Submitted 5/09; Published 2/10

Iterative Scaling and Coordinate Descent Methods for Maximum Entropy Models

Fang-Lan Huang D93011@CSIE.NTU.EDU.TW
Cho-Jui Hsieh B92085@CSIE.NTU.EDU.TW
Kai-Wei Chang B92084@CSIE.NTU.EDU.TW
Chih-Jen Lin CJLIN@CSIE.NTU.EDU.TW
Department of Computer Science
National Taiwan University
Taipei 106, Taiwan

Editor: Michael Collins

Abstract

Maximum entropy (Maxent) is useful in natural language processing and many other areas. Iterative scaling (IS) methods are one of the most popular approaches to solve Maxent. With many variants of IS methods, it is difficult to understand them and see the differences. In this paper, we create a general and unified framework for iterative scaling methods. This framework also connects iterative scaling and coordinate descent methods. We prove general convergence results for IS methods and analyze their computational complexity. Based on the proposed framework, we extend a coordinate descent method for linear SVM to Maxent. Results show that it is faster than existing iterative scaling methods.

Keywords: maximum entropy, iterative scaling, coordinate descent, natural language processing, optimization

1. Introduction

Maximum entropy (Maxent) is widely used in many areas such as natural language processing (NLP) and document classification. It is suitable for problems needing probability interpretations. For many NLP tasks, given a word sequence, we can use Maxent models to predict the label sequence with the maximal probability (Berger et al., 1996). Such tasks are different from traditional classification problems, which assign label(s) to a single instance.

Maxent models the conditional probability as:

$$\begin{aligned} &P_{\boldsymbol{w}}(y|x) \equiv \frac{S_{\boldsymbol{w}}(x,y)}{T_{\boldsymbol{w}}(x)}, \\ &S_{\boldsymbol{w}}(x,y) \equiv e^{\sum_t w_t f_t(x,y)},\ T_{\boldsymbol{w}}(x) \equiv \sum_y S_{\boldsymbol{w}}(x,y), \end{aligned} \tag{1}$$

where x indicates a context, y is the label of the context, and $\boldsymbol{w} \in R^n$ is the weight vector. A real-valued function $f_t(x,y)$ denotes the t-th feature extracted from the context x and the label y. We assume a finite number of features. In some cases, $f_t(x,y)$ is 0/1 to indicate a particular property. $T_{\boldsymbol{w}}(x)$ is a normalization term applied to make $\sum_y P_{\boldsymbol{w}}(y|x) = 1$.

Given an empirical probability distribution $\tilde{P}(x,y)$ obtained from training samples, Maxent minimizes the following negative log-likelihood:

$$\min_{\boldsymbol{w}} \; -\sum_{x,y} \tilde{P}(x,y) \log P_{\boldsymbol{w}}(y|x),$$

or equivalently,

$$\min_{\boldsymbol{w}} \; \sum_{x} \tilde{P}(x) \log T_{\boldsymbol{w}}(x) - \sum_{t} w_t \tilde{P}(f_t), \tag{2}$$

where $\tilde{P}(x,y) = N_{x,y}/N$, $N_{x,y}$ is the number of times that (x,y) occurs in training data, and N is the total number of training samples. $\tilde{P}(x) = \sum_y \tilde{P}(x,y)$ is the marginal probability of x, and $\tilde{P}(f_t) = \sum_{x,y} \tilde{P}(x,y) f_t(x,y)$ is the expected value of $f_t(x,y)$. To avoid overfitting the training samples, some add a regularization term to (2) and solve:

$$\min_{\boldsymbol{w}} \; L(\boldsymbol{w}) \equiv \min_{\boldsymbol{w}} \; \sum_{x} \tilde{P}(x) \log T_{\boldsymbol{w}}(x) - \sum_{t} w_t \tilde{P}(f_t) + \frac{1}{2\sigma^2} \sum_{t} w_t^2, \tag{3}$$

where σ is a regularization parameter. More discussion about regularization terms for Maxent can be seen in, for example, Chen and Rosenfeld (2000). We focus on (3) in this paper because it is strictly convex. Note that (2) is convex, but may not be strictly convex. We can further prove that a unique global minimum of (3) exists. The proof, omitted here, is similar to Theorem 1 in Lin et al. (2008).

