Semi-Supervised Learning with Graphs

DOCTORAL THESIS

Xiaojin Zhu
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
zhuxj@cs.cmu.edu

THESIS COMMITTEE
John Lafferty (co-chair)
Ronald Rosenfeld (co-chair)
Zoubin Ghahramani
Tommi Jaakkola

April 15, 2005

Abstract

In traditional machine learning approaches to classification, one uses only a labeled set to train the classifier. Labeled instances however are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabeled data may be relatively easy to collect, but there has been few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice.

We present a series of novel semi-supervised learning approaches arising from a graph representation, where labeled and unlabeled instances are represented as vertices, and edges encode the similarity between instances. They address the following questions: How to use unlabeled data? (label propagation); What is the probabilistic interpretation? (Gaussian fields and harmonic functions); What if we can choose labeled data? (active learning); How to construct good graphs? (hyperparameter learning); How to work with kernel machines like SVM? (graph kernels); How to handle complex data like sequences? (kernel conditional random fields); How to handle scalability and induction? (harmonic mixtures). An extensive literature review is also included at the end.

Chapter 1

Introduction

1.1 What is Semi-Supervised Learning?

The field of machine learning has traditionally been divided into three sub-fields:

- unsupervised learning. The learning system observes an unlabeled set of items, represented by their features $\{x_1, \ldots, \mathbf{x}_n\}$. The goal is to organize the items. Typical unsupervised learning tasks include clustering that groups items into clusters; outlier detection which determines if a new item x is significantly different from items seen so far; dimensionality reduction which maps x into a low dimensional space, while preserving certain properties of the dataset.

- supervised learning. The learning system observes a labeled training set consisting of (feature, label) pairs, denoted by $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. The goal is to predict the label y for any new input with feature x. A supervised learning task is called regression when $y \in \mathbb{R}$, and classification when y takes a set of discrete values.

- reinforcement learning. The learning system repeatedly observes the environment x, performs an action a, and receives a reward r. The goal is to choose the actions that maximize the future rewards.

This thesis focuses on classification, which is traditionally a supervised learning task. To train a classifier one needs the labeled training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. However the labels y are often hard, expensive, and slow to obtain, because it may require experienced human annotators. For instance,

- Speech recognition. Accurate transcription of speech utterance at phonetic level is extremely time consuming (as slow as 400$\times$RT, i.e. 400 times longer

than the utterance duration), and requires linguistic expertise. Transcription at word level is still time consuming (about 10×RT), especially for conversational or spontaneous speech. This problem is more prominent for foreign languages or dialects with less speakers, when linguistic experts of that language are hard to find.

- Text categorization. Filtering out spam emails, categorizing user messages, recommending Internet articles – many such tasks need the user to label text document as 'interesting' or not. Having to read and label thousands of documents is daunting for average users.

- Parsing. To train a good parser one needs sentence / parse tree pairs, known as treebanks. Treebanks are very time consuming to construct by linguists. It took the experts several years to create parse trees for only a few thousand sentences.

- Video surveillance. Manually labeling people in large amount of surveillance camera images can be time consuming.

- Protein structure prediction. It may take months of expensive lab work by expert crystallographers to identify the 3D structure of a single protein.

On the other hand, unlabeled data x, without labels, is usually available in large quantity and costs little to collect. Utterances can be recorded from radio broadcast; Text documents can be crawled from the Internet; Sentences are everywhere; Surveillance cameras run 24 hours a day; DNA sequences of proteins are readily available from gene databases. The problem with traditional classification methods is: **they cannot use unlabeled data to train classifiers**.

The question *semi-supervised learning* addresses is: given a relatively small labeled dataset $\{(x, y)\}$ and a large unlabeled dataset $\{x\}$, can one devise ways to learn from both for classification? The name "semi-supervised learning" comes from the fact that the data used is between supervised and unsupervised learning. Semi-supervised learning promises higher accuracies with less annotating effort. It is therefore of great theoretic and practical interest. A broader definition of semi-supervised learning includes regression and clustering as well, but we will not pursued that direction here.

1.2 A Short History of Semi-Supervised Learning

There has been a whole spectrum of interesting ideas on how to learn from both labeled and unlabeled data. We give a highly simplified history of semi-supervised

learning in this section. Interested readers can skip to Chapter 11 for an extended literature review. It should be pointed out that semi-supervised learning is a rapidly evolving field, and the review is necessarily incomplete.

