An Overview of Semi-Supervised Learning: Algorithms, Trends, and Challenges

Shruti Patel, Dr. Meera Trivedi*
Department of Computer Science and Engineering, Gujarat Technological University, Ahmedabad, Gujarat, India

Semi Supervised Learning involves using both labeled and unlabeled data to train a classifier or for clustering. Semi supervised learning finds usage in many applications, since labeled data can be hard to find in many cases. Currently, a lot of research is being conducted in this area. This paper discusses the different algorithms of semi supervised learning and then their advantages and limitations are compared. The differences between supervised classification and semi-supervised classification, and unsupervised clustering and semi-supervised clustering are also discussed.

KEYWORDS: Machine Learning, Semi-supervised learning

INTRODUCTION

Supervised learning uses labeled data to train a model that would give accurate predictions on data that the model has never seen before, e.g. classification, regression. Unsupervised learning takes in unlabeled data as its input and prepares a model that is based on the pattern or structure of the test dataset e.g. clustering, outlier detection.

Semi supervised learning is midway between supervised and unsupervised learning. We are provided with unlabeled data and labeled data. The data of Semi Supervised Learning can be divided into two parts: the labeled data $\{x_i, y_i\}_{i=1}^{l}$ and the unlabeled data $\{x\}_{i=l+1}^{l+u}$. It is assumed that unlabeled data is much more than labeled data.

Semi Supervised Learning can be seen as supervised learning with additional information on the distribution of examples. Alternatively, it can be viewed as an extension of unsupervised learning that is guided by constraints. Semi supervised Learning can be divided into two areas:

- Semi Supervised Classification
  Classifier is trained on labeled and unlabeled data, resulting in a more accurate classifier. The goal is to train a classifier from both labeled and unlabeled data such that it is better than the classifier trained on labeled data alone.
- Semi Supervised Clustering
  Labeled data is used to aid and bias the clustering of unlabeled data.

In this paper, we refer to semi supervised classification as semi supervised learning.

In many situations, there can be a dearth of labeled data. The labels may be difficult to obtain since it might require human annotators, special devices or expensive and slow experiments. Semi Supervised learning can be extremely useful in such situations. Semi Supervised find tremendous use in the following applications:

- Speech Recognition
- Natural Language Parsing
- Spam Filtering
- Video Surveillance
- Protein 3D structure prediction
- Image Categorization

For semi supervised learning, certain assumptions will have to be used. Semi supervised learning algorithms can utilize at least one of the following assumptions:

- Smoothness

Points close together in a high density region should share the same label. That is, if the points are separated by a low-density region, then their outputs need not be close.

- Cluster

Points in the same cluster are likely to be of the same class. Equivalently, it can also be stated that the decision boundary should lie in a low-density region.

- Manifold

The high-dimensional data lies on a low-dimensional manifold.

ALGORITHMS

Semi Supervised learning algorithms can be broadly divided into the following categories:

- Self Training
- Generative models
- Co-training
- Graph Based Algorithms
- Semi Supervised Support Vector Machines (S3VMs)

Self Training

This is a wrapper algorithm and is the most commonly used technique. In self training, a classifier is trained on labeled data. Then, this classifier is used to classify all unlabeled items. The unlabeled items that are predicted with the highest confidence are added to the training set. Now the classifier is trained again on the training set and the above process is repeated.

However, this algorithm assumes that the its high confidence predictions are correct.

Generative Models

In this method, we assume the form of joint probability $p(x,y \mid \theta) = p(y \mid \theta)p(x \mid y,\theta)$ for semi supervised learning. Parameters of joint probability are represented by $\theta \in \Theta$. Predictors f_θ use Bayes rule:

$$f_\theta(x) \equiv \underset{y}{\operatorname{argmax}}\, p(y|x,\theta) = \underset{y}{\operatorname{argmax}} \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)}$$

Consider unlabeled training data$\{x\}_{i=l+1}^{l+u}$. To estimate the parameters of the model, we calculate the likelihood of the data:

$$\log p(\{x_i\}_{i=l+1}^{l+u} \mid \theta) = \sum_{i=l+1}^{l+u} \log\Big(\sum_{y \in Y} p(x_i, y \mid \theta)\Big)$$

The model with parameters θ that best fits the unlabeled data will have the highest value of the above equation (MLE).