Iterative scaling (IS) methods are popular in training Maxent models. They all share the same property of *solving a one-variable sub-problem at a time.* Existing IS methods include generalized iterative scaling (GIS) by Darroch and Ratcliff (1972), improved iterative scaling (IIS) by Della Pietra et al. (1997), and sequential conditional generalized iterative scaling (SCGIS) by Goodman (2002). The approach by Jin et al. (2003) is also an IS method, but it assumes that every class uses the same set of features. As this assumption is not general, in this paper we do not include this approach for discussion. In optimization, coordinate descent (CD) is a popular method which also solves a one-variable sub-problem at a time. With these many IS and CD methods, it is difficult to see their differences. In Section 2, we propose a unified framework to describe IS and CD methods from an optimization viewpoint. We further analyze the theoretical convergence as well as computational complexity of IS and CD methods. In particular, general linear convergence is proved. In Section 3, based on a comparison between IS and CD methods, we propose a new and more efficient CD method. These two results (a unified framework and a faster CD method) are the main contributions of this paper.

Besides IS methods, numerous optimization methods have been applied to train Maxent. For example, Liu and Nocedal (1989), Bottou (2004), Daumé (2004), Keerthi et al. (2005), McDonald and Pereira (2006), Vishwanathan et al. (2006), Koh et al. (2007), Genkin et al. (2007), Andrew and Gao (2007), Schraudolph et al. (2007), Gao et al. (2007), Collins et al. (2008), Lin et al. (2008) and Friedman et al. (2010). They do not necessarily solve the optimization problem (3). Some handle more complicated log linear models such as Conditional Random Fields (CRF), but their approaches can be modified for Maxent. Some focus on logistic regression, which is a special form of Maxent if the number of labels is two. Moreover, some consider the L1 regularization term $\sum_t |w_t|$ in (3). Several papers have compared optimization methods for Maxent, though it is difficult to have a complete study. Malouf (2002) compares methods for NLP data, while Minka (2003) focuses on logistic regression for synthesis data. In this paper, we are interested in a detailed investigation of IS methods because they remain one of the most used approaches to train Maxent. This fact can be easily seen from popular NLP software. The Stanford Log-linear POS Tagger[1] supports two optimization methods, where one is IIS. The OpenNLP Maxent package (Baldridge et al., 2001) provides only one optimization method, which is GIS.

This paper is organized as follows. In Section 2, we present a unified framework for IS/CD methods and give theoretical results. Section 3 proposes a new CD method. Its advantages over existing IS/CD methods are discussed. In Section 4, we investigate some implementation issues for IS/CD methods. Section 5 presents experimental results. With a careful implementation, our CD outperforms IS and quasi-Newton techniques. Finally, Section 6 gives discussion and conclusions.

Part of this work appears in a short conference paper (Huang et al., 2009).

Notation X, Y, and n are the numbers of contexts, class labels, and features, respectively. The total number of nonzeros in training data and the average number of nonzeros per feature are respectively

$$\#\text{nz} \equiv \sum_{x,y} \sum_{t:f_t(x,y)\neq 0} 1 \quad \text{and} \quad \bar{l} \equiv \frac{\#\text{nz}}{n}. \tag{4}$$

In this paper, we assume non-negative feature values:

$$f_t(x,y) \geq 0, \ \forall t, x, y. \tag{5}$$

Most NLP applications have non-negative feature values. All existing IS methods use this property.

2. A Framework for Iterative Scaling and Coordinate Descent Methods

An important characteristic of IS and CD methods is that they solve a one-variable optimization problem and then modify the corresponding element in $\boldsymbol{w}$. Conceptually, the one-variable sub-problem is related to the function reduction

$$L(\boldsymbol{w} + z_t \boldsymbol{e}_t) - L(\boldsymbol{w}),$$

where $\boldsymbol{e}_t \equiv [\underbrace{0, \ldots, 0}_{t-1}, 1, 0, \ldots, 0]^T$. Then IS methods differ in how they approximate the function reduction. They can also be categorized according to whether $\boldsymbol{w}$'s components are updated in a sequential or parallel way. In this section, we create a framework for these methods. A hierarchical illustration of the framework is in Figure 1.