Early work in semi-supervised learning assumes there are two classes, and each class has a Gaussian distribution. This amounts to assuming the complete data comes from a mixture model . With large amount of unlabeled data, the mixture components can be identified with the expectation-maximization (EM) algorithm. One needs only a single labeled example per component to fully determine the mixture model. This model has been successfully applied to text categorization.

A variant is self-training : A classifier is first trained with the labeled data. It is then used to classify the unlabeled data. The most confident unlabeled points, together with their predicted labels, are added to the training set. The classifier is re-trained and the procedure repeated. Note the classifier uses its own predictions to teach itself. This is a 'hard' version of the mixture model and EM algorithm. The procedure is also called self-teaching , or bootstrapping[1] in some research communities. One can imagine that a classification mistake can reinforce itself.

Both methods have been used since long time ago. They remain popular because of their conceptual and algorithmic simplicity.

Co-training reduces the mistake-reinforcing danger of self-training. This recent method assumes that the features of an item can be split into two subsets. Each sub-feature set is sufficient to train a good classifier; and the two sets are conditionally independent given the class. Initially two classifiers are trained with the labeled data, one on each sub-feature set. Each classifier then iteratively classifies the unlabeled data, and teaches the other classifier with its predictions.

With the rising popularity of support vector machines (SVMs), transductive SVMs emerge as an extension to standard SVMs for semi-supervised learning. Transductive SVMs find a labeling for all the unlabeled data, and a separating hyperplane, such that maximum margin is achieved on both the labeled data and the (now labeled) unlabeled data. Intuitively unlabeled data guides the decision boundary away from dense regions.

Recently graph-based semi-supervised learning methods have attracted great attention. Graph-based methods start with a graph where the nodes are the labeled and unlabeled data points, and (weighted) edges reflect the similarity of nodes. The assumption is that nodes connected by a large-weight edge tend to have the same label, and labels can propagation throughout the graph. Graph-based methods enjoy nice properties from spectral graph theory. This thesis mainly discusses graph-based semi-supervised methods.

We summarize a few representative semi-supervised methods in Table 1.1.

[1] Not to be confused with the resample procedure with the same name in statistics.

Method	Assumptions
mixture model, EM	generative mixture model
transductive SVM	low density region between classes
co-training	conditionally independent and redundant features splits
graph methods	labels smooth on graph

Table 1.1: Some representative semi-supervised learning methods

1.3 Structure of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 starts with the simple *label propagation* algorithm, which propagates class labels on a graph. This is the first semi-supervised learning algorithm we will encounter. It is also the basis for many variations later.

Chapter 3 discusses how one constructs a graph. The emphasis is on the intuition – what graphs make sense for semi-supervised learning? We will give several examples on various datasets.

Chapter 4 formalizes label propagation in a probabilistic framework with Gaussian random fields. Concepts like graph Laplacian and harmonic function are introduced. We will explore interesting connections to electric networks, random walk, and spectral clustering. Issues like the balance between classes, and inclusion of external classifiers are also discussed here.

Chapter 5 assumes that one can choose a data point and ask an oracle for the label. This is the standard active learning scheme. We show that active learning and semi-supervised learning can be naturally combined.

Chapter 6 establishes the link to Gaussian processes. The kernel matrices are shown to be the smoothed inverse graph Laplacian.

Chapter 7 no longer assumes the graph is given and fixed. Instead, we parameterize the graph weights, and learn the optimal hyperparameters. We will discuss several methods: evidence maximization, entropy minimization, and minimum spanning tree.

Chapter 8 turns semi-supervised learning problem into kernel learning. We show a natural family of kernels derived from the graph Laplacian, and find the best kernel via convex optimization.

Chapter 9 discusses kernel conditional random fields, and its potential application in semi-supervised learning, for sequences and other complex structures.

Chapter 10 explores scalability and induction for semi-supervised learning.

Chapter 11 reviews the literatures on semi-supervised learning.

Chapter 2

Label Propagation

In this chapter we introduce our first semi-supervised learning algorithm: Label Propagation. We formulate the problem as a form of propagation on a graph, where a node's label propagates to neighboring nodes according to their proximity. In this process we fix the labels on the labeled data. Thus labeled data act like sources that push out labels through unlabeled data.

