# LogSumExp for Unlabeled Data Processing

Taocheng Hu, Jinhui Yu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, China
hutaocheng@gmail.com, jhyu@cad.zju.edu.cn

***Abstract*—There are two types of data in semi-supervised learning: feature vector with the corresponding label and feature vector without label, where labeled data processing has been well studied in supervised learning. In this paper, we derive the LogSumExp function for unlabeled data processing. This derivation establishes a unified view of labeled data processing and unlabeled data processing in semi-supervised learning. Moreover, the LogSumExp function is the kernel component of unlabeled data processing framework, which should not be restricted to the semi-supervised learning, we can easily extend the framework to the unsupervised learning situation. Interestingly, our proposed unlabeled data processing framework could also cover the k-means method.**

***Index Terms*—LogSumExp, unlabeled data processing, semi-supervised learning, k-means**

## I. Introduction

Given a training set consisting of instances from the direct product of the feature vector space and the label space $X \times Y$

$$D_{tr} = \{(\boldsymbol{x}_t, y_t) \in X \times Y : t = 1, \cdots, T\}, \tag{1}$$

the classification learning algorithm tries to find an appropriate classifier in the hypothesis space, hoping that the classifier can accurately predict a label for the unseen feature vector. This is the classical scene of the supervised learning problem, which has been well studied.

However, obtaining a feature vector with the corresponding label typically requires manual marking, special equipment, or an expensive but slow experiment, these overheads may make the complete markup training sample set infeasible. Nonetheless there are usually a large number of unlabeled sample instances, or these Marked samples are relatively easy to obtain. In this case, semi-supervised learning shows a large considerable practical value. In addition, many studies have found that a small amount of labeled feature vectors adding to the unlabeled data set, may produce unexpected improvement, in comparison with the performance obtained by supervising learning of discarding these unlabeled data, or by discarding the label data for unsupervised learning[1], [2], [3]. Therefore, semi-supervised learning also has certain theoretical value in machine learning.

Computational overhead and prediction performance are important indicators of the classification issue. Generally s-peaking, the more complex the hypothesis space is, the more accurate the prediction on the training samples is, and the more derivation the predictive accuracy on unseen data is, also the greater corresponding computational overhead is. Thus, choosing the appropriate hypothesis space is essential to the semi-supervised learning. For example, graph-based semi-supervised learning methods use graphs to represent the data, and each tag and non-tagged data instance corresponds to the node [4]. Some recent studies have took the Gaussian process or the Markov random walk [5], and Laplacian graphs are also used to solve the semi-supervised multi-classification problem [2]. Although these methods take advantage of semi-supervised and supervised relationship between sample instances, the entire solution usually requires significant computational overhead, such as most of the computational complexity of $\mathbb{O}(n^3)$, where n is sample size[6], [1].

This would become more complex when comes to multi-classification situation[7]. For example, Transductive Support Vector Machine (TSVM) is one of low-density segmentation methods, which attempt to place the boundary in areas where fewer sample instances are available, and mark unlabeled data with decision boundaries. While the SVM seeks to have the maximum interval between different classes in the supervised learning situation, the TSVM tries to mark unlabeled data with the decision edges with the largest interval between all the data. Extending TSVM to handle multiple classes of unlabeled data is given by [8]. However, since the unlabeled data and the tag data use different metrics, the corresponding objective function is biased.

Recently, a series of semi-supervised multi-class learning studies have attempted to break through these limitations. Most of these semi-supervised multi-classifications are boosting-based methods. The main difference between these methods is the loss function and regularization technology[9]. But these methods are lack of ability to use the correlations between the feature vector and the label, in particular the unlabeled data[10].

In our previous work[11], we proposed an incremental method of a Bayesian supervised learning model[12] for semi-supervised learning, which took both efficiency and the accuracy into account. Nonetheless we also found that, the incremental method depends on the semi-supervised learning environment setting: the Bayesian classifier could enter the proper state only when it is given labeled samples, and the incremental method fails when it is given completely unlabeled data. Thus, in this paper, we try to derive a framework for unlabeled data processing. The remainder of this article is organized as follows: Section II is the mathematical basis of this paper, in this part, we analyze the Fenchel conjugation

978-1-5090-5756-6/17/$31.00 ©2017 IEEE
SERA 2017, June 7-9, 2017, London, UK

and constrained optimization problem. We use these results to deduce the LogSumExp framework in section III by analyzing the structure of the Bayesian classification objective function[12], and then we instantiate the framework into a semi-supervised learning method. The empirical experiment is placed in section IV. Section V summarizes the work of this paper.

## II. Preliminaries

Our goal is to derive a framework for unlabeled data processing, before that, we need to introduce some tools for subsequent derivations.

