Robust Graph Transduction

Chen Gong

Centre for Quantum Computation & Intelligent Systems

University of Technology, Sydney

A thesis submitted for the degree of

Doctor of Philosophy

May, 2016

I would like to dedicate this thesis to my loving parents and wife.

Acknowledgements

I would like to express my gratitude to all those who helped me during the writing of this thesis.

My deepest gratitude goes first and foremost to my supervisor Prof. Dacheng Tao, for his patient and valuable guidance. I knew little about research when I began my PhD study in UTS. It was Prof. Dacheng Tao who taught me how to find interesting ideas, how to develop solid algorithms, and how to write technical papers all from scratch. He was always ready to help, and was very willing to teach everything he knows to me. Without his illuminating instructions, insightful inspiration, consistent encouragement, and expert guidance, I would not have published papers on the leading journals or conferences in my research field. Therefore, I feel extremely grateful for Prof. Dacheng Tao's effort.

I am also greatly indebted to the Centre for Quantum Computation & Intelligent Systems (QCIS) directed by Prof. Chengqi Zhang. In QCIS, I got the opportunities to learn from many world-famous experts; I met many colleagues with great enthusiasm for scientific research; and I was also permitted to attend many prestigious international conferences related to my research. Studying in QCIS and also UTS will be a fantastic memory that I will never forget.

I would also like to thank Prof. Jie Yang from Shanghai Jiao Tong University. Without his introduction, I would not have the opportunity to study in UTS under the supervision of Prof. Dacheng Tao.

Last but not the least, my gratitude also extends to my family who have been consistently supporting, encouraging and caring for me all of my life!

Abstract

Given a weighted graph, graph transduction aims to assign unlabeled examples explicit class labels rather than build a general decision function based on the available labeled examples. Practically, a dataset usually contains many noisy data, such as the "bridge points" located across different classes, and the "outliers" that incur abnormal distances from the normal examples of their classes. The labels of these examples are usually ambiguous and also difficult to decide. Labeling them incorrectly may further bring about erroneous classifications on the remaining unlabeled examples. Therefore, their accurate classifications are critical to obtaining satisfactory final performance.

Unfortunately, current graph transduction algorithms usually fall short of tackling the noisy but critical examples, so they may become fragile and produce imperfect results sometimes. Therefore, in this thesis we aim to develop a series of robust graph transduction methodologies via iterative or non-iterative way, so that they can perfectly handle the difficult noisy data points. Our works are summarized as follows:

In Chapter 2, we propose a robust non-iterative algorithm named "**L**abel **P**rediction via **D**eformed **G**raph **L**aplacian" (LPDGL). Different from the existing methods that usually employ a traditional graph Laplacian to achieve label smoothness among pairs of examples, in LPDGL we introduce a deformed graph Laplacian, which not only induces the existing pairwise smoothness term, but also leads to a novel local smoothness term. This local smoothness term detects the ambiguity of each example by exploring the associated degree, and assigns confident labels to the examples with large degree, as well as allocates "weak labels" to the uncertain examples with small degree. As a result, the negative effects of outliers and bridge points are suppressed, leading to more robust transduction performance than some existing representative algorithms. Although LPDGL is designed for transduction purpose, we show that it can be easily extended to inductive settings.

In Chapter 3, we develop an iterative label propagation approach, called "**F**ick's **L**aw **A**ssisted **P**ropagation" (FLAP), for robust graph transduction.

To be specific, we regard label propagation on the graph as the practical fluid diffusion on a plane, and develop a novel label propagation algorithm by utilizing a well-known physical theory called Fick's Law of Diffusion. Different from existing machine learning models that are based on some heuristic principles, FLAP conducts label propagation in a "natural" way, namely when and how much label information is received or transferred by an example, or where these labels should be propagated to, are naturally governed. As a consequence, FLAP not only yields more robust propagation results, but also requires less computational time than the existing iterative methods.

In Chapter 4, we propose a propagation framework called "**T**eaching-to-**L**earn and **L**earning-to-**T**each" (TLLT), in which a "teacher" (*i.e.* a teaching algorithm) is introduced to guide the label propagation. Different from existing methods that equally treat all the unlabeled examples, in TLLT we assume that different examples have different classification difficulties, and their propagations should follow a simple-to-difficult sequence. As such, the previously "learned" simple examples can ease the learning for the subsequent more difficult examples, and thus these difficult examples can be correctly classified. In each iteration of propagation, the teacher will designate the simplest examples to the "learner" (*i.e.* a propagation algorithm). After "learning" these simplest examples, the learner will deliver a learning feedback to the teacher to assist it in choosing the next simplest examples. Due to the collaborative teaching and learning process, all the unlabeled examples are propagated in a well-organized sequence, which contributes to the improved performance over existing methods.

