LEARNING WITH FEW-SHOT COMPLEMENTARY LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Complementary-label (CL) learning deals with the weak supervision scenario where each training instance is associated with one complementary label, which specifies the class label that the instance does not belong to. Since these CL algorithms rely on the assumption of a large amount of labeled/unlabeled training data, they cannot be applied in few-shot scenarios and perform well. To bridge the gap, we propose a Few-shot CL training pattern with three kinds of surrogate loss, which is based on the Model-Agnostic Meta-Learning and bilevel optimization. We demonstrate the effectiveness of our approach in an extensive empirical study and theoretical analysis.

1 INTRODUCTION

Multi-classification tasks usually require a large amount of data with high-quality labels, but correctly labeling large-scale data is time-consuming and expensive. In order to alleviate such problems, weakly supervised learning frameworks have been studied in recent years, including but not limited to, semi-supervised learning (Chapelle et al., 2006; Miyato et al., 2019), noisy-label learning (Ghosh et al., 2017; Zhang & Sabuncu, 2018; Han et al., 2020), positive-unlabeled learning (du Plessis et al., 2014; Chapel et al., 2020), and partial label learning (Zhang et al., 2017; Lv et al., 2020).

Complementary-label learning (CLL), as a recently proposed weakly supervised learning framework, learns from training instances with a complementary label which specifies a class that the pattern does not belong to is available, thereby replacing any ordinary label. In practice, although the expert marking of ordinary labels is extremely time-consuming and even impossible when the number of categories is very large, it is obviously easy to choose one label and determine whether an object belongs to the chosen class or not. In addition, another potential application are related to data privacy. In some scenarios, for protecting unpublic labels, one strategy provides a complementary label transformed from the true label. To solve the CLL problem, previous approaches mostly concentrate on unbiased risk estimator and surrogate loss through assuming the unbiased relationship between the ground-truth label y and the complementary label $\bar{y}$ based on an uniform distribution, i.e. $\bar{p}(\boldsymbol{x}, \bar{y}) = \frac{1}{K-1} \sum_{y \neq \bar{y}} p(\boldsymbol{x}, y)$ (K refers to the number of classes) (Ishida et al., 2017).

Since these CL algorithms rely on the assumption of a large amount of labeled/unlabeled training data, they cannot be applied in few-shot scenarios and perform well. Furthermore, *few-shot learning (FSL)* mainly assumes the data from similar domain and without noisy labels, which seriously diminishes the performance of the FLS algorithm shown by our experiments. This is an urgent and valuable problem that makes it possible to effectively unite complementary labels to the FSL methods. To bridge the gap, we propose a Few-shot CL training pattern with three kinds of surrogate loss, which is based on the *Model-Agnostic Meta-Learning (MAML)* (Finn et al., 2017) and bilevel optimization (Sinha et al., 2018).

Contributions. (1) We propose a practical and general CL setting, where focuses on few-shot scenarios of training samples. (2) We introduce three kinds of surrogate loss in meta-training and meta testing of model-agnostic meta-Learning, which enforce predictive gap between potential ground-truth label and complementary label. (3) We demonstrate the effectiveness of our approach in an extensive empirical study and theoretical analysis.

2 BACKGROUND AND FORMULATION

In this section, we give notations used in this paper, and briefly discuss ordinary multi-class classification, complementary-label learning and few-shot learning.

Ordinary Multi-Class Classification In ordinary multi-class classification, let X R d be the instance space and Y = [c] be the label space, where d is the feature space dimension, [c] := 1, 2, . . . , c and c ¿ 2 is the number of classes. Let p(x, y) be the underlying joint density of random variables (X, Y) X × Y. The target c is the number of classes and c 2. Let p(x, y) be the unknown probability density function over random variables (X, Y) X × Y, and D = (xi , yi) n i=1 be a set of n training examples each associated with a ground-truth label. Ordinary multi-class classification tasks aim to learn a classifier that maps from the feature space to the label space f : X → R c , which is trained by minimizing the following classification risk: R(f) = E(X,Y)p(x,y)

‘(f(X), eY)

