FRANKFURT UNIVERSITY OF APPLIED SCIENCE

FB 2 INFORMATIK UND INGENIEURWISSENSCHAFTEN

M. ENG INFORMATION TECHNOLOGY

Computational Intelligence
Semi-Supervised Learning
Winter Semester 2018

Authors:	*Matriculation No:*
Nguyen Phuong Trinh Tran	1105425

First Instructor:
Prof. Dr. Andreas Pech

Frankfurt, Germany
January 14, 2019

Declaration of Authorship

I am Nguyen Phuong Trinh Tran. I hereby certify that this report has been composed by me and is based on my own work unless stated otherwise. No other person's work has been used without due acknowledgment in this project. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.

Signed:

Nguyen Phuong Trinh Tran

January 14, 2019

Contents

List of Figures

Acknowledgments

I would like to express my sincere appreciations to the Frankfurt University of Applied Science for providing me with enormous valuable academic supports.

I would like to express a sincere gratitude to my instructors, Prof. Dr. Andreas Pech who has given me this interesting project and also followed to my project at the very first stage, give me many helpful, honest advice and constructive evaluation. Thanks to their instruction and evaluation, I have gained not only a great extent of knowledge and experience in machine-learning algorithm development during the time I carried out the project, but also a mature vision for my future career.

Finally, I would like to give all people who helped me during the project time the best wishes for health, family, and career. Thank you so much.

Abstract

Over the past few year, AI - Artificial Intelligence, specifically as Machine Learning, is emerging as an evidence of The Fourth Industrial Revolution (4IR): First - Engine Steam; Second - Electrical Energy; Third - Information Technology. As many of you probably aware, Artificial Intelligence Applications are sharply developing in all areas of the life. Self-driving vehicles of Google and Tesla, Facebook's Tag Suggestions System, Apple's Siri Virtual Assistant, Amazon's Product Suggestion System, Netflix's Movie Recommendation System, Google DeepMind's AlphaGo [1], etc are just a few example of the many AI / Machine Learning Applications. (See also Jarvis - smart assistant for Mark Zuckerberg's house [2]).

Immerse ourselves in the worldwide wonderful development, this document is addressed to give the readers an overview of Semi-supervised Learning (SSL), one of the fundamental ideas contributing to AI / Machine Learning Applications. SSL is a powerful solution which is using in most of the recent applications, because it reduces the cost while remaining the output quality. Hence, in this report, we highly recommend to implement the idea of Semi-supervised Learning by discussing from its Fundamental Theory to its Basic Application.

Not only theoretical expression, the report but also describes in detail a simple experiment which is, based on Semi-supervised Learning Theory, carried out from the first step of Feature Extracting Strategies to Perceptron Learning Algorithm (PLA) Implementation, then final to the Result Assessment. To approach SSL's example application, we firstly choose a supervised machine learning algorithm. In this case, the chosen one is PLA, because its theory is simple enough to understand and develop. Then the Sef-Training, a simplest implementation of Semi-Supervise Learning, is used to increase the performance of the PLA predict model.

The application will be developed by Python Language version 2.7 together with NumPy version 1.15.4 Library which is used in Linear Algebra Calculation and Matplotlib version 2.2.2 Library which is used to show the result on a Cartersian Coordinate Window. For more details, the reader is recommend to this Git Lab address link[3] .

[1] https://deepmind.com/research/alphago/

[2] https://www.facebook.com/zuck/posts/10103351073024591

[3] https://gitlab.com/trinhTran/semi-learning

Chapter 1

Introduction

1.1 General Motivation

Machine Learning is a subset of AI. By Wikipedia's definition, Machine learning is subfield of computer science that "gives computers the ability to learn without being explicitly programmed". Simply stated, Machine Learning is a small area of Computer Science, it is capable of self-learning based on input data without having to be programmed specifically. In recent years, when the computing power of computers has been raised to new heights and the huge amount of data collected by large technology firms, Machine Learning has gone a long way and a new field was born called as Deep Learning. Deep Learning has helped computers perform things that seemed impossible 10 years ago: categorizing thousands of different objects in photographs, creating captions for images, imitating human voices and scripts , communicating with people, or even writing or music.

There are two perspectives to group Machine Learning Algorithms. One is based on the learning path, the others is based on the algorithm functions. Semi-Supervised Learning is belong to the first group, learning path, which is composed by four participants:

- **Supervised Learning**: Supervised learning is an algorithm to predict the output of a new input based on known pairs input - output. This data pair is also called as labeled data. Supervised Learning is the most popular group in Machine Learning algorithms.

  Mathematically, Supervised Learning is when we have a set of input variables $X = x_1, x_2, \ldots, x_N$ and a corresponding set of labels $Y = y_1, y_2, \ldots, y_N$, where x_i, y_i are vectors. Data pairs know in advance $(x_i, y_i) \in X \times Y$ as training data. From this data collection, we need to create a function that maps each element from the set X to a corresponding (approximation) element of the set Y:

  $$y_i \approx f(x_i), \forall i = 1, 2, \ldots, N$$

  The goal of this above approximation function f is used for a new data x, the algorithm can calculate its corresponding label $y = f(x)$. One of the common example for the Supervised Learning Application is to detect faces in a photo. It has been developed for a long time. At first, Facebook uses this algorithm to indicate faces in a photo and asks users to tag friends which means they ask their users label their faces. The larger of the number of data pairs (faces and names), the greater of the accuracy at the next auto tags.

- **Unsupervised Learning**: The principle of the Unsupervised Learning is that the application do not know either label or output but only the input data. The algorithms will rely on their data structure to perform a certain task, such as grouping (clustering) or dimension reduction (data reduction) for storage and calculation. Mathematically, Unsupervised Learning has the input X together with unknown output Y. Unsupervised learning problems are divided into two categories: Clustering and Association

- **Reinforcement Learning**: Reinforcement Learning is to maximize the performance. It automatically chooses the best decision based on a certain situation. Currently, Reinforcement learning is mainly applied to Game Theory, algorithms need to determine the next move to achieve the highest score. There are two examples using this principle to optimize the performance: AlphaGo wins human [1] and Training for computers to play Mario games [2]

- **Semi-Supervised Learning**: In the case that the data set has a large amount of data X but only part of them is labeled, the others is unknown, as Semi-Supervised Learning. The problems of this group lie between the two groups mentioned above: Supervised Learning and Unsupervised Learning.
  A typical example of this group is a part of labeling photos or texts. Specifically, a photo of a person, animal or scientific or political text and most of them could be collected from the internet. In fact, many Machine Learning problems belong to this group because collecting data with labels is very time consuming and has a high cost. Many types of data even require a specialist to label such that medical images. Hence, this concept could solve the problems of Supervise Learning in labeling data.

In summary, SSL stands out by its practical value in learning faster, and cheaper but better. SSL is faster and cheaper because there are a large amount of unlabeled samples which are available with no expense, for instance, images can be obtained from surveillance cameras, documents can be crawled from the Internet, and speech can be collected from broadcast. Nevertheless, people have to annotate or set up many expensive laboratory experiments their corresponding labels such as intrusion detecting system, or medical images as ECG and MRI. The scarce of labeled data and a surplus of unlabeled data lead to the emergence of Semi-Supervised Learning Concept in utilizing the unlabeled data. Moreover, as a computational model for human learning, semi-supervised Learning finds itself in cognitive psychology applications. There is an evidence that human beings can combine labeled and unlabeled data to facilitate learning. For example, a child could learn by observing an object by herself (unlabeled data) then she combines from the words of her teacher (labeled data).