If both labeled and unlabeled data are available, we calculate the joint log likelihood of both labeled data $\{x_i, y_i\}_{i=1}^{l}$ and unlabeled data$\{x\}_{i=l+1}^{l+u}$.

$$\underset{\theta}{\operatorname{argmax}} \log p\big(\{x_i, y_i\}_{i=1}^{l} \mid \theta\big) + \lambda \log p(\{x\}_{i=l+1}^{l+u} \mid \theta)$$

Here, λ is the balancing weight. Since unlabeled data is available in large quantities, it often happens that the labeled data is ignored. Therefore, we add λ, to reweight the term of the unlabeled data.

The above equation is not concave. A local maxima can be found using the Expectation Maximization(EM) algorithm.
Commonly used generative models are:

- Gaussian Mixture Models(GMMs)
- Multinomial Mixture Models
- Hidden Markov Models(HMMs)

Co-training

The idea of co-training is to train two classifiers which then teach each other. It is a wrapper algorithm. There are two assumptions in co-training:

1. Data x can be split into two views$[x^{(1)}, x^{(2)}]$. Each view alone is enough to train a classifier, given enough labeled data.
2. The two views are conditionally independent.

The co-training algorithm works as follows. Consider labeled training data$\{x_i, y_i\}_{i=1}^{l}$. Two separate classifiers are trained on the labeled data. One classifier is trained on $\{x_i^{(1)}, y_i\}_{i=1}^{l}$, while the second is trained on $\{x_i^{(2)}, y_i\}_{i=1}^{l}$. The unlabeled data is classified with the two classifiers separately.

Then, classifier one's most confident predictions are added to classifier two's training set. Similarly, classifier two's most confident predictions are added to classifier one's training set. This way, the two classifiers teach each other. Both classifiers are retrained with their respective datasets (that now contain added training examples), and the above process repeats.

Multiview Learning

It is often difficult to split data into two views according to the assumptions of co-training. Multiview learning generalizes co-training over many predictors/classifiers and the assumptions of co-training are not applicable to multiview learning. Multiview learning predicts the final outcome based on the agreement between different predictors. Many predictors/learners of different types (e.g. decision trees, neural networks etc) are trained on the same labeled data training set and are required to predict the unlabeled data labels. The final prediction is obtained from a (confidence weighted) average or vote among the predictors/learners.

Graph Based Methods

In this method, a graph is constructed. The nodes comprise of the labeled and unlabeled examples of the dataset. The edges are generally weighted and undirected and it is assumed that the examples connected by heavy edges have the same label. The edge weight w_{ij} reflects how close the two nodes x_i and x_j are. The heavier the edge, the closer they are to each other. The edge weights can be computed in any one of the following heuristics:

- Fully connected graph: the edge weight decreases as the Euclidean distance between x_i and x_j increases. The weight function is

  $$w = \exp\left(\frac{-\| x_i - x_j \|^2}{\epsilon^2}\right)$$

  Where ϵ is the bandwidth parameter.

- K nearest Neighbour graph: each node defines its k nearest neighbour based on the Euclidean distance between them. If xi and xj are connected, then w_{ij} will be 1 otherwise 0.
- ϵ radius neighbours: all nodes within a distance of ϵ (radius) of a node are assigned an edge weight.

The algorithms differ primarily in their loss functions and regularizers. A loss function estimates the loss on labeled data. A regularizer is a function defined on a graph that estimates the smoothness of the graph on labeled and unlabeled data. It is also an estimate of the 'energy' of the graph. The lower the energy, the more accurate are its predictions.