(1) where e Y 0, 1 c is the one-hot encoded label of X, and the Y -th element of e Y is one with all other elements being zero. E and ‘ denote the expectation and the loss function, respectively. Accordingly, the most possible predicted label yb of an instance x is determined as yb = argmax kY fk(x) (2) where fk(·) denotes the k-th element of f(·), referring to the posterior probability of the k-th label being the ground-truth one, i.e., fk(X) = P(Y = k—X). The optimal classifier f in function class F corresponds to the minimizer of classification risk R(f): f = argminfF R(f). As the underlying distribution p(x, y) is unknown, the classification risk in Eq.(1) is usually approximated by the empirical risk Rn(f), i.e. Rn(f) = 1 n Pn i=1 ‘(f(xi), eyi). Similarly, the optimal classifier w.r.t. the empirical risk corresponds to: fn = argminfF Rn(f).

Complementary-Label Learning Different from ordinary multi-class classification, each instance only has one complementary label in CLL. Let D¯ = (xi , y¯i) n i=1 denote the set of complementarily labeled training examples, where y¯i Y yi is the complementary label of the instance xi and each example is sampled from p¯(x, y¯) which denotes an unknown probability distribution. As discussed in Section 1, existing approaches generally aim at modeling generative relationship between P(Y¯ = ¯y — X = x) and P(Y = y — X = x) (WLOG, we rewrite these terms as P(¯y — x) and P(y — x) in the rest of this paper), which can be categorized into the unbiased generative assumption and the biased one, respectively. The work of Ishida et al. (2017) follows the first assumption to define P(¯y — x) as p¯(x, y¯) = 1 c 1 X y6=¯y p(x, y) P(¯y — x)¯p(x) = 1 c 1 X y6=¯y P(y — x)p(x). (3) Since p¯(x) = p(x), we have P(¯y — x) = 1 c1 P y6=¯y p(y — x). Based on Eq.(3), the OVA loss and PC loss for CLL, which naturally lead to an URE serving as an alternative formulation to Eq.(1), are defined as

Few-shot Learning FSL [Li et al., 2006] is an example of meta-learning [Huisman et al., 2020], where a learner is trained on several related tasks during the meta-training phase, so that it can generalize well to unseen (but related) tasks using just few samples with supervision during the meta-testing phase. Existing FSL solutions mainly focus on supervised learning problems, and usually one may term as N-way K-shot classification, where N stands for the number of classes and K means the number of training samples per class, so each task contains KN samples. Given limited support samples for training, unreliable empirical risk minimization is the core issue of FSL, and existing solutions for FSL can be grouped from the perspective of data, model and algorithm [Wang et al., 2020]. Data augmentation-based FSL methods aim to acquire more supervised training samples by generating more samples from original few-shot samples, weakly-labeled/unlabeled data or similar datasets [Douze et al., 2018], and thus to reduce the uncertainty of empirical risk minimization. Model-based FSL methods typically manage to shrink the ambient hypothesis space into a smaller one by extracting prior knowledge in the meta-training phase [Snell et al., 2017; Ren et al., 2018], so empirical risk minimization becomes more reliable and overfitting issue is reduced. Algorithm-based FSL approaches use prior knowledge to guide the seek of optimal model parameters by providing a good initialized parameter or directly learning an optimizer for new tasks [Finn et al., 2017]. Unfortunately, most FSL methods ideally assume the support samples in meta-testing set is with accurate supervision, namely, these samples are precisely annotated with labels. But these support samples are PL ones with irrelevant labels, which mislead the adaption of FSL methods toward the target task (as shown in Fig. 1) and cause a compromised performance. To address this problem, our FsPLL performs the optimization of embedding network and prototype rectification

therein in an iterative manner. In this way, the learnt embedding network and prototypes are less impacted by irrelevant labels of PL samples, and can credibly adapt to new tasks.

