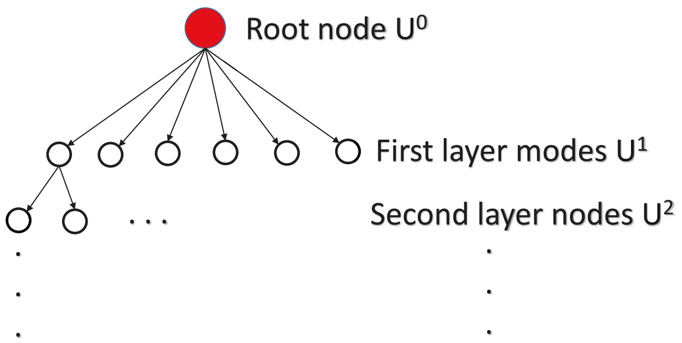


Figure 3.7 : Illustration of tree-likeness splitting. The first layer nodes are k global focal points. A binary tree splitting strategy begins from the second layer of the tree.

- **Initialization:** establish a virtual root node U^0 to begin the splitting;
- **Begin splitting:** the first split follows a global strategy that finds k Lorentzian focal points employing Algorithm 2, collects them in U^1, and hangs them on the first layer of the tree;
- **Apply the conditions for splitting:** at t-time of splitting, for any newly updated node $\mu_i \in U^{t-1}$, its associated subgraph G_i can be split into two subtrees $\mathcal{T}_1'$ and $\mathcal{T}_2'$ only if G_i has more than $\mathcal{M}$ data. Then, we update U^{t-1} into U^t. If U^t has more than K nodes, the splitting stops;
- **Select the optimal samples and conduct training:** select the top K focal points from U as the set of Lorentzian focal collection to be sent for human annotation. Once annotated, add them to network training set.

Figure 3.7 illustrates the splitting process. At the initial splitting, we have a remark to split the root node of the tree by:

Remark 2. *The first splitting of the tree-likeness splitting algorithm adopts an unsupervised Lorentzian focal approximation, which is a typical unsupervised representation learning strategy. The splitting setting can refer to the selection of* k *of unsupervised clustering [Pham et al., 2005], and the multi-class splitting rule [Buntine and Niblett, 1992] of decision tree [Fayyad and Irani, 1992], etc. To uniformly draw the distribution of the input data of* $\mathcal{X}$*,* k *is usually set as the class number of* $\mathcal{X}$ *to split the root node, which conducts the first layer of the tree, i.e.* $\|U^1\|_0 = k$*. Note that the first splitting of the tree may stop with inconsistent results in a large-scale data set. With multi-layer splitting, the bottom of the tree with more leaf nodes, will converge into more similar results.*

By adopting Algorithm 2, Algorithm 3 presents the tree-likeness splitting to implement the steps of the above procedure, in which the output focal points will be trained by a deep neural network; other classifiers are also feasible. Specifically, $\|G\|_0$ denotes the number of its collected subgraphs.

Algorithm 3: Tree-likeness Splitting

```
Input: Data set $\mathcal{X}$, splitting threshold $\mathcal{M}$.
Initialization: A tree $U$ with a virtual root node $U^0$, $j = 1$, $\mathcal{T}', \mathcal{T}_1', \mathcal{T}_2' = \varnothing$.
Splitting $\mathcal{X}$ into $k$ subgraphs $G = \{G_1, G_2, ..., G_k\}$ by adopting Algorithm 1 and
  collecting the $k$ focal points into $U^1$.
while $j \leq T$ do
    for $i = 1, 2, ..., \|G\|_0$ do
        if $\|G_i\|_0 \geq \mathcal{M}$ then
            Split subgraph $G_i$ into $\mathcal{T}_1'$ and $\mathcal{T}_2'$ by adopting Algorithm 1.
            Collect focal points $\mu_1'$ of $\mathcal{T}_1'$ and $\mu_2'$ of $\mathcal{T}_2'$ into $U^{j+1}$, respectively.
            Remove $\mu_1'$, $\mu_2'$ from $\mathcal{T}_1'$, $\mathcal{T}_2'$, respectively.
            Collect $\mathcal{T}_1', \mathcal{T}_2'$ into $\mathcal{T}'$.
            $\mathcal{T}_1', \mathcal{T}_2' = \varnothing$.
        end
    end
    $G = \mathcal{T}'$, $\mathcal{T}' = \varnothing$, $j = j + 1$.
    if $\|U^{j+1}\| \geq K$ then
        break.
    end
end
Update each node of $U$ into its nearest data in $\mathcal{X}$.
Output: final update on $U$.
```

3.3 Experiments

We tested our method with four benchmark datasets, each of them is designed for different image classification tasks.

In Section 3.6.1, we describe the tested datasets and baselines. In Section 3.6.2, we compare the centroid and focal representations on MNIST dataset. Section 3.6.3 compares the classification results of training ResNet20 with different representation features generated from different baselines. Section 3.6.4 presents the learning curves and statistical results of the trained epochs. In Section 3.6.5, we visualize the Lorentzian focal points on CIFAR-10 dataset. In Section 3.6.6, we present the batch performance of deep active learning tests.

3.3.1 Experimental Setup

The four selected datasets were MNIST, CIFAR-10, CIFAR-100, and SVHN. MNIST is an image dataset of handwritten digits with 60,000 images over 10 classes. CIFAR-10 is built for a coarse-grained image classification task with 60,000 images across 10 classes. CIFAR-100 contains 60,000 images distributed over 100 classes for a fine-grained image classification task. SVHN contains 99,289 images of numbers distributed across 10 coarse-grained classes.

Our work focuses on geometric exploration of deep active learning in hyperbolic space. The main compared experimental baselines also keep consistent geometric interests. Besides this, generalization baselines of error disagreement is necessary to verify our theoretical results. We thus firstly selected four geometric approaches involved with centroid representation as baselines for comparison: k-means, k-medoids, greedy core-set [Sener and Savarese, 2018a], and hierarchical tree clustering. Descriptions of these baselines are as follows.

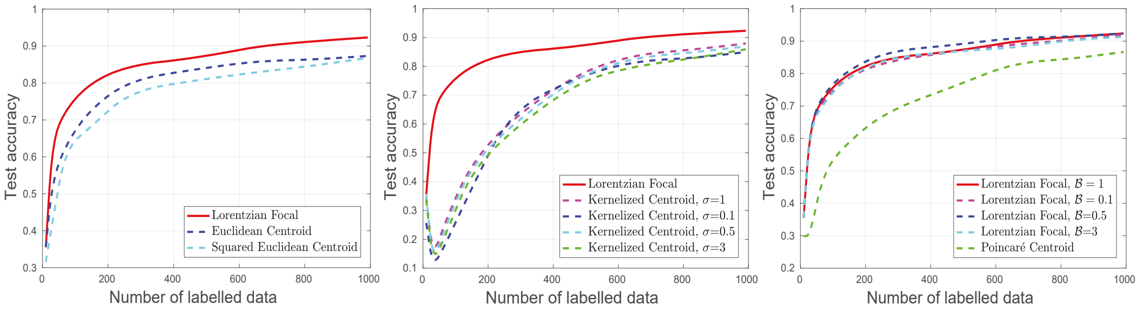
- The k-means approach on active learning setting estimates the label of some queried unlabeled data by rounding the average labels of the members of a cluster, applying its unsupervised results.
- Core-set is nonparametric and we employ a greedy selection strategy of k-center to produce its best performance following [Sener and Savarese, 2018a].
- Hierarchical clustering is with a tree-likeness splitting manner, highly relate to our approach. Only the number of desired queries (also known as querying budget) is set as the clustering number or stopping criteria of sampling.
- In addition, we selected a generalized error disagreement, i.e. error entropy-based deep active learning [Gal et al., 2017]. Error Entropy is also nonparametric as the above four baselines.
- Another generalization of error disagreement-based active learning is to maximizing the variation ratios [Gal et al., 2017] that further be used in the batch performance of deep active learning tests. It is referred to as Error Variation in experiments.
- For our tree-likeness splitting algorithm, splitting threshold $\mathcal{M}$ is set as 50.

All experiments were performed on a 2x 2.4GHz Intel Xeon E5-2680 v4 (14 Cores) with a 35MB L3 Cache 9.6GT/s QPI (Max Turbo Freq. 3.3GHz, Min 2.9GHz) and 2x NIVDIA Quadro P5000 16GB Graphics Card (GPUs) (2560 Cores). The compiler environments are Matlab 2016 and Python 3.6, where Error Entropy and Error Variation call the deep neural network model at each iteration of sampling.

3.3.2 Centroid vs. Focal Representations

We compare the Euclidean, Gaussian kernelized, and Poincaré centroids and our Lorentzian focal representations on MNIST with aspherical distributions. The selected distance metrics follow Table 3.3 and their parameter settings follow Section 3.4. To observe more differential parameter perturbations, we select SVM as the classifier, not a deep neural network, where LIBSVM is used to implement the SVM model following its default parameter setting. (Deep neural network narrows their performance disagreements due to expressive modeling on MNIST.)

With a similar setting of Section 3.4 , Euclidean, squared Euclidean, Gaussian kernelized and Poincaré centroids use k-medoids to optimize the centroid representations. Lorentzian focal points use Algorithm 3 to split the MNIST dataset. Figure 3.8 presents



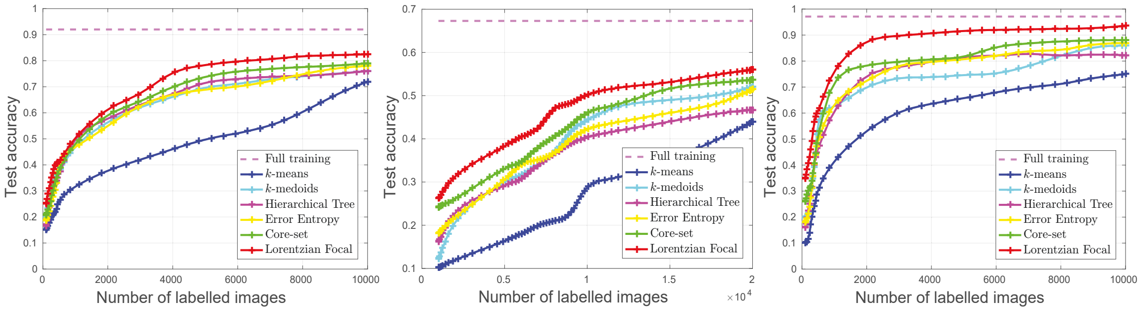
(a) Lorentzian Focal vs. Euclidean Centroids (b) Lorentzian Focal vs. Gaussian Kernelized Centroids (c) Lorentzian Focal vs. Poincaré Centroid

Figure 3.8 : Test accuracies of Euclidean, squared Euclidean, Gaussian kernelized, Poincaré centroids and our Lorentzian focal points with varying parameters on MNIST.

the active learning curves of training the centroids and focal points derived from different distance metrics. The results clearly show that our focal points achieve much higher test accuracies than the other centroid expressions due to its better representation on boundaries of aspherical distributions. Moreover, the adopted tree-likeness splitting ensures the subsequent split focal points can uniformly match each cluster over the input dataset. Parameter perturbations of Gaussian kernelized centroids and Lorentzian focal are similar after querying sufficient training data. However, test accuracies of Gaussian kernelized centroids are lower than Euclidean centroids and its squared type.

3.3.3 Deep Active Learning with ResNet20

Our experimental settings for the image classification tasks were as follows. We used ResNet20 as the deep leaning model with its default hyperparameters, i.e., batch size=32, epochs = 200, depth =20, learning rate=0.001, filter number=16, etc. The network architecture was implemented in Keras 2.2.3. With each dataset, we initiated the Entropy baseline with the first 20 samples.



(a) CIFAR-10 (b) CIFAR-100 (c) SVHN

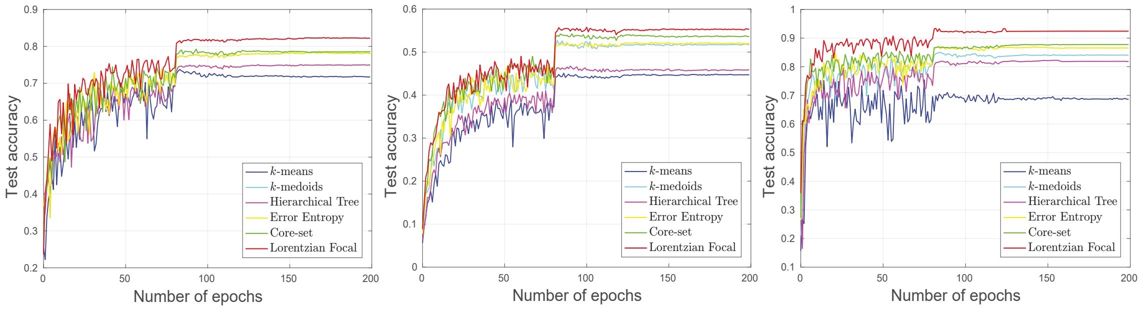
Figure 3.9 : Test accuracies of training ResNet20 with different active learning outputs on CIFAR-10, CIFAR-100, and SVHN.

The active learning results for each of the baselines for each of the three classification tasks are shown in Figure 3.9. k-means produced the least accurate results all round since nonoe of the three datasets offer an intuitive clustering structure. The centers generated in the clustering optimization process are virtual points that may not properly fit the real data distribution. Then, the tightness of the subsequent fitting over each cluster likely degrades with each iteration. The reason is that k-means rounds the average value of the ground-truth labels of each member in the cluster and annotates the label of the virtual center with that estimation. So, if the initial clusters do not roughly reflect the actual distribution, the estimated annotations will be wrong, which feeds misleading information into the next iteration and the cycle continues. The test accuracies of the k-medoids baseline was much higher than that of k-means on all three datasets. This is because the global optimization process of the k-medoids algorithm samples the nearest neighbor of each cluster center and queries all the ground-truth labels for the final update of the cluster center. More simply put, k-medoids does not estimate its owned label and asks for one from a human. Geometrically, both k-means and k-medoids are based on Euclidean centroids. However, the feature space of the image datasets are non-Euclidean. As a result, the baselines built on hierarchical tree structures performed better than these two.

Error entropy is typical of active learning approaches based on error disagreement. However, if there are not enough labels when training begins, the estimated prediction of an unlabeled sample will always tend toward a certain class label, which means the samples selected for human annotation have a higher probability of representing that class. With estimating entropy, what begins as a small deviation can degenerate into either a biased or a random selection index. In our experiment settings, the number of the initialized labels to start Entropy is 20. The core-set approach returns a collection of global representation examples for the deep model. However, these three image datasets have very high dimension feature spaces at 3072, which may drive sampling update toward the boundary of the feature space. As a result, some isolated or noisy data will always be selected. Our Lorentzian tree-likeness algorithm appeared to select informative and representative samples with good representations of the distribution at both the global and local levels from the beginning. Thus, the test accuracies of the Lorentzian centroids were the highest on all three datasets.

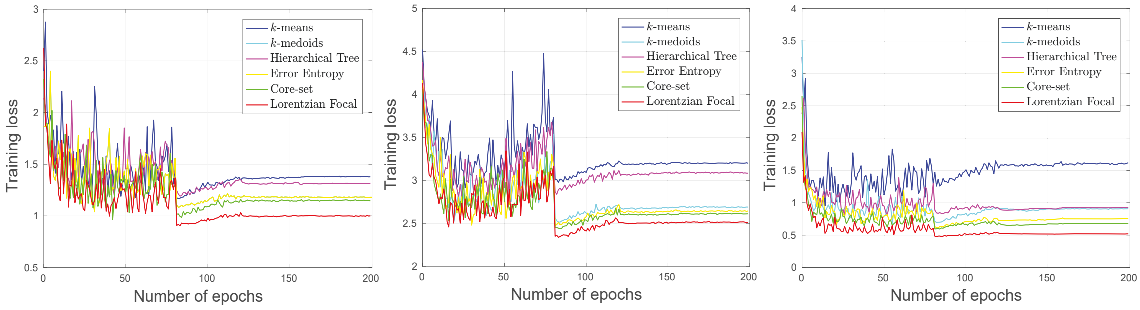
3.3.4 Learning Curves of ResNet20

Figure 3.10 shows the learning curves at each epoch of training with different active learning outputs on each of the three datasets. Initially, at a learning rate of 0.001, all baselines were unstable. However, at the 80th epoch of the training process (learning rate>0.1), most had reached optimal test accuracy. Hence, the drawn learning curves in Figure 3.10 show that the active learning outputs of all baselines had achieved their best performance under this parameterized ResNet20 model. The training losses at each epoch appear in Figure 3.11. Again, it took around 80 epochs to reach the minimum loss. The mean±standard deviation for the classification accuracy and training loss after the 80th epoch, averaged over three runs, are provided in Tables 3.4 and 3.5. Specifically, test accuracy/training loss of each epoch is averaged over 3 runs; the 'mean' accuracy/training loss denotes the averaged test accuracy/training loss of the 80th, 81th,...,200th epoch; the 'standard deviation' is over the test accuracies/training looses of these epochs. Our



(a) CIFAR-10 (b) CIFAR-100 (c) SVHN

Figure 3.10 : Test accuracies of each epoch of training ResNet20 with different active learning outputs on CIFAR-10, CIFAR-100, and SVHN. (Before 80 epochs, the learning rate is 0.001, and afterwards it is 0.1.)