[1] https://www.engadget.com/2017/05/23/google-alphago-ai-wins-best-human-player/

[2] https://www.youtube.com/watch?v=qv6UVOQ0F44

1.2 Paper Objectives

Semi-Supervised Learning is a huge topic which could be expressed in a couple of hundred pages. The document is proposed to only give the reader its fundamental knowledge in the Semi-Supervised Learning concept. This document follows a Supervised Learning Analysis method which is composed of Generative and Diagnostic Step. However, it expands to describe a number of beyond models and their algorithms. The implementation part is discussed at the end of this document to give the readers a simple experiment, however, it is expected as a foundation for anyone who start researching this topic. In conclusion, the target of this document is introduce the definition of Semi-Supervised Learning and prove its performance when using together with a Supervised Learning Algorithm.

1.3 Contents Overview

This report is divided into four chapters: Introduction, Semi-supervised Learning Theory, Semi-supervised Learning Implementation, Conclusion.

The Introduction aims to provide an overview of Machine Learning / Semi-supervised Learning and their motivation in the modern life. The second chapter is using mathematical explanation, especially the probability knowledge to explain and prove the work of Semi-Supervised Learning Methods. The target of next chapter is to give the reader a simple exampled application which applies Semi-supervised Learning principle. Last but not least, the final one is an assessment on the results that we collected after going through the theoretical and practical parts. Based on this assessment, it is promised to express a further development sector where a couple of optimization approaches are proposed.

Chapter 2

Semi-supervised Learning Theory

2.1 An Initial Idea

Since 1960s, staring from an idea of Scudder [1], one of the oldest approach in Semi-Supervised Learning classification is introduced as self-learning, also known as self-training, self-labeling, or decision-directed learning. This technique stand out from a simple reasonable desire: how the unlabeled inputs could be made use of, together with the labeled inputs, to reduce the quantity of necessary labeled data but still remain the supervised learning performance. The algorithm is applied the unlabeled data set on a supervised method to predict their labels called as pseudo-labeled data set. Then, in each iterative, the application puts the pseudo-labeled data in the labeled data set. After N times iterations, the quantity of labeled data automatically is enlarged. This method is simple to implement but it has a couple of disadvantages. In Self-Labeled Methods Taxonomy, Self-learning is Single View, Single Learning, Single Classifier which means there are a lot of practical problems are refused to solve by it. An another important weakness is about the degradation of the self-labeling process. In the case if the confidence prediction function is not correct, the error will be much increased.

In contrast to inductive inference, concept of transductive inference is a near relation of SSL concept. Transduction was introduced by Vladimir Vapnik [2] in 1990s, which not provide a decision, it labels the unlabeled data only. Based on the concept, in 1967, Hartley and Rao [3] proposed a maximum-likelihood estimation which is used in the labelling test process to increase the quality of the predicting models.

[1] Scudder, H.J. Probability of Error of Some Adaptive Pattern-Recognition Machines. IEEE Transactions on Information Theory, 11:363–371 (1965). Cited in Chapelle et al. 2006, page 3.

[2] https://en.wikipedia.org/wiki/Vladimir_Vapnik

[3] https://www.jstor.org/stable/2333854?seq=1#page_scan_tab_contents

It could be considered the time, at which the generalization of Fisher's linear discriminant [4] was introduced, is the Semi-Supervised Learning era as a highly potential topic. Basically, two classes are called discriminative if the two classes are far apart (big between-class variance) and the data in each class tends to be the same to each other(small within-class variance). Linear Discriminant Analysis is to find a projection such that the ratio of the between-class variance over within-class is as maximized as possible. Using the concept in an iterative approach, the expectation-maximization algorithm [5] maximize the likelihood of the model by using the labeled and unlabeled data.

In the investigation of Ratsaby and Venkatesh (1995) [6], to train a classification model, the trade-off between labeled and unlabeled has been discussed as a parametric two-class problem. Their discovery is a premise of the probably approximately correct (PAC) framework which has been derived for the semi-supervised learning. In the same age, belong to pattern recognition topic, Castelli and Cover [7] proposed the relative value between labeled and unlabeled data with unknown mixing parameters. The result showed that a much large number of unlabeled points with response to the certain number of labeled points leads to an exponential convergence in the error probability.

Finally, in Natural Language problems and Text Classification made SSL concept taking off among the Machine Learning community such as Yarowsky, 1995; Blum and Mitchell, 1998; Joachims, 1999; etc.

2.2 SSL Working Domain

Likewise any mathematical defined domain, SSL could work together with a prerequisite: the distribution of training unlabeled data. In the most case, this distribution could be used to derive $P(y|x)$. Otherwise, the SSL concept could not improve the performance of any supervised learning algorithm. Be worse than that, because SSL may perform an inaccurate measure which pushes a couple of mislabeled data to the labeled set, leading to a performance degradation in training process.

- **Smoothness Assumption**: This assumption is applied to both regression and classification algorithm. Consider that there are two points $x_1 - y_1$ and $x_2 - y_2$. In the high density region, the distribution of $x_1 - x_2$ are near to each other then $y_1 - y_2$ are near as well, in contrast to, $x_1 - x_2$ are far to each other than $y_1 - y_2$ are expected not to be near, called as low density region. To generalize, the supervised learning produces the outputs which varies depending on the distance of input variation.

[4] https://en.wikipedia.org/wiki/Linear_discriminant_ analysis

[5] https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

[6] https://dl.acm.org/citation.cfm?id=225348

[7] http://www-isl.stanford.edu/ cover/papers/castelli_cover_96.pdf

- **Cluster Assumption**: Consider a cluster as a high density region, combining to the above assumption, it claims that two points are belong to a certain cluster, they potentially are in the same class. In additional, any cluster boundary should not cross an high density region but an low density region instead. The assumption is used to implement SSL concept to find the cluster boundaries.
- **The Manifold Assumption**: To a couple of data set, such as natural images, there not exist any distribution to cover them, if exist, the distribution is too complex to perform. Then the manifold is created and defined to capture data. The assumption is that the different manifolds, in which the different features of the data have been learned, are defined by using labeled data. Based on this concept, the high dimensional data is reduced to a low dimensional data in manifolds region. The growth of dimension leads to the exponential growth of required training data quantity and algorithm complexity. Taking the Manifold Assumption, the dimension is reduce which is made used of in many regression algorithm.
- **Transduction**: "When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one." - Vladimir Vapnik [8]. In the most of high-dimensional estimation problems, the above philosophy has been applied as a truth. To implement the SSL concept, there are a couple of algorithms which have a transductive setting, while the others could be inferred from the induction. In the case that the transductive algorithm produces an output used for an inductive algorithm which is trained by the same labeled set. There are two reasons affect the output quality: the generality of transduction compared to the induction following to the Vapnik's principle; the transductive algorithm and the semi-supervised learning algorithm are used in the similar way. However, the transductive algorithm and the semi-supervised learning algorithm are difference; while transduction is about to label samples, the SSL concept is trying to build a data-driven model, then derive a general rule for the entire region/ cluster.