3 THE PROPOSED METHODOLOGY

Suppose we are given a small support/training set of n PL samples D = (xi , yi)—1 i n and its corresponding label space and feature space are Y = 0, 1 l and X R d , respectively. The goal of FsPLL is to induce a multi-class classifier f : X → Y, which can precisely predict the groundtruth label of an unseen instance x under this few-shot classification scenario. Different from existing PLL methods, FsPLL should and can utilize the knowledge previously acquired from meta-training phase to quickly adapt to the new classification task D in the meta-testing phase. In the metatraining phase, FsPLL learns an embedding network (metaknowledge) to project PL samples more nearby with their ground-truth prototypes and apart from their non ground-truth prototypes by iteratively rectifying these prototypes in this embedding space. In the meta-testing phase, it rectifies the prototypes of support PL samples using the embedding network and then classifies new samples by their distance to rectified prototypes in the embedding space. In this paper, we take Prototypical Network (PN) [Snell et al., 2017] as the base of our embedding network. The framework overview of FsPLL is given in Fig. 1. The following subsections elaborate on the two phases.

3.1 META-TRAINING PHASE

The meta-training phase mainly aims to extract prior knowledge from multiple relevant tasks for the target task. Suppose we are given T 1 few-shot datasets (tasks) denoted as Dt train (1 t T). For each dataset Dt train = Xt s , X˜ t q , Yt, where Xt s = (x t 1 , x t 2 , . . . , x t ns) R d×ns denotes the data matrix of support samples, X˜ t q =. (x˜ t 1 , x˜ t 2 , · · · , x˜ t nq) R d×nq denotes data matrix of query samples, Yt = (y t 1 , y t 2 , · · · , y t ns) R l×ns is the corresponding label matrix of support samples, and ns + nq ¡ n. Yt ci = 1 means the c-th label is a candidate label of the i-th sample; Yt ci = 0 otherwise. Let Qt R l×ns denotes the underlying label confidence matrix of support samples and it is initialized as Yt , where Qt ci indicates the confidence of the c-th label as the ground-truth label of the i-th sample. From these datasets, we aim at learning an embedding network, i.e., f : R d → R m, by which we can obtain the representation of every label in the embedding space and can be more robust to irrelevant labels of support samples therein. Suppose Pt = (p t 1 , p t 2 , . . . , p t l) R m×l is the prototype/representation matrix of l class labels of the t-th task, where p t c denotes the prototype of the c-th label in the embedding space. PN [Snell et al., 2017] computes the prototype by p t c = Pns i=1 Yt ci×f(x t i P) ns i=1 Yt ci , while Semi-PN [Ren et al., 2018], a variant of PN, further uses unlabeled examples to improve the prototype learning. They both simply take all PL samples annotated with the c-th label to induce the prototype, ignoring that some PL samples actually not annotated with this label. Therefore, PN and Semi-PN give contaminated prototypes. For example, prototype of goose ('circle with 1') in Fig. 1 is misled by irrelevant labels, which consequently compromises the classification performance, especially when support PL samples with excessive irrelevant labels. To address this issue, FsPLL performs prototype rectification and label confidence update in an iterative way to seek noise-robust embedding network and prototypes in the embedding space, as shown in Fig. 1. FsPLL defines each prototype based on the confidence weighted mean of corresponding support samples in the embedding space as follows: p t c = Pns i=1 Qt ci × f(x t i) Pns i=1 Qt ci . (1) Unlike prototypes optimized by PN, FsPLL rectifies the prototypes using iterative updated label confident matrix Qt , and thus explicitly accounts for the irrelevant labels of samples. It is expected for a sample to be closer to its ground-truth prototype in the embedding space; this would enable a confident label prediction in this space. Given this, we use a softmax to update the label confidence matrix Qt as follows: Qt ci = (exp(d(f(x t i),p t c P)) l c=1exp(d(f(x t i),pt c))×Yt ci , if Yt ci = 1 0, otherwise , (2) where d(f(x t i), p t c) quantifies the Euclidean distance between sample x t i and prototype p t c in the embedding space. The labels of a PL sample can be disambiguated by referring to labels of its neighborhood samples [Wang et al., 2019]. We observe that PN and Eq. (1) disregard the neighborhood support samples when computing the prototype. Unlike these PLL methods that disambiguate in the original feature space or linearly projected subspace, FsPLL further updates the label confidence matrix in the embedding space as follows: Qt ci = Qt ci + —Nk(x t i)— X x t jNk(x t i) Qt cj , if Yt ci = 1, (3). where Nk(x t i) includes the k-nearest samples of x t i , and the neighborhood is determined by Euclidean distance in the embedding space. trade-offs the confidence from the sample itself and those from neighborhood