### A. Fenchel Conjugate Relationship Measure

Given a convex function f defined on $X$, and its conjugate function $f^*$ on the dual space $X^*$, we have the following inequality [1]:

$$\langle \boldsymbol{x}, \boldsymbol{x}^* \rangle \leq f(\boldsymbol{x}) + f^*(\boldsymbol{x}^*), \quad \forall \boldsymbol{x} \in X \textit{ and } \boldsymbol{x}^* \in X^*. \tag{2}$$

This inequality is called Fenchel inequality, which could be easy to show with the definition of Fenchel conjugate function:

$$\begin{aligned} \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle &= f(\boldsymbol{x}) + \Big[ \langle \boldsymbol{x}^*, \boldsymbol{x} \rangle - f(\boldsymbol{x}) \Big] \\ &\leq f(\boldsymbol{x}) + \overbrace{\sup_{\boldsymbol{x}' \in X} \Big[ \langle \boldsymbol{x}^*, \boldsymbol{x}' \rangle - f(\boldsymbol{x}') \Big]}^{\textit{definition of } f^*} \\ &= f(\boldsymbol{x}) + f^*(\boldsymbol{x}^*). \end{aligned} \tag{3}$$

The equality $\langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = f(\boldsymbol{x}) + f(\boldsymbol{x}^*)$ holds if and only if $\boldsymbol{x}^* = \nabla_{\boldsymbol{x}} f$, while the maximum value of the variable $\boldsymbol{x}'$ in Fenchel function definition is the same as the input argument $\boldsymbol{x}$.

The Fenchel inequality gives a inner product form lower bound $\langle \boldsymbol{x}, \boldsymbol{x}^* \rangle$ of the sum of conjugate functions $f(\boldsymbol{x}) + f^*(\boldsymbol{x}^*)$. Only when there is a gradient relationship between $\boldsymbol{x}$ and $\boldsymbol{x}^*$, the sum reaches the inner product bound. Thus, we can create the following expression to measure if there is a gradient relationship between $\boldsymbol{x}$ and $\boldsymbol{x}^*$

$$Conj_{f,f^*}(\boldsymbol{x}, \boldsymbol{x}^*) = \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle - \Big( f(\boldsymbol{x}) + f^*(\boldsymbol{x}) \Big). \tag{4}$$

This expression is called the (Fenchel) conjugate relationship measure. Formally, conjugate function of f and $f^*$ is included in the conjugate relationship measure: it is the objective function for defining conjugate function $f^*$ if we omit $f(\boldsymbol{x})$; when $f^*(\boldsymbol{x})$ is omitted, it corresponds to the objective function for defining $f^*$'s conjugate function definition. So It can be used to study interactions of $\boldsymbol{x}$ and $\boldsymbol{x}^*$ when conjugate functions f and $f^*$ are given in the context of the conjugate relationship measure.

### B. Constrained optimization problem revisited

Let's look back at the Karush-Kuhn-Tucker (KKT) condition, we know that for the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) = 0, \quad i \in [I] \\ & h_j(\boldsymbol{x}) \leq 0, \quad j \in [J], \end{aligned} \tag{5}$$

the KKT condition for the minimum is:

$$\begin{aligned} & \nabla f(\boldsymbol{x}) - \sum_i \Big[ \boldsymbol{\lambda}_i \nabla g_i \Big] + \sum_j \Big[ \boldsymbol{\mu}_j \nabla h_j(\boldsymbol{X}) \Big] = 0 \\ & g_i(\boldsymbol{x}) = 0, \quad i \in [I] \\ & \boldsymbol{\mu}_i h_i(\boldsymbol{x}) = 0, \quad j \in [J], \end{aligned} \tag{6}$$

where $\boldsymbol{\lambda}_{1:I}, \boldsymbol{\mu}_{1:J}$ are new variables called the KKT multiplier, which correspond to equality constraints and inequality constraints separately. There are too many equations for the KKT, we can include them in the following equivalence optimization problem which is relatively simple

$$\arg \max_{\boldsymbol{\lambda}_{1:I}, \boldsymbol{\mu}_{1:J} \geq 0, \boldsymbol{x}} \quad L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = -f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, g(\boldsymbol{x}) \rangle + \langle \boldsymbol{\mu}, h(\boldsymbol{x}) \rangle, \tag{7}$$

where the objective function is called augmented Lagrangian function. The reason for equivalence is that variables of the optimization problem follow the law of association and exchange when the objective function is convex. Also remember that KKT multipliers are combined with constraints in terms of inner products

$$\begin{aligned} & \max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0, \boldsymbol{x}} \quad L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ = & \max_{\boldsymbol{\mu} \geq 0} \Big[ \max_{\boldsymbol{\lambda} \geq 0, \boldsymbol{x}} \quad L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \cdots + \langle \boldsymbol{\mu}, h(\boldsymbol{x}) \rangle \Big] \\ = & \max_{\boldsymbol{\lambda}} \Big[ \max_{\boldsymbol{\mu}, \boldsymbol{x}} \quad L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \cdots + \langle \boldsymbol{\lambda}, g(\boldsymbol{x}) \rangle + \cdots \Big], \end{aligned} \tag{8}$$

note that the objective function is linear w.r.t KKT multipliers, if the corresponding inequality constraint is not satisfied $h_j(\boldsymbol{x}) \geq 0$, $\boldsymbol{\mu}_j$ would be $\infty$ as $\boldsymbol{\mu}_j$ is positive, which makes the optimization problem meaningless. Similarly, if the equality constraint $g_i(\boldsymbol{x}) \neq 0$ is not satisfied, specifically, such as $g_i(\boldsymbol{x}) > 0$, $\boldsymbol{\lambda}_i$ takes value $-\infty$, causing the optimization problem to crash.

