# Graph-Based Semi-Supervised Learning

# Synthesis Lectures on Artificial Intelligence and Machine Learning

A Concise Introduction to Models and Methods for Automated Planning
Hector Geffner and Blai Bonet
2013

Essential Principles for Autonomous Robotics
Henry Hexmoor
2013

Case-Based Reasoning: A Concise Introduction
Beatriz López
2013

Answer Set Solving in Practice
Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub
2012

Planning with Markov Decision Processes: An AI Perspective
Mausam and Andrey Kolobov
2012

Active Learning
Burr Settles
2012

Computational Aspects of Cooperative Game Theory
Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge
2011

Representations and Techniques for 3D Object Recognition and Scene Interpretation
Derek Hoiem and Silvio Savarese
2011

A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice
Francesca Rossi, Kristen Brent Venable, and Toby Walsh
2011

Human Computation
Edith Law and Luis von Ahn
2011

Trading Agents
Michael P. Wellman
2011

Visual Object Recognition
Kristen Grauman and Bastian Leibe
2011

Learning with Support Vector Machines
Colin Campbell and Yiming Ying
2011

Algorithms for Reinforcement Learning
Csaba Szepesvári
2010

Data Integration: The Relational Logic Approach
Michael Genesereth
2010

Markov Logic: An Interface Layer for Artificial Intelligence
Pedro Domingos and Daniel Lowd
2009

Introduction to Semi-Supervised Learning
XiaojinZhu and Andrew B.Goldberg
2009

Action Programming Languages
Michael Thielscher
2008

Representation Discovery using Harmonic Analysis
Sridhar Mahadevan
2008

Essentials of Game Theory: A Concise Multidisciplinary Introduction
Kevin Leyton-Brown and Yoav Shoham
2008

A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence
Nikos Vlassis
2007

Intelligent Autonomous Robotics: A Robot Soccer Case Study
Peter Stone
2007

Graph-Based Semi-Supervised Learning

Amarnag Subramanya and Partha Pratim Talukdar

www.morganclaypool.com

ISBN: 9781627052016 paperback
ISBN: 9781627052023 ebook

DOI 10.2200/S00590ED1V01Y201408AIM029

A Publication in the Morgan & Claypool Publishers series
*SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING*

Lecture #29
Series Editors: Ronald J. Brachman, *Yahoo! Labs*
William W. Cohen, *Carnegie Mellon University*
Peter Stone, *University of Texas at Austin*
Series ISSN
Print 1939-4608 Electronic 1939-4616

# Graph-Based Semi-Supervised Learning

Amarnag Subramanya
Google Research, Mountain View, USA

Partha Pratim Talukdar
Indian Institute of Science, Bangalore, India

*SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING #29*

MORGAN & CLAYPOOL PUBLISHERS

## ABSTRACT

While labeled data is expensive to prepare, ever increasing amounts of unlabeled data is becoming widely available. In order to adapt to this phenomenon, several semi-supervised learning (SSL) algorithms, which learn from labeled as well as unlabeled data, have been developed. In a separate line of work, researchers have started to realize that graphs provide a natural way to represent data in a variety of domains. Graph-based SSL algorithms, which bring together these two lines of work, have been shown to outperform the state-of-the-art in many applications in speech processing, computer vision, natural language processing, and other areas of Artificial Intelligence. Recognizing this promising and emerging area of research, this synthesis lecture focuses on graph-based SSL algorithms (e.g., label propagation methods). Our hope is that after reading this book, the reader will walk away with the following: (1) an in-depth knowledge of the current state-of-the-art in graph-based SSL algorithms, and the ability to implement them; (2) the ability to decide on the suitability of graph-based SSL methods for a problem; and (3) familiarity with different applications where graph-based SSL methods have been successfully applied.

## KEYWORDS

semi-supervised learning, graph-based semi-supervised learning, manifold learning, graph-based learning, transductive learning, inductive learning, nonparametric learning, graph Laplacian, label propagation, scalable machine learning

*To the loving memory of my father*

*A.S.*

*To the loving memory of Bamoni, my beloved sister*

*P.P.T.*

# Contents

CHAPTER 1

# Introduction

Learning from limited amounts of labeled data, also referred to as *supervised learning*, is the most popular paradigm in machine learning. This approach to machine learning has been very successful with applications ranging from spam email classification to optical character recognition to speech recognition. Such labeled data is usually prepared based on human inputs, which is expensive to obtain, difficult to scale, and often error prone. At the same time, unlabeled data is readily available in large quantities in many domains. In order to benefit from such widely available unlabeled data, several *semi-supervised learning (SSL) algorithms* have been developed over the years [Zhu and Goldberg, 2009]. SSL algorithms thus benefit from both labeled as well as unlabeled data.