samples. In this way, FsPLL utilizes local manifold of samples to rectify prototypes. Based on the rectified prototypes and embedding network f, we can predict the label of a query sample with a softmax over its distances to all prototypes in the embedding space as: p(z t j = c — x˜ t j) = exp(d(f(x˜ t j), p t c)) Pl i=1 exp(d(f(x˜ t j), p t i)) , (4) where z t j is the unknown ground-truth label of the j-th query sample. To make the representation of every query sample in the embedding space closer to its ground-truth prototype and apart from its non ground-truth prototypes, FsPLL minimizes the negative log-probability of the most likely label of a query example as follows: J(, x˜ t i) = log(max c=1,··· ,l p(z t j = c — x˜ t i)). (5) By minimizing the above equation, FsPLL can obtain the rectified prototypes Pt and the corresponding embedding network parameterized by f for task Dt train. We want to remark that the l-th labels for different tasks is not always the same. The meta-training phase involves a lot of different tasks, each of which is composed of support/query samples. To enable a good generalization ability, it attempts to gain the optimal mode parameter by minimizing the average negative log-probability of the most likely labels of all query samples over T tasks as follows: = arg min XT t=1 1 nq Xnq i=1 J(, x˜ t i). (6) To this end, FsPLL obtains an embedding network f that is robust to irrelevant labels of PL samples across T tasks. Via this network, a PL sample in the embedding space is made closer to its ground-truth prototype than to other prototypes, and the generalization and fast adaption ability are pursued among T different tasks

3.2 Meta-testing Phase

In the meta-testing phase, we are only given a small set of PL samples, which compose the target task with support and query samples. These support samples are overly-annotated with irrelevant labels, while query samples are without label information. We want to highlight that the labels of these PL samples are disjoint with the labels used in the meta-training phase. In other words, the PL samples are few-shot ones. Here, FsPLL aims to use the meta-knowledge (embedding network f) acquired in the meta-training phase to precisely annotate the query samples based on the inaccurately supervised few-shot support examples. Formally, FsPLL aims to quickly generalize to a new task Dtest = Xs, X˜ q, Y, where Xs R d×ns , X˜ q R d×nq and Y R l×ns denote the data matrices of support examples, of query examples, and of labels of query examples, respectively. Alike the meta-training phase, FsPLL first computes the prototypes P R m×l of this new task in the embedding space using the confidence-weighted mean of support samples Xs and label confidence matrix Q as in Eq. (1). Then the label confidence matrix Q of the support samples is updated based on a softmax over their distances to prototypes as in Eq. (2) and local manifold as in Eq. (3). FsPLL repeats the above two steps to rectify the prototypes and update label confidence matrix for adapting to the target task. Note, the embedding network f is fixed during the above repetitive optimization. Given a query sample xi , FsPLL classifies its label zi using its distance to rectified prototypes P R m×l as follows: zi = arg max q p (zi = q — xi) (q = 1, · · · , l). (7)

4 Theoretical Analysis

5 Experiments

6 Submission of conference papers to ICLR 2022

ICLR requires electronic submissions, processed by `https://openreview.net/`. See ICLR's website for more instructions.

If your paper is ultimately accepted, the statement `\iclrfinalcopy` should be inserted to adjust the format to the camera ready requirements.

The format for the submissions is a variant of the NeurIPS format. Please read carefully the instructions below, and follow them faithfully.

6.1 Style

Papers to be submitted to ICLR 2022 must be prepared according to the instructions presented here.

Authors are required to use the ICLR LaTeX style files obtainable at the ICLR website. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

6.2 RETRIEVAL OF STYLE FILES

The style files for ICLR and other conference information are available online at:

`http://www.iclr.cc/`

The file `iclr2022_conference.pdf` contains these instructions and illustrates the various formatting requirements your ICLR paper must satisfy. Submissions must be made using LaTeX and the style files `iclr2022_conference.sty` and `iclr2022_conference.bst` (to be used with LaTeX2e). The file `iclr2022_conference.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in sections 7, 8, and 9 below.

7 GENERAL FORMATTING INSTRUCTIONS

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing of 11 points. Times New Roman is the preferred typeface throughout. Paragraphs are separated by 1/2 line space, with no indentation.