Some of the graph based algorithms are:

- Mincut

The unlabeled data is assumed to have binary labels. Positive labeled instances are termed as 'source' vertices and negative labeled instances are termed as 'sink' vertices. The objective is to find a path with minimum set of edges whose removal stops all the flow from the source to the sink. The optimization problem for mincut is:

$$\min_{y_i \in \{0,1\}} \infty \sum_{i \in L} (y_i - y_{i|L})^2 + \sum_{ij} w_{ij}(y_i - y_j)^2$$

The first term denotes the loss function with infinity weight and the second term denotes the regularizer.

- The Harmonic Function

  The harmonic function f assigns continuous values in R instead of discrete labels to unlabeled data. For labeled data points, the harmonic function equals the respective label. The harmonic property means that the value of f at each unlabeled data point is the average of f at neighboring points:

  $$f(x_i) = y_i \text{ for } i = 1 \dots l$$

  $$f(x_j) = \frac{\sum_{k=1}^{l+u} w_{jk} f(x_k)}{\sum_{k=1}^{l+u} w_{jk}}, j = 1 \dots l + u$$

  The harmonic solution minimizes the following equation. The loss function has infinity weight, so that labeled data is fixed at the given label values. The second term denotes the regularizer or the energy. It can also be expressed in terms of Laplacian Delta.

  $$\min_f \infty \sum_{i \in L} (f(x_i) - y_i)^2 + \sum_{i,j} w_{ij}(f(x_i) - f(x_j))^2$$

- Manifold Regularization

  Mincut and harmonic function are both transductive learning algorithms. That is, they define a function that is restricted to the labeled and unlabeled data of the dataset. They will not be able to predict on a test instance that doesn't belong to the database. Also, it might happen that some of the labels in the labeled dataset are wrong, i.e. label noise. Therefore, it would be good to have an algorithm that occasionally disagrees with the labels. Manifold regularization addresses the above two concerns. It is an inductive learning algorithm that defines a function f:X->R in the entire feature space. The following is the manifold regularization framework:

  $$\frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_K^2 + \gamma_I \| f \|_I^2$$

V is an arbitrary loss function. It has two regularization terms. K is a base kernel and I is a regularization term caused by the labeled and unlabeled data.

Semi Supervised Support Vector Machines (S3VM)

Semi Supervised Support Vector Machines can be thought of as an extension of Support Vector Machines with unlabeled data. In a standard Support Vector Machine, labeled data is used to find a maximum margin linear boundary in the Reproducing Kernel Hilbert Space. In an S3VM, the unlabeled data guides the placement of the decision boundary. Labeled data is used to find a labeling of the unlabeled data, so that a linear boundary has the maximum distance from both the original labeled data and the (now labeled) unlabeled data. The assumption in this model is that the decision boundary is situated in a low density region, between two classes $y \in \{-1, 1\}$. S3VMs can be viewed as SVM with an additional regularization term for the unlabeled data.

COMPARISON BETWEEN THE ALGORITHMS

In this section, all the advantages and limitations of the algorithms discussed are compared in Table 1, Table 2, Table 3, Table 4 and Table 5.

Tables:

Table 1. Self Training

ADVANTAGES	LIMITATIONS
1. Simplest of all semi supervised learning algorithms.	1. Mistakes reinforce or strengthen themselves
2. Wrapper method. Applies to almost all existing classifiers	2. In terms of convergence, cannot give too much information.

Table 2. Generative Models

ADVANTAGES	LIMITATIONS
1. If the model is close to correct, it can give efficient predictions.	1. They often don't provide good solutions to classification problems.
2. The knowledge of the structure of the problem or data can be included by modelling it.	2. There can be a problem balancing the impact of labeled and unlabeled data when the unlabeled data is much, much more than labeled data.
	3. Local optima of the EM algorithm.
	4. Modelling effort is much more demanding than discriminative models.
	5. Since generative models are very precise, there is a high likelihood of them being incorrect.
	6. Unlabeled data will hurt the prediction if the model is wrong.

Table 3. Co-training

ADVANTAGES	LIMITATIONS
1. It is a wrapper method. Can use any classifier.	1. The feature set might not be able to split.
2. Less susceptible to mistakes than self training.	