2.1 Problem Setup

Let $\{(x_1, y_1) \dots (x_l, y_l)\}$ be the labeled data, $y \in \{1 \dots C\}$, and $\{x_{l+1} \dots x_{l+u}\}$ the unlabeled data, usually $l \ll u$. Let $n = l + u$. We will often use L and U to denote labeled and unlabeled data respectively. We assume the number of classes C is known, and all classes are present in the labeled data. In most of the thesis we study the *transductive* problem of finding the labels for U. The inductive problem of finding labels for points outside of $L \cup U$ will be discussed in Chapter 10.

Intuitively we want data points that are similar to have the same label. We create a graph where the nodes are all the data points, both labeled and unlabeled. The edge between nodes i, j represents their similarity. For the time being let us assume the graph is fully connected with the following weights:

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\alpha^2}\right) \tag{2.1}$$

where α is a bandwidth hyperparameter. The construction of graphs will be discussed in later Chapters.

2.2 The Algorithm

We propagate the labels through the edges. Larger edge weights allow labels to travel through more easily. Define a $n \times n$ probabilistic transition matrix P

$$P_{ij} = P(i \to j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \tag{2.2}$$

where P_{ij} is the probability of transit from node i to j. Also define a $l \times C$ label matrix Y_L, whose ith row is an indicator vector for $y_i, i \in L$: $Y_{ic} = \delta(y_i, c)$. We will compute soft labels f for the nodes. f is a $n \times C$ matrix, the rows can be interpreted as the probability distributions over labels. The initialization of f is not important. We are now ready to present the algorithm.

The label propagation algorithm is as follows:

1. Propagate $f \leftarrow Pf$
2. Clamp the labeled data $f_L = Y_L$.
3. Repeat from step 1 until f converges.

In step 1, all nodes propagate their labels to their neighbors for one step. Step 2 is critical: we want persistent label sources from labeled data. So instead of letting the initially labels fade away, we clamp them at Y_L. With this constant 'push' from labeled nodes, the class boundaries will be pushed through high density regions and settle in low density gaps. If this structure of data fits the classification goal, then the algorithm can use unlabeled data to help learning.

2.3 Convergence

We now show the algorithm converges to a simple solution. Let $f = \begin{pmatrix} f_L \\ f_U \end{pmatrix}$. Since f_L is clamped to Y_L, we are solely interested in f_U. We split P into labeled and unlabeled sub-matrices

$$P = \begin{bmatrix} P_{LL} & P_{LU} \\ P_{UL} & P_{UU} \end{bmatrix} \tag{2.3}$$

It can be shown that our algorithm is

$$f_U \leftarrow P_{UU} f_U + P_{UL} Y_L \tag{2.4}$$

which leads to

$$f_U = \lim_{n\to\infty} (P_{UU})^n f_U^0 + \left(\sum_{i=1}^{n} (P_{UU})^{(i-1)} \right) P_{UL} Y_L \tag{2.5}$$

where f_U^0 is the initial value for f_U. We need to show $(P_{UU})^n f_U^0 \to 0$. Since P is row normalized, and P_{uu} is a sub-matrix of P, it follows

$$\exists \gamma < 1, \sum_{j=1}^{u} (P_{UU})_{ij} \leq \gamma, \forall i = 1 \ldots u \tag{2.6}$$

Therefore

$$\begin{aligned} \sum_j (P_{UU})^n{}_{ij} &= \sum_j \sum_k (P_{UU})^{(n-1)}{}_{ik} (P_{UU})_{kj} && (2.7)\\ &= \sum_k (P_{UU})^{(n-1)}{}_{ik} \sum_j (P_{UU})_{kj} && (2.8)\\ &\leq \sum_k (P_{UU})^{(n-1)}{}_{ik} \gamma && (2.9)\\ &\leq \gamma^n && (2.10) \end{aligned}$$

Therefore the row sums of $(P_{UU})^n$ converges to zero, which means $(P_{UU})^n f_U^0 \to 0$. Thus the initial value f_U^0 is inconsequential. Obviously