Paper title is 17 point, in small caps and left-aligned. All pages should start at 1 inch (6 picas) from the top of the page.

Authors' names are set in boldface, and each name is placed above its corresponding address. The lead author's name is to be listed first, and the co-authors' names are set to follow. Authors sharing the same address can be on the same line.

Please pay special attention to the instructions in section 9 regarding figures, tables, acknowledgments, and references.

There will be a strict upper limit of 9 pages for the main text of the initial submission, with unlimited additional pages for citations.

8 HEADINGS: FIRST LEVEL

First level headings are in small caps, flush left and in point size 12. One line space before the first level heading and 1/2 line space after the first level heading.

8.1 HEADINGS: SECOND LEVEL

Second level headings are in small caps, flush left and in point size 10. One line space before the second level heading and 1/2 line space after the second level heading.

8.1.1 HEADINGS: THIRD LEVEL

Third level headings are in small caps, flush left and in point size 10. One line space before the third level heading and 1/2 line space after the third level heading.

9 CITATIONS, FIGURES, TABLES, REFERENCES

These instructions apply to everyone, regardless of the formatter being used.

9.1 CITATIONS WITHIN THE TEXT

Citations within the text should be based on the `natbib` package and include the authors' last names and year (with the "et al." construct for more than two authors). When the authors or the publication are included in the sentence, the citation should not be in parenthesis using `\citet{}` (as in "See **?** for more information."). Otherwise, the citation should be in parenthesis using `\citep{}` (as in "Deep learning shows promise to make progress towards AI (**?**).").

The corresponding references are to be listed in alphabetical order of authors, in the REFERENCES section. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

9.2 FOOTNOTES

Indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).[2]

9.3 FIGURES

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction; art work should not be hand-drawn. The figure number and caption always appear after the figure. Place one line space before the figure caption, and one line space after the figure. The figure caption is lower case (except for first word and proper nouns); figures are numbered consecutively.

Make sure the figure caption does not get separated from the figure. Leave sufficient space to avoid splitting the figure and figure caption.

You may use color figures. However, it is best for the figure captions and the paper body to make sense if the paper is printed either in black/white or in color.

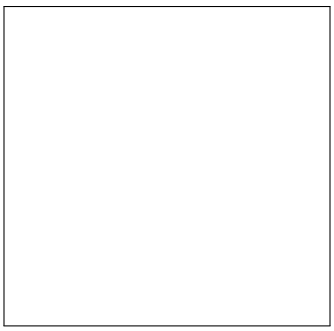

Figure 1: Sample figure caption.

9.4 TABLES

All tables must be centered, neat, clean and legible. Do not use hand-drawn tables. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

[1] Sample of the first footnote

[2] Sample of the second footnote

Table 1: Sample table title

PART	DESCRIPTION
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

10 DEFAULT NOTATION

In an attempt to encourage standardized notation, we have included the notation file from the textbook, *Deep Learning* **?** available at `https://github.com/goodfeli/dlbook_notation/`. Use of this style is not required and can be disabled by commenting out `math_commands.tex`.

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\mathsf{\mathbf{A}}$	A tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
$\boldsymbol{e}^{(i)}$	Standard basis vector $[0, \dots, 0, 1, 0, \dots, 0]$ with a 1 at position i
$\text{diag}(\boldsymbol{a})$	A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \dots, n\}$	The set of all integers between 0 and n
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b
$\mathbb{A} \backslash \mathbb{B}$	Set subtraction, i.e., the set containing the elements of $\mathbb{A}$ that are not in $\mathbb{B}$
$\mathcal{G}$	A graph
$Pa_{\mathcal{G}}(\mathrm{x}_i)$	The parents of x_i in $\mathcal{G}$

Indexing

a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
a_{-i}	All elements of vector $\boldsymbol{a}$ except for element i
$A_{i,j}$	Element i, j of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$\mathsf{A}_{i,j,k}$	Element (i, j, k) of a 3-D tensor A
$\mathsf{A}_{:,:,i}$	2-D slice of a 3-D tensor
a_i	Element i of the random vector $\mathbf{a}$