2.3 SSL Methods & Algorithms

Semi-Supervised Learning is a principle which is used and inherit to establish a couple of mathematical models. Each of the model is more specific to solve a certain problem type. The SSL concept is a premise on many models, the followings are the most common examples which are used in various fields.

[8] https://en.wikipedia.org/wiki/Transduction_(machine_learning)

2.3.1 Generative Models

Suppose that x_1 is an input point (unlabeled data) which is required to be labeled Y. In Machine Learning, the most of algorithms is to find the probability if Y is true in the case of observing x_1; revise Bayes' Theorem [9] for $P(Y \mid X = x_1)$:

$$P(Y \mid X = x_1) = \frac{P(X = x_1 \mid Y)\,P(Y)}{P(X = x_1)}, \tag{2.1}$$

where $P(X) \neq 0$. $P(Y \mid X)$ and $P(X \mid Y)$ is a conditional probability. The likelihood of event X when Y is true. Vice versa, the likelihood of event Y in the case that $X = x_1$ appears. $P(X)$ and $P(Y)$ are the marginal probability.They are independently observed X and Y.

Consider two random variables X and Y. If we observe a lot of pairs of outputs X and Y, there are a couple of combinations of outputs that occur more often than others. This information is represented by a distribution called the joint probability of X and Y, written as $P(X,Y)$:

$$P(X) = \sum_{Y=y} P(X,Y) \tag{2.2}$$

$$P(X,Y) = P(Y \mid X)P(X) = P(X \mid Y)P(Y) \tag{2.3}$$

From the equation (2.1), it is derived:

$$P(Y \mid X = x_1) = \frac{P(X = x_1 \mid Y)P(Y)}{\sum_{Y=y} P(X = x_1 \mid Y)P(Y)} \tag{2.4}$$

The above equation is one of the most important model in Machine Learning and SSL as well. It is no doubt that once the $P(X \mid Y)$ is determined, the probability $P(Y \mid X)$ could be found. In the case that there is an input x_1, the output is the probability that x_1 is belong to class y. The Generative Models is defined as $P(X \mid Y)P(Y)$, determine how labeled data could be generated and distributed.

2.3.1.1 Generative Paradigm

As described above, to calculate $P(Y \mid X)$, it is required to know the $P(X \mid Y)P(Y)$. Currently, there is no solution to directly calculate this probability; hence; this paradigm proposes an architecture which helps to find an estimation. Assume that the input point set has a certain distribution, which is denoted as a vector $\boldsymbol{\theta}$. Hence, the class conditional distribution should be $P(X \mid Y, \boldsymbol{\theta})$. By using Chain Rule[10]:

$$P(X \mid Y, \theta) = \frac{P(X, Y \mid \theta)}{P(Y \mid \theta)} \tag{2.5}$$

[9] https://en.wikipedia.org/wiki/Bayes%27_theorem

[10] https://en.wikipedia.org/wiki/Chain_rule_(probability)

Then, assume that the output has its own a distribution called as $\boldsymbol{\pi}$. Hence, the $P(Y)$ in the equation 2.4 is described as $P(Y \mid \boldsymbol{\pi})$. The equation 2.4 is generalized to:

$$P(Y \mid X,\theta,\pi_Y) = \frac{P(X \mid Y,\theta)P(Y \mid \pi)}{\sum_{y=1}^{M} P(X \mid y,\theta)P(y \mid \pi)}, \tag{2.6}$$

where M is the quantity of labels. There are no any direct solution to calculate $P(Y \mid X,\theta,\pi_Y)$ but an estimation approach by maximizing this probability. Investigate this equation, the denominator is a constant sum which should not affect the function direction. Therefore, the numerator is desired to maximize. From Joint Probability, the marginal probability of X by marginalization:

$$\sum_{y=1}^{M} P(X,Y \mid \theta) = P(X \mid \theta) \tag{2.7}$$

According to equation 2.5, this expression is:

$$\sum_{y=1}^{M} P(X \mid Y,\theta)P(Y \mid \theta) = P(X \mid \theta) \tag{2.8}$$

Consider $x_1, x_2, ...x_n$ is labeled data, and $x_1, x_2, ...x_m$ is unlabeled data. Assume data points are independent events, the above expression must contains many multiplied parts:

$$P(x_1, x_2, ...x_N \mid \theta) = P(x_1 \mid \theta)P(x_2 \mid \theta)...P(x_N \mid \theta) \tag{2.9}$$

Unfortunately, the multiplication makes the maximum method becomes very hard to solve. This could be solve if this function is transferred to logarithm function in which the multiplication becomes a summation. Moreover, logarithm is a strictly monotonically increasing function, which means this function reach the maximum at the maximum variable.

$$\theta = \operatorname*{argmax}_{\theta} \sum_{i=1}^{n} \log \pi_{y_i} P(x_i \mid y_i,\theta) + \sum_{i=n+1}^{m} \log \sum_{y=1}^{M} \pi_y P(x_i \mid y,\theta) \tag{2.10}$$

The above function could be used to train θ with data training set by Gradient Descent Method or Expectation Maximization. Finally, the output is best-matched vector θ such that the probability $P(Y \mid X,\theta,\pi_Y)$ is maximum. The above expression is fitted for both labeled and unlabeled data set.

2.3.1.2 Diagnostic Paradigm

In any Machine Learning topic, the Diagnostic method focus on modeling the conditional distribution $P(Y \mid X)$, while the Generative method estimate $P(Y \mid X)$ by modelling the joint distribution $P(Y,X)$, then using its characteristics to fit training data set with $P(Y \mid X)$. In Generative method, the $P(X)$ could be estimated by marginalizing, while $P(X)$ could not be derived from the Diagnostic method.

As the above introduction, the method directly model the conditional distribution family $P(Y \mid X,\theta)$. Denote the distribution $P(X \mid \mu)$ to express the sampling model for input data, where θ and μ are a priori independent. Recall the problem of SSL, there is a small labeled sample set $D_l = \{(xi, yi)|i = 1,...,n\}$ and a larger unlabeled sample set $D_u = \{x_{n+j}|j = 1,...,m\}$ using the marginal $P(X \mid \mu)$. Then denote:

- $X_l = (x_1,...,x_n)$ are the labeled samples.
- $Y_l = (y_1,...,y_n)$ are the labels of labeled samples.
- $X_u = (x_{n+1},...,x_{n+m})$ are the unlabeled samples.
- $Y_u = (y_{n+1},...,y_{n+m})$. are the labels of unlabeled samples. Note that they have not been observed yet.

Base on Bayesian statistics, the likelihood is:

$$P(D_l, D_u \mid \theta, \mu) = P(Y_l \mid X_l, \theta)P(X_l, D_u \mid \mu) \tag{2.11}$$

Recall that Posterior Probability $\propto$ Likelihood x Prior Probability. Based on that, it is derived as: $P(\theta|D_l, D_u) \propto P(Y_l|X_l,\theta)P(\theta)$. Moreover, θ and μ are a independent, hence:

- $P(\theta|D_l, D_u) = P(\theta|D_l)$,
- $P(\theta|D_l, \mu) = P(\theta|D_l)$.

From these above expression, it is no doubt that both D_u and μ cannot affect the belief θ of labeled data. Therefore, Bayesian inference could not be used together with the unlabeled data.

A reason why the unlabeled data could not make use of in a straight diagnostic Bayesian methods is proposed as θ and μ are a independent. Then a new concept was created to solve this problem, called as "Regularization Depending on the Input Distribution" [11]. This solution points that if θ and μ are allowed to depend on each other by $P(\theta, \mu) = P(\theta \mid \mu)P(\mu)$, the information of μ is transferred to θ, leads to θ and unlabeled data are dependent on the given labeled data. In another words, the unlabeled data affects to update the θ. In the following part, the document is proposed to introduce a couple of implementations which are inherited this concept to their algorithm.

2.3.1.3 Generative Implementation

There are many algorithms but the followings are chosen because of their common in the Machine Learning community:

[11] O. Chapelle, A. Zien, and B. Sch¨olkopf, editors. Semi-supervised learning. MIT Press, 2006, p. 19-20

Co-Training

This algorithm was introduced in 1998 by Avrim Blum and Tom Mitchell [12]. The main idea of this approach is observing input data with different views. One of the common example is a binary classification problem on web pages, there are two views: content-text and hyperlink-text. This method is the same with the Self-Training method which has been expressed in the above section 2.1. However, instead of Self-Training could be used for one view, this method could be used with multiple views, as the Figure 2.1 illustrates below [13]. Firstly, training with labeled data set to creates a weak classifier for each view. Each classifier is used to predict the pseudo labels for the unlabeled examples then add the most confident classifications to the labeled data set.

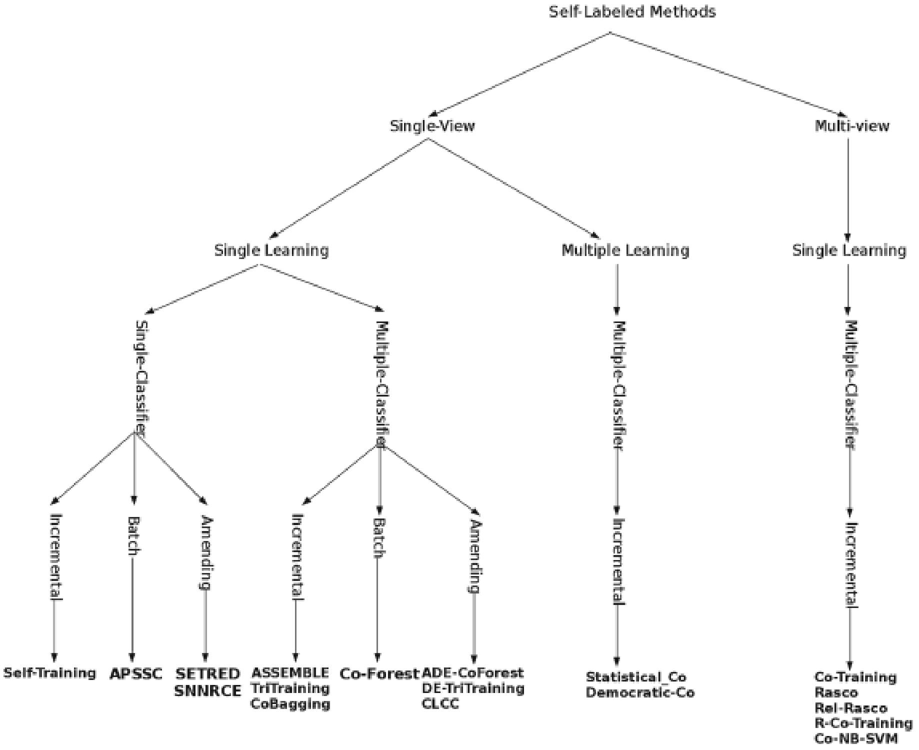

Figure 2.1: Self-labeled techniques hierarchy

The Fisher Kernel

This algorithm was introduced in 1999 by Jaakkola and Haussler [14]. Recall the generative model $P(x \mid \theta)$. Denote the log-likelihood of the generative model as $log_e P(x \mid \theta)$, then the $\nabla\theta$ is the gradient operator. Then the Fisher score, U_x, is :

$$U_x = \nabla\theta log_e P(x \mid \theta) \tag{2.12}$$

Assume to the binary classification problem, which means there are 2 classes. In this setting, $P(x \mid \theta)$ contains the knowledge of the unlabeled data set, and the Fisher

[12] Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training. COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann, 1998, p. 92-100

[13] Isaac T., Salvador G. and Francisco H. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Springer-Verlag London, 5 November 2013

[14] Tommi Jaakkola and David Haussler (1998), Exploiting Generative Models in Discriminative Classifiers. In Advances in Neural Information Processing Systems 11, p. 487–493. MIT Press

kernel is constructing a co-variance kernel which depends on this generative model.

$$K(x_i, x_j) = U_{x_i}^T I^{-1} U_{x_j} \tag{2.13}$$

where I is the Fisher information matrix:

$$I = E_{P(.|\theta)}[F_\theta(x) F_\theta(x)^T] \tag{2.14}$$

We can define a scale parameter α for the Fisher information matrix, as αI in a variant. Then, using the Fisher score, they are obtained as the feature vector for x and pluged into a standard Fisher kernel. The Fisher kernel defines a distance between the two above classes which can be used with any kernel-based classifier such as a support vector machine (SVM).

Label Propagation

This algorithm is based on the Cluster Assumption, it is introduced by Zhu and Ghahramani in 2002[15]. Among all the approaches and techniques in Machine Learning, the Label Propagation is neither the most accurate nor the most robust method, however, it is one of the simplest and fastest clustering methods. The main idea is, with a graph of a small subset of labeled vertices, but a large subset of unlabeled vertices, the method aims to diffuse labeling to all nodes on the graph based on the labels that the neighboring nodes possess. Within one degree of the nodes, this change is subject to the maximum number of labels. After creating these dense consensus groups throughout the network, they continue to expand outwards as far as possible. The Label Propagation algorithm has 5 main steps:[16]

Algorithm 1 Label Propagation

```
1: repeat
2: Y = TY
3: Row-normalize Y
4: Clamp the labeled data
5: Until Convergence
```

In above algorithm, the row dimension of matrix Y is the labeled and unlabeled data set in such order that the labeled is above of the unlabeled data. They contain the probabilities that a point belongs to a given class. The column dimension of matrix Y represent the size of the data set and the number of classes: $Y = \begin{bmatrix} Y_L \\ Y_U \end{bmatrix}$, where Y_L is labeled and Y_U is unlabeled data. The matrix T is an N × N transition matrix realizing the propagation of the labels. In the above algorithm, the equation in step 2 is:

$$\begin{bmatrix} Y_L \\ Y_U \end{bmatrix} = \begin{bmatrix} T_{LL} & T_{LU} \\ T_{UL} & T_{UU} \end{bmatrix} \cdot \begin{bmatrix} Y_L \\ Y_U \end{bmatrix} \tag{2.15}$$

[15] Zhu, Xiaojin and Ghahramani, Zoubin. Learning From Labeled and Unlabeled Data With Label Propagation, 2002.

[16] Zal´an Bodo´and Lehel Csato’. A note on label propagation for semi-supervised learning, 2015

From the expression 2.15, it is derived as following:

$$Y_U = T_{UL} Y_L + T_{UU} Y_U. \tag{2.16}$$

Finally, it is no doubt that the labels of the unlabeled data could be marked from the known information:

$$Y_U = (I - T_{UU})^{-1} T_{UL} Y_L. \tag{2.17}$$

Note that, Label Propagation is considered as a mixture of the Generative SSL Methods with Clustering Assumption and Graph-Based Methods.

2.3.2 Low-Density Separation

Inherit the concept of Self-Training and use with Maximum Margin Algorithm, the Low-Density Separation model becomes a common approach which successfully solved many practical problems. Starting from a solution to train the labeled data, then using the trained models to predict the unlabeled examples. The algorithms are iteratively retrained on all points, after each iteration, the weight of the unlabeled points are adapted. As a consequence of the smoothness assumption, the target probabilities P(y|x), as discussed in the above section, is close to either 1 or 0 with respect to the labeled and unlabeled points, hence, there are high-density region and low-density region. The low density regions are called as the class boundaries or decision boundaries which correspond to intermediate probabilities. This seems to implement the low-density separation concept, hence, all algorithms using it will be classified in this section.

Assumes that there are two high-intensity regions as -1 and 1 which are separated by a decision boundary $f(x) = 0$. It is situated in a low-density region, where x are unlabeled points. Hence, there is a simple loss function L as following:

$$L = max(1 - | f(x) |, 0) \tag{2.18}$$

The above equation is always positive with the absolute values of $f(x)$. Denote l and u is respectively the indices of labeled and unlabeled points. The loss function measures the violation in large margin separation between $f(x)$ and x, which is expected to low for labeled and high for unlabeled data. Ordering by this idea, the following expression is the average of the loss function:

$$L_{avg} = \frac{1}{u} \sum_{i=l+1}^{l+u} max(1 - | f(x_i) |, 0) \tag{2.19}$$

All algorithms implementing this concept are expected to train and choose a model $f(x)$ such that the loss function L_{avg} is minimum:

$$f(x) = \underset{f}{\text{argmin}} \frac{1}{l} \sum_{i=1}^{l} max(1 - | y_i f(x_i) |, 0) + \lambda_1 \| y \|^2 + \lambda_2 \frac{1}{u} \sum_{i=l+1}^{l+u} max(1 - | f(x_i) |, 0) \tag{2.20}$$

2.3.2.1 Low-Density Separation Implementation

In practical, the above optimization problem is difficult to solve because the loss function is non-convex. There are many algorithms but the following is chosen because of their common in the Machine Learning community:

Transductive Support Vector Machines

TSVM is not a supervised learning but a semi-supervised learning because of two main reasons. First, in the training step with the labeled data, the algorithm is not expected to exploit the general rules in labeled data but a reference to predict the unlabeled data. Second, it considers the unlabeled data as a prior which contains information in the learning step. It means a transductive learner know the geometry of the unlabeled data when defining the cluster, while an inductive learner ignores this information.

As the setting above, Vapnik, V. (1998) proposed an objective method [17] to increase the qualification of a Support Vector Classifier with a small set of labeled data and large set of unlabeled data. In one dimensional data set, the separator is a point, in two dimensional data set is a line, in three dimensional is a plane, and the above is a hyper-plane $H(x) = \psi^T x + b = 0$. In each dimension, this hyper-plane is expressed as a boundary following a linear function and the classifier $y(x) = sign(\psi^T x + b)$. Then, a margin is the distance from the nearest points to a hyper-plane in d dimensions is:

$$M_0 = \min_n \frac{|\psi^T x_n + b|}{||\psi||_2} \tag{2.21}$$

, where M_0 is its margin.The objective of SVM algorithm is to determine an optimal margin such that the distances from all nearest points to this margin is maximum. Note that, a label y(x) is expected to have the same sign with its data x, hence, the above equation is:

$$(\psi, b) = \operatorname*{argmax}_{\psi,b}(\min_n \frac{(y_n(\psi^T x_n + b)}{||\psi||_2}) \tag{2.22}$$

As a definition of "the nearest points", the information $\min_n(y_n(\psi^T x_n + b))$ is a convention and should not change, hence, the above equation become:

$$(\psi, b) = \operatorname*{argmax}_{\psi,b} \frac{1}{||\psi||_2} = \operatorname*{argmin}_{\psi,b} \frac{1}{2}||\psi||_2 \tag{2.23}$$

Note that, the above expression is valid, subject to:

- Labeled Data: $\forall_{i=1}^{l} y_i(\psi^T x_i + b) \geq 1$
- Unlabeled Data: $\forall_{j=1}^{u} y_j(\psi^T x_j + b) \geq 1$
- Classifiers: $\forall_{j=1}^{u} y_j \in 1, -1$

Beside that, this model is a fundamental to implement many techniques which are used in various fields such as Gaussian Processes and Entropy Regularization. In this document, the author would like to discuss the Semi-Supervise Learning concept, hence, it is proposed to only the highlight points of this topic. Hence, the algorithms are not described in details in this document.

[17] Vapnik, V. Statistical Learning Theory. Wiley, 1998, New York, p.434-437

2.3.3 Graph-Based Method

The Graph-Based Method, stands out in semi-learning community because of its advantages for learners' visualization. From its title, there is a graph G = V, E and a training instance set. In semi-learning, the training instances could be labeled and unlabeled data. The edges E connecting these instances i, j with weight w_{ij} and the vertices V set up the graph G. In most of algorithm, the weights reflect the data proximity, for instance, the Gaussian edge weight function [18] or the k-Nearest Neighbor edge weight function [19].

The Laplacian matrix of a simple graph G with V vertices, not containing loops or parallel edges, L_{VxV} is defined as: $L_{VxV} = D - A$, where D is the degree matrix and A is the adjacency matrix of the graph. The energy of a graph is defined as a sum of the absolute values of the eigenvalues of the adjacency matrix of the graph. Denote $\lambda_i = [1...V]$ be the eigenvalues of A. Then the energy of the graph is defined as:

$$E(G) = \sum_{i=1}^{l+u} |\lambda_i| \tag{2.24}$$

Because the adjacency matrix A is represented to a link of any vertices pair, it could be defined as an edge weight function:

$$E(G) = \frac{1}{2} \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 \tag{2.25}$$

The energy of a graph is also the spectrum of the graph. In a case that G is totally disconnected graph its energy is zero, while its energy is equal to 2(v-1) if G is a completed graph and v is its vertices. The target of this concept is to find the labelling function which is calculated from the labeled nodes, then determine that the nearby unlabeled nodes should have the same labels. To develop an algorithm which is belong to the Graph-Based Method, the smoothness assumption and manifold assumption should conventionally be applied. One of an example is called Harmonic Functions[20]:

$$f = \underset{f_L = y_L}{\operatorname{argmin}} \frac{1}{2} \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 \tag{2.26}$$

There are many algorithms that used the models as their concept such as: Label Propagation with Quadratic Criterion, Discrete Regularization, and Conditional Harmonic Mixing. To the interested readers, they are referred to the book "Semi-Supervised Learning"[21].

[18] Mitra Basu. Gaussian Based Edge Detection Methods-A Survey. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, vol. 32, no. 3, Agust 2002

[19] Kozak K, M. Kozak, K. Stapor. Weighted k-Nearest-Neighbor Techniques for High Throughput Screening Data. International Journal of Chemical, Molecular, Nuclear, Materials and Metallurgical Engineering vol.1, no.12, 2007

[20] Xiaojin Zhu, Zoubin Ghahramani, John Laffert. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. 2002,2003

[21] Olivier C., Bernhard S., and Alexander Z. Semi-Supervised Learning. The MIT Press, 2006, p. 191-273.

Chapter 3

SSL - A Simple Implementation

"What I cannot create, I do not understand."
—Richard Feynman

3.1 Perceptron Learning Algorithm

3.1.1 Introduction

Perceptron Learning Algorithm - PLA [1] is a Classification algorithm for the simplest cases such as binary classification, however, it is a cornerstone of any important major part of the later Machine Learning techniques in both Neural Networks and Deep Learning. Hence, in this section, the author choose it as an example to implement the SSL concept. Suppose there are two labeled data sets illustrated in the left side of Figure 3.1, which is blue points and red points. The problem is that: from the data of the two given labels, build a classifier so that when there is a new gray triangle data point, the algorithm can predict the color as its label. In other words, each class has its own territory so that, for each new point coming, the algorithm can determine the class it belongs to. These territories are separated by a boundary, therefore, classification problem can be considered as boundary finding problem between classes. The simplest boundary in two-dimensional space is a straight line, in a three-dimensional space is a plane, in a multi-dimensional space is a hyper-plane. These flat boundaries are simple because they can be expressed mathematically with a simple function of linear form. In this case which is illustrated in the right side of Figure 3.1, a line that divides two classes in the plane. The section with a blue background is considered as the territory of the blue class, the others is the territory of the red class. In this case, the new triangle data point is assigned to the red class.

[1] https://machinelearningcoban.com/2017/01/21/perceptron/

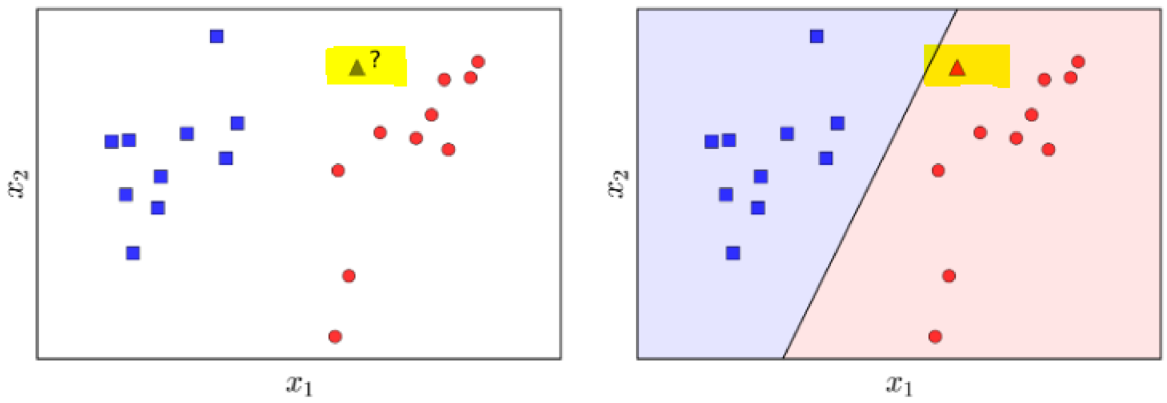

Figure 3.1: PLA Example

In summary, the PLA problem is stated as: For two labeled classes, find a flat line such that all of the class 1 points are on one side, all the points of class 2 are on the other side of the flat line.

3.1.2 Algorithm Establishment

3.1.2.1 Mathematical Model

Assume that $X = [x_1, x_2, ...x_N] \in \mathbb{R}^{dxN}$ is a matrix containing data points, each column $x_i \in \mathbb{R}^{dx1}$ is a point in d-dimensional space. Denote $Y = [y_1, y_2, ...y_N] \in \mathbb{R}^{1xN}$ is the label set for each data point, with y_i = 1 if x_i is belong to blue class, otherwise y_i = -1 for red class. In each iteration, the boundary line is:

$$f_w(x_1) = w_1 x_{11} + w_2 x_{12} + ... + w_d x_{1d} + w_0 = w^T \bar{X}, \tag{3.1}$$

where $\bar{X}$ is X adding one more value, called as $x_0 = 1$. In the above case, the d = 2, hence the above equation is $w_1 x_1 + w_2 x_2 + w_0 = 0$ is the below boundary in Figure 3.2.

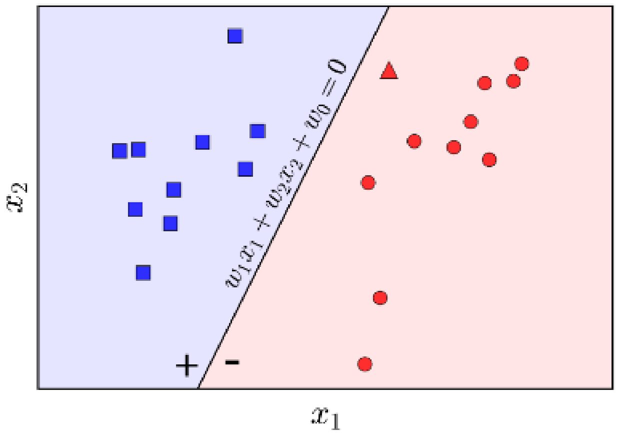

Figure 3.2: PLA Boundary Function

From the Figure 3.2 above, if the boundary is true, all residents in blue region should have the same positive sign and opposite to the red region with negative sign. Therefore, mathematically, labelling a data point can be expressed as following:

$$Lable(x) = \begin{cases} 1, & \text{if } w^T x \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{3.2}$$

In a short form, the above equation is Label(x) = $sgn(w^T x)$.

3.1.2.2 Loss Function

It is clear that the algorithm can not reach the best boundary right at the first iteration, hence, there must be an idea such that after each iteration, the result approaches directly towards the prior desire, instead of far away to it. Based on the above example, there is a boundary function with d = 2. In the Figure 3.3, the circled points are misclassified points, while the objective is no mis-classification. The simplest loss function is proposed to count the number of mis-classied points and minimize this quantity:

$$L(x) = \sum_{x_i \in M} (-y_i sgn(w^T x_i)) \tag{3.3}$$

, where M is the mis-classified data set. The minimum of the above L(x) = 0 when there is no any mis-classified points and the algorithm should be stop.

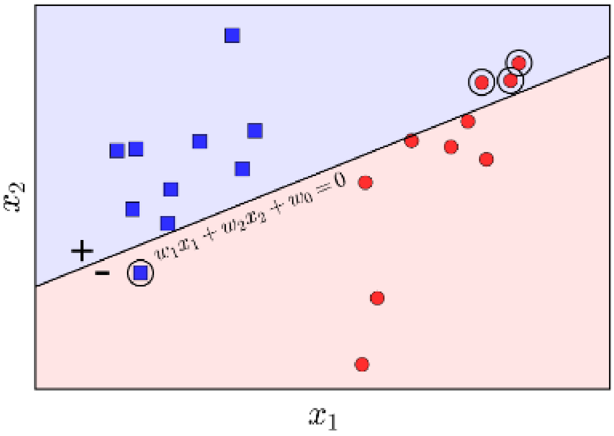

Figure 3.3: PLA Mis-classification

It is clear that the above expression containing sgn function is hard to take its derivative, hence, in the most of practical applications, the sgn function usually not be realized in the Loss function. Based on the Loss Function to assess the quality of the output result, after each iteration, the weight w should be updated following to:

$$w_{t+1}^T x_i = (w_t + y_i x_i)^T x_i = w_t^T x_i + y_i ||x_i||_2^2 \tag{3.4}$$

Assume that $y_i = 1$, leading to $w_t^T x_i < 0$ (because $x_i \in M$ and recall the Equation 3.2). In the Equation 3.4, $y_i ||x_i||_2^2 \geq 1$ because $x_0 = 1$, hence, $w_{t+1}^T x_i > w_t^T x_i$. In other words, updating w is to move the boundary towards a position such that the mis-classified is correctly classified.

3.2 PLA with Self-Learning

3.2.1 Introduction

As the above introduction, Self-Learning is belong to SSL concept but it is not a certain algorithm. Its idea could be used together with a couple of Supervised Learning Algorithms to improve the quality of their output in the case that there are a small labeled data set and a huge unlabeled samples. The combination between Self-Learning and PLA is simple enough to individually implement with a basic set up. Moreover, the two are the fundamental idea which is inherited and expanded among many applications later. Hence, this example is desired to highlight the characteristics of Semi-Supervised Learning. The Self-Learning approach is described as these following step:

Input:

- Labelled data-set L
- Unlabelled data-set U
- Confidence threshold θ

Output:

- Predictive Model H'

Algorithm 2 Self - learning

```
Train model H by labeled data-set L with PLA (Supervised Learning)
Use model H to predict unlabeled data-set U
Select samples with threshold $\theta$, then $U \leftarrow U'$
Train model H' by labeled data-set L + sub-set unlabeled U'
```

3.2.2 Finding a Diamond

This is the name of this set up - FaD. The objective of FaD is to classify a rhombus out of the normal quadrilaterals. A rhombus is a quadrilateral whose four sides all have the same length, as shown in the Figure 3.4[2]. Hence, the feature vector of input data set are three ratio - pairs of the edge lengths.

[2] https://en.wikipedia.org/wiki/Rhombus

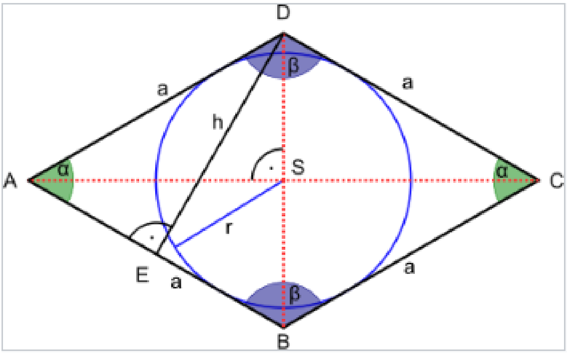

Figure 3.4: A Rhombus

First of all, the application generates a 10 by 10 grid, where it chooses 4 points set to make up a normal convex. This is a combination of 4 in 100 points:

$$^{100}C_4 = \frac{n(n-1)...(n-k+1)}{k(k-1)...1} = \frac{100*99*98*97}{1*2*3*4} \tag{3.5}$$

The size of data set is about 4 million quadrilaterals, but because of the simplicity of this problem, the application choose only 10 thousands samples in them.

Algorithm 3 Finding a Diamond

```
1: Generate 10 thousand samples
2: Add random noise
3: Assign 1 percent as Training Set and 99 percent as Test Set
4: Hide 90 percent labeled data to get unlabeled data
5: Run Algorithm 2: Self-Learning with the Training Set
6: Test predictive output model from Algorithm 2 by Test Set
```

3.3 Result & Discussion

To validate the performance of the proposed Semi-Supervised Learning concept, the experiment is set up to run many times with a couple of targets.

3.3.1 Algorithm Convergence

In 50 running times, ninety percent of both Perceptron Learning Algorithm (Supervised Learning) and Self-Learning (Semi-Supervised Learning) are converge after a couple of iterations.

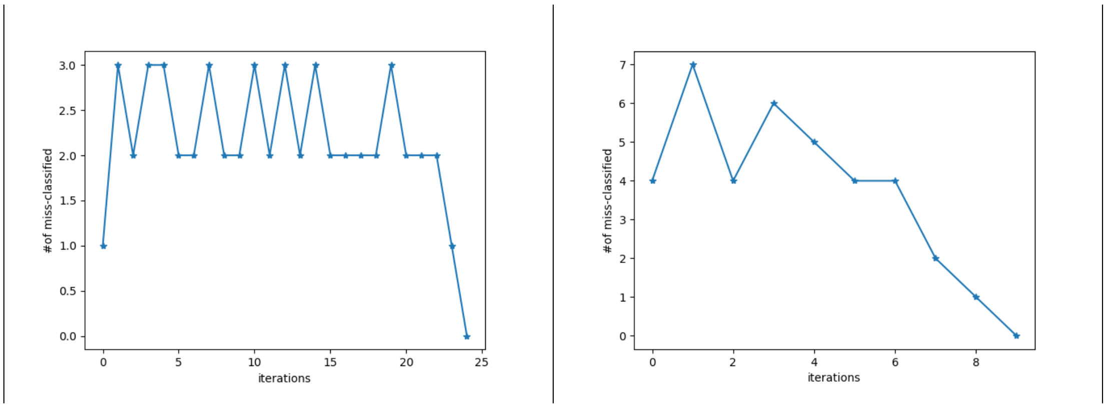

Figure 3.5: One of the running times. The left figure is using PLA with 10 percent of labeled data in training set. The right figure is using PLA, improved by Self-Learning with 10 percent of labeled and 90 percent of unlabeled data in training set.

As above discussion, the main idea of PLA is setting an arbitrary boundary, then counting the number of mis-classified samples. Based on this number, the weight has been updated until the number reduces to zero. Hence, in the Figure 3.5, the number of mis-classified samples do not decrease smoothly but fluctuate a lot. However, the performance of PLA is not good, because 10 percent of the training data set is small, then this algorithm need many iterations before completing to calculate the predictive model. On the other hand, the iterations of Self-Learning is not as much as PLA. This is expected result. The reason is that the predictive model has already been trained one times, and continue training one more time in Self-Learning step. Moreover, the second training has more data than the first one. It is clear that the execution time of PLA before Self-Learning is faster than the combination between PLA and Self-Learning, however, the second training step should be fast convergent. Therefore, the execution time of the combination should not run too much more slowly than the PLA.

3.3.2 Algorithm Robustness & Accuracy

The robustness and accuracy of any algorithm or approach is the most important value that people concern. The robustness is measured by continuously running several times, then as more fluctuation as less robustness. On the other hand, the accuracy is followed this establishment but it calculates the mean value of number of error samples along with the running times. The next experiment aims to analysis the robustness and the accuracy of the two above algorithms. This application is set up to run 20 times continuously in the training set and measure the error with the test set.