We rearrange the Lagrangian function, and found this form is consistent with the objective function used to define the Fenchel conjugate function

$$\overbrace{\left\langle \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{pmatrix}, \begin{pmatrix} g(\boldsymbol{x}) \\ h(\boldsymbol{x}) \end{pmatrix} \right\rangle}^{\text{inner product form}} - f(\boldsymbol{x}), \tag{9}$$

that is, a definition of a conjugate function with $(g, h)$ as optimization variable (vector) and $f(\boldsymbol{x}) = f(\boldsymbol{x}(g, h))$ as the primal function, while we introduce the Fenchel conjugate relationship measure to include the definition in the former part. Thus, we establish the conjugate function definition view of the constrained optimization problem, which reminds us to

[1] For the sake of convenience, the rest of the paper assumes $\boldsymbol{x} \in X$, and $\boldsymbol{x}^* \in X^*$.

think of its equivalent constrained optimization problem when comes to the Fenchel conjugate relationship measure

$$\begin{aligned} &\textit{constrained optimization problem} \\ \Longleftrightarrow &\textit{Fenchel conjugate relationship measure} \end{aligned} \tag{10}$$

## III. From labeled data processing framework$\langle Q, \log(P)\rangle$ to unlabeled data processing framework LogSumExp

In [12], we give a probabilistic graphical model for multi-classification, whose learning procedure corresponds to the following regularized optimization problem

$$\begin{aligned} \max_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}; \{\boldsymbol{x}_t, y_t\}_{t=1}^{T}) = & \overbrace{-\frac{1}{2}\|\boldsymbol{w}\|_{\mathcal{F}}^2}^{\text{regularization term}} \\ & + \sum_{t=1}^{T} \overbrace{\langle \boldsymbol{e}_{y_t}, \log\phi(\boldsymbol{w}^\top \boldsymbol{x}_t)\rangle}^{\text{data term}}, \end{aligned} \tag{11}$$

where $\boldsymbol{x}_t$ is the feature vector instance and $y_t$ is the corresponding label, $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius norm of the matrix, $\boldsymbol{\phi}(\mathbf{u} \in \mathbb{R}^k) = \frac{\exp(\mathbf{u})}{\sum_{j=1}^{k}\exp(\mathbf{u}_j)}$ denotes the probability assignment, we use the data term $\langle \boldsymbol{e}_{y_t}, \log\phi(\boldsymbol{w}^\top \boldsymbol{x}_t)\rangle$ to measure the correlation of the assignment and the given label.

We abstract the data term as $\langle Q, \log(P)\rangle$, where P and Q are probabilities. If not stated, the given probability is denoted with Q, the one corresponding to the parameter of model is denoted with P. With the convention, we know that Q is $\boldsymbol{e}_y$ and $\phi(\boldsymbol{w}^\top \boldsymbol{x})$ is P.

$$\overbrace{\Big\langle \boldsymbol{e}_y, \log\phi(\boldsymbol{w}^\top \boldsymbol{x})\Big\rangle}^{\langle Q, \log(P)\rangle}, \tag{12}$$

In the following part, we do a further analysis of the structure $\langle Q, \log(P)\rangle$

$$\Big\langle Q, \log\phi(\boldsymbol{\theta} \in \mathbb{R}^d)\Big\rangle, \tag{13}$$

the logarithm structure on the right of the inner product form could be rewritten as follows

$$\begin{aligned} \log\phi(\boldsymbol{\theta}) &= \log\Big(\frac{\exp(\boldsymbol{\theta})}{\sum_k \exp(\boldsymbol{\theta})}\Big) \\ &= \log(\exp(\boldsymbol{\theta})) - \log(\sum_k \exp(\boldsymbol{\theta})) \\ &= \boldsymbol{\theta} - \text{LogSumExp}(\boldsymbol{\theta}). \end{aligned} \tag{14}$$

This is a different structure, we substitute the difference expression into the data term and get the following equation:

$$\begin{aligned} \langle Q, \log\phi(\boldsymbol{\theta})\rangle &= \langle Q, \boldsymbol{\theta} - \text{LogSumExp}(\boldsymbol{\theta})\rangle \\ &= \langle Q, \boldsymbol{\theta}\rangle - \text{LogSumExp}(\boldsymbol{\theta}), \end{aligned} \tag{15}$$