2.1 The Framework

To introduce the framework, we separately discuss coordinate descent methods according to whether $\boldsymbol{w}$ is sequentially or parallelly updated.

[1] Stanford Log-linear POS Tagger can be found at `http://nlp.stanford.edu/software/tagger.shtml`.

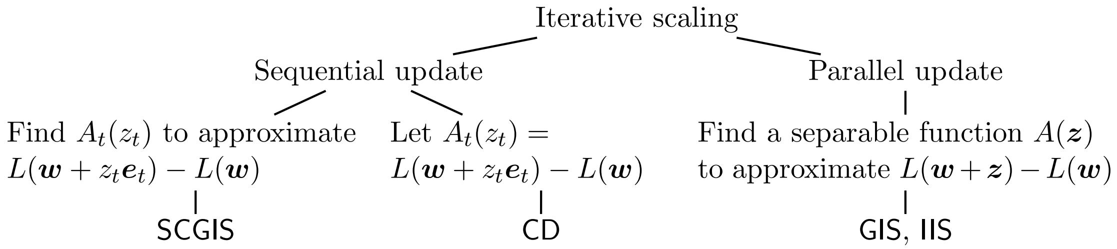

Iterative scaling
Sequential update
Parallel update
Find $A_t(z_t)$ to approximate $L(\boldsymbol{w} + z_t\boldsymbol{e}_t) - L(\boldsymbol{w})$
Let $A_t(z_t) = L(\boldsymbol{w} + z_t\boldsymbol{e}_t) - L(\boldsymbol{w})$
Find a separable function $A(\boldsymbol{z})$ to approximate $L(\boldsymbol{w}+\boldsymbol{z}) - L(\boldsymbol{w})$
SCGIS
CD
GIS, IIS

Figure 1: An illustration of various iterative scaling methods.

2.1.1 Sequential Update

For a sequential-update algorithm, once a one-variable sub-problem is solved, the corresponding element in $\boldsymbol{w}$ is updated. The new $\boldsymbol{w}$ is then used to construct the next sub-problem. The procedure is sketched in Algorithm 1. If the t-th component is selected for update, a sequential IS method solves the following one-variable sub-problem:

$$\min_{z_t} \; A_t(z_t),$$

where $A_t(z_t)$ is twice differentiable and bounds the function difference:

$$A_t(z_t) \geq L(\boldsymbol{w} + z_t\boldsymbol{e}_t) - L(\boldsymbol{w}), \; \forall z_t. \tag{6}$$

We hope that by minimizing $A_t(z_t)$, the resulting $L(\boldsymbol{w} + z_t\boldsymbol{e}_t)$ can be smaller than $L(\boldsymbol{w})$. However, (6) is not enough to ensure this property, so we impose an additional condition

$$A_t(0) = 0 \tag{7}$$

on the approximate function $A_t(z_t)$. The explanation below shows that we can strictly decrease the function value. If $A_t'(0) \neq 0$ and assume $\bar{z}_t \equiv \arg\min_{z_t} A_t(z_t)$ exists, with the condition $A_t(0) = 0$, we have $A_t(\bar{z}_t) < 0$. This property and (6) then imply $L(\boldsymbol{w} + \bar{z}_t\boldsymbol{e}_t) < L(\boldsymbol{w})$. If $A_t'(0) = 0$, we can prove that $\nabla_t L(\boldsymbol{w}) = 0$,[2] where $\nabla_t L(\boldsymbol{w}) = \partial L(\boldsymbol{w})/\partial w_t$. In this situation, the convexity of $L(\boldsymbol{w})$ and $\nabla_t L(\boldsymbol{w}) = 0$ imply that we cannot decrease the function value by modifying w_t, so we should move on to modify other components of $\boldsymbol{w}$.