With the explosive growth of the World Wide Web, graph structured datasets are becoming widespread, e.g., online social networks, hyperlinked web graph, user-product graph, user-video watched graph, etc. In a separate line of work, researchers have started to realize that graphs provide a natural way to represent data in a variety of other domains. In such datasets, nodes correspond to data instances, while edges represent relationships among nodes (instances).

Graph-based SSL techniques bring together these two lines of research. In particular, starting with the graph structure and label information about a subset of the nodes, graph SSL algorithms classify the remainder of the nodes in the graph. Graph-based SSL algorithms have been shown to outperform *non* graph-based SSL approaches [Subramanya and Bilmes, 2010]. Furthermore, majority of the graph-based SSL approaches can be optimized using convex optimization techniques and are both easily scalable and parallelizable. Graph-based SSL algorithms have been successfully used in applications as diverse as phone classification [Subramanya and Bilmes, 2010], part-of-speech (POS) tagging [Subramanya et al., 2010], statistical machine translation (SMT) [Alexandrescu and Kirchhoff, 2009], word sense disambiguation, semantic parsing [Das and Smith, 2012], knowledge acquisition [Wijaya et al., 2013], sentiment analysis in social media [Lerman et al., 2009], text categorization [Subramanya and Bilmes, 2008], and many others. Recognizing this promising and emerging area of research, this book provides an introduction to graph-based SSL techniques.

To give a concrete example, let us consider the graph shown in Figure 1.1. This graph consists of six nodes, where edge weight represents the degree of similarity between the two nodes connected by the edge. The nodes *Microsoft* and *Bangalore* are known to have the labels *Company* and *City*, respectively. Starting from this setting, a graph-based SSL algorithm aims to classify the remaining initially unlabeled four nodes.

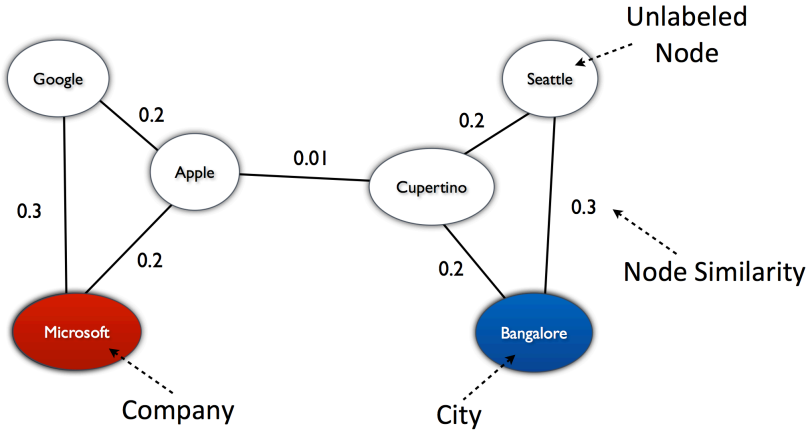


**Figure 1.1:** Starting with the setting shown above, a graph-based SSL may be used to classify the four unlabeled nodes. Ideally, we would like to infer that *Seattle* and *Cupertino* are both *Cities*, while *Google* and *Apple* are *Companies*.

In order to understand Graph-based SSL techniques, it might be informative to first understand their position in the ontology of learning algorithms. To this end, we first present some conceptual definitions, and motivations behind different learning settings.

**Definition 1.1 Instance.** An *instance* $\mathbf{x}$ represents a specific object. The instance is often represented by a $d$-dimensional *feature vector* $\mathbf{x} = [x_1, \ldots x_d] \in \mathcal{R}^{d \times 1}$, where each dimension is commonly referred to as feature. Each instance may be associated with one or more labels from the label set $\mathcal{Y}$. An instance with known label assignment is called a *labeled (or training) instance*, and otherwise called an *unlabeled instance*.

Based on the amount of supervisory information in the training data, machine learning can be categorized into *unsupervised learning*, *supervised learning*, and *semi-supervised learning*.