In Chapter 5, we apply the TLLT framework proposed in Chapter 4 to accomplish saliency detection, so that the saliency values of all the superpixels are decided from simple superpixels to more difficult ones. The difficulty of a superpixel is judged by its informativity, individuality, inhomogeneity, and connectivity. As a result, our saliency detector generates manifest saliency maps, and outperforms baseline methods on the typical public datasets.

Contents

Contents v

List of Figures viii

List of Tables xiii

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Related Work . . . 5
  - 1.2.1 Non-iterative Methods . . . 6
  - 1.2.2 Iterative Methods . . . 8
- 1.3 Motivations and Contributions . . . 9
- 1.4 Thesis Structure . . . 13
- 1.5 Publications during PhD Study . . . 14

2 Label Prediction Via Deformed Graph Laplacian 16
- 2.1 Transduction In Euclidean Space . . . 18
  - 2.1.1 Sensitivity of γ . . . 21
  - 2.1.2 Sensitivity of β . . . 22
- 2.2 Induction In RKHS . . . 22
  - 2.2.1 Robustness Analysis . . . 23
  - 2.2.2 Generalization Risk . . . 26
  - 2.2.3 Linearization of Kernelized LPDGL . . . 27
- 2.3 Relationship Between LPDGL and Existing Methods . . . 29
- 2.4 Experiments . . . 31
  - 2.4.1 Toy Data . . . 31
    - 2.4.1.1 Transduction on 3D Data . . . 31
    - 2.4.1.2 Visualization of Generalizability . . . 32
  - 2.4.2 Real Benchmark Data . . . 33
  - 2.4.3 UCI Data . . . 34
  - 2.4.4 Handwritten Digit Recognition . . . 39

2.4.5 Face Recognition . . . 40
2.4.5.1 *Yale* . . . 40
2.4.5.2 *LFW* . . . 42
2.4.6 Violent Behavior Detection . . . 43
2.5 Summary of This Chapter . . . 45

3 Fick's Law Assisted Propagation **46**
3.1 Model Description . . . 47
3.2 Convergence Analysis . . . 50
3.3 Interpretation and Connections . . . 53
3.3.1 Regularization Networks . . . 53
3.3.2 Markov Random Fields . . . 54
3.3.3 Graph Kernels . . . 56
3.4 Experimental Results . . . 56
3.4.1 Synthetic Data . . . 57
3.4.2 Real Benchmarks Data . . . 59
3.4.3 UCI Data . . . 61
3.4.4 Handwritten Digit Recognition . . . 62
3.4.5 Teapot Image Classification . . . 64
3.4.6 Face Recognition . . . 64
3.4.7 Statistical Significance . . . 65
3.4.8 Computational Cost . . . 66
3.4.9 Parametric Settings . . . 67
3.4.9.1 Choosing η . . . 67
3.4.9.2 Choosing K . . . 69
3.5 Summary of This Chapter . . . 70

4 Label Propagation Via Teaching-to-Learn and Learning-to-Teach **74**
4.1 A Brief Introduction to Machine Teaching . . . 75
4.2 Overview of the Proposed Framework . . . 76
4.3 Teaching-to-Learn Step . . . 77
4.3.1 Curriculum Selection . . . 78
4.3.2 Optimization . . . 81
4.4 Learning-to-Teach Step . . . 83
4.5 Efficient Computations . . . 86
4.5.1 Commute Time . . . 86
4.5.2 Updating $\mathbf{\Sigma}_{\mathcal{L},\mathcal{L}}^{-1}$ . . . 87
4.6 Robustness Analysis . . . 88
4.7 Physical Interpretation . . . 92
4.8 Experimental Results . . . 94
4.8.1 Synthetic Data . . . 94

4.8.2 UCI Benchmark Data . . . 96
4.8.3 Text Categorization . . . 97
4.8.4 Object Recognition . . . 98
4.8.5 Fine-grained Image Classification . . . 99
4.8.6 Parametric Sensitivity . . . 100
4.9 Summary of This Chapter . . . 101