Table 4. Graph Based Algorithms

ADVANTAGES	LIMITATIONS
1. Lucid mathematical framework	1. Bad performance if graph doesn't fit the task.
2. Good performance if the graph fits the task.	2. Performance is vulnerable to graph structure and edge weights
3. It is can be applied directed graphs	

Table 5. Semi Supervised Support Vector Machines (S3VMs)

ADVANTAGES	LIMITATIONS
1. They are valid wherever Support Vector Machines are valid.	1. Optimization is difficult since algorithm can be caught in bad local optima.
2. Lucid mathematical framework	

Table 6. Comparison between semi-supervised classification and supervised classification

SEMI-SUPERVISED CLASSIFICATION	SUPERVISED CLASSIFICATION
1. It consists of both labeled and unlabeled data.	1. It consists of only labeled data.
2. If the algorithm is transductive, then the algorithm assigns labels to the unlabeled data in the training set. If the algorithm is inductive, then the algorithm assigns labels to the unlabeled data both in the training and test set.	2. The algorithms predicts or assigns labels to unlabeled data in the test set after learning from labeled data in the training set.
3. Some unsupervised learning techniques might be used to discover the structure of the input data.	3. It is purely supervised learning.
4. It can lead to worse performance if an algorithm with the wrong assumption is chosen	4. Labeled data does not the hurt the performance of the algorithm.

Table 7 compares semi-supervised clustering with unsupervised clustering

SEMI-SUPERVISED CLUSTERING	UNSUPERVISED CLUSTERING
1. It also clusters the data but with some supervision information as a guide.	1. It organizes data into clusters such that the points in one cluster are more similar to each other than points in another cluster.
2. It consists of unlabeled data and a very small amount of labeled data.	2. It consists of only unlabeled data
3. Some modifications are applied to unsupervised clustering. Either the similarity criterion of a clustering algorithm is changed so that the constraints of supervised information can be considered, or the clustering algorithm is itself modified so that the supervision information can affect the search for a suitable cluster.	3. A clustering algorithm is used

CONCLUSION

It can be seen from the discussion of the algorithms in the paper that the main differences in the models lay in the difference in their assumptions. The different assumptions of each model can help in determining the right model for certain data.

REFERENCES

[1] Xiaojin Zhu, “Semi Supervised Learning Tutorial”. Department of Computer Sciences, University of Wisconsin-Madison, ICML 2007.
[2] Olivier Chapelle, Bernhard Scholkopf, Alexander Zien. “Semi-supervised Learning”, MIT Press.
[3] Xiaojin Zhu. “Semi-Supervised Learning with Graphs”. Carnegie Mellon University, May 2005.
[4] Xiaojin Zhu. “Semi-Supervised Learning Literature Survey”, University of Wisconsin-Madison. 2007.
[5] Xiaojin Zhu. “Semi-Supervised Classification”, University of Wisconsin-Madison.
[6] Tobias Scheffer. “Semi-Supervised Learning”, Encyclopedia of Data Warehousing and Mining, Second Edition.
[7] Zoubin Ghahramani. “Graph based Semi-Supervised Learning”, Department of Engineering, University of Cambridge, UK, 2012.
[8] Saroj K. Meher. “Semi Supervised Learning” , Systems Science and Informatics Unit, Indian Statistical Institute, Bangalore.
[9] Xiaojin Zhu. “Semi-Supervised Learning”, University of Wisconsin-Madison.
[10] Marco Loog. “Semi-Supervised Learning”, Pattern Recognition Laboratory Delft University of Technology
[11] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty. “Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions”
[12] Xiaojin Zhu. “Tutorial on Semi-Supervised Learning”. Department of Computer Sciences, University of Wisconsin-Madison. Videolectures.net.
[13] “Intoduction to Semi Supervised Learning”. What-when-how.com
[14] Nizar Grira, Michel Crucianu, Nozha Boujemaa. “Unsupervised and Semi-supervised Clustering: a Brief Survey”. INRIA Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France. August 15, 2005