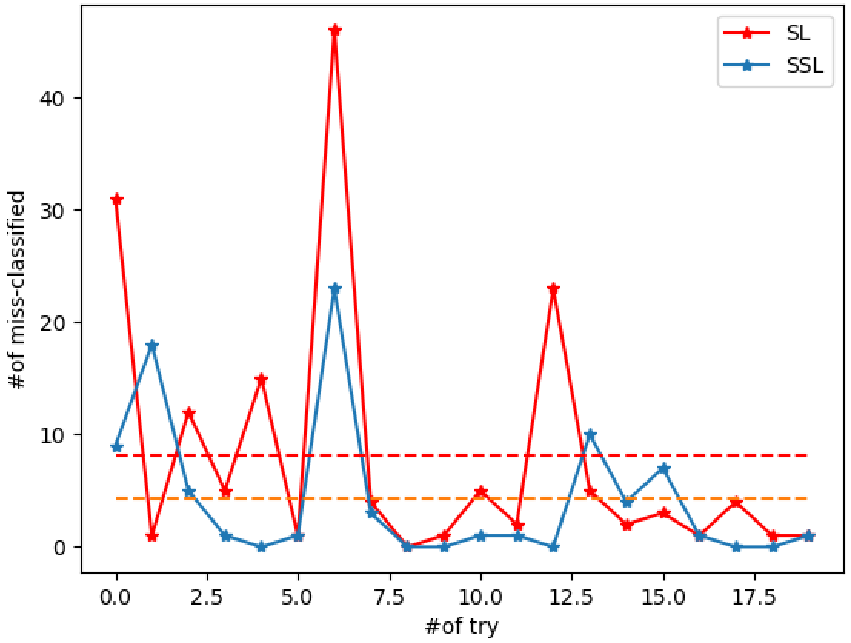

Figure 3.6: The horizontal axis is the number of running times. The vertical axis is the number of mis-classified samples in test set. The red line is the result when using PLA with 10 percent of labeled data in training set. The blue line is the result when using PLA, improved by Self-Learning with 10 percent of labeled and 90 percent of unlabeled data in training set. The red dash line is the mean value of PLA, the orange dash line is the mean value of the combination of PLA and Self-Learning.

In the Figure 3.6 below, the fluctuation of PLA with the same training set and test set but different running times is higher than the combination which is used PLA with Self-Learning method. After a couple of experiments, the pattern of the above figure is likely to the other. Hence, it could be concluded that the Self-Learning improved the robustness and accuracy of PLA in the case when there is a small labeled data set and the big unlabeled data set.

3.3.3 Quantity - Quality Correlation

In the two above experiments, the labeled data takes account of 10 percent of training set and the other is unlabeled data. In this experiment, the quantity of labeled and unlabeled data are changed and then measure the mis-classified samples in test set. This experiment could illustrate the correlation of the data quantity and accuracy in the both approaches. Hence, the section is expected to give the readers an overview of the utilization when using Self-Learning in PLA.

Table 3.1: Quantity - Quality Correlation

% Labeled	% Unlabeled	PLA	PLA + SL
10	90	5.90	1.20
20	80	4.85	0.85
30	70	2.65	0.80
40	60	0.95	0.55

From the table 3.1, it is no doubt that together with the increase of labeled quantity, the mean value of the number of mis-classified samples after 50 running times decrease in both PLA and the combination of PLA and Self-Learning (SL). When the labeled data set increases, the model H in the Algorithm 2 is stronger, which makes the classifier H' stronger as well. Hence, this result is as our expectation. The table also shows us that with using Self-Learning method, people do not need to spend time and cost to notate or label their input samples. Therefore, Self-Learning is one of the most potential method which could use in case there are not enough labeled data for a Supervise Learning algorithm.

Chapter 4

Conclusion

Since the mid-nineties, the Semi-Supervised Learning concept was born with the essential desire among Machine Learning community. Together with the Internet improvement and many digital technique appearance, the waste of the unlabeled data could not be ignored anymore. Hence, many mathematicians, scientists and developers created and inherited the Semi-Supervised concept to solve their problems with the more efficient approaches.

This principle quickly emerges in various scientific fields, because of not only its capability in improving the learning quality, but also its interesting philosophy. In the second chapter, the document theoretically discusses a couple of its models and their algorithms in both programming and mathematics. This is a premise on the next chapter as an implementation. The implement is a combination of two fundamental ideas in both Semi-Supervised Learning and Supervised Learning. Hence, they are the simplest but nearest by their both concepts.

The document is also expressed three experiments with different target. Firstly, the result proved that the Semi-Supervise Learning is a convergence method, which means the classifier should be derived after a number of iterations. Secondly, the result proved that the Semi-Supervised Learning improves the performance of the Supervised Learning in the case there are not enough labels. Finally, when the labels are collected more and more the efficiency of the Semi-Supervised Learning is not significant, compared to the Supervised Learning. This indicates that by using Semi-Supervised Learning, people could decrease the time and expense to label data. By using both the mathematical and implemented proof, the author could conclude that Semi-Supervised Learning works efficiently and it could improve the performance of many Supervise Learning algorithms in a certain case.

Reference

[1] Scudder, H.J. Probability of Error of Some Adaptive Pattern-Recognition Machines. IEEE Transactionson Information Theory, 11:363–371 (1965). Cited in Chapelle et al. 2006, page 3.
[2] https://en.wikipedia.org/wiki/Vladimir_Vapnik
[3] https://www.jstor.org/stable/2333854?seq=1page_scan_tab_contents
[4] https://en.wikipedia.org/wiki/Linear_discriminant_ analysis
[5] https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm
[6] https://dl.acm.org/citation.cfm?id=225348
[7] http://www-isl.stanford.edu/ cover/papers/castelli_cover_96.pdf
[8] https://en.wikipedia.org/wiki/Transduction_(machine_learning)
[9] https://en.wikipedia.org/wiki/Bayes%27_theorem
[10] https://en.wikipedia.org/wiki/Chain_rule_(probability)
[11] O. Chapelle, A. Zien, and B. Sch ¨olkopf, editors. Semi-supervised learning. MIT Press, 2006, p. 19-20
[12] Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training. COLT: Proceedings ofthe Workshop on Computational Learning Theory, Morgan Kaufmann, 1998, p. 92-100
[13] Isaac T., Salvador G. and Francisco H. Self-labeled techniques for semi-supervised learning: taxon-omy, software and empirical study. Springer-Verlag London, 5 November 2013
[14] Tommi Jaakkola and David Haussler (1998), Exploiting Generative Models in Discriminative Classi-fiers. In Advances in Neural Information Processing Systems 11, p. 487–493. MIT Press
[15] Zhu, Xiaojin and Ghahramani, Zoubin. Learning From Labeled and Unlabeled Data With Label Prop-agation, 2002
[16] Zal ´an Bodo ´and Lehel Csato'. A note on label propagation for semi-supervised learning, 2015
[17] Vapnik, V. Statistical Learning Theory. Wiley, 1998, New York, p.434-437
[18] Mitra Basu. Gaussian Based Edge Detection Methods-A Survey. IEEE Transactions on Systems, Man,and Cybernetics—Part C: Applications and Reviews, vol. 32, no. 3, Agust 2002
[19] Kozak K, M. Kozak, K. Stapor. Weighted k-Nearest-Neighbor Techniques for High Throughput Screen-ing Data. International Journal of Chemical, Molecular, Nuclear, Materials and Metallurgical Engineeringvol.1, no.12, 2007
[20] Xiaojin Zhu, Zoubin Ghahramani, John Laffert. Semi-Supervised Learning Using Gaussian Fields andHarmonic Functions. 2002,2003
[21] Olivier C., Bernhard S., and Alexander Z. Semi-Supervised Learning. The MIT Press, 2006, p. 191-273.