Calculus

$\frac{dy}{dx}$	Derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	Matrix derivatives of y with respect to $\boldsymbol{X}$
$\nabla_{\mathsf{X}} y$	Tensor containing derivatives of y with respect to X
$\frac{\partial f}{\partial \boldsymbol{x}}$	Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \to \mathbb{R}^m$
$\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x})$ or $\boldsymbol{H}(f)(\boldsymbol{x})$	The Hessian matrix of f at input point $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$

Probability and Information Theory

$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathrm{x} \sim P}[f(x)]$ or $\mathbb{E} f(x)$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$
$H(\mathrm{x})$	Shannon entropy of the random variable x
$D_{\mathrm{KL}}(P \Vert Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Functions

$f : \mathbb{A} \to \mathbb{B}$	The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f \circ g$	Composition of the functions f and g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. (Sometimes we write $f(\boldsymbol{x})$ and omit the argument $\boldsymbol{\theta}$ to lighten notation)
$\log x$	Natural logarithm of x
$\sigma(x)$	Logistic sigmoid, $\dfrac{1}{1 + \exp(-x)}$
$\zeta(x)$	Softplus, $\log(1 + \exp(x))$
$\|\|\boldsymbol{x}\|\|_p$	L^p norm of $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|$	L^2 norm of $\boldsymbol{x}$
x^+	Positive part of x, i.e., $\max(0, x)$
$\mathbf{1}_{\text{condition}}$	is 1 if the condition is true, 0 otherwise

11 FINAL INSTRUCTIONS

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the REFERENCES section; see below). Please note that pages should be numbered.

12 PREPARING POSTSCRIPT OR PDF FILES

Please prepare PostScript or PDF files with paper size "US Letter", and not, for example, "A4". The -t letter option on dvips will produce US Letter files.

Consider directly generating PDF files using `pdflatex` (especially if you are a MiKTeX user). PDF figures must be substituted for EPS figures, however.

Otherwise, please generate your PostScript and PDF files with the following commands:

```
dvips mypaper.dvi -t letter -Ppdf -G0 -o mypaper.ps
ps2pdf mypaper.ps mypaper.pdf
```

12.1 MARGINS IN LATEX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the graphicx package. Always specify the figure width as a multiple of the line width as in the example below using .eps graphics

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.eps}
```

or

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

for .pdf graphics. See section 4.4 in the graphics bundle documentation (`http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command.

AUTHOR CONTRIBUTIONS

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Laetitia Chapel, Mokhtar Z. Alaya, and Gilles Gasso. Partial optimal tranport with applications on positive-unlabeled learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1e6e25d952a0d639b676ee20d0519ee2-Abstract.html.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (eds.). *Semi-Supervised Learning*. The MIT Press, 2006. ISBN 9780262033589. doi: 10.7551/mitpress/9780262033589.001.0001. URL https://doi.org/10.7551/mitpress/9780262033589.001.0001.

Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 703–711, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/35051070e572e47d2c26c241ab88307f-Abstract.html.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017. URL http://proceedings.mlr.press/v70/finn17a.html.

Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In Satinder P. Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 1919–1925. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14759.

Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor W. Tsang, and Masashi Sugiyama. SIGUA: forgetting may make learning with noisy labels more robust. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4006–4016. PMLR, 2020. URL http://proceedings.mlr.press/v119/han20c.html.

Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5639–5649, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/1dba5eed8838571e1c80af145184e515-Abstract.html.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6500–6510. PMLR, 2020. URL http://proceedings.mlr.press/v119/lv20a.html.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993, 2019. doi: 10.1109/TPAMI.2018.2858821. URL https://doi.org/10.1109/TPAMI.2018.2858821.

Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Trans. Evol. Comput.*, 22(2):276–295, 2018. doi: 10.1109/TEVC.2017.2712906. URL https://doi.org/10.1109/TEVC.2017.2712906.

Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Trans. Knowl. Data Eng.*, 29(10):2155–2167, 2017. doi: 10.1109/TKDE.2017.2721942. URL https://doi.org/10.1109/TKDE.2017.2721942.

Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8792–8802, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/f2925f97bc13ad2852a7a551802feea0-Abstract.html.

A Appendix

You may include other additional sections here.