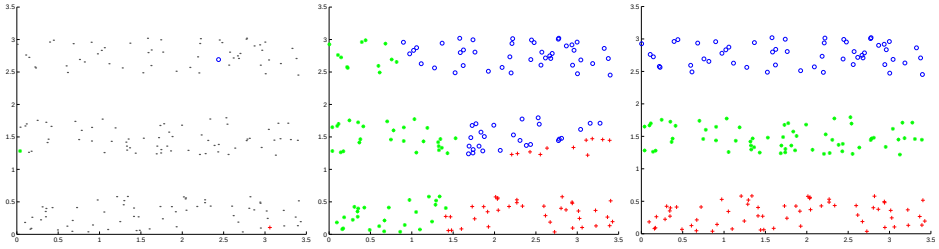
$$f_U = (I - P_{UU})^{-1} P_{UL} Y_L \tag{2.11}$$

is a fixed point. Therefore it is the unique fixed point and the solution to our iterative algorithm. This gives us a way to solve the label propagation problem directly without iterative propagation.

Note the solution is valid only when $I - P_{UU}$ is invertible. The condition is satisfied, intuitively, when every connected component in the graph has at least one labeled point in it.

2.4 Illustrative Examples

We demonstrate the properties of the Label Propagation algorithm on two synthetic datasets. Figure 2.1(a) shows a synthetic dataset with three classes, each being a narrow horizontal band. Data points are uniformly drawn from the bands. There are 3 labeled points and 178 unlabeled points. 1-nearest-neighbor algorithm, one of the standard supervised learning methods, ignores the unlabeled data and thus the

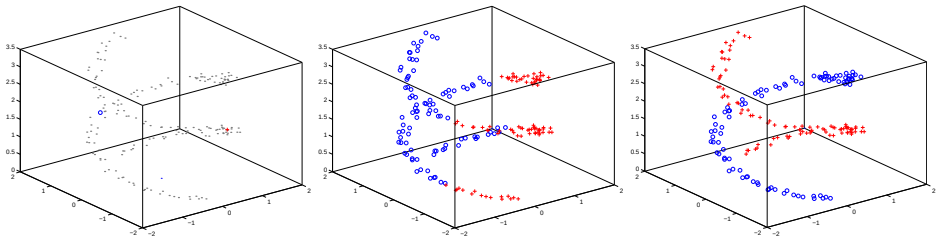


(a) The data (b) 1NN (c) Label Propagation

Figure 2.1: The Three Bands dataset. Labeled data are marked with color symbols, and unlabeled data are black dots in (a). 1NN ignores unlabeled data structure (b), while Label Propagation takes advantage of it (c).

band structure (b). On the other hand, the Label Propagation algorithm takes into account the unlabeled data (c). It propagates labels along the bands. In this example, we used $\alpha = 0.22$ from the minimum spanning tree heuristic (see Chapter 7).

Figure 2.2 shows a synthetic dataset with two classes as intertwined three-dimensional spirals. There are 2 labeled points and 184 unlabeled points. Again, 1NN fails to notice the structure of unlabeled data, while Label Propagation finds the spirals. We used $\alpha = 0.43$.



(a) The data (b) 1NN (c) Label Propagation

Figure 2.2: The Springs dataset. Again 1NN ignores unlabeled data structure, while Label Propagation takes advantage of it.

Chapter 3

What is a Good Graph?

In Label Propagation we need a graph , represented by the weight matrix W. How does one construct a graph? What is a good graph? In this chapter we give several examples on different datasets. The goal is not to rigorously define 'good' graphs, but to illustrate the assumptions behind graph based semi-supervised learning.

A good graph should reflect our prior knowledge about the domain. At the present time, its design is more of an art than science. It is the practitioner's responsibility to feed a good graph to graph-based semi-supervised learning algorithms, in order to expect useful output. The algorithms in this thesis do not deal directly with the design of graphs (with the exception of Chapter 7).

We look at the graphs constructed for handwritten digits, document categorization and the FreeFoodCam. Then we explore a few common ways to create graphs.

Chapter 4

Gaussian Random Fields and Harmonic Functions

In this chapter we formalize label propagation with a probabilistic framework. Without loss of generality we assume binary classification $y \in \{0, 1\}$. We assume the $n \times n$ weight matrix W is given, which defines the graph. W has to be symmetric with non-negative entries, but otherwise need not to be positive semi-definite. Intuitively W specifies the 'local similarity' between points. Our task is to assign labels to unlabeled nodes.