(a) CIFAR-10 (b) CIFAR-100 (c) SVHN

Figure 3.11 : Training losses of each epoch of training ResNet20 with different active learning outputs on CIFAR-10, CIFAR-100, and SVHN. (Before 80 epochs, the learning rate is 0.001, and afterwards it is 0.1.)

Lorentzian focal points had the highest mean values and the lowest training losses, which we attribute to the powerful representation ability of Lorentzian focal points.

3.3.5 Visualization of Lorentzian Focal Representations

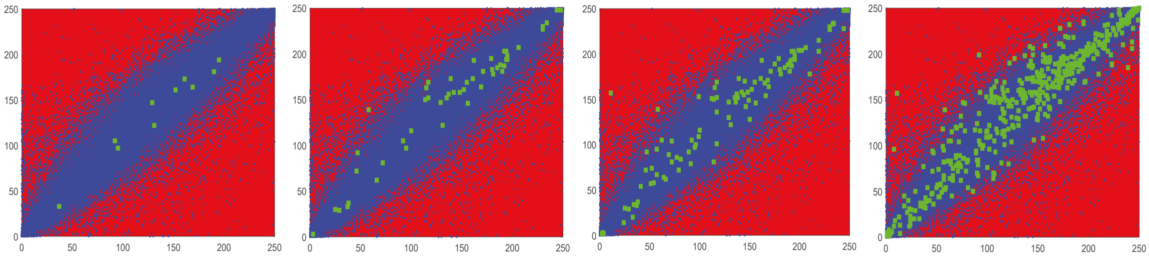
To illustrate the distribution of the Lorentzian focal representations, Figure 3.12 visualizes the focal points for CIFAR-10 in its first-two dimensional feature space. The blue points represent the original features of CIFAR-10, and the green squares represent the Lorentzian focal points. Figure 3.12(a) shows that all the initially-selected focal points are distributed around the central regions of the feature space with very high global representativeness. These first nodes of the tree then continue to split with most of the newly-generated focal points, which still are distributed tightly around the central region of the input feature space (see Figures 3.12(c) and 3.12(d)).

Table 3.4 : Mean±standard deviation of the test accuracies on CIFAR-10, CIFAR-100 and SVHN after 80 epochs over 3 runs.

Datasets	Algorithms					
	k-means	k-medoids	Hierarchical Tree	Error Entropy	Core-set	Lorentzian Focal
CIFAR-10	0.7199± 0.0055	0.7778±0.0097	0.7466±0.0077	0.7779±0.7639	0.7837±0.0086	**0.8184±0.0105**
CIFAR-100	0.4444±0.0097	0.5149±0.0129	0.4573±0.0108	0.5176±0.0125	0.5353±0.0100	**0.5503±0.0130**
SVHN	0.6881±0.0100	0.8396±0.0053	0.8160±0.0100	0.8647±0.0070	0.8728±0.0084	**0.9233±0.0043**

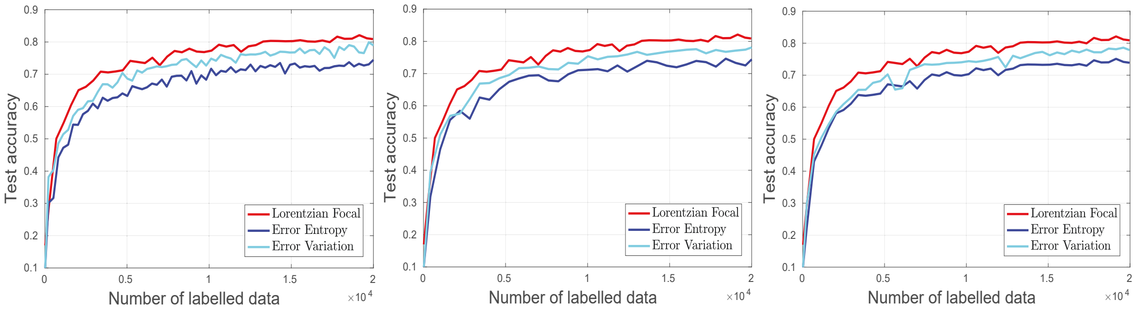
Table 3.5 : Mean±standard deviation of the training losses on CIFAR-10, CIFAR-100 and SVHN after 80 epochs over 3 runs.

Datasets	Algorithms					
	k-means	k-medoids	Hierarchical Tree	Error Entropy	Core-set	Lorentzian Focal
CIFAR-10	1.3450±0.0575	1.1756±0.0479	1.3016±0.0361	1.1756±0.0479	1.1300±0.0457	**0.9926±0.0496**
CIFAR-100	3.1748±0.0890	2.6738±0.1052	3.0602±0.0858	2.6273±0.0923	2.5915±0.0787	**2.4948±0.0948**
SVHN	1.5436±0.0927	0.8720±0.0578	0.9149±0.0446	0.7289±0.0378	0.6691±0.0258	**0.5169±0.0156**



(a) K=10 (b) K=50 (c) K=100 (d) K=500

Figure 3.12 : Projection of Lorentzian focal points in the first two dimensional feature space of CIFAR-10.



(a) Batch budget=100 (b) Batch budget=200 (c) Batch budget=300

Figure 3.13 : Lorentzian focal vs. batch performance of error entropy and error variation on CIFAR-10.

3.3.6 Batch Performance of Deep Active Learning

We next compare the performance of Lorentzian focal points and batch settings of error disagreement-based active learning algorithms. Note, batch active learning [Guo and Schuurmans, 2008] is another different topic involved with diverse sampling. The general active learning algorithms can be extended into batch returns to accelerate the sampling, but they are not special-purpose batch sampling strategies.

In this task setting, the selected error disagreement baselines are Error Entropy and Error Variation. The deep network architecture follows a three-layer perceptron (MLP) with three blocks of [convolution, dropout, max-pooling, relu], with 32, 64, and 128 3x3 convolution filters, 5x5 max pooling, and 0.5 dropout rate. The network architecture still was implemented in Keras 2.2.3. The labeled set to start the error disagreement-based active learning algorithms are 20 random samples drawn from the training sets of the input datasets.

The learning curves of batch performance of deep active learning on CIFAR-10 are presented in Figure 3.13, where the batch budgets are set as 100, 200, and 300, respectively. For error disagreement-based active learning, the test accuracies do not keep consistent results to the batch budgets since the error estimations with different budget settings make the hypothesis update coarsely. Table 3.6 presents their mean±standard deviation of the test accuracies against the learning breakpoints. Observing the standard deviation finds that, the larger budget settings have more obvious perturbations to the querying process of active learning (see deviation change). However, the distribution disagreement-based focal representation approach does not estimate the hypothesis update. Therefore, it has no perturbations to the batch budget settings. This can be one advantage of our focal representation.

3.4 Discussions

3.4.1 Deep Active Learning with Weak-supervision

With active learning methods based on error disagreement, the initial state of the trained model directly affects the subsequent sampling process. In our experiments, we

Table 3.6 : Mean±standard deviation of the test accuracies of different baselines with batch settings on CIFAR-10.

Batch budget	Algorithms		
	Error Entropy	Error Variation	Lorentzian Focal
100	0.6644±0.0091	0.7070±0.0098	0.7394±0.0124
200	0.6687±0.0110	0.7053±0.0112	-
300	0.6694±0.1115	0.7007±0.1145	-

Table 3.7 : Time computations of selecting 10,000 examples from CIFAR-10 using different baselines.

Datasets	Algorithms						
	k-means	k-medoids	Hierarchical Tree	Error Entropy	Error Variation	Core-set	Lorentzian Focal
CIFAR-10	30 mins	5 hours	8 hours	3 days	2.5 days	2 days	**20 mins**

selected 20 labeled samples as a prior training set. However, had we began with full supervision on all classes and generated sufficient prior labels, the error-disagreement baselines would have shown better performance, if not the best performance. However, this would not be a fair comparison since the other baselines do not utilize prior labels. Therefore, there exists an open question of how to properly start the error disagreement-based active learning baselines. This is also the reason of why we do not introduce more generalization baselines of error disagreement in our experiments.

3.4.2 Limitations of Our Approach

Our tree splitting process is very fast, but, as Figure 3.12 shows, the first layer of the established tree has a significant influence over the subsequent splitting positions. Therefore, beginning the splits from an inappropriate starting position will degenerate performance

3.4.3 Computational Complexities

It costs $O(nd)$ time to establish the first layer of the tree. Then the algorithm continues to split each subtree in a shorter time. It is thus, the total time complexity of the proposed algorithm is $O(nd)$. For the compared baselines, k-means has a time complexity of $O(nd)$, and k-medoids has a time complexity of $O(n^2)$. Core-set selection with a greedy search costs about $O(n^3)$. Hierarchical tree costs $O(n^2)$. Further, the total time taken to select 10,000 samples from CIFAR-10 with each of the baselines are reported in Table 3.7. As the results show, the function of Lorentzian focal representation optimized with the tree splitting is the fastest.

3.5 Summary of This Chapter

This chapter presents a new direction to embed active learning onto a non-Euclidean hyperbolic geometry. The proposed distribution disagreement graph coefficient (DDGC), based on distribution, was derived to present a tighter bound on label complexity than typical error disagreement coefficient; a generalization test subsequently verified the tightness of the approximate inequality relationship of the bounds. With these theoretical guarantees, the Lorentzian focal representation was proposed as a generalization of DDGC in practical active learning tasks, where the squared Lorentzian distance is used to derive a closed-form update on the focal points. The overall deep active learning approach which derives the Lorentzian focal points, then shows effective accuracy improvement against the centroid representation methods and generalized error disagreement algorithms in three image classification datasets.

Chapter 4

Distribution Matching

This chapter discusses the third question of the thesis: "how to improve deep active learning based on the theoretical results of halfspace learning?". Specifically, we present geometric Bayesian active learning by disagreements (GBALD), a framework that performs BALD on its geometric interpretation, matching the distribution to interact with a deep learning model. There are two main components in GBALD: initial acquisitions based on core-set construction and model uncertainty estimation with those initial acquisitions. Our key innovation is to construct the core-set on an ellipsoid, not typical sphere, preventing its updates towards the boundary regions of the distributions.

The main improvements over BALD are twofold: relieving sensitivity to uninformative prior and reducing redundant information of model uncertainty. Experiments on acquisitions with several scenarios demonstrate that, yielding slight perturbations to noisy and repeated samples, GBALD further achieves significant accuracy improvements than BALD, BatchBALD and other baselines.

The rest of this chapter is organized as follows. In Section 4.1, we elaborate BALD and its two interpretations. In Section 4.2, we present our GBALD framework. Experimental results are presented in Section 4.3. Finally, we conclude this chapter in Section 4.4.

4.1 Bayesian Active Learning by Disagreements

Bayesian active learning by disagreements (BALD) [Houlsby et al., 2011] expresses the information gain in terms of the predictive entropy over the model. It has two interpretations: model uncertainty estimation and core-set construction. To estimate the model uncertainty, a greedy strategy is applied to select those data that maximize the parameter disagreements between the current training model and its subsequent updates as [Gal et al., 2017]. However, naively interacting with BALD using uninformative prior leads to unstable biased acquisitions [Gao et al., 2020]. Moreover, the similarity or consistency of those acquisitions to the previous acquired samples, brings redundant information to the model and decelerates its training.

Core-set construction [Campbell and Broderick, 2018] avoids the greedy interaction to the model by capturing characteristics of the data distributions. By modeling the complete data posterior over the distributions of parameters, BALD can be deemed as a core-set construction process on a sphere [Kirsch et al., 2019], which seamlessly solicits a compact subset to approximate the input data distribution, and efficiently mitigates the sensitivity to uninformative prior and redundant information. From the view of geometry,

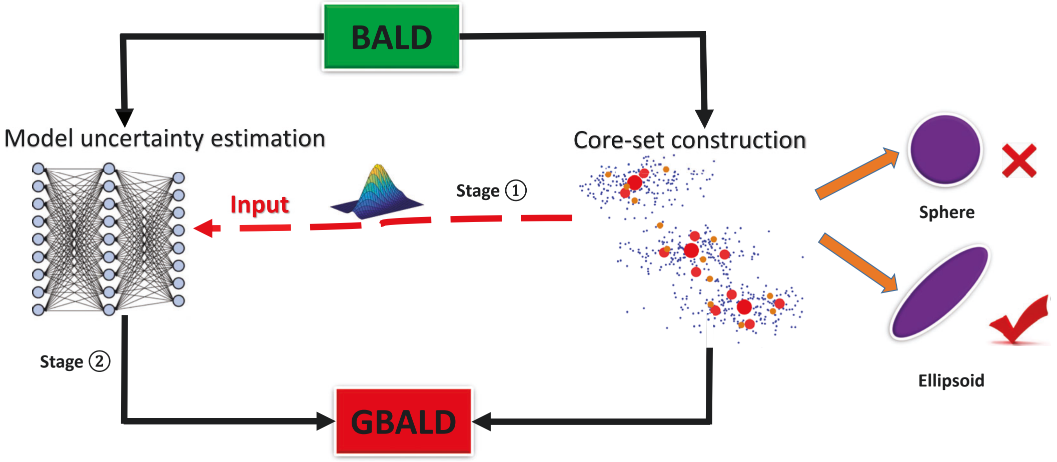


Figure 4.1 : Illustration of two-stage GBALD framework. BALD has two types of expression: model uncertainty estimation and core-set construction where the deeper the color of the core-set elements, the higher the representation; GBALD integrates them to an uniform framework. Stage ①: core-set construction is with an ellipsoid, not typical sphere, representing the original distribution to indicate the input features of DNN. Stage ②: model uncertainty estimation of DNN then derives the subsequent highly informative and representative samples.

updates of core-set construction is usually optimized with sphere geodesic as [Nie et al., 2013; Wang et al., 2019]. Once the core-set is obtained, deep active learning immediately seeks annotations from experts and starts the training. However, data points located at the boundary regions of the distribution, usually win uniform distribution, cannot be highly-representative candidates for the core-set. Therefore, constructing the core-set on a sphere may not be the optimal choice for deep active learning.

This chapter presents a novel active learning framework, namely Geometric BALD (GBALD), over the geometric interpretation of BALD that, interpreting BALD with core-set construction on an ellipsoid, initializes an effective matching on distribution to drive a DNN model. The goal is to seek for significant accuracy improvements against an uninformative prior and redundant information. Figure 4.1 describes this two-stage framework. In the first stage, geometric core-set construction on an ellipsoid initializes effective matching on the distributions to start a DNN model regardless of the uninformative prior. Taking the core-set as the input features, the next stage ranks the batch acquisitions of model uncertainty according to their geometric representativeness, and then solicits some highly-representative examples from the batch. With the representation constraints, the ranked acquisitions reduce the probability of sampling nearby samples of the previous acquisitions. It is thus the ranking rejects unnecessary redundant acquisitions.

4.1.1 Interpretations

BALD has two different interpretations: model uncertainty estimation and core-set construction. We simply introduce them in this section.

4.1.1.1 Model Uncertainty Estimation

We consider a discriminative model $p(y|x,\theta)$ parameterized by θ that maps $x \in \mathcal{X}$ into an output distribution over a set of $y \in \mathcal{Y}$. Given an initial labeled (training) set $\mathcal{D}_0 \in \mathcal{X} \times \mathcal{Y}$, the Bayesian inference over this parameterized model is to estimate the posterior $p(\theta|\mathcal{D}_0)$, i.e. estimate θ by repeatedly updating $\mathcal{D}_0$. Active learning adopts this setting from a Bayesian view.