The second step removes LogSumExp($\boldsymbol{\theta}$) from the inner product structure because $P$ is the probability that the sum of the components is equal to 1. The new data term is still a different structure, and it is easy to see that the final expression is the Fenchel conjugate relationship measure structure. According to the II-B section, the constrained optimization problem and the Fenchel conjugate relationship measure are interpreted. We know that this corresponds to the following constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \text{LogSumExp}(\boldsymbol{\theta}) \\ \textit{s.t.} \quad & \phi(\boldsymbol{\theta}) = Q, \end{aligned} \tag{16}$$

where the objective function is the LogSumExp. Generally speacking, a Bayesian network is associated with a MLE (Maximum Likelihood Estimation) form optimization problem, here we turn the above minimum optimization problem to the following maximum optimization without constraint.

$$\max_{\boldsymbol{\theta}} \quad \text{LogSumExp}(\boldsymbol{\theta}). \tag{17}$$

Let's back to the semi-supervised learning. In the above derivation, we extend the unit vector $\boldsymbol{e}_y$ to the probability simple $P$, and replace the product $\boldsymbol{w}^\top \boldsymbol{x}$ with the symbol $\boldsymbol{\theta}$ for convenience, we know that the supervised learning corresponds to an optimization problem with the LogSumExp as objective function and the label data as the constraint

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \text{LogSumExp}(\boldsymbol{w}^\top \boldsymbol{x}) \\ \textit{s.t.} \quad & \phi(\boldsymbol{w}^\top \boldsymbol{x}) = \boldsymbol{e}_y. \end{aligned} \tag{18}$$

Naturally, unlabeled data corresponds to an unconstrained problem optimization problem

$$\max_{\boldsymbol{w}} \quad \text{LogSumExp}(\boldsymbol{w}^\top \boldsymbol{x}), \tag{19}$$

Combining the above two expression together, we can obtain the following optimization problems for semi-supervised learning

$$\max_{\boldsymbol{w}} \quad \overbrace{-\frac{\lambda}{2}\|\boldsymbol{w}\|_F^2}^{\textit{regularization term}} \quad + \overbrace{\Big[\sum_{l}\langle \boldsymbol{e}_{y_l}, \log(\phi(\boldsymbol{w}\boldsymbol{x}_l))\rangle + \sum_{u}\text{LogSumExp}(\boldsymbol{w}\boldsymbol{x}_u)\Big]}^{\textit{data term}}, \tag{20}$$

where letter 'l' denotes the iteration of labeled data and 'u' denotes that of unlabeled data.

Recalling the relevant steps of the classical unlabeled data processing algorithm k-means, it selects the centroid based on the minimum of the distances between input data and the parameters, which are essentially based on the objective function

$$\text{LogSumExp}(-\frac{1}{2}\|\boldsymbol{x}_u - \boldsymbol{w}\|^2). \tag{21}$$

So our proposed LogSumExp framework can cover k-means method, the difference is that between dealing with unlabeled data in this paper is that we use the inner product connection parameter $\boldsymbol{w}$ and the data $\boldsymbol{x}_u$, while the k-means uses $L_2$ distance $-\frac{1}{2}\|\boldsymbol{x}_u - \boldsymbol{w}\|^2$ as input of the LogSumExp function. But both are semantically equivalent, as the conjugate function of $L_2$ norm is itself, and corresponding conjugate relationship measure is

$$\begin{aligned} conj_{\frac{1}{2}\|\boldsymbol{x}_u\|^2, \frac{1}{2}\|\boldsymbol{w}_k\|^2} &= \langle \boldsymbol{w}_k, \boldsymbol{x}_u \rangle - (\frac{1}{2}\|\boldsymbol{w}_k\|^2 + \frac{1}{2}\|\boldsymbol{x}_u\|^2) \\ &= -\frac{1}{2}\|\boldsymbol{x}_u - \boldsymbol{w}_k\|^2. \end{aligned} \tag{22}$$

We can see that the $L_2$ distance is identical to the Fenchel relationship measure of the $L_2$ norm.

*1) Solving the optimization problem:* We extend the Duality Aggregation algorithm [12] to solve the optimization problem (20) of the semi-supervised learning. The reason is that the data term of the objective function is approximately linear. As the data term with labeled data had proved in [12], we only need to prove the approximate linearity of the unlabeled data term: the $\boldsymbol{w}$ is a $|X| \times K$ matrix, there is no difference when processing each column vector of matrix $\boldsymbol{w}$ in the above objective function, we use the shorthand $\boldsymbol{\theta}_k = \langle \boldsymbol{w}_{\cdot,k}, \boldsymbol{x} \rangle$, the analysis of the component $\boldsymbol{\theta}_k$ represents the analysis of each element of the column vector $\boldsymbol{w}_{\cdot,k}$ because $\boldsymbol{\theta}_k$ is linear to each element of column vector $\boldsymbol{w}_{\cdot,k}$.

$$d\text{LogSumExp}(\theta) = \Big\langle \frac{\exp(\boldsymbol{\theta})}{\sum_k \exp(\boldsymbol{\theta}_k)} = \phi(\boldsymbol{\theta}), d\boldsymbol{\theta} \Big\rangle \tag{23}$$

$$d^2\text{LogSumExp}(\theta) = \Big(\delta_k^{k'} \phi(\theta)_k - \phi(\boldsymbol{\theta})_k \phi(\boldsymbol{\theta})_{k'}\Big) d\boldsymbol{\theta}_k d\boldsymbol{\theta}_{k'}. \tag{24}$$

The Hessian matrix of the LogSumExp has the following structure

$$H_{\text{LogSumExp}} = \Lambda(\phi(\boldsymbol{\theta})) - \phi(\boldsymbol{\theta})\phi(\boldsymbol{\theta})^{\mathrm{T}}. \tag{25}$$

Any $\boldsymbol{x} \in \mathbb{R}^k$ multiply by the Hessian matrix

$$\boldsymbol{\theta}^{\mathrm{T}} H_{\text{LogSumExp}} \boldsymbol{\theta} = \sum_k \left[\phi_k(\boldsymbol{\theta})\boldsymbol{\theta}_k^2 - (\phi_k(\boldsymbol{\theta}_k)\boldsymbol{\theta}_k)^2\right] \approx 0. \tag{26}$$

When $\boldsymbol{\theta}_k$ is the maximum of $\boldsymbol{\theta}$, the corresponding $\phi_k$ is very close to 1, and $\phi_k\boldsymbol{\theta}_k^2 - (\phi_k\boldsymbol{\theta}_k)^2 \approx 1 * \boldsymbol{\theta}_k^2 - (1 * \boldsymbol{\theta}_k)^2 = 0$ when $\boldsymbol{\theta}_k$ is relative small, the corresponding $\phi_k$ is approximately equal to 0, which also leads to 0: $\phi_k\boldsymbol{\theta}_k^2 - (\phi_k\boldsymbol{\theta}_k)^2 \approx 0 * \boldsymbol{\theta}_k^2 - (0 * \boldsymbol{\theta}_k)^2 = 0$. Thus we know the data term is approximately linear, and the strongly convexity is provided by the regularization term of the objective function.

In addition to the gradient calculation of unlabeled data, the most important difference is the introduction of the counter, because the trained multi-classifier predict a label based on the inner product operation while scaling column vectors $\boldsymbol{w}_{\cdot,k}$ would obviously affect the classification. We introduce an intermediate variable $\boldsymbol{cnt}$ to storage instance number of each class to reduce the impact of historical records. The details of the algorithm are shown in the algorithm 1.

**Algorithm 1** Extended Duality Aggregation Algorithm for LogSumExp based Semi-Supervised Muti-classifcation Learning

```
Input: Training set $D_{tr} = \Big\{(\boldsymbol{x}_t, y_t)\Big|(\boldsymbol{x}_t, )\Big\}_{t=1}^{T}$, learning rate $\sigma^2$
Output: A multi-classifier configured with $\boldsymbol{w}^*$
$\boldsymbol{w}_0 \leftarrow \boldsymbol{0}, \boldsymbol{cnt}_0 \leftarrow \boldsymbol{1}$
repeat
    for $t = 1$ to $T$ do
        if $\boldsymbol{x}_t$ is labeled then
            $\boldsymbol{w}_t \leftarrow \frac{1}{\sigma^2}\big[\boldsymbol{e}_{y_t} - \phi(\boldsymbol{w}_t\boldsymbol{x}_t)\big]\boldsymbol{x}_t^{\mathrm{T}}$
        else
            $\boldsymbol{q}_t = \phi(\boldsymbol{w}_{t-1}\boldsymbol{x}_t)$
            $\boldsymbol{cnt}_t \leftarrow \boldsymbol{cnt}_{t-1} + \frac{1}{\sigma^2}\boldsymbol{q}_t$
            $\boldsymbol{w}_t \leftarrow \frac{\boldsymbol{cnt}_{t-1}}{\boldsymbol{cnt}_t}\boldsymbol{w}_{t-1} + \frac{1}{\sigma^2}\big[\boldsymbol{q}_t\boldsymbol{x}_t^{\mathrm{T}}\big]$
        end if
    end for
until Convergence
$\boldsymbol{w}^* \leftarrow \boldsymbol{w}_T$
```

## IV. Experiments

This paper uses the four datasets which had been used in the context of supervised learning: MNIST, pre-processed MNIST with PCA + RBF, COIL-20 and COIL-100, to evaluate various semi-supervised learning's responses under changes of data representations and class numbers[12]. In the context of semi-supervised learning, we are required to construct the data set with both labeled data and unlabeled data for training.

The method is as follows: given a training set, we randomly select labeled data whose number is controlled by the configuration "labeled data number of each class" which follows the exponential law, while the other samples with the label discarded. The paper uses the "labeled data number of each class" as the configuration parameter to minimize the impact when the number of classes and instances of different data sets is not the same. When the training is ended, the evaluation of the semi-supervised learning is in the same way to that of the supervised learning, that is, we feed the trained model with instances of test set, the model returns $\hat{y}_t = h(\boldsymbol{x}_t)$, we compares whether $\hat{y}_t$ and the real label $y_t$ are the same. So this paper still uses $\hat{y}_t$ and the real label $y_t$ to measure the prediction accuracy $Accuracy(h; D) = \frac{\sum_t 1(\hat{y}_t = y_t)}{\|\text{Test Samples}\|}$.

This paper evaluates two indicators: prediction accuracy and time overhead of training. We do not illuminate the time overhead of prediction, because this reflects the completion of the model training, and the speed of data processing indicators changes little in the context of semi-supervised learning method.

As a comparison, the experiments not only include the incremental[11] and the LogSumExp semi-supervised learning methods, but also the k-means, which has been referred in section III, and is used to associate parameters and data for semi-supervised learning method. In addition, because the goal of semi-supervised learning is to obtain better prediction accuracy, rather than relatively good or bad, we use supervised learning as a benchmark, that is, the Bayesian classifier of [12], which uses only the portion of the labeled data for training, Mark the data to understand the performance improvement of semi-supervised learning because it only uses the labeled data part, and in this paper analyzes the linear relationship between the training time and the amount of tagged data, The number of data in the mark is very small, so the training time is the least cost. In all experiments, the corresponding algorithms perform 20 rounds on the training samples in order to achieve continuous improvement, and each experiment runs 20 times to allow the methods to be adequately trained while reducing the effect of environmental instability. In addition, for the convenience of writing, this article uses ”Incremental SSL”[2], ”LogSumExp SSL”, ”k-means SSL” and ”Bayes Classifier” to represent the above four classification learning methods.

### A. On the effect of data representations

Performances of each algorithm using different data representations of the MNIST data set are shown in Figure (1a), (1b) and table I. Although the data set pre-processed by PCA + RBF under the supervised learning scenario gives the classifier a greater improvement in prediction accuracy (see experiment of [12] for more detail), the situation is different in the context of semi-supervised learning: prediction accuracy on the raw MNIST is obvious better than on the pre-processed MNIST when given less labeled data[3], the pre-processed MNIST with PCA+RBF demonstrates its advantages only when each class is given 32 labels. Thus, we can see that the same data set has different effect for the supervised learning and the semi-supervised learning, which suggests us to select the appropriate data representation according to the application scenario.

The k-means SSL method has the best predictive accuracy when the labeled data is less, but its training cost is also the largest. That is because the k-means uses $L_2$ distance correlation input data and model parameters, the calculation is greater compared with the product of this model. The LogSumExp SSL performs poorly when the number of labeled data is low, but with the increase of labeled data, it gradually has the best predictive accuracy.

### B. On the Effect of Class Numbers

Because the COIL data set is not divided into training set and test set, we randomly select instance into training set and test set with ratio 7:3. The performance data for each algorithm in COIL-20 and COIL-100 are shown in the figure (2a), (2b) and table II. It can be seen that although this article takes the ”average number of tag data types contained in each type” as a parameter, it still can not completely shield the effect of the label space on the prediction accuracy, and the prediction accuracy in the COIL-20 data set is better than in the COIL-100 data set. Similar to the case of the MNIST data set, the LogSumExp SSL performs poorly when the number of labeled data is small, but with the increase of labeled data, it progressively outperform other semi-supervised learning methods.

## V. Conclusions

Because there are many difficulties in completely marking data in practical applications, learning from a data set with small number of labeled data has important practical and theoretical extremes.

This paper is the follow-up work of our previous work[11], which could not work when there are only unlabeled data. By analysing the objective function of Bayesian multiclassifier[12], we know it has the form $\langle Q, \log(P)\rangle$, which corresponds to the objective function used to define the Fenchel function of the LogSumExp. For this optimization problem, we find that this corresponds to the constraint function with LogSumExp as the objective function, and the term related to $Q$ corresponds to the equation constraint part. Thus, we derive the LogSumExp framework for unlabeled data processing. The Bayesian multi-classifier use the product operator connection parameter $\boldsymbol{w}$ and the input feature data $\boldsymbol{x}$, we take this for input of the LogSumExp and keep the labeled data intact, which is an instance of applying the LogSumExp framework to the semi-supervised learning; If we take the $L_2$ distance as the input of the LogSumExp, which is the classical unsupervised learning method k-means.

## References

[1] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[2] A. Goldberg, B. Recht, J. Xu, R. Nowak, and X. Zhu, “Transduction with matrix completion: Three birds with one stone,” in *Advances in neural information processing systems*, 2010, pp. 757–765.

[3] J. Zhu, E. P. Xing, and B. Zhang, “Partially observed maximum entropy discrimination markov networks.” in *NIPS*, 2008, pp. 1977–1984.

[4] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu, “Non-negative low rank and sparse graph for semi-supervised learning,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2328–2335.

[5] A. Azran, “The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks,” in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 49–56.