We show that Gaussian random fields capture the notion of label propagation on the graph. We define the graph Laplacian. We derive the harmonic functions, which has interesting interpretations as random walks or electric networks. We discuss ways to incorporate class prior knowledge, and combination with other classifiers.

Chapter 5

Active Learning

In this chapter, we take a brief detour to look at the active learning problem. We combine semi-supervised learning and active learning naturally and efficiently.

Chapter 6

Connection to Gaussian Processes

Gaussian random fields are equivalent to Gaussian processes that are restricted to a finite set of points. Thus, the standard machineries for Gaussian processes can be used for semi-supervised learning. Through this connection, we establish the link between the graph Laplacian and kernel methods in general.

Chapter 7

Graph Hyperparameter Learning

Previously we assumed that the weight matrix W is given and fixed. In this chapter we investigate *learning* the weights from both labeled and unlabeled data. We present three methods. The first one is evidence maximization in the context of Gaussian processes. The second is entropy minimization, and the third one is based on minimum spanning trees. The latter ones are heuristic but also practical.

Chapter 8

Kernels from the Spectrum of Laplacians

We used the inverse of a smoothed Laplacian as kernel matrix in Chapter 6. In fact, one can construct a whole family of graph kernels from the spectral decomposition of graph Laplacians. These kernels combine labeled and unlabeled data in a systematic fashion. In this chapter we devise the best one (in a certain sense) for semi-supervised learning.

Chapter 9

Sequences and Beyond

So far, we have treated each data point individually. However in many problems the data has complex structures. For example in speech recognition the data is sequential. Most semi-supervised learning methods have not addressed this problem. We use sequential data as an example in the following discussion because it is simple. Nevertheless the discussion applies to other complex data structures like grids, trees etc.

It is important to clarify the setting. By sequential data we do not mean each data item x is a sequence and we give a *single label* y to the whole sequence. Instead we want to give individual labels to the constituent data points in the sequence.

There are generative and discriminative methods that can be used for semi-supervised learning on sequences.

The Hidden Markov Model (HMM) is such a generative methods. Specifically the standard EM training with forward-backward algorithm (also known as Baum-Welch) is a sequence semi-supervised learning algorithm, although it is usually not presented that way. The training data typically consists of a small labeled set with l labeled sequences $\{X_L, Y_L\} = \{(\mathbf{x}_1, \mathbf{y}_1) \dots (\mathbf{x}_l, \mathbf{y}_l)\}$, and a much larger unlabeled set of sequences $X_U = \{\mathbf{x}_{l+1} \dots \mathbf{x}_{l+u}\}$. We use bold font $\mathbf{x}_i$ to represent the i-th sequence with length m_i, whose elements are $x_{i1} \dots x_{im_i}$. Similarly $\mathbf{y}_i$ is a sequence of labels $y_{i1} \dots y_{im_i}$. The labeled set is used to estimate initial HMM parameters. The unlabeled data is then used to run the EM algorithm on, to improve the HMM likelihood $P(X_U)$ to a local maximum. The trained HMM parameters thus are determined by both the labeled and unlabeled sequences. This parallels the mixture models and EM algorithm in the *i.i.d.* case. We will not discuss it further in the thesis.

For discriminative methods one strategy is to use a kernel machine for se-

quences, and introduce semi-supervised dependency via the kernels in Chapter 8. Recent kernel machines for sequences and other complex structures include Kernel Conditional Random Fields (KCRFs) and Max-Margin Markov Networks, which are generalization of logistic regression and support vector machines respectively to structured data. These kernel machines by themselves are not designed specifically for semi-supervised learning. But we can use a kernel, which is derived from some graph Laplacian as in Chapter 8 with the machines. This results in semi-supervised learning methods on sequential data.

The idea is straightforward. The remainder of the chapter focuses on KCRFs, describing the formalism and training issues, with a synthetic example on semi-supervised learning.

Chapter 10

Harmonic Mixtures: Handling Unseen Data and Reducing Computation

There are two important questions to graph based methods:

1. The graph is constructed on the labeled and unlabeled data. Many such methods are transductive in nature. How to handle unseen new data points?

2. They often involve expensive manipulation on large matrices, for example matrix inversion which can be $O(n^3)$. Because unlabeled data is relatively easy to obtain in large quantity, the matrix could be too big to handle. How to reduce computation when the unlabeled dataset is large?