With AL, the learner can choose unlabeled data from $\mathcal{D}_u = \{x_i\}_{j=1}^{N} \in \mathcal{X}$, to observe the outputs of the current model, maximizing the uncertainty of the model parameters. Houlsby et al. [2011] proposed a greedy strategy termed BALD to update $\mathcal{D}_0$ by estimating a desired data x^* that maximizes the decrease in expected posterior entropy:

$$x^* = \arg\max_{x \in \mathcal{D}_u} \mathrm{H}[\theta|\mathcal{D}_0] - \mathbb{E}_{y \sim p(y|x,\mathcal{D}_0)}\Big[\mathrm{H}[\theta|x,y,\mathcal{D}_0]\Big], \tag{4.1}$$

where the labeled and unlabeled sets are updated by $\mathcal{D}_0 = \mathcal{D}_0 \cup \{x^*, y^*\}, \mathcal{D}_u = \mathcal{D}_u \backslash x^*$, and y^* denotes the output of x^*. In deep AL, y^* can be annotated as a label from experts and θ yields a DNN model.

4.1.1.2 Core-set Construction

Let $p(\theta|\mathcal{D}_0)$ be updated by its log posterior $\log p(\theta|\mathcal{D}_0, x^*)$, $y^* \in \{y_i\}_{i=1}^{N}$, assume the outputs are conditional independent of the inputs, i.e.

$$p(y^*|x^*, D_0) = \int_\theta p(y^*|x^*,\theta)p(\theta|D_0)\mathrm{d}\theta,$$

then we have the *complete data log posterior following* [Pinsler et al., 2019]:

$$\begin{aligned}
\mathbb{E}_{y^*}[\log p(\theta|\mathcal{D}_0, x^*, y^*)] &= \mathbb{E}_{y^*}[\log p(\theta|\mathcal{D}_0) + \log p(y^*|x^*,\theta) - \log p(y^*|x^*,\mathcal{D}_0)] \\
&= \log p(\theta|\mathcal{D}_0) + \mathbb{E}_{y^*}\Big[\log p(y^*|x^*,\theta) + \mathrm{H}[y^*|x^*,\mathcal{D}_0]\Big] \\
&= \log p(\theta|\mathcal{D}_0) + \sum_{i=1}^{N}\left(\mathbb{E}_{y_i}\Big[\log p(y_i|x_i,\theta) + \mathrm{H}[y_i|x_i,\mathcal{D}_0]\Big]\right).
\end{aligned} \tag{4.2}$$

Core-set. The key idea of core-set construction is to approximate the log posterior of Eq. (4.2) by a subset of $D_u' \subseteq D_u$ such that:

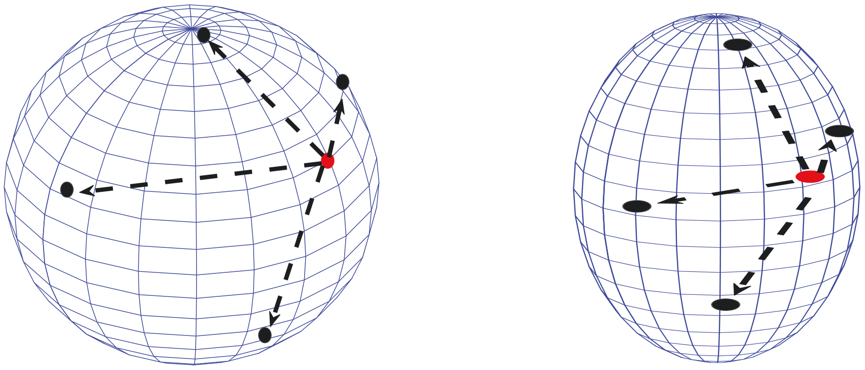
$$\mathbb{E}_{\mathcal{Y}_u}[\log p(\theta|\mathcal{D}_0, \mathcal{D}_u, \mathcal{Y}_u)] \approx \mathbb{E}_{\mathcal{Y}_u'}[\log p(\theta|\mathcal{D}_0, \mathcal{D}_u', \mathcal{Y}_u')],$$

where $\mathcal{Y}_u$ and $\mathcal{Y}_u'$ denote the predictive labels of $\mathcal{D}_u$ and $\mathcal{D}_u'$ respectively by the Bayesian discriminative model, that is,

$$p(\mathcal{Y}_u|\mathcal{D}_u, D_0) = \int_\theta p(\mathcal{Y}_u|\mathcal{D}_u,\theta)p(\theta|D_0)\mathrm{d}\theta,$$

and

$$p(\mathcal{Y}_u'|\mathcal{D}_u', D_0) = \int_\theta p(\mathcal{Y}_u'|\mathcal{D}_u',\theta)p(\theta|D_0)\mathrm{d}\theta.$$



(a) Sphere geodesic (b) Ellipsoid geodesic

Figure 4.2 : Optimizing BALD with sphere and ellipsoid geodesics. Ellipsoid geodesic rescales the sphere geodesic to prevent the updates of the core-set towards the boundary regions of the sphere where the characteristics of the distribution cannot be captured. Black points denote the feasible updates of the red points. Dash lines denote the geodesics.

Here D'_u can be indicated by a core-set [Pinsler et al., 2019] that highly represents $\mathcal{D}_u$. Optimization tricks such as Frank-Wolfe optimization [Vavasis, 1992] then can be adopted to solve this problem.

Motivations. Eqs. (4.1) and (4.2) provide the Bayesian rules of BALD over model uncertainty and core-set construction respectively, which further attract the attention of the deep learning community. However, the two interpretations of BALD are limited by: 1) redundant information and 2) uninformative prior, where one major reason which causes these two issues is the poor initialization on the prior, i.e. $p(\mathcal{D}_0|\theta)$. For example, unbalanced label initialization on $\mathcal{D}_0$ usually leads to an uninformative prior, which further conducts the acquisitions of active learning to select those unlabeled data from one or some fixed classes; highly-biased results with [Gao et al., 2020] redundant information are inevitable. Therefore, these two limitations affect each other.

4.2 Framework

GBALD consists of two components: initial acquisitions based on core-set construction and model uncertainty estimation with those initial acquisitions.

4.2.1 Geometric Interpretation of Core-set

Modeling the complete data posterior over the parameter distribution can relieve the above two limitations of BALD. Typically, finding the acquisitions of active learning is equivalent to approximating a core-set centered with spherical embeddings [Sener and Savarese, 2018b]. Let w_i be the sampling weight of x_i, $\|w_i\|_0 \leq N$, the core-set construction is to optimize:

$$\min_{w} \left\| \underbrace{\sum_{i=1}^{N} \mathbb{E}_{y_i}\Big[\log p(y_i|x_i,\theta) + \mathrm{H}[y_i|x_i,\mathcal{D}_0]\Big]}_{\mathcal{L}} - \underbrace{\sum_{i=1}^{N} w_i \mathbb{E}_{y_i}\Big[\log p(y_i|x_i,\theta) + \mathrm{H}[y_i|x_i,\mathcal{D}_0]\Big]}_{\mathcal{L}(w)} \right\|^2, \tag{4.3}$$

where $\mathcal{L}$ and $\mathcal{L}(w)$ denote the full and expected (weighted) log-likelihoods, respectively. Specifically, $\sum_{i=1}^{N}\mathrm{H}[y_i|x_i,\mathcal{D}_0] = -\sum_{y_i} p(y_i|x_i,\mathcal{D}_0)\log(p(y_i|x_i,\mathcal{D}_0)$, where $p(y_i|x_i,\mathcal{D}_0) = \int_\theta p(y_i|x_i,\theta)p(\theta|\mathcal{D}_0)\mathrm{d}\theta$. Note $\|\cdot\|$ denotes the ℓ^2 norm.

The approximation of Eq. (4.3) implicitly requires that the complete data log posterior of Eq. (4.2) w.r.t. $\mathcal{L}$ must be close to an expected posterior w.r.t. $\mathcal{L}(w)$ such that approximating a sparse subset for the original inputs by sphere geodesic search is feasible (see Figure 4.2(a)). Generally, solving this optimization is intractable due to cardinality constraint [Pinsler et al., 2019]. Campbell and Broderick [2019] proposed to relax the constraint in Frank–Wolfe optimization, in which mapping $\mathcal{X}$ is usually performed in a Hilbert space (HS) with a bounded inner product operation. In this solution, the sphere embedded in the HS replaces the cardinality constraint with a polynomial constraint. However, the initialization on $\mathcal{D}_0$ affects the iterative approximation to $\mathcal{D}_u$ at the beginning of the geodesic search. Moreover, the posterior of $p(\theta|\mathcal{D}_0)$ is uninformative, if the initialized $\mathcal{D}_0$ is empty or not correct. Therefore, the typical Bayesian core-set construction of BALD cannot ideally fit an uninformative prior. The another geometric interpretation of core-set construction, such as k-centers [Sener and Savarese, 2018b], is not restricted to this setting. We thus follow the construction of k-centers to find the core-set.

***k*-centers.** Sener and Savarese [2018b] proposed a core-set representation approach for active deep learning based on k-centers. This approach can be adopted in core-set construction of BALD without the help of the discriminative model. Therefore, the uninformative prior has no further influence on the core-set. Typically, the k-centers approach uses a greedy strategy to search the data $\widetilde{x}$ whose nearest distance to elements of $\mathcal{D}_0$ is the maximal:

$$\widetilde{x} = \arg\max_{x_i\in\mathcal{D}_u}\min_{c_i\in\mathcal{D}_0}\|x_i - c_i\|, \tag{4.4}$$

then $\mathcal{D}_0$ is updated by $\mathcal{D}_0 \cup \{\widetilde{x},\widetilde{y}\}$, $\mathcal{D}_u$ is updated by $\mathcal{D}_u \backslash \widetilde{x}$, where $\widetilde{y}$ denotes the output of $\widetilde{x}$. This max-min operation usually performs k times to construct the centers.

From the view of geometry, k-centers can be deemed as the core-set construction via spherical geodesic search [Bādoiu et al., 2002; Har-Peled and Mazumdar, 2004]. Specifically, the max-min optimization guides $\mathcal{D}_0$ to be updated into one data, which draws the longest line segment from $x_i, \forall i$ across the sphere center. The iterative update on $\widetilde{x}$ is then along its unique diameter through the sphere center. However, this greedy optimization has large probability that yields the core-set to fall into boundary regions of the sphere, which cannot capture the characteristics of the distribution.

4.2.2 Initial Acquisitions based on Core-set Construction

We present a novel greedy search which rescales the geodesic of a sphere into an ellipsoid following Eq. (4.4), in which the iterative update on the geodesic search is rescaled (see Figure 4.2(b)). We follow the importance sampling strategy to begin the search.

Initial prior on geometry. Initializing $p(\mathcal{D}_0|\theta)$ is performed with a group of internal spheres centered with $D_j, \forall j$, subjected to $D_j \in \mathcal{D}_0$, in which the geodesic between $\mathcal{D}_0$ and the unlabeled data is over those spheres. Since $\mathcal{D}_0$ is known, specification of θ plays the key role for initializing $p(\mathcal{D}_0|\theta)$. Given a radius R_0 for any observed internal sphere,

$p(y_i|x_i,\theta)$ is firstly defined by

$$p(y_i|x_i,\theta) = \begin{cases} 1, & \exists j, \|x_i - D_j\| \leq R_0, \\ \max\left\{\dfrac{R_0}{\|x_i - D_j\|}\right\}, & \forall j, \|x_i - D_i\| > R_0, \end{cases} \tag{4.5}$$

thereby θ yields the parameter R_0. When the data is enclosed with a ball, the probability of Eq. (4.5) is 1. The data near the ball, is given a probability of $\max\left\{\frac{R_0}{\|x_i - D_j\|}\right\}$ constrained by $\min\|x_i - D_j\|, \forall j$, i.e. the probability is assigned by the nearest ball to x_i, which is centered with D_j. From Eq. (4.3), the information entropy of $y_i \sim \{y_1, y_2, ..., y_N\}$ over $x_i \sim \{x_1, x_2, ..., x_N\}$ can be expressed as the integral regarding $p(y_i|x_i,\theta)$:

$$\sum_{i=1}^{N} \mathrm{H}(y_i|x_i,\mathcal{D}_0) = -\sum_{i=1}^{N}\int_\theta p(y_i|x_i,\theta)p(\theta|D_0)d\theta \log\Big(\int_\theta p(y_i|x_i,\theta)p(\theta|D_0)\Big)d\theta, \tag{4.6}$$

which can be approximated by $-\sum_{i=1}^{N} p(y_i|x_i,\theta)\log\Big(p(y_i|x_i,\theta)\Big)$ following the details of Eq. (4.3). In short, this indicates an approximation to the entropy over the entire outputs on $\mathcal{D}_u$ that assumes the prior $p(D_0|\theta)$ w.r.t. $p(y_i|x_i,\theta)$ is already known from Eq. (4.5).

Max-min optimization. Recalling the max-min optimization trick of k-centers in the core-set construction of [Sener and Savarese, 2018b], the minimizer of Eq. (4.3) can be divided into two parts: $\min_{x^*}\mathcal{L}$ and $\max_w \mathcal{L}(w)$, where $\mathcal{D}_0$ is updated by acquiring x^*. However, updates of $\mathcal{D}_0$ decide the minimizer of $\mathcal{L}$ with regard to the internal spheres centered with $D_i, \forall i$. Therefore, minimizing $\mathcal{L}$ should be constrained by an unbiased full likelihood over $\mathcal{X}$ to alleviate the potential biases from the initialization of $\mathcal{D}_0$. Let $\mathcal{L}_0$ denote the unbiased full likelihood over $\mathcal{X}$ that particularly stipulates $\mathcal{D}_0$ as the k-means centers written as $\mathcal{U}$ of $\mathcal{X}$ which jointly draw the input distribution. We define $\mathcal{L}_0 = |\sum_{i=1}^{N}\mathbb{E}_{y_i}[\log p(y_i|x_i,\theta) + \mathrm{H}[y_i|x_i,\mathcal{U}]]|$ to regulate $\mathcal{L}$, that is

$$\min_{x^*}\|\mathcal{L}_0 - \mathcal{L}\|^2, \ \text{s.t.}\ \mathcal{D}_0 = \mathcal{D}_0 \cup \{x^*, y^*\}, \mathcal{D}_u = \mathcal{D}_u \backslash x^*. \tag{4.7}$$

The other sub optimizer is $\max_w \mathcal{L}(w)$. We present a greedy strategy following Eq. (4.1):

$$\begin{aligned} \max_{1\leq i\leq N}\ \min_{w_i} & \sum_{i=1}^{N} w_i \mathbb{E}_{y_i}[\log p(y_i|x_i,\theta) + \mathrm{H}[y_i|x_i,\mathcal{D}_0]] \\ &= \sum_{i=1}^{N} w_i \log p(y_i|x_i,\theta) - \sum_{i=1}^{N} w_i p(y_i|x_i,\theta)\log p(y_i|x_i,\theta), \end{aligned} \tag{4.8}$$

which can be further written as: $\sum_{i=1}^{N} w_i \log p(y_i|x_i,\theta)(1 - \log p(y_i|x_i,\theta))$. Let $w_i = 1, \forall i$ for unbiased estimation of the likelihood $\mathcal{L}(w)$, Eq. (4.8) can be simplified as

$$\max_{x_i \in \mathcal{D}_u}\ \min_{D_j \in \mathcal{D}_0} \log p(y_i|x_i,\theta), \tag{4.9}$$

where $p(y_i|x_i,\theta)$ follows Eq. (4.5). Combining Eqs. (4.7) and (4.9), the optimization of

Eq. (4.3) is then transformed as

$$x^* = \underset{x_i \in \mathcal{D}_u}{\arg\max} \ \min_{D_i \in \mathcal{D}_0} \left\{ \|\mathcal{L}_0 - \mathcal{L}\|^2 + \log p(y_i|x_i, \theta) \right\}, \tag{4.10}$$

where $\mathcal{D}_0$ is updated by acquiring x^*, i.e. $\mathcal{D}_0 = \mathcal{D}_0 \cup \{x^*, y^*\}$.

Geodesic line. For a metric geometry M, a geodesic line is a curve γ which projects its interval I to M: $I \to M$, maintaining everywhere locally a distance minimizer [Lou et al., 2020a]. Given a constant $\nu > 0$ such that for any $a, b \in I$ there exists a geodesic distance $d(\gamma(a), \gamma(b)) := \int_a^b \sqrt{g_{\gamma(t)}(\gamma'(t), \gamma'(t))}dt$, where $\gamma'(t)$ denotes the geodesic curvature, and g denotes the metric tensor over M. Here, we define $\gamma'(t) = 0$, then $g_{\gamma(t)}(0, 0) = 1$ such that $d(\gamma(a), \gamma(b))$ can be generalized as a segment of a straight line: $d(\gamma(a), \gamma(b)) = \|a - b\|$.

Ellipsoid geodesic distance. For any observation points $p, q \in M$, if the spherical geodesic distance is defined as $d(\gamma(p), \gamma(q)) = \|p - q\|$. The affine projection obtains its ellipsoid interpretation: $d(\gamma(p), \gamma(q)) = \|\eta(p - q)\|$, where η denotes the affine factor subjected to $0 < \eta < 1$.

Optimizing with ellipsoid geodesic search. The max-min optimization of Eq. (4.10) is performed on an ellipsoid geometry to prevent the updates of core-set towards the boundary regions, where ellipsoid geodesic line scales the original update on the sphere. Assume x_i is the previous acquisition and x^* is the next desired acquisition, the ellipsoid geodesic rescales the position of x^* as $x_e^* = x_i + \eta(x^* - x_i)$. Then, we update this position of x_e^* to its nearest neighbor x_j in the unlabeled data pool, i.e. $\arg\min_{x_j \in \mathcal{D}_u} \|x_j - x_e^*\|$, also can be written as

$$\underset{x_j \in \mathcal{D}_u}{\arg\min} \left\| x_j - [x_i + \eta(x^* - x_i)] \right\|. \tag{4.11}$$

To study the advantage of ellipsoid geodesic search, Appendix B presents our generalization analysis.

4.2.3 Model Uncertainty Estimation with Core-set

GBALD starts the model uncertainty estimation with those initial core-set acquisitions, in which it introduces a ranking scheme to derive both informative and representative acquisitions.

Single acquisition. We follow [Gal et al., 2017] and use MC dropout to perform Bayesian inference on the model of the neural network. It then leads to ranking the informative acquisitions with batch sequences is with high efficiency. We first present the ranking criterion by rewriting Eq. (4.1) as batch returns:

$$\{x_1^*, x_2^*, ..., x_b^*\} = \underset{\{\hat{x}_1, \hat{x}_2, ..., \hat{x}_b\} \subseteq \mathcal{D}_u}{\arg\max} \mathrm{H}[\theta|\mathcal{D}_0] - \mathbb{E}_{\hat{y}_{1:b} \sim p(\hat{y}_{1:b}|\hat{x}_{1:b}, \mathcal{D}_0)} \Big[\mathrm{H}[\theta|\hat{x}_{1:b}, \hat{y}_{1:b}, \mathcal{D}_0] \Big], \tag{4.12}$$

where $\hat{x}_{1:b} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_b\}$, $\hat{y}_{1:b} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_b\}$, $\hat{y}_i$ denotes the output of $\hat{x}_i$. The informative acquisition x_t^* is then selected from the ranked batch acquisitions $\hat{x}_{1:b}$ due to the highest representation for the unlabeled data:

$$x_t^* = \operatorname*{arg\,max}_{x_i^* \in \{x_1^*, x_2^*, ..., x_b^*\}} \left\{ \max_{D_j \in \mathcal{D}_0} p(y_i|x_i^*, \theta) := \frac{R_0}{\|x_i^* - D_j\|} \right\}, \tag{4.13}$$

where t denotes the index of the final acquisition, subjected to $1 \leq t \leq b$. This also adopts the max-min optimization of Eq. (4.4), i.e. $x_t^* = \arg\max_{x_i^* \in \{x_1^*, x_2^*, ..., x_b^*\}} \min_{D_j \in \mathcal{D}_0} \|x_i^* - D_j\|$.

Batch acquisitions. The greedy strategy of Eq. (4.13) can be written as a batch acquisitions by setting its output as a batch set, i.e.

$$\{x_{t_1}^*, ..., x_{t_{b'}}^*\} = \operatorname*{arg\,max}_{x_{t_1:t_{b'}}^* \subseteq \{x_1^*, x_2^*, ..., x_b^*\}} p(y_{t_1:t_{b'}}^* | x_{t_1:t_{b'}}^*, \theta), \tag{4.14}$$

where $x_{t_1:t_{b'}}^* = \{x_{t_1}^*, ..., x_{t_{b'}}^*\}$, $y_{t_1:t_{b'}}^* = \{y_{t_1}^*, ..., y_{t_{b'}}^*\}$, $y_{t_i}^*$ denotes the output of $x_{t_i}^*$, $1 \leq i \leq b'$, and $1 \leq b' \leq b$. This setting can be used to accelerate the acquisitions of active learning in a large dataset. Algorithm 4 presents the two-stage GBALD algorithm.

4.2.4 Two-stage GBALD Algorithm

The two-stage GBALD algorithm is described as follows: 1) construct core-set on ellipsoid (Lines 3 to 13), and 2) estimate model uncertainty with a deep learning model (Lines 14 to 21). Core-set construction is derived from the max-min optimization of Eq. (4.10), then updated with ellipsoid geodesic w.r.t. Eq. (4.11), where θ yields a geometric probability model w.r.t. Eq. (4.5). Importing the core-set into $\mathcal{D}_0$ derives the deep learning model to return b informative acquisitions one time, where θ yields a deep learning model. Ranking those samples, we select b' samples with the highest representations as the batch outputs using Eq. (4.14). The iterations of batch acquisitions stop until its budget is exhaust. The final update on $\mathcal{D}_0$ is our acquisition set of active learning.

4.3 Experiments

4.3.1 Experimental Setup

In experiments, we start by showing how BALD degenerates its performance with uninformative prior and redundant information, and show that how our proposed GBALD relieves theses limitations.

Our experiments discuss three questions: 1) is GBALD using core-set of Eq. (4.11) competitive with uninformative prior? 2) can GBALD using ranking of Eq. (4.14) improve informative acquisitions of model uncertainty? and 3) can GBALD outperform state-of-the-art acquisition approaches? Following the experiment settings of [Gal et al., 2017; Kirsch et al., 2019], we use MC dropout to implement the Bayesian approximation of DNNs. Three benchmark datasets are selected: MNIST, SVHN and CIFAR10.

To evaluate the performance of GBALD, several typical baselines from the latest deep active learning literatures are selected.

- Bayesian active learning by disagreement (BALD) [Houlsby et al., 2011]. It has been introduced in Section 4.4.3.

Algorithm 4: Two-stage GBALD Algorithm

```
Input: Data set $\mathcal{X}$, core-set size $N_{\mathcal{M}}$, batch returns $b$, batch output $b'$, iteration
  budget $\mathcal{A}$.
Initialization: $\alpha \leftarrow 0$, core-set $\mathcal{M} \leftarrow \varnothing$.
Stage ① begins:
Initialize $\theta$ to yield a geometric probability model w.r.t. Eq. (4.5).
Perform $k$-means to initialize $\mathcal{U}$ to $\mathcal{D}_0$.
Core-set construction begins by acquiring $x_i^*$,
for $i \leftarrow 1, 2, ..., N_{\mathcal{M}}$ do
    $x_i^* \leftarrow \arg\max_{x_i \in \mathcal{D}_u} \min_{D_i \in \mathcal{D}_0} \left\{ \left\| \mathcal{L}_0 - \mathcal{L} \right\|^2 + \log p(y_i|x_i, \theta) \right\}$, where
      $\mathcal{L}_0 \leftarrow \left| \sum_{i=1}^{N} \mathbb{E}_{y_i} [\log p(y_i|x_i, \theta) + \mathrm{H}[y_i|x_i, \mathcal{U}]] \right|$.
    Ellipsoid geodesic line scales $x_i^*$: $x_i^* \leftarrow \arg\min_{x_j \in \mathcal{D}_u} \left\| x_j - [x_i + \eta(x^* - x_i)] \right\|$.
    Update $x_i^*$ into core-set $\mathcal{M}$: $\mathcal{M} \leftarrow x_i^* \cup \mathcal{M}$.
    Update $N \leftarrow N - 1$.
end
Import core-set to update $\mathcal{D}_0$: $\mathcal{D}_0 \leftarrow \mathcal{M} \cup \mathcal{U}'$, where $\mathcal{U}'$ updates each element of $\mathcal{U}$
  into their nearest samples in $\mathcal{X}$.
Stage ② begins:
Initialize $\theta$ to yield a deep learning model.
while $\alpha < \mathcal{A}$ do
    Return $b$ informative deep learning acquisitions in one budget:
      $\{x_1^*, x_2^*, ..., x_b^*\} \leftarrow \arg\max_{x \in \mathcal{D}_u} \mathrm{H}[\theta|\mathcal{D}_0] - \mathbb{E}_{y \sim p(y|x, \mathcal{D}_0)} \Big[ \mathrm{H}[\theta|x, y, \mathcal{D}_0] \Big]$.
    Rank $b'$ informative acquisitions with the highest geometric
      representativeness:
      $\{x_{t_1}^*, ..., x_{t_{b'}}^*\} = \arg\max_{x_{t_1:t_{b'}}^* \subseteq \{x_1^*, x_2^*, ..., x_b^*\}} p(y_{t_1:t_{b'}}^* | x_{t_1:t_{b'}}^*, \theta)$.
    Update $\{x_{t_1}^*, ..., x_{t_{b'}}^*\}$ into $\mathcal{D}_0$: $\mathcal{D}_0 \leftarrow \mathcal{D}_0 \cup \{x_{t_1}^*, ..., x_{t_{b'}}^*\}$.
    $\alpha \leftarrow \alpha + 1$.
end
Output: final update on $\mathcal{D}_0$.
```

- Maximize Variation Ratio (Var) [Gal et al., 2017]. The algorithm chooses the unlabeled data that maximizes its variation ratio of the probability:

$$x^* = \arg\max_{x \in \mathcal{D}_u} \Big\{ 1 - \max_{y \in \mathcal{Y}} \mathrm{Pr}(y|, x, \mathcal{D}_0)) \Big\}. \tag{4.15}$$

- Maximize Entropy (Entropy) [Gal et al., 2017]. The algorithm chooses the unlabeled data that maximizes the predictive entropy:

$$x^* = \arg\max_{x\in\mathcal{D}_u}\Big\{-\sum_{y\in\mathcal{Y}}\Pr(y|x,\mathcal{D}_0))\log\Big(\Pr(y|x,\mathcal{D}_0)\Big)\Big\}. \tag{4.16}$$

- k-modoids [Park and Jun, 2009]. A classical unsupervised algorithm that represents the input distribution with k clustering centers:

  $$\{x_1^*, x_2^*, ..., x_k^*\} = \arg\min_{z_1,z_2,\dots,z_k}\Big\{\sum_{i=1}^{k}\sum_{z_i\in\mathcal{X}^k}\|x_i - z_i\|\Big\}, \tag{4.17}$$

  where $\mathcal{X}^k$ denotes the k-th subcluster centered with z_i, and $z_i \in \mathcal{X}, \forall i$..

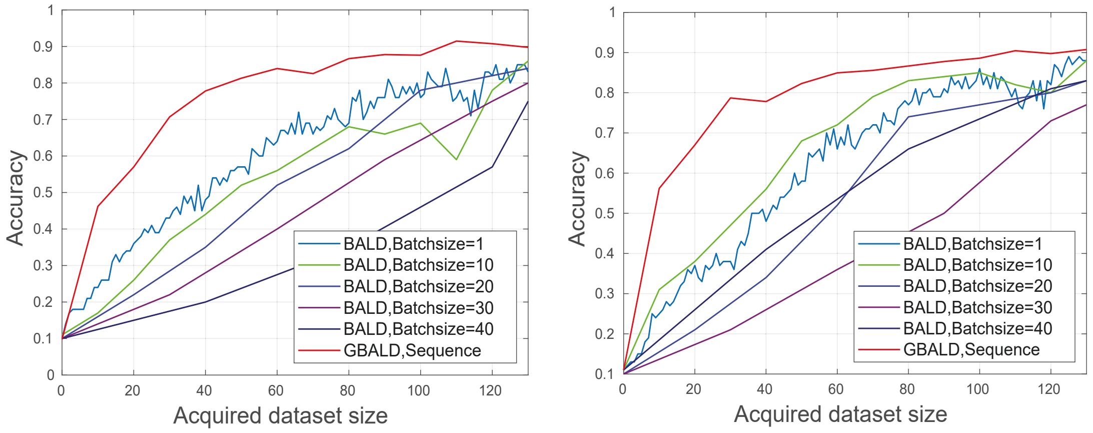
- Greedy k-centers (k-centers) [Sener and Savarese, 2018b]. A geometric core-set interpretation on sphere. See Eq. (4.4).

- BatchBALD [Kirsch et al., 2019]. A batch extension of BALD which incorporates the diversity, not maximal entropy as BALD, to rank the acquisitions:

  $$\{x_{t_1}^*, ..., x_{t_b}^*\} = \arg\max_{x_{t_1},\dots,x_{t_b}} \mathrm{H}(y_{t_1}, ..., y_{t_b}) - \mathrm{E}_{p(\theta|\mathcal{D}_0)}[\mathrm{H}(y_{t_1}, ...y_{t_b}|\theta)], \tag{4.18}$$

  where $\mathrm{H}(y_{t_1}, ..., y_{t_b})$ denotes the entropy over all possible labels from y_{t_1} to y_{t_b} such that $\mathrm{H}(y_{t_1}, ..., y_{t_b}) = \mathrm{E}_p(y_{t_1}, ...y_{t_b})[-\log p(y_{t_1}, ...y_{t_b}]$, and the expected entropy over $\mathrm{E}_{p(\theta|\mathcal{D}_0)}[\mathrm{H}(y_{t_1}, ...y_{t_b}|\theta)]$ is estimated by MC sampling [Roy and McCallum, 2001] [Osborne et al., 2012] a subset from $\mathcal{X}$ which approximates the parameter distributions of θ.

The parameter settings of Eq. (4.5) are $R_0 = 2.0e + 3$ and η =0.9. Accuracy of each acquired dataset size of the experiments are averaged over 3 runs.



(a) Digit ‘0’

(b) Digit ‘1’

Figure 4.3 : Acquisitions with uninformative priors from digit ‘0’ and ‘1’.

4.3.2 Uninformative Priors

As discussed in the introduction, BALD is sensitive to an uninformative prior, i.e. $p(\mathcal{D}_0|\theta)$. We thus initialize $\mathcal{D}_0$ from a fixed class of the tested dataset to observe its acquisition performance. Figure 4.3 presents the prediction accuracies of BALD with an acquisition budget of 130 over the training set of MNIST, in which we randomly select 20 samples from the digit '0' and '1' to initialize $\mathcal{D}_0$, respectively. The classification model of active learning follows a convolutional neural network with one block of [convolution, dropout, max-pooling, relu], with 32, 3x3 convolution filters, 5x5 max pooling, and 0.5 dropout rate. In the active learning loops, we use 2,000 MC dropout samples from the unlabeled data pool to fit the training of the network following [Kirsch et al., 2019].

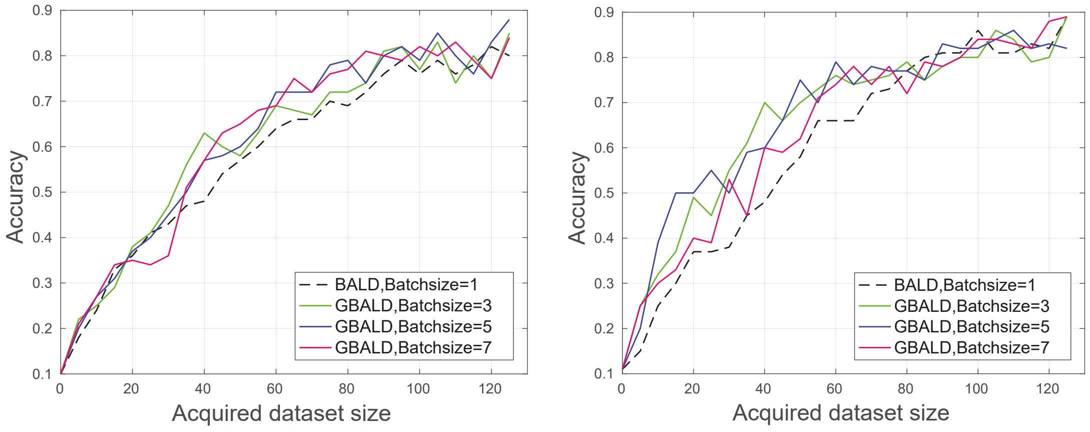
The results show BALD can slowly accelerate the training model due to biased initial acquisitions, which cannot uniformly cover all the label categories. Moreover, the uninformative prior guides BALD to unstable acquisition results. As the shown in Figure 4.3(b), BALD with Bathsize = 10 shows better performance than that of Batchsize =1; while BALD in Figure 4.3(a) keeps stable performance. This is because the initial labeled data does not cover all classes and BALD with Batchsize =1 may further be misled to select those samples from one or a few fixed classes at the first acquisitions. However, Batchsize >1 may result in a random acquisition process that possibly covers more diverse labels at its first acquisitions. Another excursive result of BALD is that the increasing batch size cannot degenerate its acquisition performance in Figure 4.3(b). Specifically, Batchsize =10 $\succ$ Batchsize =1 $\succ$ Batchsize =20,40 $\succ$ Batchsize =30, where '$\succ$' denotes 'better' performance; Batchsize = 20 achieves similar results of Batchsize =40. This undermines the acquisition policy of BALD: its performance would degenerate when the batch size increases.

Different to BALD, the core-set construction of GBALD using Eq. (4.11) provides a complete label matching against all classes. Therefore, it outperforms BALD with the batch sizes of 1, 10, 20, 30, and 40. As the shown learning curves in Figure 4.3, GBALD with a batch size of 1 and sequence size of 10 (i.e. breakpoints of acquired size are 10, 20, ..., 130) achieves significantly higher accuracies than BALD using different batch sizes since BALD misguides the network updating using poor prior.

4.3.3 Improved Informative Acquisitions

Repeated or similar acquisitions delay the acceleration of the model training of BALD. Following the experiment settings of Section 4.5.1, we compare the best performance of BALD with a batch size of 1 and GBALD with different batch size parameters. Following Eq. (4.14), we set $b = \{3, 5, 7\}$ and b'=1, respectively, that means, we output the highest representative data from a batch of highly-informative acquisitions. Different settings on b and b' are used to observe the parameter perturbations of GBALD.

Training by the same parameterized CNN model as Section 4.3.2, Figure 4.4 presents the acquisition performance of parameterized BALD and GBALD. As the learning curves shown, BALD cannot accelerate the model as fast as GBALD due to the repeated information over the acquisitions. For GBALD, it ranks the batch acquisitions of the highly-informative samples and selects the highest representative ones. By employing this special ranking strategy, GBALD can reduce the probability of sampling nearby data of the



(a) Digit ‘0’ (b) Digit ‘1’

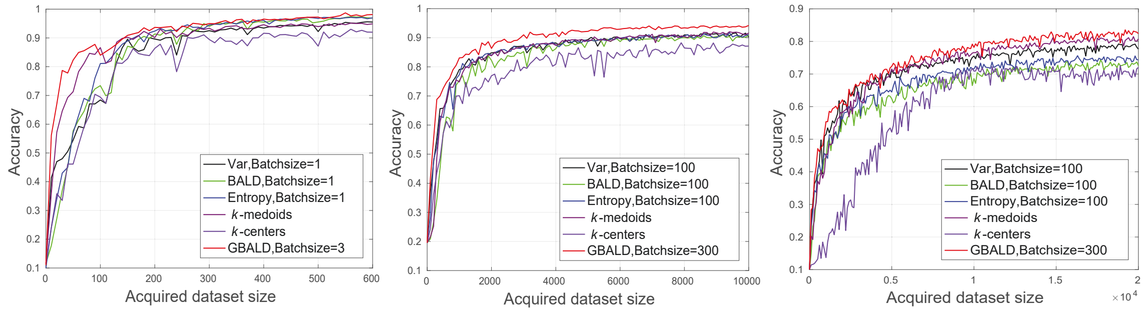
Figure 4.4 : GBALD outperforms BALD using ranked informative acquisitions which cooperate with representation constraints.

previous acquisitions. It is thus GBALD significantly outperforms BALD, even if we progressively increase the ranked batch size b.

4.3.4 Active Acquisitions

GBALD using Eqs. (4.11) and (4.14) has been demonstrated to achieve successful improvements over BALD. We thus combine these two components into a uniform framework. Figure 4.5 reports the active learning accuracies using different acquisition algorithms on the three image datasets. The selected baselines follow [Gal et al., 2017] including 1) maximizing the variation ratios (Var), 2) BALD, 3) maximizing the entropy (Entropy), 4) k-medoids, and one greedy 5) k-centers approach [Sener and Savarese, 2018b]. The network architecture is a three-layer MLP with three blocks of [convolution, dropout, max-pooling, relu], with 32, 64, and 128 3x3 convolution filters, 5x5 max pooling, and 0.5 dropout rate. In the active learning loops, the MC dropout still randomly samples 2,000 data from the unlabeled data pool to approximate the training of the network architecture following [Kirsch et al., 2019]. The initial labeled data of MNIST, SVHN and CIFAR-10 are 20, 1000, 1000 random samples from their full training sets.

The batch size of the compared baselines is 100, where GBALD ranks 300 acquisitions to select 100 data for the training, i.e. $b = 300, b' = 100$. As the learning curves shown in Figure 4.5, 1) k-centers algorithm performs more poorly than other compared baselines because the representative optimization with sphere geodesic usually falls into the selection of boundary data; 2) Var, Entropy and BALD algorithms cannot accelerate the network model rapidly due to highly-skewed acquisitions towards few fixed classes at its first acquisitions (start states); 3) k-medoids approach does not interact with the neural network model while directly imports the clustering centers into its training set; 4) The accuracies of the acquisitions of GBALD achieve better performance at the beginning than the Var, Entropy and BALD approaches which fed the training set of the network model via acquisition loops. In short, the network is improved faster after drawing the distri-



(a) MNIST (b) SVHN (c) CIFAR10

Figure 4.5 : Active acquisitions on MNIST, SVHN, and CIFAR10 datasets.

Table 4.1 : Mean±std of the test accuracies of the breakpoints of the learning curves on MNIST, SVHN, and CIFAR-10.

Datasets	Algorithms					
	Var	BALD	Entropy	k-medoids	k-centers	GBALD
MNIST	0.8419± 0.1721	0.8645±0.1909	0.8498±0.2098	0.8785±0.1433	0.8052±0.1838	**0.9106±0.1296**
SVHN	0.8535±0.1098	0.8510±0.1160	0.8294±0.1415	0.8498±0.1294	0.7909±0.1235	**0.8885±0.1054**
CIFAR-10	0.7122±0.1034	0.6760±0.1023	0.6536±0.1038	0.71837±0.1245	0.5890±0.1758	**0.7440±0.1087**

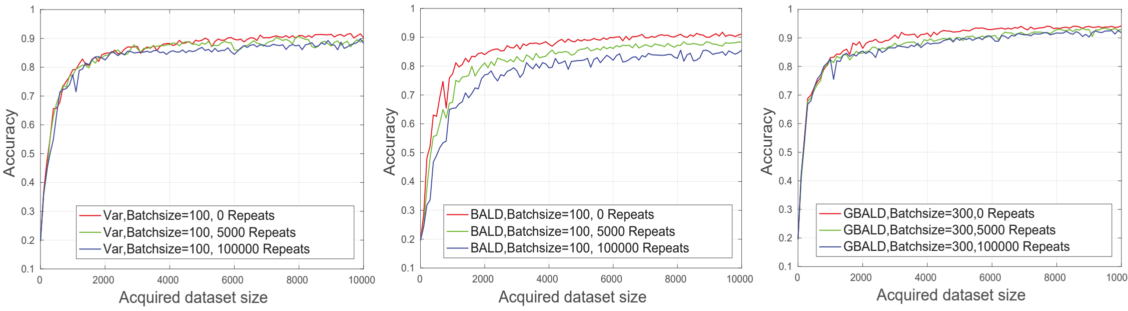
bution characteristics of the input dataset with sufficient labels. GBALD thus consists of the representative and informative acquisitions in its uniform framework. Advantages of these two acquisition paradigms are integrated and present higher accuracies than any single paradigm.

Table 4.1 reports the mean±std values of the test accuracies of the breakpoints of the learning curves in Figure 4.5, where breakpoints of MNIST are $\{$0,10,20,30,...,600$\}$, breakpoints of SVHN are $\{0, 100, 200, ..., 10000\}$, and the breakpoints of CIFAR10 are $\{$ 0,100,200,...,20000 $\}$. We then calculate their average accuracies and std values over these acquisition points. As the shown in Table 1, all std values around 0.1, yielding a norm value. Usually, an average accuracy on a same acquisition size with different random seeds of DNNs, will result a small std value. Our mean accuracy spans across the whole learning curve.

The results show that 1) GBALD achieves the highest average accuracies; k-medoids is ranked the second amongst the compared baselines; 2) k-centers has ranked the worst accuracies amongst these approaches; 3) the others, which iteratively update the training model are ranked at the middle including BALD, Var and Entropy algorithms. Table 4.2 shows the acquisition numbers of achieving the accuracies of 70%, 80%, and 90% on the three datasets. The three numbers of each cell are the acquisition numbers over MNIST, SVHN, and CIFAR10, respectively. The results show that GBALD can use fewer acquisitions to achieve a desired accuracy than the other algorithms.

Table 4.2 : Number of acquisitions on MNIST, SVHN and CIFAR10 until 70%, 80%, and 90% accuracies are reached.

Algorithms	Accuracies 70%	 80%	 90%
Var	140/1,700/5,700	150/2,200/>20,000	210/>10,000/>6,100
BALD	110/1,700 /8,800	120 /2,300/>20,000	190/7,100 / >20,000
Entropy	110/1,900/11,200	150/2,400/>20,000	200/8,600/>20,000
k-modoids	70/1,700/5,900	90/2,200/16,000	**170**/6,200 />20,000
k-centers	110/2,000/10,100	150/3,800/>20,000	280/>10,000/>20,000
GBALD	**50/1,400/4,800**	**70/1,900/12,200**	**170/3,900**/>20,000



(a) Var (b) BALD (c) GBALD

Figure 4.6 : Active acquisitions on SVHN with 5,000 and 10,000 repeated samples.

Table 4.3 : Mean±std of active acquisitions on SVHN with 5,000 and 10,000 repeated samples.

Algorithms	Accuracies 0 repeated samples	 5,000 repeated samples	 10,000 repeated samples
Var	0.8535±0.1098	0.8478±0.1074	0.8281±0.1082
BALD	0.8510±0.1160	0.8119±0.1216	0.7689±0.1288
GBALD	**0.8885±0.1054**	**0.8694±0.1032**	**0.8630±0.1002**

4.3.5 Active Acquisitions with Repeated Samples

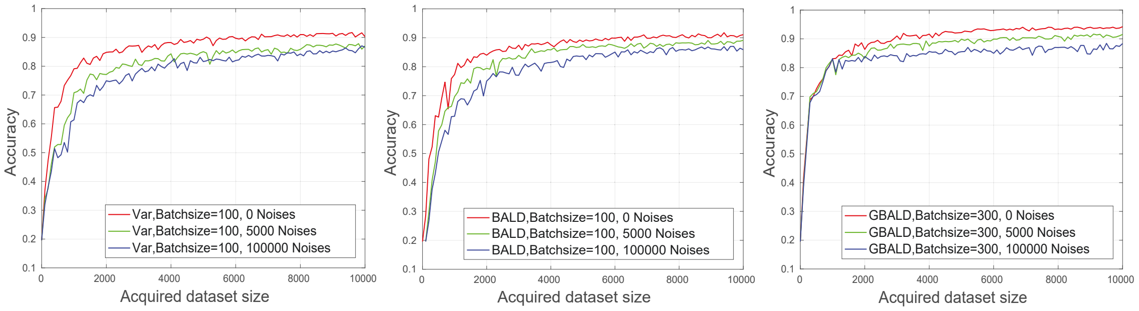
Repeatedly collecting samples in the establishment of a database is very common. Those repeated samples may be continuously evaluated as the primary acquisitions of active learning due to the lack of one or more kinds of class labels. Meanwhile, this situation may lead the evaluation of the model uncertainty to fall into repeated acquisitions. To respond this collecting situation, we compare the acquisition performance of BALD, Var, and GBALD using 5,000 and 10,000 repeated samples from the first 5,000 and 10,000 unlabeled data of SVHN, respectively. In addition, the unsupervised algorithms which do not interact with the network architecture, such as k-medoids and k-centers, have been shown that they cannot accelerate the training in terms of the experiment results of Section 4.3.4. Thus, we are no longer studying their performance. The network architecture still follows the settings of the MLP as Section 4.3.4.

The acquisition results over the repeated SVHN datasets are presented in Figure 4.6 The batch sizes of the compared baselines are 100, where GBALD ranks 300 acquisitions to select 100 data for the training, i.e. $b = 300, b' = 100$. The mean±std values of these baselines of the breakpoints (i.e. $\{0, 100, 200, ..., 10000\}$) are reported in Table 4.3. Results demonstrate that GBALD shows slighter perturbations on repeated samples than Var and BALD because it draws the core-set from the input distribution as the initial acquisition, leading small probability to sample from one or more fixed class. In GBALD, the informative acquisitions constrained with geometric representations further scatter the acquisitions spread in different classes. However, Var and BALD algorithms have no particular schemes against the repeated acquisitions. The maximizer on the model uncertainty may be repeatedly produced by those repeated samples. In additional, the unsupervised algorithms such as k-medoids and k-centers don not have these limitations, but cannot accelerate the training since there has no interactions with the network architecture.

4.3.6 Active Acquisitions with Noisy Samples

Noisy labels [Golovin et al., 2010; Han et al., 2018] are inevitable due to human errors in data annotation. Training on noisy labels, the neural network model will degenerate its inherent properties. To assess the perturbations of the above acquisition algorithms against noisy labels, we organize the following experiment scenarios: we select the first 5,000 and 10,000 samples respectively from the unlabeled data pool of the MNIST dataset and reset their labels by shifting $\{$'0','1',...,'8'$\}$ to $\{$'1','2',...,'9'$\}$, respectively. The network architecture follows MLP of Section 4.3.4. The selected baselines are Var and BALD.

Figure 4.7 presents the acquisition results of those baseline with noisy labels. The batch sizes of the compared baselines are 100, where GBALD ranks 300 acquisitions to select 100 data for the training, i.e. $b = 300, b' = 100$. Table 4.4 presents the mean±std values of the breakpoints (i.e. $\{0, 100, 200, ..., 10000\}$) over learning curves of Figure 4.7. The results further show that GBALD has smaller noisy perturbations than other baselines. For Var and BALD, model uncertainty leads high probabilities to sample those noisy data due to their greatly updating on the model.



(a) Var (b) BALD (c) GBALD

Figure 4.7 : Active noisy acquisitions on SVHN with 5,000 and 10,000 noisy labels.

Table 4.4 : Mean±std of active noisy acquisitions on SVHN with 5,000 and 10,000 noises.

Algorithms	Accuracies		
	0 noises	5,000 noises	10,000 noises
Var	0.8535±0.1098	0.7980±0.1203	0.7702±0.1238
BALD	0.8510±0.1160	0.8205±0.1185	0.7849±0.1239
GBALD	**0.8885±0.1054**	**0.8622±0.0991**	**0.8301±0.0916**

4.3.7 BatchBALD vs. GBALD

Batch active deep learning was recently proposed to accelerate the training of a DNN model. In recent literature, BatchBALD [Kirsch et al., 2019] extended BALD with a batch acquisition setting to converge the network using fewer iteration loops. Different to BALD, BathBALD introduces diversity to avoid repeated or similar output acquisitions.

How to set the batch size of the acquisitions attracted our eyes before starting the experiments. It involves with whether our experiment settings are fair and reasonable. From a theoretical view, the larger the batch size, the worse the batch acquisitions will be. Experiments results of [Kirsch et al., 2019] also demonstrated this phenomenon. We thus set different batch sizes to run BatchBALD. Figure 4.8 reports the comparison results of BALD, BatchBALD, and our proposed GBALD following the experiment settings of Section 4.3.4. As the shown in this figure, BatchBALD degenerates the test accuracies if we progressively increase the bath sizes, where BatchBALD with a batch size of 10 keeps similar learning curves as BALD. This means BatchBALD actually can accelerate BALD with a similar acquisition result if the batch size is not large. That means, if the batch size is between 2 to 10, BatchBALD will degenerate into BALD and maintains highly-consistent results.

Also because of this, BatchBALD has the same sensitivity to the uninformative prior. For our GBALD, the core-set solicits sufficient data which properly matches the input distribution (w.r.t. acquired data set size ≤ 100), providing powerful input features to start the DNN model (w.r.t. acquired data set size > 100). Table 4.5 then presents the mean±std

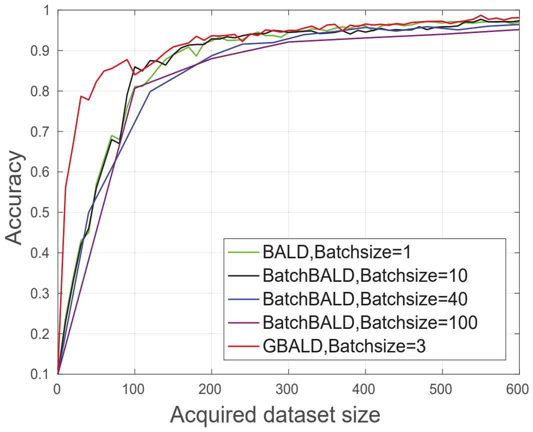


Figure 4.8 : Comparisons of BALD, BatchBALD, and GBALD of active acquisitions on MNIST with bath settings.