## 1.1 UNSUPERVISED LEARNING

The training data in the case of unsupervised learning contains *no supervisory* information. Here, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. It is assumed that each $\mathbf{x}_i$ is sampled independently from a distribution $P(\mathbf{x}), \mathbf{x} \in \mathcal{X}$. Thus, these samples are independent and identically distributed or *i.i.d.* Note that the underlying distribution $P(\mathbf{x})$ is not revealed to the learning algorithm.

Examples of unsupervised learning problems include, clustering, dimensionality reduction, and density estimation. The goal of clustering is to group (cluster) similar instances in $\mathcal{D}$, while dimensionality reduction aims to represent each sample with a lower dimensional feature vector

with as little loss of information as possible. Density estimation is the problem of estimating of the parameters of the underlying distribution that generated the $\mathcal{D}$.

## 1.2 SUPERVISED LEARNING

In supervised learning, every sample in $\mathcal{D}$ has both the input $\mathbf{x} \in \mathcal{X}$ and the corresponding expected response $y \in \mathcal{Y}$. Here, $\mathcal{Y}$ represents the set of possible outputs. The expected response is also referred to as the *label*. Formally, given a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the goal of a supervised learning algorithm is to train a function $f : \mathcal{X} \to \mathcal{Y}$ where $f \in \mathcal{F}$. Here, $\mathcal{F}$ is some pre-defined family of functions. Each training sample $(\mathbf{x}_i, y_i)$ is assumed to be sampled independently from a joint distribution $P(\mathbf{x}, y), \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$, which is not revealed to the learning algorithm. Once we have a trained model, $f^*(\mathbf{x})$, given a new input $\mathbf{x}' \in \mathcal{X}$ we can predict the label $\hat{y} = f^*(\mathbf{x}')$.

When one uses probabilistic models, the decision function $f(x)$ is given by $p(\mathbb{Y}|\mathbb{X}; \Theta)$.[1] Here, $\mathbb{Y}$ and $\mathbb{X}$ are random variables defined over the output and input domains respectively and $p(\mathbb{Y}|\mathbb{X}; \Theta)$ is a conditional distribution parameterized by $\Theta \in \mathcal{R}^{|\Theta|}$. Thus, training in this case involves learning $\Theta$ given $\mathcal{D}$. Once we have a trained model $\Theta^*$, given a new input $\mathbf{x}' \in \mathcal{X}$, the most likely class can be computed by $y^* = \text{argmax}_{y \in \mathcal{Y}}\, p(y|\mathbb{X} = \mathbf{x}'; \Theta^*)$.

Based on the nature of the output domain $\mathcal{Y}$, supervised learning is categorized into classification and regression. Supervised learning problems where $\mathcal{Y}$ is discrete are referred to as *classification* while those in which $\mathcal{Y}$ is continuous are called *regression*. In the classification case, each $y \in \mathcal{Y}$ is referred to as a *class*. Problems with only two classes, i.e., $|\mathcal{Y}| = 2$, are called *binary classification* while *multi-class* problems have $|\mathcal{Y}| > 2$. In the case of binary classification, $y \in \mathcal{R}$ can be used to represent the result of classification (or annotation) but in the multi-class setting $y \in \mathcal{R}_+^{|\mathcal{Y}|}$. In other words, the size of $y$ is equal to the number of classes. The values may either represent a score indicating how likely it is for a particular input to belong to a particular class or may be a valid probability distribution.

Examples of supervised learning algorithms include, support vector machines (SVM) [Scholkopf and Smola, 2001] and maximum entropy classifiers [Mitchell, 1997]. Supervised learning is perhaps one of the most well researched areas of machine learning and a large number of machine learning applications are built using supervised learning [Duda et al., 2001, Kotsiantis, 2007, Turk and Pentland, 1991].

Depending on the complexity of the function class $\mathcal{F}$, the number of samples required to accurately learn the mapping, $f$, can sometimes be extremely large. Training with insufficient amounts of data can lead to poor performance on unseen data. This is a drawback of supervised learning as annotating large amounts of data requires extensive human supervisory efforts. This can be both time consuming and often error prone. Unsupervised learning, on the other hand, requires no

[1]While discriminative models directly model this distribution, in the case of generative models, the Bayes rule is used to compute the class conditional distribution $p(y|x)$. For more details, see Mitchell [1997].

"labeled" training data but suffers from the inability for one to specify the expected output for a given input.

## 1.3 SEMI-SUPERVISED LEARNING (SSL)

Semi-supervised learning (SSL) lies somewhere between supervised and unsupervised learning. It combines the positives of both supervised and unsupervised learning. Here only a small amount of the training set $\mathcal{D}$ is labeled while a relatively large fraction of the training data is left unlabeled. The goal of a SSL algorithm is to learn a function $f : \mathcal{X} \to \mathcal{Y}$, $f \in \mathcal{F}$ given a training set $\mathcal{D} = \{\mathcal{D}_l, \mathcal{D}_u\}$ where $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$ represents the labeled portion of the training data while $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=1}^{n_u}$ are the unlabeled samples. Thus, we have $n_l$ labeled examples and $n_u$ unlabeled examples. We assume that the total number of training samples $n \triangleq n_l + n_u$. In most practical settings, $n_u \gg n_l$. Each labeled training sample $(\mathbf{x}_i, y_i)$ is assumed to be sampled independently from a joint distribution $P(\mathbf{x}, y), \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ while the unlabeled examples are sampled from $P(\mathbf{x}) = \sum_y P(\mathbf{x}, y)$. Like in the cases above, these distributions are not revealed to the learning algorithm. Note that like in the case of supervised learning, based on the type of the output space $\mathcal{Y}$, we have *semi-supervised classification* and *semi-supervised regression*. In this book we will be focussing on semi-supervised classification. Unless stated otherwise, SSL implies semi-supervised classification.

How does unlabeled data help? The goal of an SSL algorithm is to learn the mapping $f : \mathcal{X} \to \mathcal{Y}$. However, it appears that the unlabeled data contains no information about this mapping. In general, SSL algorithms make one or more of the following assumptions so that information available in the unlabeled data can influence $f : \mathcal{X} \to \mathcal{Y}$.

1. *Smoothness Assumption*—if two points in a high-density region are close then their corresponding outputs are also close. In regression problems, for example, the above assumption implies that the function $f : \mathcal{X} \to \mathcal{Y}$ is continuous.

2. *Cluster Assumption*—if two points are in the same cluster, they are likely to be of the same class. Another way to state this assumption would be to say that the decision boundary should lie in a low-density region. Transductive SVMs and some of the graph-based SSL algorithms are based on this assumption.

3. *Manifold Assumption*—high-dimensional data lies within a low-dimensional manifold. This is very important owing to the fact that most machine learning algorithms suffer from the "curse of dimensionality." Thus, being able to handle the data on a relatively low-dimensional manifold can often be very advantageous for the algorithms.

One of the earliest SSL algorithm is *self-training* (or self-learning or self-labeling) [Scudder, 1965]. In many instances, *expectation-maximization* (EM) [Dempster et al., 1977] can also be seen as an SSL algorithm. EM is a general procedure to maximize the likelihood of the data

given a model with hidden variables and is guaranteed to converge to a local maxima. EM lends itself naturally to SSL as the labels for the unlabeled data can be treated as missing (hidden) variables. Example of algorithms that use EM for SSL include [Hosmer, 1973, McLachlan and Ganesalingam, 1982, Nigam, 2001]. Co-training is another SSL algorithm [Blum and Mitchell, 1998] that is related to self-training and takes advantage of multiple views of the data. Transductive support vector machines (TSVM) [Vapnik, 1998] are based on the premise that the decision boundary must avoid high density regions in the input space. They are related to the fully supervised support vector machines. For more details on SSL algorithms, see [Chapelle et al., 2007].

## 1.4 GRAPH-BASED SEMI-SUPERVISED LEARNING

Graph-based SSL algorithms are an important sub-class of SSL techniques that have received much attention in the recent past [Chapelle et al., 2007, Zhu, 2005]. Here, one assumes that the data (both labeled and unlabeled) is embedded within a low-dimensional manifold that may be reasonably expressed by a graph. Each data sample is represented by a vertex in a weighted graph with the weights providing a measure of similarity between vertices. Thus, taking a graph-based approach to solving a SSL problem involves the following steps:

1. graph construction (if no input graph exists),
2. injecting seed labels on a subset of the nodes, and
3. inferring labels on unlabeled nodes in the graph.

Graph construction will be the topic of Chapter 2 while inference algorithms will be discussed in Chapter 3. There are a number of reasons why graph-based SSL algorithms are very attractive.

1. *Graphs are everywhere*: As stated earlier, many data sets of current interest are naturally represented by graphs. For example, the Web is a hyperlinked graph, social network is a graph, communication networks are graphs, etc.
2. *Convexity*: As we will see in Chapter 3, majority of the graph-based SSL algorithms involve optimizing a convex objective.
3. *Ease of scalability*: As SSL is based on the premise that large amounts of unlabeled data improve performance, it is very important for SSL algorithms to scale. This is crucial in many application domains (see Chapter 5) where ability to handle large datasets is a prerequisite. As we will see in Chapter 4, compared to other (non-graph-based) SSL algorithms, many graph-based SSL techniques can be easily parallelized. For example, state-of-the-art TSVMs can only handle tens of thousands of samples when using an arbitrary kernel; Karlen et al. [2008] report that for a problem with about 70,000 samples (both labeled and unlabeled included), it took about 42 h to train a TSVM.

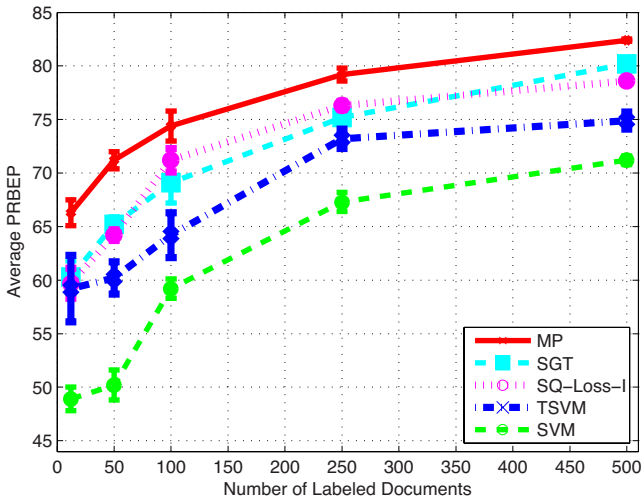


**Figure 1.2:** Performance of various SSL techniques on a text categorization problem. The y-axis represents average precision-recall break even point (PRBEP) and so larger is better. Measure propagation (MP), spectral graph transduction (SGT), and SQ-Loss-I are all graph-based SSL algorithms. SVM is fully supervised and TSVM is a SSL algorithm that does not make use of a graph. It can be seen that the graph-based approaches outperform other SSL and supervised techniques.

4. *Effective in practice*: Finally, graph SSL algorithms tend to be effective in practice. For example, Figure 1.2 shows a comparison of the performance of a number of SSL algorithms on a text categorization task [Subramanya and Bilmes, 2010]. The input here is a document (e.g., news article) and the goal is to classify it into a one of many classes (e.g., sport, politics). It can be seen that the graph-based SSL algorithms (MP, SGT, and SQ-Loss-I) outperform the SSL approaches and supervised techniques.

Next, we define a number of terms that we will use in this book.

**Definition 1.2 Graph.** A graph is an ordered pair, $G = (V, E)$ where $V = \{1, \ldots, |V|\}$ is the set of vertices (or nodes) and $E \subseteq \{V \times V\}$ is the set of edges.

**Definition 1.3 Directed and Undirected Graphs.** A graph in which the edges have no orientation is called an undirected graph. In the case of a directed graph the edges have a direction associated with them. Thus, the set of edges $E$ in a directed graph consists of ordered pairs of vertices.

As explained above, graph-based SSL algorithms start by representing the data (labeled and unlabeled) as a graph. We assume that vertex $i \in V$ represents input sample $\mathbf{x}_i$. We will be using both $i$ and $\mathbf{x}_i$ to refer to the $i^{th}$ vertex in the graph.

**Definition 1.4 Weighted Graph.** A graph is said to be weighted if there is a number or weight associated with every edge in the graph. Given an edge $(i, j)$, where $i, j \in V$, we use $W_{ij}$ to denote the weight of the edge. We will use the matrix $W \in \mathcal{R}^{n \times n}$ to denote all the edge weights, i.e., $W_{ij} = [W]_{ij}$. Also, $G = (V, E, W)$ represents a weighted graph.

While, in general, there are no constraints on $W_{ij}$, for all graphs in this book we assume that $W_{ij} \geq 0$. Further we assume that $W_{ij} = 0$ if and only if there is no edge between vertices $i$ and $j$. Also in the case of an undirected graph $W_{ij} = W_{ji}$ and thus $W$ is a symmetric matrix. Majority of the graphs in this book will be weighted undirected graphs.

**Definition 1.5 Connected component.** Given an undirected graph, a connected component is a subgraph in which all pairs of vertices are connected to each other by a path.