5 Teaching-to-Learn and Learning-to-Teach For Saliency Detection 103
5.1 A Brief Introduction to Saliency Detection . . . 103
5.2 Saliency Detection Algorithm . . . 105
5.2.1 Image Pre-processing . . . 106
5.2.2 Coarse Map Establishment . . . 107
5.2.3 Map Refinement . . . 107
5.3 Teaching-to-learn and Learning-to-teach For Saliency Propagation . . . 108
5.3.1 Teaching-to-learn . . . 108
5.3.2 Learning-to-teach . . . 111
5.3.3 Saliency Propagation . . . 113
5.4 Experimental Results . . . 114
5.4.1 Experiments on Public Datasets . . . 114
5.4.2 Parametric Sensitivity . . . 117
5.4.3 Failed Cases . . . 117
5.5 Summary of This Chapter . . . 119

6 Conclusion and Future Work 120
6.1 Thesis Summarization . . . 120
6.2 Relationship and Differences among Algorithms . . . 121
6.3 Future Work . . . 122

References 124

List of Figures

1.1 The illustration of graph, where the circles represent the vertices $\mathbf{x}_1 \sim \mathbf{x}_7$ and the lines are edges. The red circle denotes the positive example with label 1, while blue circle denotes the negative example with label -1. The numbers near the edges are weights evaluating the similarity between the two connected vertices. 3
1.2 The transductive results of some representative methods on the *DoubleMoon* dataset. (a) is the initial state with marked labeled examples and difficult bridge points. (b), (c), (d), (e), (f) are incorrect results produced by Zhu and Ghahramani [2002], Zhou and Bousquet [2003], Wang et al. [2009b], Wang et al. [2013] and Wang et al. [2008a], respectively. 10
1.3 The structure of this thesis. 14

2.1 The illustration of local smoothness constraint on *DoubleLine* dataset. A k-NN graph with $k = 2$ is built and the edges are shown as green lines in (a). (b) shows the result without incorporating the local smoothness, and (c) is the result produced by the proposed LPDGL. The labels of "bridge point" under two different simulations are highlighted in (b) and (c), respectively. 17
2.2 The evolutionary process from LPDGL to other typical SSL methods. The dashed line means "infinitely approach to". Note that our LPDGL is located in the central position and other algorithms are derived from LPDGL by satisfying the conditions alongside the arrows. 29
2.3 Transduction on two 3D datasets: (a) and (d) show the initial states of *Cylinder&Ring* and *Knot*, respectively, in which the red triangle denotes a positive example and the blue circle represents a negative example. (b) and (e) are the transduction results of developed LPDGL on these two datasets. (c) and (f) present the results of LPDGL (Linear). 32

2.4 Induction on *DoubleMoon* and *Square&Ring* datasets. (a) and (d) show the initial states with the marked labeled examples. (b) and (e) are induction results, in which the decision boundaries are plotted. (c) and (f) are induction performances produced by LPDGL (Linear). . . 33
2.5 Experimental results on four UCI datasets. (a) and (e) are *Iris*, (b) and (f) are *Wine*, (c) and (g) are *BreastCancer*, and (d) and (h) are *Seeds*. The sub-plots in the first row compare the transductive performance of the algorithms, and the sub-plots in the second row compare their inductive performance. 34
2.6 Empirical studies on the parametric sensitivity of LPDGL. (a) and (e) are *Iris*, (b) and (f) are *Wine*, (c) and (g) are *BreastCancer*, and (d) and (h) are *Seeds*. The sub-plots in the first row show the transductive results, and the sub-plots in the second row display the inductive results. 37
2.7 Experimental results on *USPS* dataset. (a) shows the transductive results, and (b) shows the inductive results. 39