In this chapter we address these questions by combining graph method with a mixture model.

Chapter 11

Literature Review

There has been a whole spectrum of interesting ideas on how to learn from both labeled and unlabeled data. In this chapter we review some of the literature. The review is by no means comprehensive. The field of semi-supervised learning is evolving rapidly. The author welcomes comments on incorrect description or missing papers.

The common assumption is that the distribution of unlabeled data is linked to their labels. In fact this is a necessary condition for semi-supervised learning to work. Different methods make different assumptions about the linkage.

Chapter 12

Discussions

We have presented a series of semi-supervised learning algorithms, based on a graph representation of the data. Experiments show that they are able to take advantage of the unlabeled data to improve classification. Contributions of the thesis include:

- We proposed a harmonic function and Gaussian field formulations for semi-supervised problems. This is not the first graph-based semi-supervised method. The first one was graph mincut. However our formulation is a continuous relaxation to the discrete labels, resulting in a more benign problem. Several variations of the formulation were proposed independently by different groups shortly after.
- We addressed the problem of graph construction, by setting up parametric edge weights and performing edge hyperparameter learning. Since the graph is the input to all graph-based semi-supervised algorithms, it is important that we construct graphs that best suit the task.
- We combined an active learning scheme that reduces expected error instead of ambiguity, with graph-based semi-supervised learning. We believe that active learning and semi-supervised learning will be used together for practical problems, because limited human annotation resources should be spent wisely.
- We defined optimal semi-supervised kernels by spectral transformation of the graph Laplacian. Such optimal kernels can be found with convex optimization. We can use the kernels with any kernel machine, e.g. support vector machines, for semi-supervised learning. The kernel machines in general can handle noisy labeled data, which is an improvement over the harmonic function solution.

- We kernelized conditional random fields. CRFs were traditionally feature based. We derived the dual problem and presented an algorithm for fast sparse kernel CRF training. With kernel CRFs, it is possible to use a semi-supervised kernel on instances for semi-supervised learning on sequences and other structures.

- We proposed to solve large-scale problems with harmonic mixtures. Harmonic mixtures reduce computation cost significantly by grouping unlabeled data into soft clusters, then carrying out semi-supervised learning on the coarser data representation. Harmonic mixtures also handle new data points naturally, making the semi-supervised learning method inductive.

Semi-supervised learning is a relatively new research area. There are many open questions and research opportunities:

- The graph is the single most important quantity for graph-based semi-supervised learning. Parameterizing graph edge weights, and learning weight hyperparameters, should be the first step of any graph-based semi-supervised learning methods. Current methods in Chapter 7 are not efficient enough. Can we find better ways to learn the graph structure and parameters?

- Real problems can have millions of unlabeled data points. Anecdotal stories and experiments indicate that conjugate gradient with a suitable preconditioner is one of the fastest algorithms in solving harmonic functions. Harmonic mixture works along an orthogonal direction by reducing the problem size. How large a dataset can we process if we combine conjugate gradient and harmonic mixture? What can we do to handle even larger datasets?

- Semi-supervised learning on structured data, e.g. sequences and trees, is largely unexplored. We have proposed the use of kernel conditional random fields plus semi-supervised kernels. Much more work is needed in this direction.

- In this thesis we focused on classification problems. The spirit of combining some human effort with large amount of data should be applicable to other problems. Examples include: regression with both labeled and unlabeled data; ranking with ordered pairs and unlabeled data; clustering with cluster membership knowledge. What can we do beyond classification?

- Because labeled data is scarce, semi-supervised learning methods depend more heavily on their assumptions (see e.g. Table 1.1). Can we develop novel semi-supervised learning algorithms with new assumptions?

- Applications of semi-supervised learning are emerging rapidly. These include text categorization, natural language processing, bioinformatics, image processing, and computer vision. Many others are sure to come. Applications are attractive because they solve important practical problems, and provide fertile test bed for new ideas in machine learning. What problems can we apply semi-supervised learning? What applications were too hard but are now feasible with semi-supervised learning?

- The theory of semi-supervised learning is almost absent in both the machine learning literature and the statistics literature. Is graph-based semi-supervised learning consistent? How many labeled and unlabeled points are needed to learn a concept with confidence?

We expect advances in research will address these questions. We hope semi-supervised learning become a fruitful area for both machine learning theory and practical applications.