A CD method can be viewed as a sequential-update IS method. Its approximate function $A_t(z_t)$ is simply the function difference:

$$A_t^{\mathsf{CD}}(z_t) = L(\boldsymbol{w} + z_t\boldsymbol{e}_t) - L(\boldsymbol{w}). \tag{8}$$

Other IS methods consider approximations so that $A_t(z_t)$ is simpler for minimization. More details are in Section 2.2. Note that the name "sequential" comes from the fact that each sub-problem $A_t(z_t)$ depends on $\boldsymbol{w}$ obtained from the previous update. Therefore, sub-problems must be sequentially solved.

[2] Define a function $D(z_t) \equiv A(z_t) - (L(\boldsymbol{w}+z_t\boldsymbol{e}_t) - L(\boldsymbol{w}))$. We have $D'(0) = A'(0) - \nabla_t L(\boldsymbol{w})$. If $\nabla_t L(\boldsymbol{w}) \neq 0$ and $A_t'(0) = 0$, then $D'(0) \neq 0$. Since $D(0) = 0$, we can find a z_t such that $A(z_t) - (L(\boldsymbol{w}+z_t\boldsymbol{e}_t) - L(\boldsymbol{w})) < 0$, a contradiction to (6).

Algorithm 1 A sequential-update IS method

```
While $\boldsymbol{w}$ is not optimal
    For $t = 1, \ldots, n$
        1. Find an approximate function $A_t(z_t)$ satisfying (6)-(7).
        2. Approximately solve $\min_{z_t} A_t(z_t)$ to get $\bar{z}_t$.
        3. $w_t \leftarrow w_t + \bar{z}_t$.
```

Algorithm 2 A parallel-update IS method

```
While $\boldsymbol{w}$ is not optimal
    1. Find approximate functions $A_t(z_t)\ \forall z_t$ satisfying (9).
    2. For $t = 1, \ldots, n$
           Approximately solve $\min_{z_t} A_t(z_t)$ to get $\bar{z}_t$.
    3. For $t = 1, \ldots, n$
           $w_t \leftarrow w_t + \bar{z}_t$.
```

2.1.2 Parallel Update

A parallel-update IS method simultaneously constructs n independent one-variable sub-problems. After (approximately) solving all of them, the whole vector $\boldsymbol{w}$ is updated. Algorithm 2 gives the procedure. The function $A(\boldsymbol{z})$, $\boldsymbol{z} \in R^n$, is an approximation of $L(\boldsymbol{w}+\boldsymbol{z}) - L(\boldsymbol{w})$ satisfying

$$A(\boldsymbol{z}) \geq L(\boldsymbol{w}+\boldsymbol{z}) - L(\boldsymbol{w}),\ \forall \boldsymbol{z}, \quad A(\mathbf{0}) = 0, \quad \text{and} \quad A(\boldsymbol{z}) = \sum_{t=1}^{n} A_t(z_t). \tag{9}$$

The first two conditions are similar to (6) and (7). By a similar argument, we can ensure that the function value is strictly decreasing. The last condition indicates that $A(\boldsymbol{z})$ is separable, so

$$\min_{\boldsymbol{z}} A(\boldsymbol{z}) = \sum_{t=1}^{n} \min_{z_t} A_t(z_t).$$

That is, we can minimize $A_t(z_t)$, $\forall z_t$ simultaneously, and then update $w_t\ \forall t$ together. We show in Section 4 that a parallel-update method possesses some nicer implementation properties than a sequential method. However, as sequential approaches update $\boldsymbol{w}$ as soon as a sub-problem is solved, they often converge faster than parallel methods.

If $A(\boldsymbol{z})$ satisfies (9), taking $\boldsymbol{z} = z_t \boldsymbol{e}_t$ implies that (6) and (7) hold for $A_t(z_t)$, $\forall t = 1, \ldots, n$. A parallel-update method could thus be transformed to a sequential-update method using the same approximate function. Contrarily, a sequential-update algorithm cannot be directly transformed to a parallel-update method because the summation of the inequality in (6) does not imply (9).

2.2 Existing Iterative Scaling Methods

We introduce GIS, IIS and SCGIS via the proposed framework. GIS and IIS use a parallel update, but SCGIS is sequential. