# Research Proposal

Ye Tian

June 2022

**Title:** Semi-superised Classification

**Author:** YE TIAN

**Supervisor:** Professor Zhiqiang Tan

**Degree:** PhD in Statistics

## Background

Let $\mathcal{X} = \mathbb{R}^d$ for certain $d \in \mathbb{N}_+$ denotes the feature space, and $\mathcal{Y} = \{0, \cdots, M\} \subset \mathbb{N}$ for some $M \geq 1$ denotes the label space. As a "standard" setup of semi-supervised learning (SSL) classification tasks, suppose $\mathcal{X} \times \mathcal{Y} \ni (x_i, y_i) \overset{i.i.d}{\sim} P(x, y), i \in [n]$, we have a training dataset $\mathcal{S} = \mathcal{S}^\ell \cup \mathcal{S}^u$, where $\mathcal{S}^\ell = \{(x_i, y_i) : i \in [l]\}$ denotes the labelled dataset and $\mathcal{S}^u = \{x_i : i = l+1, \cdots, n\}$ denotes the unlabeled dataset. A semi-supervised algorithms tries to learn a map $F_\mathcal{S} : \mathcal{X} \to \mathcal{Y}$ based on the training set. The performance of $F_\mathcal{S}$ is typically evaluated by the zero-one risk on test data,

$$E\{L^{zo}(y_0, F_\mathcal{S}(x_0))\}, \tag{1}$$

where $(x_0, y_0) \sim P(x, y)$, independent of training data, and $L^{zo}(y, F_\mathcal{S}(x))$ is the zero-one loss, defined as 0 if the $y$th component of $F_\mathcal{S}(x)$ is a maximum and 1 otherwise. The goal of an SSL algorithm is to minimize (1). Apparently, a bottom line of SSL methods is the corresponding supervised models, people try to develop SSL algorithms to exploit data structures by using unlabeled data to alleviate the need for labeled data.

There is another learning task which is closely related to SSL, named transductive learning, which we need to discriminate. Basically, if we replace $\mathcal{S}^u$ with $\mathcal{S}^t = \{x_i | x_i \text{ in testing set}\}$, then we transfer to a transductive learning task. In literatures, those two tasks are often analyzed together, but we will focus on inductive SSL in this research.

In previous literatures, before the recent boom of deep neural network techniques, researchers didn't focus much on how to generate the feature space $\mathcal{X}$, ideally, we can suppose that we have a pre-defined feature space where the data are linearly

separable, which can come from kernel methods, manually created features based on certian prior knowledge and even raw data.

In recent years, modern large, deep neural network (DNN) models achieve human- or super-human-level performances on various kinds of supervised classification tasks, successes of which depend greatly on representational capacity of deep neural networks and the existence of large labeled datasets.

However, such large labeled datasets are not always available. On the one hand, collecting labeled data is expensive for many learning tasks because it necessarily involves expert knowledge. This is perhaps best illustrated by NLP tasks where treebanks are fruits of a time-consuming analysis that draws from multiple linguists. On the other hand, data labels may contain private information so that we do not have access to.

In comparison, in many tasks, large amounts of unlabeled data are ready to be exploited, which encourages the deep learning community to look into SSL. This has led to a plethora of SSL methods that are designed for deep neural networks. Some recent results have shown that in certain cases, SSL approaches the performance of purely supervised learning, even when only few labeled data are provided, especially on standard image datasets. However, compared with huge success in application, many current SSL methods are still ad hoc, brute-force, or lack of statistical justifications, which makes it uncertain that apart from those standard datasets, whether the performance of these methods can still be good enough on real-world data.

The goal of our research is to find an simple SSL methods with good statistical properties and application performances. Currently, we will focus on image classification tasks.

# Literature Review

There is a rich literature on SSL, we will only focus on those closely related to this research. At first, we will introduce some classical methods to provide an idea of how SSL methods leverage unlabeled data. Then we will introduce some modern deep semi-supervised learning methods to reveal how researchers combine DNN and SSL to achieve state-of-art performances. Furthermore, we want to find out flaws of current methods through this review, so that by modifying current methods we can develop better SSL methods.

There are two main paradigms of SSL methods, generative and discriminative. Generative methods assume there is a generative model $P(x, y) = P(y)P(x|y)$, where $P(x|y)$ is an identifiable mixture distribution, with large amount of unlabeled data, mixture components can be identified, then, we only need one labeled data per class to determine the mixture distribution. On the contrary, discriminative methods work on $P(y|x)$ directly, which makes discriminative methods more efficient than Generative methods in general. However, if $P(y|x)$ and $P(x)$ do not share parameters, this would leave $P(x)$ outside of the estimating procedure, which brings extra danger, since $P(x)$ is almost all we can get from unlabeled data and it is believed that SSL would not help if $P(y|x)$ and $P(x)$ do not share parameters. In this research,

we will focus on discriminative models.

For discriminative models, we need some extra assumptions on the distribution of the data to characterized the connection between $P(y|x)$ and $P(x)$. A classical assumption is semi-supervised smoothness assumption, which reduces to cluster assumption in the classification case, which can the equivalently formulated as low-density separation assumption: The decision boundary should lie in a low-density region. The above assumption is often encoded into a regularization term, adding to the loss on labeled data, where unlabeled data play a role, formulating the following kind of loss functions:

$$L = \sum_{i=1}^{l} L_\ell(y_i, F(x_i)) + \lambda R(\mathcal{S}^u), \tag{2}$$

where $\lambda$ is a tuning parameter.

Entropy Regularization is a typical example of such kind of methods. In the MAP framework, labeled part becomes cross-entropy loss, and assumptions can be encoded by means of a prior on model parameters. In [5], by putting a prior prefers minimal overlapping, the regularization term becomes label entropy on unlabeled data. The whole loss is equivalent to the following:

$$L = \frac{1}{l}\sum_{i=1}^{l} \ln P(y_i|x_i;\theta) + \lambda \frac{1}{n-l} \sum_{i=1+1}^{n} \sum_{k=1}^{M} P(k|x_i;\theta) \ln P(k|x_i;\theta). \tag{3}$$

It is not necessarily that only unlabeled data play a role in the regularization term, labeled data can also be included, then loss (2) can be extended to

$$L = \sum_{i=1}^{l} L_\ell(y_i, F(x_i)) + \lambda R(\mathcal{S}). \tag{4}$$

The above framework includes a large sub-class of semi-supervised graph models. Semi-supervised graph models rely on the geometry of the data induced by both labeled and unlabeled examples to improve performances of supervised methods that use only labeled data. The geometry can be represented by an empirical graph $g = (V, E)$ where nodes $V = \{1, \cdots, n\}$ represent the training data and edges $E$ represent similarities among them. These similarities are given by a weight matrix $\mathbf{W}$. In general, we assume $\mathbf{W}_{i,j}$ is given by a symmetric non-negative function $\mathbf{W}_X$ (possibly dependent on the data set $X = (x_1, \cdots, x_n)$) by $\mathbf{W}_{i,j} = \mathbf{W}_X(x_i, x_j) \geq 0$. Let $\mathbf{D}$ denote the diagonal degree matrix:

$$D_{i,j} = \begin{cases} \sum_j \mathbf{W}_{i,j} & i = j, \\ 0 & i \neq j. \end{cases} \tag{5}$$

and $\hat{Y} = (\hat{Y}_l, \hat{Y}_u)$ an estimated labeling of $X$, [1] showed that, in the binary case, under certain conditions, Label Propagation, Markov Random Walks and Electric Networks methods are equivalent to minimize the following loss

$$C(\hat{Y}) = \|\hat{Y}_l - Y_l\|^2 + \mu \hat{Y}^\top L \hat{Y} + \mu\epsilon\|\hat{Y}\|^2, \tag{6}$$

where $L = \mathbf{D} - \mathbf{W}$ is the un-normalized graph Laplacian, $Y_l$ is the true labels of labeled data, $\mu$ and $\epsilon$ are both tuning parameters. If we ignore the last term, which is added to prevent degenerate situations, the first term in (6) corresponds to labeled loss, while the second term including both labeled and unlabeled data, corresponds to penalization on rapid changes in $\hat{Y}$ between points that are close (as given by the predefined similarity matrix $\mathbf{W}$), where the low-density assumption is coded in.

An assumption which is different from but closely related to low-density assumption that forms the basis of several semi-supervised learning methods is the manifold assumption:

The (high-dimensional) data lie (roughly) on a low-dimensional manifold.

[17] developed a framework to introduce manifold regularization into labeled loss functions. Basically, they proposed to minimize the following loss function:

$$\begin{aligned} \min L = \sum_{i=1}^{l} L_\ell(y_i, F(x_i)) + \gamma_A \|F\|_K + \gamma_I \|F\|_I, \\ s.t. \quad F \in \mathcal{H}_K, \end{aligned} \tag{7}$$

where $K : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ is a Mercer kernel, $\mathcal{H}_K$ denotes the corresponding RKHS, $\|\cdot\|_K$ denotes the corresponding norm, $\|\cdot\|_I$ is an appropriate penalty term that reflects the intrinsic structure of $P(x)$. $\gamma_A$ and $\gamma_I$ are tunning parameters.

When $P(x)$ is unknown, for certain choice of $\|\cdot\|_I$ and after necessary approximations, optimization problem (7) can be reduced to

$$\begin{aligned} \min C = \sum_{i=1}^{l} L_\ell(y_i, F(x_i)) + \gamma_A \|F\|_K + \gamma_I \mathbf{F}^\top L \mathbf{F}, \\ s.t. \quad F \in \mathcal{H}_K, \end{aligned} \tag{8}$$

where $\mathbf{F} = (F(x_1, \cdots, F(x_n)^\top)$ and $L$ again, the un-normalized graph Laplacian. Furthermore, the solution of (8) $F^\star$ admits the form

$$F^\star = \sum_{i=1}^{n} \alpha_i K(x_i, x). \tag{9}$$

Graph methods is naturally transductive, there is a lot of works focusing on these models like graph convolutional networks and so on, which we would not introduce here. Two mainstreams of techniques proposed in SSL papers from the computer vision community are consistency regularization and Pseudo-labeling. Almost all of them use loss functions of the form (2) or (4).

Consistency regularization seeks to implement the low-density assumption by encouraging the model $F$ to be invariant to perturbations $\delta$ to the data $x$, so that the decision boundaries can be pushed to low-density regions. Mathematically, given some data perturbing function $\mu : \mathcal{X} \to \mathcal{X}$ , such that $u(x) = x + \delta$, consistency based approaches seek to minimize the regularization term in the general form of

$$\sum_i d \circ F(u(x_i), x_i), \quad i \in [n] \text{ or } [n] \setminus [l]. \tag{10}$$

where $d \circ F$ is a non-negative function measure the difference at certain output level(logit, probability, or label) of model $F$. A large number of papers have applied this idea to SSL including the $\Pi-$Model and temporal ensembling [6], Virtual Adversarial Training (VAT) [8], Mean Teacher [14], the Interpolation Consistency Training (ICT) [16], MixMatch [3] and RemixMatch [2], etc. Differences among those methods come from three aspects: how to define the perturbation function $u$, what $d \circ F$ use and how to compose $u$ with $F$. For example, Virtual Adversarial Training [8] uses adversarial training to define $u$ pointwisely; $\Pi-$Model measures the difference of logits using MSE. Mean Teacher [15] applies a perturbation to the model itself, and replaces $F(u(x))$ with an exponential moving average of the model $F_{EMA}(x)$. Except for differences in the regularization part, almost all of them choose cross-entropy loss as the labeled loss.

Another family of methods, termed pseudo-label approaches, focus on estimating labels for the unlabelled data points and then using them in a (modified) labeled loss function. The loss functions have the following formula:

$$L = \sum_{i=1}^{l} L_\ell(y_i, F(x_i)) + \lambda \sum_{i=l+1}^{n} L'_\ell(\hat{y}_i, F(x_i)). \tag{11}$$

In general, $L_l$ is cross-entropy loss. Differences among those methods come from different ways of creating pseudo labels and choices of $L'_\ell$. The first application of this idea to the deep learning setting was presented by Lee [7]. Treating the output of the neural network $F(x)$ as a discrete probability distribution, Lee assigned a hard pseudo-label $\hat{y}_i = \arg\max F(x_i)$ for each unlabelled data point and choose $L'_l$ to cross-entropy loss. [11] also uses cross-entropy loss and neural network output $F(x)$, but only use unlabled data with $\max F(x_i) > \tau$, a pre-defined threshold. Except directly use neural network outputs as pseudo-labels, [10] only uses neural network to extract features,m while use graph models to propagate labels. For this kind of method, $L'_\ell$ can be treated as entropy regularization term in (3) while using degenerated distribution $P(\hat{y}_i|x_i;\theta) = 1$.

Entropy regularization and pseudo-labeling can also be combined, [12] replaces $L'_\ell(\hat{y}_i, F(x_i))$ in (11) with $L'_\ell(\hat{y}_i, F(u(x_i)))$.

From the previous part, we can see that, in most of those methods, unlabeled data only play a role in the regularization term, encoding prior knowledge of low-density assumption, but not included in the probability system.

Besides, when combining with deep neural network methods, current loss functions do not escape from the classical framework, although they share the same formula, except for encoding the low-density assumption, we conjecture that the regularization term in deep SSL methods has an extra role, guide deep neural networks to create a proper feature space.

A classifier $F$ can be viewed as the composition of two functions $Z$ and $G$ such that $F(x) = G \circ Z(x)$. $Z : \mathcal{X} \to \mathbb{R}^{d_p}$ is the embedding function mapping original data to some $d_p-$dimensional feature space and $G : \mathbb{R}^{d_p} \to \mathcal{Y}$ projects from the feature space to the label space. Deep SSL methods usually combine with large base models, thus, $Z(x)$ usually has a large number of parameters. For old works on SVHN and CIFAR-10 datasets, the community usually uses the "13-CNN" architecture [15].

For current state-of-art models, WideResNet (WRN) 28-2 [9] and WRN-28-8 [18] is often used on CIFAR-10 and CIFAR-100. For such models, even in purely supervised case, we need to take care of overfitting problem, while in SSL case, only several, tens of at most hundreds of labeled data are provided. Without the regularization term, deep neural networks easily fall into overfitting. However, in most previous methods, a whole regularization term that plays dual roles is used, sometimes make things blurred.

# Method

By analyzing previous literatures, we see that no matter classical SSL methods or current Deep SSL methods, most of them treat unlabeled data to be a regularization to labeled model, which seems to be ad hoc. Alternatively, we can also "augment" labeled data, treat them to be an extra, isolated class of data, so that we can include unlabeled data into our probability system, which is different from classical SSL methods. Besides, from previous anaylsis, we also see that, to combine with neural network techniques, to avoid overfitting, we need extra mechanisms to regularize the neural nets to project the data to a reasonable feature space, where both labeled and unlabeled data should also play a role. Our final loss would have the following form:

$$L = C(\mathcal{S}, G) + \lambda R(\mathcal{S}, Z), \tag{12}$$

where the second term comes from the regularization of the neural nets, and the first term comes from the classification error on the feature space. The first part is what we have been focusing on until now. This part is what we have been focusing on, until now.

We first consider the binary classification problem, i.e., $\mathcal{Y} = \{0, 1\}$.

Our idea is to drop one labeled class each time and do binary classification among labeled and unlabeled data and creating $m + 1$ branches of binary loss functions, and then combine them together.

Suppose data follow exponential tilt model:
$P(x|y = 0) \sim G_0(x)$ and $P(x|y = 1) \sim G_1(x)$, $G_0(x)$ and $G_1(x)$ satisfy the following condition

$$\frac{dG_1(x)}{dG_0(x)} = \exp\left(\phi_0 + x_i^\top \phi_x\right). \tag{13}$$

Suppose class proportions in labelled, unlabelled data and population are all $\lambda$, then, the exponential tilt model is equivalent to the logistic regression model:

$$P(y = 1|x) = \frac{1}{1 + \exp\left(-\phi_0^c - x_i^\top \phi_x\right)}, \tag{14}$$

where $\phi_0^c = \log \frac{\lambda}{1-\lambda} + \phi_0$. Therefore, after estimating the parameter, we can (14) to make prediction directly.

Denote $R_0 = \{x_i \in \mathcal{S}^\ell | y_i = 0\}$, $R_1 = \{x_i \in \mathcal{S}^\ell | y_i = 1\}$ and $R_2 = \mathcal{S}^u$.

Suppose we only have $R_0 \cup R_2$, then we define the conditional probability

$$\begin{aligned} P_{0,2} &= P(x \in R_0 | x \in R_0 \cup R_2) \\ &= \frac{(1-\lambda) dG_0(x)}{(1-\lambda) dG_0(x) + \delta(\lambda dG_1(x) + (1-\lambda) dG_0(x))} \\ &= \frac{1}{1 + \delta + \delta \frac{\lambda}{1-\lambda} \exp\left(\phi_0 + x_i^\top \phi_x\right)} \\ &= \frac{1}{1 + \delta + \delta \exp\left(\phi_0^c + x_i^\top \phi_x\right)}. \end{aligned} \tag{15}$$

where $\delta$ characterizes the weight of unlabled data from prior knowledge.

For a convex function $f$, consider the discrimination (negative) loss[13]:

$$L_f(y, D_\phi) = y f^{'}(U_\phi(x)) - (1-y)\{U_\phi(x) f^{'}(U_\phi(x)) - f^{'}(U_\phi(x))\}, \tag{16}$$

where $f^{'}$ is the derivative of $f$, $\phi = (\phi_0, \phi_x^\top)^\top$, and $U_\phi(x) = \frac{P_\phi(y=1|x)}{P_\phi(y=0|x)}$.
Substitute our conditional odds $\tilde{U}_\phi(x) = \frac{P_{0,2}}{1-P_{0,2}}$ for $U_\phi(x)$, we can get a family of loss functions, the empirical formula is as the following:

$$L_{f,0}(\phi) = -\frac{1}{l} \sum_{i=1}^{l} (1-y_i) f^{'}(U_\phi(x_i)) + \frac{\delta}{n-l} \sum_{i=l+1}^{n} \{\tilde{U}_\phi(x_i) f^{'}(\tilde{U}_\phi(x_i)) - f^{'}(\tilde{U}_\phi(x_i))\}, \tag{17}$$

Suppose we have only $R_1 \cup R_2$, by similar trick, we have corresponding $P_{1,2}$ and $L_{f,1}(\phi)$. We combine 2 branches of losses simply by adding them together and get our loss function:

$$L^s(\phi) = L_{f,0}(\phi) + L_{f,1}(\phi). \tag{18}$$

Our loss function (18) has main properties that are summarized in the following proposition and theorem.

**Proposition 1** *Suppose the supervised loss defined by Equation* (16) *is Fisher consistent, the semi-supervised loss defined by* (18) *is Fisher consistent.*

Denote parameter $\phi \in \mathbb{R}^{n+1} = (\phi_b, (\phi_w)^\top)^\top$, where $\phi_b \in \mathbb{R}$ is the bias term, $\phi_w \in \mathbb{R}^n$ are weight coefficients.
Define $f^{\#}(t) = t f^{'}(t) - f(t)$,
and

$$\hat{y}^\phi(x) = \begin{cases} 2(y(x) - 1), \text{ if } x \in X^l \\ sign(x^\top \phi), \text{ if } x \in X^u. \end{cases}$$

We use abbreviation $\hat{y}_i^\phi$ for $\hat{y}^\phi(x_i)$. Given a dataset, define

$$\Phi_0 = \{\phi \in \mathbb{R}^{n+1} | \hat{y}_i^\phi x_i^\top \phi > 0, \ \forall i \in [n], \ \|\phi^x\| = 1\},$$

$$\Phi^\dagger = \{\phi \in \mathbb{R}^{n+1} | \phi = \arg\max_{\phi \in \Phi_0} \min_{i \in [n]} \hat{y}_i^\phi x_i^\top \phi\},$$

$$\hat{\Phi}(\alpha) = \{\phi | \phi = \arg \min_{\|\phi_w\| \leq \alpha} L_{\text{mix}}(\phi)\},$$

$\forall \phi \in \hat{\Phi}(\alpha)$ is denoted as $\hat{\phi}(\alpha)$.Denote $\min_{i\in[n]} \hat{y}_i^{\phi} x_i^{\top}\phi$ as $m(\phi)$.
Assume the data is separable, i.e., $\exists \phi$ s.t. for $\forall i \in [n]$, $\hat{y}_i^{\phi} x_i^{\top}\phi > 0$. Additionally, assume that $\exists\ M > 0$, such that, for all such $\phi$, $\max_{i\in[l]} y_i x_i^{\top}\phi \leq M\|\phi_w\|$.
Suppose that $f : \mathbb{R}_{++} \to \mathbb{R}$ is convex and differentiable with $f(1) = 0$ and satisfies the following conditions:

1. $f(t)$ twice differentiable and $f^{''}(t) > 0$ fot $t \in (0, 1]$.
2. $\lim_{t\to 0^+} f^{\#}(t) = c,\ c \in \mathbb{R}$.
3. For $\forall \gamma \in (0, \frac{1}{2}]$, $t \in (0, 1]$, $f^{\#}(\gamma t) + f^{\#}((1-\gamma)t) > f^{\#}(t) + c$.
4. (1) Condition for maximum margin among unlabelled data.
   (i) For $\forall \phi$, separating candidate, for $\forall x_i,\ i \in [l]$, $\exists x_j,\ j \in [l+1, n]$, s. t., $|x_j^{\top}\phi| \leq |x_i^{\top}\phi|$ (Strictly contained condition.)
   (ii) i) $\lim_{t\to o^+} \frac{f(t)-c}{t} = \infty$.
   ii) $f^{\#}(t) > c$, for $\forall t \in \mathbb{R}_{++}$.
   iii) If $\lim_{t\to 0} \frac{h(t)}{g(t)} = c_1 \neq 0$, then $\lim_{t\to 0} \frac{(f^{\#})'(h(t))}{(f^{\#})'(g(t))} = c_2 \neq 0$. If $\lim_{t\to 0} \frac{h(t)}{g(t)} = 0$, then $\lim_{t\to 0} \frac{f^{\#}(h(t))-c}{f^{\#}(g(t))-c} = 0$.
   (2) Condition for maximum margin attains among labelled data and unlabelled data.
   $(f^{\#})'(0) = s > tf^{''}(t),$ for $\forall t \in (0, 1]$.

**Theorem 1** *Under previous conditions 1.-4., $L^s(\phi)$ defined by* (18) *achieves maximum margin property in the sense that if $\exists\{\hat{\phi}(\alpha)\}_{\alpha\in\mathbb{R}_{++}}$, for $\forall \alpha \in \mathbb{R}_{++}$, $\hat{\phi}(\alpha) \in \hat{\Phi}(\alpha)$, and $\frac{\hat{\phi}(\alpha)}{\|\hat{\phi}_w(\alpha)\|} \to \phi^{\star}$, then*

$$\phi^{\star} \in \Phi^{\dagger}. \tag{19}$$

Except for previous properties, we figure out the sub-family of the binary classification loss function family, Beta Family, defined in [4], which can achieve the maximum margin property.

As an alternative of summation, we can also minimize the maximum of our 2 branches of losses, which leads to the following loss:

$$L^m(\phi) = \max\{L_{f,0}(\phi), L_{f,1}(\phi)\}. \tag{20}$$

Loss defined in (20) can also achieve the maximum margin property defined in **Theorem 1**.

Equipped with the maximum margin property, if our dataset satisfies the low-density assumption, our method would push the decision boundary to the separation region.

Fix $f(u) = u\log(u) - (1+u)\log(1+u)$, which corresponds to cross entropy loss and achieve maximum margin property among labeled and unlabeled data, we extend

our loss function into the multi-class classification task, i.e., $\mathcal{Y} = \{0, \cdots, M\}$, where $M > 1$.

From the generative modeling perspective, the training data $\mathcal{S}$ can be viewed as a pool of $M+1$ samples $\mathcal{S}^M \cup \{\mathcal{S}_j\}_{j=0}^{M-1}$ where

$\mathcal{S}_j = \{(x_i, y_i) : y_i = j, i = 1, \cdots, l\}$ drawn from $P_m(x) = P(x|y=j), 0 \le j \le M-1$,
$\mathcal{S}_M = \{x_i :, i = l+1, \cdots, n\}$ drawn from $P_M(x) = P^u(x)$.

Also let $\mathcal{S}^\ell = \{\mathcal{S}_j\}_{j=0}^{M-1}$ and $\mathcal{S}_u = \mathcal{S}^M$.

An exponential tilt mixture model for the $M+1$ samples $\{\mathcal{S}_j\}_{j=0}^{M}$ postulates a mixture model assumption that

$$\begin{aligned} dP_j(x) &= dG_J(x), 0 \le j \le M-1, \\ dP_M(x) &= \lambda_0 dG_0(x) + \lambda_1 dG_1(x) + \cdots, \lambda_{M-1} dG_{M-1}(x), \end{aligned} \tag{21}$$

where $dG_j(x)$ represents the conditional density $P(x|y=j)$ and $\lambda_j = P^u(y=j)$ is the unknown proportion of $y=j$ underlying the unlabeled data. In addition, these component conditional distributions $dG_j(x)$ satisfy

$$\frac{dG_j(x)}{dG_0(x)} = \exp(\phi_{b,j} + \phi_{w,j}^\top x), \quad 0 \le j \le M-1, \tag{22}$$

where $dG_0$ is an unknown baseline distribution, $\phi_0 = (\phi_{b,0} + \phi_{w,0}) = 0$ and $\{\phi_j = (\phi_{b,j} + \phi_{w,j})\}_{j=1}^{M}$ are unknown parameters satisfying $\phi_{w,j} = -\log\{\int \exp \phi_{w,j}^\top x\} dG_0(x)$ for $1 \le j \le M-1$. Let $\phi$ denote $\{\phi_j\}_{j=0}^{M-1}$ For a branch $s \in \{0, 1, \cdots, M-1\}$, suppose we have training data $\mathcal{S}_{-s} = \mathcal{S} \setminus \mathcal{S}_s$, the corresponding loss

$$\begin{aligned} L_{-s}(\phi) = -\frac{1}{l} \sum_{i=1}^{l} \sum_{j \ne s} y_{i,j} \log \frac{\exp(x_i^\top \phi_j)}{\sum_{k=1}^{M}(1+\delta)\exp(x_i^\top \phi_k) - \exp(x_i^\top \phi_s)} \\ - \frac{1}{n-l} \sum_{i=1+1}^{n} \log \frac{\sum_{k=1}^{M} \delta \exp(x_i^\top \phi_k)}{\sum_{k=1}^{M}(1+\delta)\exp(x_i^\top \phi_k) - \exp(x_i^\top \phi_s)} \end{aligned}, \tag{23}$$

then, we can define total losses as in the binary case by

$$\begin{aligned} L^s(\phi) &= \sum_{s=0}^{M} L_{-s}(\phi) \\ &= -\sum_{s=0}^{M} \Big( \frac{1}{l} \sum_{i=1}^{l} \sum_{j \ne s} y_{i,j} \log \frac{\exp(x_i^\top \phi_j)}{\sum_{k=1}^{M}(1+\delta)\exp(x_i^\top \phi_k) - \exp(x_i^\top \phi_s)}, \\ &\quad + \frac{1}{n-l} \sum_{i=1+1}^{n} \log \frac{\sum_{k=1}^{M} \delta \exp(x_i^\top \phi_k)}{\sum_{k=1}^{M}(1+\delta)\exp(x_i^\top \phi_k) - \exp(x_i^\top \phi_s)} \Big) \end{aligned} \tag{24}$$

and

$$L^m(\phi) = \max_{s \in \{0,1,\cdots,M-1\}} L_{-s}(\phi) \quad , \tag{25}$$

where $y_{i,j} = I_{y_i = j}$.

Similar to the binary case, $L^s(\phi)$ defined in (24) has following properties.

**Proposition 2** *The semi-supervised loss defined by* (24) *is Fisher consistent.*

Besides, given a dataset $\mathcal{S}$ classifier $\phi$, and $x$, let the corresponding $\hat{y} = \arg\max_{j\in\{0,\cdots,M-1\}} x^\top \phi_j$ and $\tilde{y} = \arg\max_{j\in\{0,\cdots,M-1\}\setminus\{\hat{y}\}} x^\top \phi_j$, define

$$\beta_\phi(x) = \begin{cases} I_{(y=\hat{y})} \cdot (x^\top(\phi_{\hat{y}} - \phi_{\tilde{y}})), \text{ if } (x,y) \in \mathcal{S}^\ell \\ (x^\top(\phi_{\hat{y}} - \phi_{\tilde{y}})), \text{ if } x \in \mathcal{S}^u. \end{cases} \tag{26}$$

define multi-class margin in the following way,

$$m(\phi) = \min_{x\in\mathcal{S}} \beta_\phi(x). \tag{27}$$

Besides, define

$$\Phi_0 = \{\phi | m(\phi) > 0,\ \|\phi_{w,j}\| = 1, j \in [M-1]\},$$
$$\Phi^\dagger = \{\phi | \phi = \arg\max_{\phi\in\Phi_0} m(\phi)\},$$
$$\hat{\Phi}(\alpha) = \{\phi | \phi = \arg \min_{\|\phi_{w,j}\|\le\alpha,\, j\in[M-1]} L^s(\phi)\},$$

$\forall \phi \in \hat{\Phi}(\alpha)$ is denoted as $\hat{\phi}(\alpha)$. Suppose $\exists N > 0$, such that $\max_{\phi\in\Phi_0} \max_{x\in\mathcal{S}} \beta_\phi(x) < N$, we have the following theorem

**Theorem 2** *$L^s(\phi)$ defined by* (24) *achieves maximum margin property in the sense that if $\exists\{\hat{\phi}(\alpha)\}_{\alpha\in\mathbb{R}++}$, for $\forall\alpha \in \mathbb{R}_{++}$, $\hat{\phi}(\alpha) \in \hat{\Phi}(\alpha)$, and $\frac{\hat{\phi}(\alpha)}{\|\hat{\phi}_w(\alpha)\|} \to \phi^\star$, then*

$$\phi^\star \in \Phi^\dagger, \tag{28}$$

where $\frac{\phi}{\|\phi_w\|} = \{\frac{\phi_j}{\|\phi_{w,j}\|}\}_{j=1}^{M-1}$.

# Regularization

The problem of previous introduced loss function is that, without extra regularization term, there is no guarantee that the generated feature space by deep neural nets are smooth enough so that the data are linearly separable. Here we introduce our regularization term, the idea is that when we change the parametrization of the embedding function $Z$, we want it to change smoothly. Based on this idea, we introduce the following objective function:

$$L_s(x_\star, \theta) := D(Z_{\hat{\theta}}(x_\star), Z_\theta(x_\star + r_v)), \tag{29}$$

$$r_v := \arg\max_{r:\|r\|_2\le\epsilon} D(Z_{\hat{\theta}}(x_\star), Z_\theta(x_\star + r)), \tag{30}$$

where $D$ is a distance function in the feature space, here we choose Euclidean distance.

For simplicity, we denote $D(Z_{\hat{\theta}}(x_\star), Z_\theta(x_\star + r))$ by $D(r, x_\star, \theta)$. We assume $Z_\theta(x_\star + r)$ is twice differentiable w.r.t $\theta$ and $x$ almost everywhere. Since $D(r, x_\star, \hat{\theta})$ takes the minimal value equals to 0 at $r = 0$, we have $nabla_r D(r, x, \hat{\theta})|_{r=0} = 0$.

# Future work

First, the property of binary loss function have been relatively well studied, we need to take more care about the difference between binary case and multi-class extension. For example, low-density assumption becomes tricky in the multi-class case, since decision boundary passing through low-density region between two classes can passing through high-density region of the third party. Even the concept of linearly separability is more complicated than binary case. So, we need to look into more about the property of multi-class extension. Moreover, we need to look into more about what kind of property in binary classes can be extended to multi-class case and what cannot. For example, we haven't prove the maximum margin property of $L^m(\phi)$ defined in (24). Besides, we can look into more general problem setup. For example, we only proved maximum margin property in the case where proportions of different classes of labeled and unlabeled data are the same, but the case where proportions of different classes of labeled and unlabeled data are different also worth to look into.

Second, currently, overfitting appears when combine our loss function with neural networks, we conjecture that current loss function itself cannot effectively regularize neural nets, so we need to explore the missing part $R(\mathcal{S}, Z)$ so that we can guarantee that the neural net generate a good feature space.

Furthermore, we may develop some tools to analyze the generalization error, convergence rate and other statistical properties of our method, and compare our methods with other methods in more theoretical ways.

# References


[1] Y. Bengio, O. Delalleau, and N. Le Roux. *Label Propagation and Quadratic Criterion*, pages 193–216. MIT Press, semi-supervised learning edition, January 2006.

[2] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[3] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[4] A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. *Working draft, November*, 3:13, 2005.

[5] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, page 529–536, Cambridge, MA, USA, 2004. MIT Press.

[6] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[7] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[8] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[9] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018.

[10] P. Sellars, A. I. Aviles-Rivero, and C.-B. Schönlieb. Laplacenet: A hybrid energy-neural model for deep semi-supervised classification. *arXiv preprint arXiv:2106.04527*, 2021.

[11] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[12] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608. Curran Associates, Inc., 2020.

[13] Z. Tan, Y. Song, and Z. Ou. Calibrated adversarial algorithms for generative modelling. *Stat*, 8(1):e224, 2019.

[14] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[15] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[16] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, A. Solin, Y. Bengio, and D. Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022.

[17] P. N. Vikas Sindhwani, Misha Belkin. The Geometric Basis of Semi-Supervised Learning. In *Semi-Supervised Learning*. The MIT Press, 09 2006.

[18] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.