Table 4.5 : Mean±std of BALD, BatchBALD, and GBALD of active acquisitions on MNIST with batch settings.

Algorithms	Batch sizes	Accuracies
BALD	1	0.8654±0.0354
BatchBALD	10	0.8645±0.0365
BatchBALD	40	0.8273±0.0545
BatchBALD	100	0.7902±0.0951
GBALD	3	0.9106±0.1296

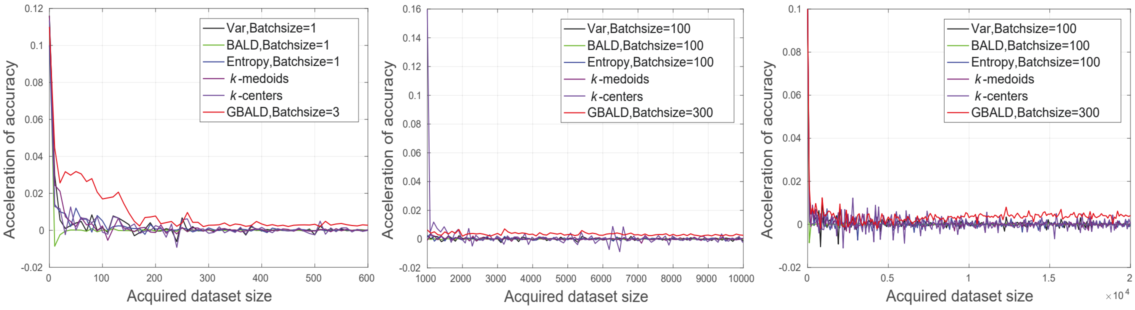
of breakpoints ($\{0, 10, 20, ..., 600\}$) of active acquisitions on MNIST with batch settings. The statistical results show GBALD has much higher mean accuracy than BatchBALD with different bath sizes. Therefore, evaluating the model uncertainty of DNN using highly-representative core-set samples can improve the performance of the neural network.

4.4 Discussions

4.4.1 Acceleration of Accuracy

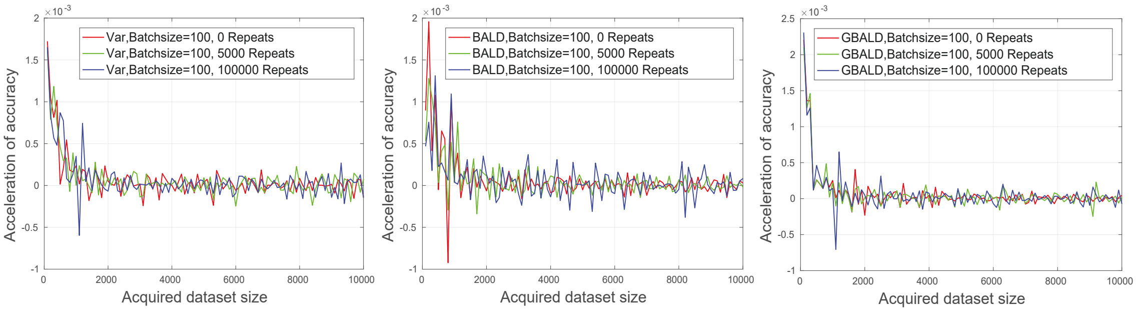
Accelerations of accuracy i.e. the first-orders of breakpoints of the learning curve, describe the efficiency of the active acquisition loops. Different to the accuracy curves, the acceleration curve reflects how active acquisitions help the convergence of the interacting DNN model.

We thus firstly present the acceleration curves of different baselines on MNIST, SVHN, and CIFAR10 datasets following the experiments of Section 4.3.4. The acceleration curves of active acquisitions are drawn in Figure 4.9. Observing those acceleration curves of different algorithms clearly finds that, GBALD always keeps higher accelerations of



(a) MNIST (b) SVHN (c) CIFAR10

Figure 4.9 : Accelerations of accuracy of different baselines on MNIST, SVHN, and CIFAR10 datasets.



(a) Var (b) BALD (c) GBALD

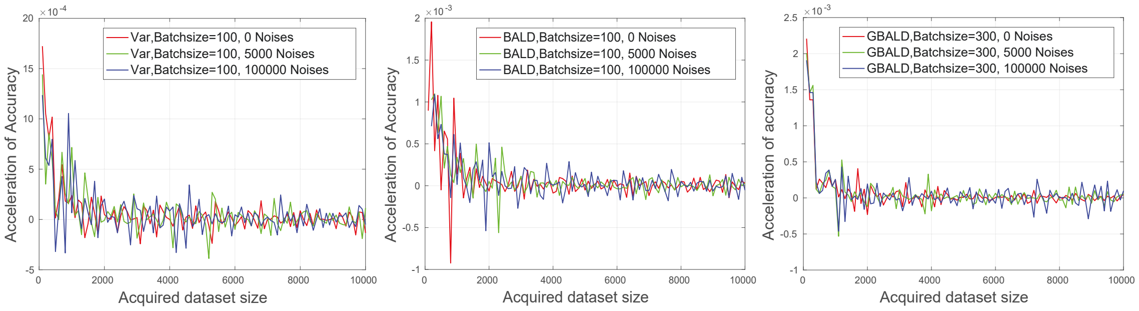
Figure 4.10 : Accelerations of accuracy of active acquisitions on SVHN with 5,000 and 10,000 repeated samples.

accuracy than the other baselines against the three benchmark datasets. This revels the reason of why GBALD can derive more informative and representative data to maximally update the DNN model.

The acceleration curves of active acquisitions with repeated samples are presented in Figure 4.10. As the shown in this figure, GBALD presents slighter perturbations to the number of repeated samples than that of Var and BALD due to its effective ranking scheme on optimizing model uncertainty of DNN. The acceleration curves of active noisy acquisitions are drawn in Figure 4.11. Compared to Figure 4.7, it presents more intuitive descriptions for the noisy perturbations to different baselines. With horizontal comparisons to acceleration curves of Var and BALD, our proposed GBALD has smaller noisy perturbations due to 1) the powerful core-set which properly captures the input distribution, and 2) highly representative and informative acquisitions of model uncertainty.

4.4.2 Hyperparameter Settings

What is the proper time to start active acquisitions using Eq. (4.14) in GBALD framework? Does the ratio of core-set and model uncertainty acquisitions affect the perfor-



(a) Var (b) BALD (c) GBALD

Figure 4.11 : Accelerations of accuracy of active noisy acquisitions on SVHN with 5,000 and 10,000 noisy labels.

Table 4.6 : Relationship of accuracies and sizes of core-set on SVHN.

Size of core-set	Accuracies		
	Start accuracy	Ultimate accuracy	Mean±std accuracy
$N_{\mathcal{M}}$ = 1,000	0.8790	0.9344	0.9134±0.0169
$N_{\mathcal{M}}$ = 2,000	0.8898	0.9212	0.9151±0.0148
$N_{\mathcal{M}}$ = 3,000	0.8848	**0.9364**	0.9173±0.0138
$N_{\mathcal{M}}$ = 4,000	0.8811	0.9271	0.9146±0.0165
$N_{\mathcal{M}}$ = 5,000	**0.8959**	0.9342	**0.9197±0.0117**

mance of GBALD?

We discuss the key hyperparameter of GBALD here: core-set size $N_{\mathcal{M}}$. Table 4.6 presents the relationship of accuracies and the sizes of core-set, where the start accuracy denotes the test accuracy over the initial core-set, and the ultimate accuracy denotes the test accuracy over up to $Q = 20,000$ training data. Let $b = 1000, b' = 500$ in GBALD, $\mathcal{N}_{\mathcal{M}}$ be the number of the core-set size, the iteration budget $\mathcal{A}$ of GBALD then can be defined as $\mathcal{A} = (Q - \mathcal{N}_{\mathcal{M}})/b'$. For example, if the number of the initial core-set labels are set as $\mathcal{N}_{\mathcal{M}} = 1,000$, we have $\mathcal{A} = (Q - \mathcal{N}_{\mathcal{M}})/b' \approx 38$; if $\mathcal{N}_{\mathcal{M}} = 2,000$, then $\mathcal{A} = (Q - \mathcal{N}_{\mathcal{M}})/b' \approx 36$.

From Table 4.6, GBALD algorithm keep stable accuracies over the start, ultimate, and mean±std accuracies when there inputs more than 1,000 core-set labels. Therefore, drawing sufficient core-set labels using Eq. (4.11) to start the model uncertainty of Eq. (4.14) can maximize the performance of our GBALD framework.

Hyperparameter settings on batch returns b and bath outputs b'. Experiments of Sections 4.3.2 and 4.3.3 used different b and b' to observe the parameter perturbations. No matter what the settings of b' and b are, GBALD still outperforms BALD. For single acquisition of GBALD, we suggest $b = 3$ and $b' = 1$. For bath acquisitions, the settings on b' and b are user-defined according the time cost and hardware resources.

Hyperparameter setting on iteration budget $\mathcal{A}$. Given the acquisition budget Q, let b' be the number of the output returns at each loop, $\mathcal{N}_{\mathcal{M}}$ be the number of the core-set size, the iteration budget $\mathcal{A}$ of GBLAD then can be defined as $\mathcal{A} = (Q - \mathcal{N}_{\mathcal{M}})/b'$.

Other hyperparameter settings. Eq. (4.5) has one parameter R_0 which describes the geometry prior from probability. The default radius of the intern balls R_0 is used to legalize the prior and has no further influences on Eq. (4.10). It is set as $R_0 = 2.0e + 3$ for those three image datasets. Ellipsoid geodesic is adjusted by η which controls how far of the updates of core-set to the boundaries of the distributions. It is set as $\eta = 0.9$ in this paper.

Advantages of GBALD. The aforementioned scenarios of active learning in Chapters 2 and Chapter 3 are insufficient labels and null hypothesis, respectively. Therefore, they are negative settings. For agnostic scenarios, GBALD may have advantages, i.e., it performs robustly whether in setting of insufficient labels or null hypothesis.

Time complexity of GBALD. In the first stage of GBALD, the core-set construction costs at most $O(NN_{\mathcal{M}})$. For the second stage, the time complexity of the Bayesian model is decided by the parameterized network configuration.

4.4.3 Two-sided t-test

We present two-sided (two-tailed) t-test for the learning curves of Figure 4.5. Different to the mean± std of Table 4.1, t-test can enlarge the significant difference of those baselines. In typical t-test, the two groups of observations usually require a degree of freedom smaller than 30. However, the numbers of breakpoints of MNIST, SVHN, and CIFAR10 are 61, 101, and 201, respectively, thereby holding a degree of freedom of 60, 100, 200, respectively. It is thus we introduce t-test score to directly compare the significant difference of pairwise baselines.

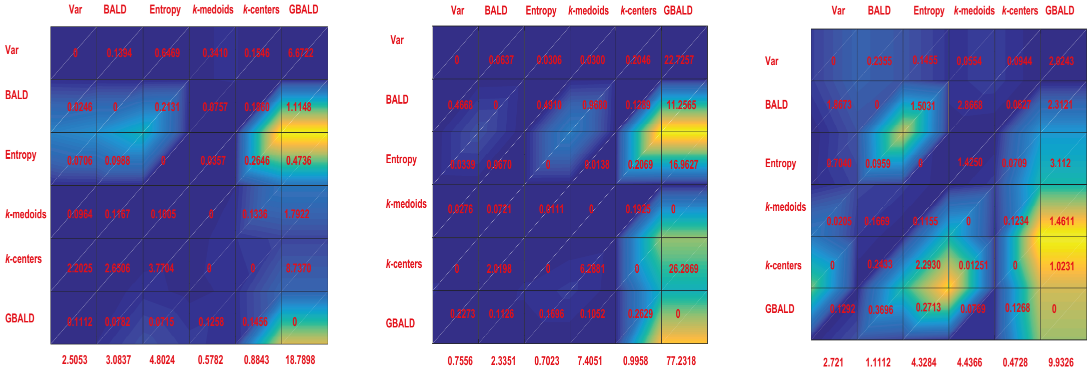
t-test score between any pair group of breakpoints are defined as follows. Let $B_1 = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ and $B_2 = \{\beta_1, \beta_2, ..., \beta_n\}$, there exists t-score of

$$t - \text{score} = \sqrt{n}\frac{\mu}{\sigma},$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n}(\alpha_i - \beta_i)$, and $\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\alpha_i - \beta_i - \mu)^2}$.

In two-sided t-test, B_1 beats B_2 on breakpoints α_i and β_i satisfying a condition of $t-\text{score} > \nu$; B_2 beats B_1 on breakpoints α_i and β_i satisfying a condition of $t-\text{score} < -\nu$, where ν denotes the hypothesized criterion with a given confidence risk. Following [Ash et al., 2019], we add a penalty of $\frac{1}{e}$ to each pair of breakpoints, which further enlarges their differences in the aggregated penalty matrix, where e denotes the number of B_1 beats B_2 on all breakpoints. All penalty values finally calculate their L_1 expressions.

Figure 4.12 presents the penalty matrix over learning curves of Figure 4.5. Column-wise values at the bottom of each matrix show the overall performance of the compared baselines. As the shown results, GBALD has significant performances than that of the other baselines over the three datasets. Especially for SVHN, it has superior performance.



(a) MNIST (b) SVHN (c) CIFAR10

Figure 4.12 : A pairwise penalty matrix over active acquisitions on MNIST, SVHN, and CIFAR10. Column-wise values at the bottom of each matrix show the overall performance of the compared baselines (larger value has more significant superior performance).

4.5 Summary of This Chapter

This chapter introduced a novel Bayesian active learning framework, GBALD, from the geometric perspective, which seamlessly incorporates representative (core-set) and informative (model uncertainty estimation) acquisitions to accelerate the training of a DNN model. Our GBALD yields significant improvements over BALD, flexibly resolving the limitations of an uninformative prior and redundant information by optimizing the acquisition on an ellipsoid. Compared to the representative or informative acquisition algorithms, experiments show that our GBALD spends much fewer acquisitions to accelerate the accuracy. Moreover, it keeps slighter accuracy reduction than other baselines against repeated and noisy acquisitions.

Chapter 5

Conclusion

5.1 Thesis Summarization

Active learning of halfspace provides theoretical guarantees for realizable supervision sampling with distribution and noise assumptions. The typical error disagreement coefficient derives iterative pruning in the hypothesis class over the version space. However, the pruning process is limited by the initialization of the input hypothesis and the estimation of the coefficient. This generates a challenging gap between those theoretical guarantees and application performance of active learning. Three questions thus arising from theoretical view to realizable framework were studied in this thesis.

- **How to reduce the typical theoretical bounds of label complexity?** Chapter 2 answered the first question by proposing a novel perspective of shattering the input distribution, which characterizes any hypothesis using a lower bound on the VC dimension. With lower generalization error and label complexity in the shattered distribution, an shattering algorithm termed SDAL is proposed, which yields slighter perturbations to adversarial and noisy samples than other typical active learning algorithms.
- **How to control hypothesis update without errors when estimating the error disagreement is infeasible?** Chapter 3 answered the second question by proposing a novel distribution disagreement graph coefficient, which is a feasible alternative against active learning without sufficient supervisions. Our theoretical results further proved that the proposed coefficient yields tighter label complexity bound than that of error disagreement. Generalization of distribution disagreement via focal representation in hyperbolic space showed significant accuracy improvements than the other related baselines.
- **How to improve deep active learning based on the theoretical results of halfspace learning?** Chapter 4 answered the third question by proposing a geometric Bayesian active learning framework, which incorporates the core-set construction and model uncertainty estimation, interacting with a deep learning model. Experiment results showed that the derived two-stage GBALD algorithm can spend fewer labels to achieve a desired accuracy than other state-of-the-art active deep learning baselines.

5.2 Future Work

There still remains some potential work in future.

- **Active teaching: explain active learning from machine teaching.** Performance disagreements of hypothesis-pruning and distribution-shattering strategies should be studied in different distribution assumptions and noise conditions. Closed-form learning functions also can be produced to observe the potential bounds on error and label complexity. This can be further studied by machine teaching [Liu et al., 2018], which is an inverse problem of machine learning. In machine teaching, the teacher steers the student learner towards its target hypothesis, which assumes the teacher has already known the learning parameters of the model. Given the parameter distributions, machine teaching can help active learner to predict its feasible error threshold and estimate the label complexity bounds. It is thus active teaching can be a possible direction to improve our work in future.