**Definition 1.6 Degree of a Vertex.** The degree $D_{ii}$ of vertex $i$ is given by $D_{ii} = \sum_j W_{ij}$. In the case of an unweighted graph, the degree of vertex is equal to its number of neighbors.

**Definition 1.7 Unnormalized Graph Laplacian.** The unnormalized graph Laplacian is given by $L = D - W$. Here $D \in \mathcal{R}^{n \times n}$ is a diagonal matrix such that $D_{ii}$ is the degree of node $i$ and $D_{ij} = 0 \; \forall i \neq j$. $L$ is a positive semi-definite matrix.

**Definition 1.8 Normalized Graph Laplacian.** The normalized graph Laplacian is given by $\mathcal{L} = D^{-1/2} L D^{1/2}$.

### 1.4.1 INDUCTIVE VS. TRANSDUCTIVE SSL

**Definition 1.9 Inductive Algorithm.** Given a training set consisting of labeled and unlabeled data, $\mathcal{D} = \{\{\mathbf{x}_i, y_i\}_{i=1}^{n_l}, \{\mathbf{x}_i\}_{i=1}^{n_u}\}$, the goal of an inductive algorithm is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$. Thus $f$ is able to predict the output $y$ for any input $\mathbf{x} \in \mathcal{X}$.

**Definition 1.10 Transductive Algorithm.** Given a training set consisting of labeled and unlabeled data, $\mathcal{D} = \{\{\mathbf{x}_i, y_i\}_{i=1}^{n_l}, \{\mathbf{x}_i\}_{i=1}^{n_u}\}$, the goal of a transductive algorithm is to learn a function $f : \mathcal{X}^n \rightarrow \mathcal{Y}^n$. In other words, $f$ only predicts the labels for the unlabeled data $\{\mathbf{x}_i\}_{i=1}^{n_u}$.

### Parametric vs. Nonparametric Models

Orthogonal to these are the concepts of parametric and nonparametric models. In case of nonparametric models, the model *structure* is not specified *a priori*, and is instead allowed to change as dictated by the data. In contrast, in case of parametric models, the data is assumed to come from a type of probability distribution, and the goal is to recover the parameters of the probability distribution. Most graph SSL techniques are nonparametric in nature.

## 1.5 BOOK ORGANIZATION

The rest of the book is organized as follows. In Chapter 2, we look at how a graph can be constructed from the data when no graph is given as input. In Chapter 3, we look at various graph SSL inference techniques. Then in Chapter 4, we look at issues related to scaling graph SSL to large datasets. Applications to various real-world problems will be the focus of Chapter 5.

# Authors' Biographies

## AMARNAG SUBRAMANYA

Google Research, 1600 Amphitheater Pkwy., Mountain View, CA 94043, USA
Email: asubram@google.com, Web: http://sites.google.com/site/amarsubramanya

**Amarnag Subramanya** is a Staff Research Scientist in the Natural Language Processing group at Google Research. Amarnag received his Ph.D. (2009) from the University of Washington, Seattle, working under the supervision of Jeff Bilmes. His dissertation focused on improving the performance and scalability of graph-based semi-supervised learning algorithms for problems in natural language, speed, and vision. Amarnag's research interests include machine learning and graphical models. In particular, he is interested in the application of semi-supervised learning to large-scale problems in natural language processing. He was the recipient of the Microsoft Research Graduate fellowship in 2007. He recently co-organized a session on "Semantic Processing" at the National Academy of Engineering's (NAE) Frontiers of Engineering (USFOE) conference.

## PARTHA PRATIM TALUKDAR

401 SERC, Indian Institute of Science, Bangalore, India 560012
Email: ppt@serc.iisc.in, Web: http://talukdar.net

**Partha Pratim Talukdar** is an Assistant Professor in the Supercomputer Education and Research Centre (SERC) at the Indian Institute of Science (IISc), Bangalore. Before that, Partha was a Postdoctoral Fellow in the Machine Learning Department at Carnegie Mellon University, working with Tom Mitchell on the NELL project. Partha received his Ph.D. (2010) in CIS from the University of Pennsylvania, working under the supervision of Fernando Pereira, Zack Ives, and Mark Liberman. Partha is broadly interested in Machine Learning, Natural Language Processing, Data Integration, and Cognitive Neuroscience, with particular interest in large-scale learning and inference over graphs. His past industrial research affiliations include HP Labs, Google Research, and Microsoft Research.

# Index