[6] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, “Nonnegative sparse coding for discriminative semi-supervised learning,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2849–2856.

[7] H. Valizadegan, R. Jin, and A. K. Jain, “Semi-supervised boosting for multi-class classification,” in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 522–537.

[8] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, “Distance metric learning with application to clustering with side-information,” in *Advances in neural information processing systems*, 2002, pp. 505–512.

[9] A. Saffari, C. Leistner, and H. Bischof, “Regularized multi-class semi-supervised boosting,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 967–974.

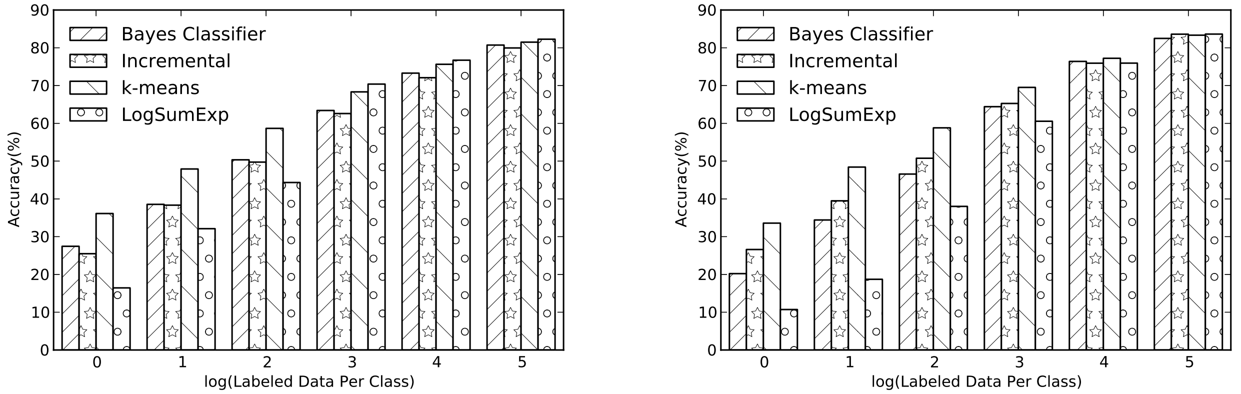
[10] B. Wang, Z. Tu, and J. K. Tsotsos, “Dynamic label propagation for semi-supervised multi-class multi-label classification,” in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 425–432.

[11] T. Hu and J. Yu, *Incremental Max-Margin Learning for Semi-Supervised Multi-Class Problem*. Cham: Springer International Publishing, 2016, pp. 31–43. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23509-7_3

[12] T. Hu and J. Yu, “Max-margin based bayesian classifier,” *Frontiers of Information Technology & Electronic Engineering*, vol. 17, pp. 973–981, 2016.

[2] SSL stands for Semi-Supervised Learning

[3] If we evaluate the accuracy with the mean value only, the Incremental SSL works better with the pre-processed MNIST. However, be aware of its variance, the pre-processed MNIST is extremely unstable when only one label is given for each class. Therefore, in terms of the effect of data representations, we should also pay attention to the variance of predictive accuracy.



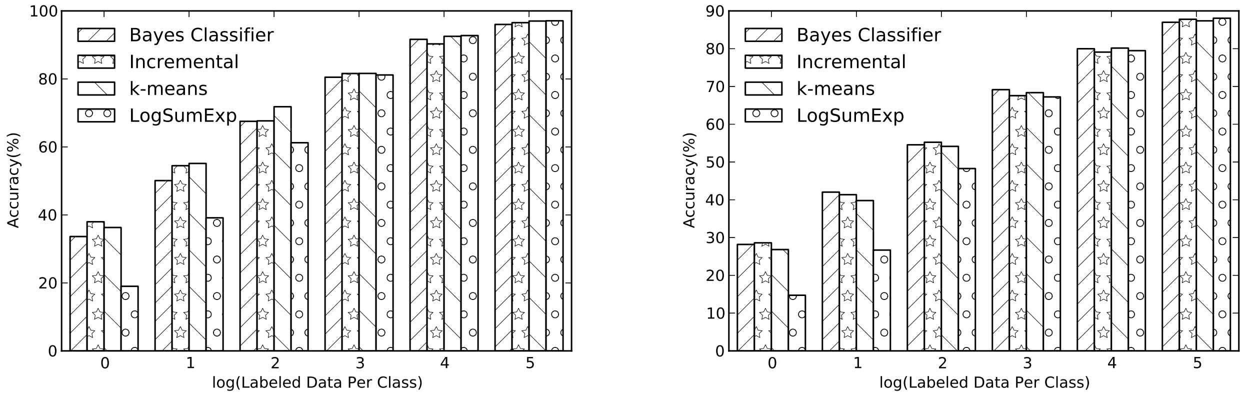
(a) MNIST(raw)

(b) MNIST(PCA-RBF)

Fig. 1: The Effect of Different Data Representation

TABLE I: Performance of various multi-classification schemes on the MNIST dataset with different data representations

| MNIST(raw) | | | | | |
|---|---|---|---|---|---|
| SSL method | | Bayes Classifier | Incremental SSL | k-means SSL | LogSumExp SSL |
| Prediction | 1 | 27.45 ± 4.02 | 25.49 ± 4.56 | **36.13 ± 8.67** | 16.44 ± 3.99 |
| Accuracy | 2 | 38.55 ± 4.71 | 38.34 ± 4.90 | **47.92 ± 6.33** | 32.12 ± 5.92 |
| with | 4 | 50.34 ± 3.76 | 49.74 ± 4.18 | **58.67 ± 4.66** | 44.34 ± 4.11 |
| Different | 8 | 63.41 ± 2.13 | 62.59 ± 2.41 | 68.33 ± 2.12 | **70.39 ± 2.61** |
| Labels | 16 | 73.29 ± 2.30 | 72.08 ± 2.88 | 75.63 ± 2.03 | **76.72 ± 1.84** |
| per Class | 32 | 80.72 ± 1.15 | 79.96 ± 1.41 | 81.50 ± 1.01 | **82.28 ± 0.84** |
| Training time(s) | | **0.02** | 3.73 | 10.29 | 4.64 |
| MNIST pre-prested with PCA-RBF | | | | | |
| SSL method | | Bayes classifier | Incremental SSL | k-means SSL | LogSumExp SSL |
| Prediction | 1 | 20.22 ± 3.45 | 26.60 ± 8.27 | **33.59 ± 8.02** | 10.71 ± 1.61 |
| Accuracy | 2 | 34.42 ± 5.56 | 39.49 ± 6.34 | **48.42 ± 6.93** | 18.72 ± 3.92 |
| with | 4 | 46.57 ± 5.52 | 50.76 ± 5.26 | **58.81 ± 7.96** | 38.02 ± 5.62 |
| Different | 8 | 64.41 ± 2.75 | 65.26 ± 3.56 | **69.52 ± 3.02** | 60.56 ± 3.74 |
| Labels | 16 | 76.39 ± 1.31 | 75.93 ± 2.15 | **77.22 ± 2.08** | 75.94 ± 2.62 |
| per Class | 32 | 82.48 ± 1.05 | 83.61 ± 0.77 | 83.34 ± 1.00 | **83.63 ± 0.65** |
| Training time(s) | | **0.04** | 15.82 | 29.00 | 17.15 |



(a) COIL-20

(b) COIL-100

Fig. 2: The effect of Different Class Numbers

TABLE II: Performance of various multi-classifcation schemes on the COIL dataset with different class numbers

| COIL-20 | | | | | |
|---|---|---|---|---|---|
| SSL Method | | Bayes Classifier | Incremental SSL | k-means SSL | LogSumExp SSL |
| Prediction | 1 | 33.64 ± 2.89 | **37.98 ± 6.12** | 36.30 ± 6.72 | 19.00 ± 3.99 |
| Accuracy | 2 | 50.09 ± 4.98 | 54.48 ± 5.27 | **55.14 ± 3.87** | 39.14 ± 5.07 |
| with | 4 | 67.52 ± 5.43 | 67.66 ± 4.99 | **71.82 ± 3.00** | 61.23 ± 6.23 |
| Different | 8 | 80.50 ± 3.58 | 81.59 ± 3.69 | **81.61 ± 2.93** | 81.16 ± 3.22 |
| Labels | 16 | 91.64 ± 2.31 | 90.30 ± 1.80 | 92.52 ± 1.88 | **92.75 ± 1.49** |
| per Class | 32 | 96.02 ± 1.56 | 96.55 ± 1.22 | 97.02 ± 1.00 | **97.09 ± 1.05** |
| Training time(s) | | **0.12** | 0.57 | 1.04 | 0.59 |
| COIL-100 | | | | | |
| SSL method | | Bayes classifier | Incremental SSL | k-means SSL | LogSumExp SSL |
| Prediction | 1 | 28.18 ± 1.96 | **28.60 ± 1.42** | 26.83 ± 1.28 | 14.74 ± 0.92 |
| Accuracy | 2 | **42.03 ± 1.20** | 41.35 ± 0.91 | 39.80 ± 2.12 | 26.69 ± 2.09 |
| with | 4 | 54.55 ± 1.43 | **55.24 ± 1.44** | 54.14 ± 1.57 | 48.27 ± 1.09 |
| Different | 8 | **69.16 ± 1.23** | 67.57 ± 1.37 | 68.36 ± 1.18 | 67.23 ± 2.17 |
| Labels | 16 | 79.99 ± 1.34 | 79.10 ± 0.69 | **80.16 ± 1.07** | 79.48 ± 1.19 |
| per Class | 32 | 86.99 ± 0.73 | 87.79 ± 0.99 | 87.37 ± 0.67 | **88.07 ± 1.08** |
| Training time(s) | | **2.42** | 11.62 | 25.87 | 11.51 |