3.1 The parallel between fluid diffusion and label propagation. The left cube with more balls is compared to the example with more label information. The right cube with fewer balls is compared to the example with less label information. The red arrow indicates the diffusion direction. 48
3.2 The propagation results on *Square&Ring* dataset. (a), (b), (c) are propagation processes of FLAP, LGC and LNP, respectively. (d), (e), (f) present the classification results brought by MinCut, HF and NTK. 58
3.3 The propagation results on *DoubleMoon* dataset. (a), (b), (c) are propagation processes of FLAP, LGC and LNP, respectively. (d), (e), (f) present the classification results brought by MinCut, HF and NTK. 59
3.4 The comparison of convergence curves. (a) is the result on *Square&Ring* and (b) is that on *DoubleMoon*. 60
3.5 Classification outputs on imbalanced *DoubleMoon*. (a), (b), (c) and (d) are results of 1:2, 1:5, 1:10 and 1:25 situations, respectively. 61
3.6 Comparison of accuracy and iteration times. (a) (b) denote *Iris*, (c) (d) denote *Wine*, (e) (f) denote *BreastCancer*, and (g) (h) denote *CNAE-9*. 63
3.7 Comparison of accuracy and iteration times on digit recognition dataset. (a), (b) are the curves of accuracy and iteration times with the growing of the labeled examples, respectively. 64
3.8 Experiment on *Teapot* dataset. (a) shows some typical images. (b) is the accuracy curve for comparison. 65
3.9 Experimental results on *LFW* face dataset. (a) shows some representative images, and (b) compares the recognition accuracy. 66

3.10 Distribution of eigenvalues on *Iris*. (a) denotes LNP, (b) denotes LGC and (c) denotes FLAP. Note that the ranges of the three x-axes are different. 70
3.11 The impact of parametric settings on accuracy and iteration times. (a), (b) investigate η, and (c), (d) evaluate K. 71

4.1 The TLLT framework for label propagation. The labeled examples, unlabeled examples, and curriculum are represented by red, grey, and green balls, respectively. The steps of Teaching-to-Learn and Leaning-to-Teach are marked with blue and black dashed boxes. 76
4.2 The toy example to illustrate our motivation. The orientation (left or right) of the spout in each image is to be determined. Labeled positive and negative examples are marked with red and blue boxes, respectively. The difficulties of these examples are illustrated by the two arrows below the images. The propagation sequence generated by the conventional methods Gong et al. [2014b]; Zhou and Bousquet [2003]; Zhu and Ghahramani [2002] is $\{1, 6\} \rightarrow \{2, 3, 4, 5\}$, while TLLT operates in the sequence $\{1, 6\} \rightarrow \{2, 5\} \rightarrow \{3\} \rightarrow \{4\}$. As a consequence, only the proposed TLLT can correctly classify the most difficult images 3 and 4. 77
4.3 The physical interpretation of our TLLT label propagation algorithm. (a) compares the propagation between two examples with equal difficulty to the fluid diffusion between two cubes with same altitude. The left cube with more balls is compared to the examples with larger label value. The right cube with fewer balls is compared to the examples with less label information. The red arrow indicates the diffusion direction. (b) and (c) draw the parallel between fluid diffusion with different altitudes and label propagation guided by curriculums. The lowland "C", highland "B", and source "A" in (b) correspond to the simple vertex $\mathbf{x}_C$, difficult vertex $\mathbf{x}_B$, and labeled vertex $\mathbf{x}_A$ in (c), respectively. Like the fluid can only flow from "A" to the lowland "C" in (b), $\mathbf{x}_A$ in (c) also tends to transfer the label information to the simple vertex $\mathbf{x}_C$. 92

4.4 The propagation process of the methods on the *DoubleMoon* dataset. (a) is the initial state with marked labeled examples and difficult bridge point. (b) shows the imperfect edges during graph construction caused by the bridge point in (a). These unsuitable edges pose a difficulty for all the compared methods to achieve accurate propagation. The second row (c)∼(i) shows the intermediate propagations of TLLT (Norm), TLLT (Entropy), GFHF, LGC, LNP, DLP, and GTAM. The third row (j)∼(p) compares the results achieved by all the algorithms, which reveals that only the proposed TLLT achieves perfect classification while the other methods are misled by the ambiguous bridge point. . . 95
4.5 Example images of *COIL20* dataset. 98
4.6 Example images of *UT Zappos* dataset. 99
4.7 Parametric sensitivity of TLLT. The first, second and third rows correspond to *RCV1*, *COIL20* and *UT Zappos* datasets, respectively. (a), (c) and (e) show the variation of accuracy w.r.t. the kernel width ξ when α is fixed to 1, and (b), (d) and (f) evaluate the influence of the trade-off α to final accuracy under $\xi = 10$. 102

5.1 The results achieved by typical propagation methods and our method on two example images. From left to right: input images, results of Yang et al. [2013b], Jiang et al. [2013], and our method. 104
5.2 The diagram of our detection algorithm. The magenta arrows annotated with numbers denote the implementations of teaching-to-learn and learning-to-teach propagation shown in Fig. 5.3. 106
5.3 An illustration of our teaching-to-learn and learning-to-teach paradigm. In the teaching-to-learn step, based on a set of labeled superpixels (magenta) in an image, the teacher discriminates the adjacent unlabeled superpixels as difficult (blue superpixels) or simple (green superpixels) by fusing their informativity, individuality, inhomogeneity, and connectivity. Then simple superpixels are learned by the learner, and the labeled set is updated correspondingly. In the learning-to-teach step, the learner provides a learning feedback to the teacher to help decide the next curriculum. 109
5.4 The illustrations of individuality (a) and inhomogeneity (b). The region $\mathbf{s}_1$ in (a) obtains larger individuality than $\mathbf{s}_2$, and $\mathbf{s}_3$ in (b) is more inhomogeneous than $\mathbf{s}_4$. 112

5.5 Visualization of the designed propagation process. (a) shows the input image with boundary seeds (yellow). (b) displays the propagations in several key iterations, and the expansions of labeled set $\mathcal{L}$ are highlighted with light green masks. (c) is the final saliency map. The curriculum superpixels of the 2nd iteration decided by informativity, individuality, inhomogeneity, connectivity, and the final integrated result are visualized in (d), in which the magenta patches represent the learned superpixels in the 1st propagation, and the regions for the 2nd diffusion are annotated with light green. 113
5.6 Comparison of different methods on two saliency detection datasets. (a) is MSRA 1000, and (b) is ECSSD. 115
5.7 Visual comparisons of saliency maps generated by all the methods on some challenging images. The ground truth (GT) is presented in the last column. 116
5.8 Parametric sensitivity analyses: (a) shows the variation of F_β^w w.r.t. θ by fixing $N = 400$; (b) presents the change of F_β^w w.r.t. N by keeping $\theta = 0.25$. 117
5.9 Failed cases of our method. (a) shows an example that the object is very similar to the background, in which the correct seed superpixels are marked with magenta. (b) is the imperfect saliency map corresponding to the image in (a). In (c), the targets are completely missed by the convex hull (blue polygon), which leads to the detection failure as revealed by (d). 118

List of Tables

2.1 Experimental results on the benchmark datasets for the variety of transduction algorithms. (The values in the table represent the error rate (%). The best three results for each dataset are marked in red, blue, and green, respectively.) 35
2.2 Summary of four UCI datasets 36
2.3 F-statistics values of inductive algorithms versus LPDGL on UCI datasets. (The records smaller than 4.74 are marked in red, which mean that the null hypothesis is accepted.) 38
2.4 Transductive comparison on *Yale* dataset 41
2.5 Inductive comparison on *Yale* dataset 41
2.6 Transductive comparison on *LFW* dataset 42
2.7 Inductive comparison on *LFW* dataset 43
2.8 Transductive results on *HockeyFight* dataset 44
2.9 Inductive results on *HockeyFight* dataset 44

3.1 FLAP vs. popular graph transduction algorithms. 55
3.2 Performances of all the methods on two synthetic datasets. Each record follows the format ”iteration time/CPU seconds/accuracy”. 57
3.3 Experimental results on the benchmark datasets for the variety of graph transduction algorithms. (The values in the table represent accuracy (%).) 62
3.4 F-statistics values of baselines versus FLAP on four UCI datasets. (The records smaller than 4.74 are marked in red, which mean that 1) the null hypothesis is accepted, and 2) the corresponding baseline algorithm performs comparably to FLAP.) 72
3.5 CPU time (unit: seconds) of various methods. (For each l, the smallest record among iterative methods is marked in red, while the smallest record among non-iterative methods is highlighted in blue.) 73

4.1 Experimental results of the compared methods on four UCI benchmark datasets. (Each record in the table represents the "accuracy ± standard deviation". The highest result obtained on each dataset is marked in bold.) 97
4.2 Accuracy of all methods on the *RCV1* dataset (the highest records are marked in bold). 98
4.3 Accuracy of all methods on the *COIL20* dataset (the highest records are marked in bold). 99
4.4 Accuracy of all methods on *UT Zappos* dataset (highest records are marked in bold). 100

5.1 Average CPU seconds of all the approaches on ECSSD dataset 116