- **Safety guarantees of aspherical focal representation.** The focal representations adopted in a tree-likeness splitting manner can accelerate the training of a deep neural network using fewer labels than centroid-based active learning algorithms. However, the input distribution is assumed as aspherical, which lacks theoretical guarantees. Moreover, the first splitting of the tree-likeliness algorithm affects the subsequent splitting process, which needs more safety guarantees. Otherwise, the splitting would be misled. In importance sampling, learning in surrogate representations of input distribution can keep consistent properties for the machine learning model but eliminates the perturbations from low-informative samples. Theoretically, a desired safety guarantee [Beygelzimer et al., 2009b] expects that the performance of a machine learning algorithm keeps a provable consistency on its inherent optimal hypothesis. Therefore, developing a set of safety guarantee theories can reduce the gaps between aspherical focal and spherical centroid representations.

- **Auto-active learning.** What is the proper time to start the model uncertainty estimation of a deep learning mode? It also means that how to settle the size of the initial core-set? Introducing auto-machine learning may derive an explicit discriminator to maximize the potential performance of GBALD. This discriminator can be generalized from automated machine learning [Feurer et al., 2015], which establishes a uniform framework to develop a complete pipeline from input distribution, deriving the best model parameters, desired training set, etc. It is thus building an auto-active learning pipeline can allow non-expert learners to intelligently select their desired annotations without the iterative querying from the unlabeled data pool.

References

I. M. Alabdulmohsin, X. Gao, and X. Zhang. Efficient active learning of halfspaces via query synthesis. In *Association for the Advancement of Artificial Intelligence*, pages 2483–2489, 2015.

J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2019.

A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2019.

P. Awasthi, M. F. Balcan, and P. M. Long. The power of localization for efficiently learning linear separators with noise. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pages 449–458. ACM, 2014.

M. Bādoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 250–257, 2002.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *International Conference on Machine Learning*, pages 65–72. ACM, 2006.

M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Computational Learning Theory*, pages 35–50. Springer, 2007.

M.-F. Balcan, S. Hanneke, and J. W. Vaughan. The true sample complexity of active learning. *Machine learning*, 80(2-3):111–139, 2010.

M.-F. F. Balcan and V. Feldman. Statistical active learning algorithms. In *Advances in neural information processing systems*, pages 1295–1303, 2013.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 49–56, 2009a.

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 49–56, 2009b.

A. Beygelzimer, D. J. Hsu, J. Langford, and T. Zhang. Agnostic active learning without constraints. In *Neural Information Processing Systems*, pages 199–207, 2010.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *In ICML*. Citeseer, 2001.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622, 2015.

S. Bratieres, N. Quadrianto, and Z. Ghahramani. Gpstruct: Bayesian structured prediction using gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1514–1520, 2014.

G. Brightwell and P. Winkler. Counting linear extensions is# p-complete. In *Proceedings of the twenty-third annual ACM Symposium on Theory of Computing*, pages 175–181. ACM, 1991.

W. Buntine and T. Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8(1):75–85, 1992.

S. Burkhardt, J. Siekiera, and S. Kramer. Semisupervised bayesian active learning for text classification. In *Bayesian Deep Learning Workshop at NeurIPS*, 2018.

D. Cai and X. He. Manifold adaptive experimental design for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 24(4):707–719, 2012.

T. Campbell and T. Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, pages 698–706, 2018.

T. Campbell and T. Broderick. Automated scalable bayesian inference via hilbert coresets. *The Journal of Machine Learning Research*, 20(1):551–588, 2019.

X. Cao and I. W. Tsang. Shattering distribution for active learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

X. Cao, I. W. Tsang, and G. Xu. A structured perspective of volumes on active learning. *arXiv:1807.08904*, 2018.

C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

L. Chen, S. H. Hassani, and A. Karbasi. Near-optimal active learning of halfspaces via query synthesis in the noisy setting. In *Association for the Advancement of Artificial Intelligence*, pages 1798–1804, 2017.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

C. Cortes, G. DeSalvo, C. Gentile, M. Mohri, and N. Zhang. Region-based active learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2801–2809. PMLR, 2019a.

C. Cortes, G. DeSalvo, M. Mohri, N. Zhang, and C. Gentile. Active learning with disagreement graphs. In *Proceedings of the 36th International Conference on Machine Learning, International Conference on Machine Learning 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1379–1387, 2019b.

C. Cortes, G. DeSalvo, M. Mohri, N. Zhang, and C. Gentile. Active learning with disagreement graphs. In *International Conference on Machine Learning*, pages 1379–1387, 2019c.

Z. Cranko, A. K. Menon, R. Nock, C. S. Ong, Z. Shi, and C. J. Walder. Monge blunts bayes: Hardness results for adversarial training. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1406–1415, 2019.

S. Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, volume 18, pages 235–242, 2005.

S. Dasgupta. Coarse sample complexity bounds for active learning. In *Neural Information Processing Systems*, pages 235–242, 2006.

S. Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, 2011.

S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *International Conference on Machine Learning*, pages 208–215, 2008.

S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Computational Learning Theory*, pages 249–263. Springer, 2005.

S. Dasgupta, D. J. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Neural Information Processing Systems*, pages 353–360, 2008.

S. Dasgupta, D. Hsu, S. Poulis, and X. Zhu. Teaching a black-box learner. In *International Conference on Machine Learning*, pages 1547–1555, 2019.

C. De Sa, A. Gu, C. Ré, and F. Sala. Representation tradeoffs for hyperbolic embeddings. *Proceedings of machine learning research*, 80:4460, 2018.

B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao. Exploring representativeness and informativeness for active learning. *IEEE Transactions on Cybernetics*, 47(1):14–26, 2017.

L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 47–58. SIAM, 2003.

M. Fang, J. Yin, L. O. Hall, and D. Tao. Active multitask learning with trace norm regularization based on excess risk. *IEEE Transactions on Cybernetics*, 47(11):3906–3915, 2017.

U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine learning*, 8(1):87–102, 1992.

M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. In *Advances in neural information processing systems*, pages 2962–2970, 2015.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Y. Gal, R. Islam, and Z. Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic neural networks. In *Advances in neural information processing systems*, pages 5345–5355, 2018.

M. Gao, Z. Zhang, G. Yu, S. O. Arik, L. S. Davis, and T. Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

D. Golovin, A. Krause, and D. Ray. Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems*, pages 766–774, 2010.

A. Gonen, S. Sabato, and S. Shalev-Shwartz. Efficient active learning of halfspaces: an aggressive approach. *The Journal of Machine Learning Research*, 14(1):2583–2615, 2013.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *Advances in neural information processing systems*, pages 593–600, 2008.

M. Hamann. On the tree-likeness of hyperbolic spaces. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 164, pages 345–361. Cambridge University Press, 2018.

B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *International Conference on Machine Learning*, pages 353–360. ACM, 2007a.

S. Hanneke. Teaching dimension and the complexity of active learning. In *International Conference on Computational Learning Theory*, pages 66–81. Springer, 2007b.

S. Hanneke. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014. ISSN 1935-8237. doi: 10.1561/2200000037. URL http://dx.doi.org/10.1561/2200000037.

S. Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300, 2004.

A. Harpale and Y. Yang. Active learning for multi-task adaptive filtering. In *International Conference on Machine Learning*, 2010.

D. Haussler. Probably approximately correct learning. In *Proceedings of the eighth National conference on Artificial intelligence-Volume 2*, pages 1101–1108. AAAI Press, 1990.

T. N. Hoang, B. K. H. Low, P. Jaillet, and M. Kankanhalli. Nonmyopic ε-bayes-optimal active learning of gaussian processes. In *International Conference on Machine Learning*, pages 739–747, 2014.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Y. Hu, D. Zhang, Z. Jin, D. Cai, and X. He. Active learning via neighborhood reconstruction. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1415–1421. Citeseer, 2013.

S. Javdani, Y. Chen, A. Karbasi, A. Krause, D. Bagnell, and S. S. Srinivasa. Near optimal bayesian active learning for decision making. In *AISTATS*, volume 14, pages 430–438, 2014.

K. Jedoui, R. Krishna, M. Bernstein, and L. Fei-Fei. Deep bayesian active learning for multiple correct outputs. *arXiv preprint arXiv:1912.01119*, 2019.

C. Jin, L. T. Liu, R. Ge, and M. I. Jordan. On the local minima of the empirical risk. *arXiv preprint arXiv:1803.09357*, 2018.

A. T. Kalai, A. R. Klivans, Y. Mansour, and R. A. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

S. Kanai, Y. Fujiwara, and S. Iwamura. Preventing gradient explosions in gated recurrent units. In *Advances in neural information processing systems*, pages 435–444, 2017.

A. Kirsch, J. van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7024–7035, 2019.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

J. Langford. Tutorial on practical prediction theory for classification. *Journal of machine learning research*, 6(Mar):273–306, 2005.

M. Law, R. Liao, J. Snell, and R. Zemel. Lorentzian distance learning for hyperbolic representations. In *International Conference on Machine Learning*, pages 3672–3681, 2019.

T. Le, H. Vu, T. D. Nguyen, and D. Q. Phung. Geometric enclosing networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 2355–2361, 2018.

N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.

W. Liu, B. Dai, X. Li, Z. Liu, J. Rehg, and L. Song. Towards black-box iterative machine teaching. In *International Conference on Machine Learning*, pages 3147–3155, 2018.

A. Lou, I. Katsman, Q. Jiang, S. Belongie, S.-N. Lim, and C. De Sa. Differentiating through the frechet mean. *ICML*, 2020a.

A. Lou, I. Katsman, Q. Jiang, S. J. Belongie, S.-N. Lim, and C. D. Sa. Differentiating through the fréchet mean. *ICML 2020*, 2020b.

A. Malinin and M. Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058, 2018.

P. Massart, É. Nédélec, et al. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006.

T. Matiisen, A. Oliver, T. Cohen, and J. Schulman. Teacher-student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

A. K. McCallumzy and K. Nigamy. Employing em and pool-based active learning for text classification. In *International Conference on Machine Learning*, pages 359–367. Citeseer, 1998.

S. Mohamad, A. Bouchachia, and M. Sayed-Mouchaweh. A bi-criteria active learning algorithm for dynamic data streams. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):74–86, 2016.

N. Monath, M. Zaheer, D. Silva, A. McCallum, and A. Ahmed. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 714–722. ACM, 2019.

N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Neural Information Processing Systems*, pages 1196–1204, 2013.

M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347, 2017.

M. Nickel and D. Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3776–3785, 2018.

F. Nie, H. Wang, H. Huang, and C. Ding. Early active learning via robust representation and structured sparsity. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

M. Osborne, R. Garnett, Z. Ghahramani, D. K. Duvenaud, S. J. Roberts, and C. E. Rasmussen. Active learning of model evidence using bayesian quadrature. In *Advances in neural information processing systems*, pages 46–54, 2012.

H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

D. T. Pham, S. S. Dimov, and C. D. Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.

R. Pinsler, J. Gordon, E. Nalisnick, and J. M. Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*, pages 6356–6367, 2019.

Z. Qiu, D. J. Miller, and G. Kesidis. A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):917–933, 2016.

M. H. Quang, M. San Biagio, and V. Murino. Log-hilbert-schmidt metric between positive definite operators on hilbert spaces. In *Advances in neural information processing systems*, pages 388–396, 2014.

N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018a.

O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018b. URL `https://openreview.net/forum?id=H1aIuk-RW`.

B. Settles. Active learning literature survey. *Computer Sciences Technical Report, University of Wisconsin, Madison*, 2009.

S. Shalev-Shwartz, O. Shamir, and K. Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM Journal on Computing*, 40(6):1623–1646, 2011.

A. Siddhant and Z. C. Lipton. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, 2018.

M. Tang, X. Luo, and S. Roukos. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 120–127. Association for Computational Linguistics, 2002.

Y.-P. Tang and S.-J. Huang. Self-paced active learning: Query the right thing at the right time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5117–5124, 2019.

T. Tran, T.-T. Do, I. Reid, and G. Carneiro. Bayesian generative active deep learning. In *International Conference on Machine Learning*, pages 6295–6304, 2019.

I. W. Tsang, P.-M. Cheung, and J. T. Kwok. Kernel relevant component analysis for distance metric learning. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 954–959. IEEE, 2005a.

I. W. Tsang, J. T. Kwok, and P.-M. Cheung. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(Apr):363–392, 2005b.

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer, 2015.

S. A. Vavasis. Approximation algorithms for indefinite quadratic programming. *Mathematical Programming*, 57(1-3):279–311, 1992.

Z. Wang, B. Du, W. Tu, L. Zhang, and D. Tao. Incorporating distribution matching into uncertainty for multiple kernel active learning. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

D. Wu. Pool-based sequential active learning for regression. *IEEE Transactions on Neural Networks and Learning Systems*, 30(5):1348–1359, 2018.

S. Yan and C. Zhang. Revisiting perceptron: Efficient and label-optimal learning of halfspaces. In *Neural Information Processing Systems*, pages 1056–1066, 2017.

Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015.

N. Young. *An introduction to Hilbert space*. Cambridge university press, 1988.

K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning*, pages 1081–1088, 2006.

C. Zhang and K. Chaudhuri. Beyond disagreement-based agnostic active learning. In *Advances in Neural Information Processing Systems*, pages 442–450, 2014.

Appendix

A. Proofs of Chapter 2

Proof of Lemma 1

Proof. The proof can be adapted from [Balcan et al., 2007] and [Dasgupta et al., 2005]. ■

Proof of Lemma2

Proof. In the shattered distribution over $\mathcal{H}^*$, let $N_{\mathcal{H}^*} = N(\epsilon, \delta, \mathcal{H}^*)$. Then, we obtain $N_{\mathcal{H}} > N_{\mathcal{H}^*} \geq \text{In}(\frac{1}{2\epsilon})\text{In}(\frac{16}{\epsilon^2}(\frac{1}{\sqrt{2}^{m-2}}\text{In}(\frac{6}{\epsilon}) + \text{In}(\frac{4N^2_{\mathcal{H}^*}}{\delta})))$, in which $N_{\mathcal{H}}$ is described in Balcan et al. [2006]. Then, $\delta'_{\mathcal{H}^*} < \delta'_{\mathcal{H}}$. The lemma is as stated. ■

Proof of Theorem 1

Proof. Following [Dasgupta et al., 2008], assume $\text{err}_{\mathcal{D}}(h_i, \mathcal{Z}) - \text{err}_{\mathcal{D}}(h'_i, \mathcal{Z}) = \mathcal{G} = \sqrt{\mathbb{E}_Z[y^+_{h_i,h'_i}]} + \sqrt{\mathbb{E}_Z[y^-_{h_i,h'_i}]}$ for any $Z \times \{+1, -1\}$, where $\mathcal{G} := \{\mathcal{G} : (h_i, h_i) \in \mathcal{H} \times \mathcal{H}\}$, then the i.i.d sample Z of size t from $\mathcal{D}$ satisfies

$$\begin{aligned} \text{err}_{\mathcal{D}}(h_i, \mathcal{Z}) - \text{err}_{\mathcal{D}}(h'_i, Z) &\leq \text{err}_{\mathcal{D}}(h_i) - \text{err}_{\mathcal{D}}(h'_i) \\ &+ \alpha_t^2 + \alpha_t\left(\sqrt{\mathbb{E}_Z[y^+_{h_i,h'_i}]} + \sqrt{\mathbb{E}_Z[y^-_{h_i,h'_i}]}\right). \end{aligned} \tag{A.1}$$

With a similar inequality in the shattered distribution, let $\alpha'_t = \sqrt{(8/n)\text{In}(8\mathcal{S}(\mathcal{H}^*, 2t)^2)/\delta}$, $\mathcal{G}' := \{\mathcal{G}' : (h_i, h_i) \in \mathcal{H}^* \times \mathcal{H}^*\}$,

$$\begin{aligned} \Delta' \leq & \underbrace{(\text{err}_{\mathcal{D}'}(h_i, Z) - \text{err}_{\mathcal{D}}(h_i, Z)))}_{\gamma_1} \\ & + \underbrace{(\text{err}_{\mathcal{D}'}(h'_i, Z) - \text{err}_{\mathcal{D}}(h'_i, \mathcal{Z}))}_{\gamma_2} + \underbrace{(\alpha_t^2 - \alpha_t'^2)}_{\gamma_3} \\ & + \underbrace{(\alpha_t\mathcal{G} - \alpha_t\mathcal{G}')}_{\gamma_4}. \end{aligned} \tag{A.2}$$

Let us rewrite the above equation as $\Delta' \leq \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4$ for the four parts, where each part is within a pair of brackets. Considering the number density of $\mathcal{D}'$ is smaller than that of $\mathcal{D}$, there exists $\text{err}_{\mathcal{D}'}(h_i) \leq \text{err}_{\mathcal{D}}(h_i), \forall i$ and $\text{err}_{\mathcal{D}'}(h'_i) \leq \text{err}_{\mathcal{D}}(h'_i), \forall i$. Therefore, $\gamma_1, \gamma_2 \leq 0$. For the VC dimension, since $\text{Vcdim}(\mathcal{H}^*) = d' < \text{Vcdim}(\mathcal{H}) = d$, then we have $\mathcal{S}(\mathcal{H}^*, 2n) \leq \mathcal{O}((2n)^{d'}) \leq \mathcal{S}(\mathcal{H}, 2n) \leq \mathcal{O}((2n)^d)$. Then, $\gamma_3 + \gamma_4 \leq 0$. ■

Proof of Lemma 4

Proof. Apply $\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h_{-1}) > \Delta_t$ in Lemma 4, we have $\mathrm{err}_t(h_{+1}) > \beta_t^2$. Then, there exists the inequality of

$$\begin{aligned}\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h_{-1}) = &\underbrace{(\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h^*))}_{\xi_1} \\ &+ \underbrace{(\mathrm{err}_t(h^*) - \mathrm{err}_t(h_{-1}))}_{\xi_2}.\end{aligned} \tag{A.3}$$

Let us rewrite this inequality as $\Delta'' = \xi_1 + \xi_2$, we then have

$$\begin{aligned}\xi_1 &> \sqrt{\mathrm{err}_t(h_{+1})}(\sqrt{\mathrm{err}_t(h_{+1})} - \sqrt{\mathrm{err}_t(h^*)}) \\ &> \beta_t(\sqrt{\mathrm{err}_t(h_{+1})} - \sqrt{\mathrm{err}_t(h^*)}).\end{aligned} \tag{A.4}$$

$$\begin{aligned}\xi_2 &= (\mathrm{err}_t(h^*) - \mathrm{err}_t(h_{+1})) + (\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h_{-1})) \\ &> \sqrt{\mathrm{err}_t(h_{+1})}(\sqrt{\mathrm{err}_t(h^*)} - \sqrt{\mathrm{err}_t(h_{+1})} + \Delta_t \\ &> \beta_t(\sqrt{\mathrm{err}_t(h^*)} - \sqrt{\mathrm{err}_t(h_{+1})} + \beta_t^2 + \Delta_t.\end{aligned} \tag{A.4}$$

Therefore,

$$\begin{aligned}\xi_1 + \xi_2 &> 2\beta_t^2 + \beta_t(\sqrt{\mathrm{err}_t(h_{+1})} + \sqrt{\mathrm{err}_t(h_{-1})}) \\ &> 2\beta_t^2 + \beta_t(\sqrt{\mathrm{err}_t(h_{+1})} - \sqrt{\mathrm{err}_t(h_{-1})}) \\ &> 2\beta_t^2 + \beta_t(\xi_1 + \xi_2).\end{aligned} \tag{A.5}$$

Now, we have $(1 - \beta)(\xi_1 + \xi_2) > 2\beta_t^2$. Then, the lemma follows. ■

Proof of Theorem 2

Proof. Using Lemma 4, we obtain $\mathrm{err}_{D'}(h^*) - \mathrm{err}_{D'}(h_f) \geq \beta_k^2 + \beta_k(\sqrt{\mathrm{err}_k(h_{+1})} + \sqrt{\mathrm{err}_k(h_{-1})})$. Let $\nu = \mathrm{err}_{D'}(h^*)$ then

$$\begin{aligned}\mathrm{err}_{D'}(h_f) &\leq \nu + \beta_k^2 + \beta_k(\sqrt{\nu} + \sqrt{\mathrm{err}_k(h_f)}) \\ &\leq (\sqrt{\nu} + \beta_k)^2.\end{aligned} \tag{A.6}$$

■

B. Proofs of Chapter 3

Proof of Theorem 5.

Proof. In the sampling process, Importance Weighted Active Learning (IWAL) algorithm of [Beygelzimer et al., 2009a] provided the label complexity in terms of θ. The generalization type associated with the probability observations $\mathcal{F}_t$ is defined as:

$$\theta \geq \frac{\mathbb{E}_{x\sim\mathcal{D}}[p_t|\mathcal{F}_{t-1}]}{4K_\ell(R^* + 2\Delta_{t-1})}, \tag{B.1}$$

where K_ℓ is the slope asymmetry that satisfies $K_\ell = \sup_{x,x'} \left| \frac{\max_{y\in\mathcal{Y}} \ell(h(x),y) - \ell(h'(x),y)}{\min_{y\in\mathcal{Y}} \ell(h(x),y) - \ell(h'(x),y)} \right|$. Specifically, slope asymmetry describes the sensitive of loss function $\ell(\cdot,\cdot)$ to the label complexity, that is, a sensitive loss usually derives a coarse estimation on updating hypothesis, then requesting large number of labels to achieve a desired error.

By adopting the approximation condition of Assumption 1, we know:

$$\begin{aligned} \mathfrak{L}(\mathcal{G},\mathcal{G}^*) &:= \mathbb{E}_{h\in\mathcal{H}}\, \ell(h(x),y) - \mathbb{E}_{h^*\in\mathcal{H}}\, \ell(h^*(x),y) \\ &= R(h) - R(h^*). \end{aligned} \tag{B.2}$$

Following the proof in IWAL, for any sampling time t, $R(h^*) \geq R(h) - 4\Delta_{t-1}$. We then know:

$$\mathfrak{L}(\mathcal{G},\mathcal{G}^*) \leq 4\Delta_{t-1}. \tag{B.3}$$

Let $B(\mathcal{G}, r_\mathcal{G})$ be the hypothesis ball over $\mathcal{G}$, where $r_\mathcal{G}$ denotes its hypothesis radius. For any $\mathcal{G}$ embedded in $B(\mathcal{G}, r_\mathcal{G})$, the disagreement between any pair of the hypotheses is smaller than the hypothesis diameter of the ball, i.e. $\mathfrak{L}(\mathcal{G},\mathcal{G}^*) \leq 2r_G$. We thus use $r_G^t = 2\Delta_{t-1}$ to define the radius of $B(\mathcal{G}, r_\mathcal{G})$, which means that $r_\mathcal{G}$ can be used in a sampling process without hypothesis. Further the current hypothesis set $\mathcal{H}_t$ satisfies $\mathcal{H}_t \in B(h^*, r_t) := B(\mathcal{G}^*, r_\mathcal{G})$. Hence for any t, we know:

$$\mathbb{E}_{x\sim\mathcal{D}}[p_t|\mathcal{F}_{t-1}] \leq 2 \mathbb{E}_{t\sim B(h^*,r_t)} r_t := 2\theta_G r_\mathcal{G}^t, \tag{B.4}$$

which leads to the following inequality of

$$\theta_G \geq \frac{\mathbb{E}_{x\sim\mathcal{D}}[p_t|\mathcal{F}_{t-1}]}{4\Delta_{t-1}}. \tag{B.5}$$

The second step is to prove the inequality relationship between θ and θ_G. Based on Eq. (1.2) and (1.4), we know:

$$\begin{aligned} \mathfrak{L}(\mathcal{G},\mathcal{G}') \leq \mathcal{L}(h,h') &\leq \mathbb{E}_{h\in\mathcal{H}}\, \ell(h(x),y) - \mathbb{E}_{h^*\in\mathcal{H}}\, \ell(h^*(x),y) \\ &= 2(R(h) - R(h^*)). \end{aligned} \tag{B.6}$$

In other word, the hypothesis diameter relations of $B(\mathcal{G}^*, r_G)$ and $B(h^*, r)$ satisfies: $DIA(B(\mathcal{G}^*, r_G)) < DIA(B(h^*, r))$, where $DIA(\cdot)$ denotes the hypothesis diameter function. Then, we know their volume relations: $VOL(B(\mathcal{G}^*, r_G)) < VOL(B(h^*, r))$, where $VOL(\cdot)$ denotes the volume function. Thus, we know $B(\mathcal{G}^*, r_G)) \subset B(h^*, r)$. By the definition of θ, we know $\theta \geq \theta_G$. ■

Proof of Theorem 6.

Proof. Given the linearity of Euclidean inner product, we transfer the maximization issue of Theorem 2 into:

$$\max_{\mu \in \mathcal{R}^d} \sum_{i=1}^{n} w_i \langle x_i, \mu \rangle_{\mathcal{R}} = \max_{\mu \in \mathcal{R}^d} \sum_{i=1}^{n} \langle w_i x_i, \mu \rangle_{\mathcal{R}}. \tag{B.7}$$

For any vector $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{R}}$ is maximized when $\mathbf{u} = \mathbf{v}$. Therefore, this maximization issue is equivalent to find the linear relation of $\sum_{i=1}^{n} w_i \langle x_i, \mu \rangle_{\mathcal{R}}$ and μ. Here we define a parameter ψ to describe the above mapping: $\mu = (\psi \sum_{i=1}^{n} w_i x_i)$. The maximization then is to calculate a desired ψ that makes the inner product of $\langle \sum_{i=1}^{n} w_i x_i, \mu \rangle_{\mathcal{R}} = \mathcal{B}$.

Given a vector $u \in \mathbb{R}^d$ that satisfies $\langle u, u \rangle_{\mathcal{R}} = \mathcal{B}$. We define another vector $v = \frac{1}{\|u\|_{\mathcal{R}}} u$ which satisfies $\langle v, v \rangle_{\mathcal{R}} = 1$. Then, we know $\mathcal{B} \langle v, v \rangle_{\mathcal{R}} = \left\langle \sqrt{\mathcal{B}} v, \sqrt{\mathcal{B}} v \right\rangle_{\mathcal{R}} = \mathcal{B}$. Next, we model $\psi \sum_{i=1}^{n} w_i x_i$ and μ by

$$\sqrt{\mathcal{B}} v = \psi \sum_{i=1}^{n} w_i x_i = \mu. \tag{B.8}$$

Let us calculate the norms of $\sqrt{\mathcal{B}} v$ and $\psi \sum_{i=1}^{n} w_i x_i$: $\|\sqrt{\mathcal{B}} v\|_{\mathcal{R}} = \|\psi \sum_{i=1}^{n} w_i x_i\|_{\mathcal{R}}$. We then have $\sqrt{\mathcal{B}} \|v\|_{\mathcal{R}} = \psi \| \sum_{i=1}^{n} w_i x_i\|_{\mathcal{R}}$, i.e. $\sqrt{\mathcal{B}} \langle v, v \rangle_{\mathcal{R}} = \psi \| \sum_{i=1}^{n} w_i x_i\|_{\mathcal{R}} = \sqrt{\mathcal{B}}$. We thus obtain

$$\psi = \frac{\sqrt{\mathcal{B}}}{\| \sum_{i=1}^{n} w_i x_i\|_{\mathcal{R}}}. \tag{B.9}$$

Then, we know the center is formulated as:

$$\mu = \sqrt{\mathcal{B}} \frac{\sum_{i=1}^{n} w_i x_i}{\| \sum_{i=1}^{n} w_i x_i\|_{\mathcal{R}}}. \tag{B.10}$$

■

Proof of Theorem 7.

Proof. With the invertible mapping of $h(u) = \frac{1}{1+\sqrt{1+\sum_{i=1}^{d} u_i^2}} (u_1, u_2, ..., u_d) \in \mathcal{P}^d$ in a hyperboloid model, Nickel and Kiela [2018] provided an equivalent formulation of $d_{\mathcal{P}}$ in the unit hyperboloid mode using a mapping function $h(u)$. When $\mathcal{B} = 1$, the distance function of the hyperboloid model exists following equivalence from the Poincaré distance $d_{\mathcal{P}}$:

$$\begin{aligned} d_{\mathcal{H}}(\mathbf{u},\mathbf{v}) &= d_{\mathcal{P}}(h^{-1}(\mathbf{u}), h^{-1}(\mathbf{v})) \\ &= cosh^{-1}\Big(- \big\langle h^{-1}(\mathbf{u}), h^{-1}(\mathbf{v})\big\rangle_{\mathcal{L}}\Big). \end{aligned} \tag{B.11}$$

Therefore, the minimizer of $\min_{\mu\in\mathcal{P}^{\mathrm{d}}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{w_i} \mathrm{d}_{\mathcal{P}}(\mathrm{x_i},\mu)$ equals $\min_{\mu\in\mathcal{P}^{\mathrm{d}}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{w_i} \mathrm{cosh}^{-1}\Big(- \langle \mathrm{h}^{-1}(\mathrm{x_i}), \mathrm{h}^{-1}(\mu)\rangle_{\mathcal{L}}\Big)$ with same constraints on μ. We can observe this minimization has no closed-form solution due to $cosh^{-1}(\cdot)$ is non-convex. ■

Proof of Proposition 5.

Proof. The proof follows Theorem 6 by revising $v = \frac{1}{\left\| \|u\|_{\mathcal{L}} \right\|} u$ which satisfies $\langle v, v\rangle_{\mathcal{L}} = -\mathcal{B}$ due to the negative characteristics of the Lorentzian inner product. ■

C. Specification of θ and θ_G

Specification of θ. Suppose that the initialized labeled set is the first 20 samples of the digit ‘1’ which has an the best-in-class accuracy of 0.1135 that stipulates the learning risk of h as $R(h) = 1 - 0.1135$. Based on Eq. (3.2), we generalize the hypothesis disagreement by the risks, that is, $\mathcal{L}(h^*(x), h(x)) = |(1 - 0.9980) - (1 - 0.1135)| = 0.8845$.

To derive a realizable hyperparameter θ which estimates the error disagreements, we design a set of $r \in \{1, 2, \cdots, 10\}$ to determine the minimum value of θ, and we then have $\{(r,\theta)\}$ = {(1,0.8845), (2,0.4423), (3,0.2948), (4,0.2211), (5,0.1769), (6,0.1474), (7,0.1264), (8,0.1106), (9,0.0983), (10,0.0885)}, where $\theta = \frac{0.8845}{r}$ following Eq. (3.2). Such diverse settings on the radius r make the AL sampling using θ can start properly.

Specification of θ_G. To derive a realizable θ_G, we use a simple Euclidean distance to measure the distribution disagreement of the input and selected data. Let $\mathfrak{f}(x, x') = \|x - x'\|_2$, then $\mathfrak{L}(\mathcal{G}, \mathcal{G}') = \mathop{\mathbb{E}}_{x\sim\mathcal{G}, x'\sim\mathcal{G}'} \Big[\|x - x'\|_2 \Big]$. With Assumption 1, the optimal hypothesis/subgraph is defined over the full training set. Following the specification of θ, $\mathcal{G}$ is still defined over the first 20 training samples of digit ‘1’ in MNIST.

Based on the generalized function $\mathfrak{L}(\mathcal{G}, \mathcal{G}')$, we know: $\mathfrak{L}(G, \mathcal{G}^*) = 1.2965e + 03$ and $\mathfrak{L}(G, \mathcal{G}) = 1.2603e + 03$. To generalize the proposition of Eq. (3.8) on G and $\mathcal{G}^*$, the distribution disagreement $\mathfrak{L}(G, \mathcal{G}^*)$ is defined as the hypothesis disagreement $\rho(h, h^*)$, which is assumed to be tighter than the another disagreement metric of $\mathcal{L}(h, h^*)$. That is, $\mathfrak{L}(G, \mathcal{G}^*) := \rho(h, h^*) < \mathcal{L}(h, h^*)$. To satisfy the above inequality, following the specification of θ on $\mathcal{L}(h, h^*)$, we have: $\rho(h, h^*) = 1 - \alpha < \mathcal{L}(h, h^*) = 0.8845$. We thus set $\alpha = 0.9900$ to satisfy $1 - 0.9900 < 0.8845$. Then $\mathfrak{L}(G, \mathcal{G})$ is rescaled as

$$\tilde{\mathfrak{L}}(G, \mathcal{G}) = (\mathfrak{L}(G, \mathcal{G})/\mathfrak{L}(G, \mathcal{G}^*)) * \alpha = (1.2603e + 03/1.2965e + 03) * 0.9900 = 0.9624.$$

Based on Eq. (3.3), $\mathfrak{L}(\mathcal{G}^*, \mathcal{G}') = \alpha - \tilde{\mathfrak{L}}(G, \mathcal{G}) = 0.9900 - 0.9624 = 0.0276$.

With the same setting of the radius $r_G = r$, we obtain the following diverse settings on (r_G, θ_G), that is, $\{(r_G, \theta_G)\}$ = $\{$(1,0.0276), (2,0.0138), (3,0.0092), (4,0.0069), (5,0.0055), (6,0.0046), (7,0.0039), (8,0.0034), (9,0.0031), (10,0.0028)$\}$, where $\theta_G = \frac{0.0276}{r}$ following Eq. (3.4). It is thus $\theta \geq \theta_G$ for any r holds with the above test settings. This asserts that our analysis is reasonable for the inequality in Eq. (3.8).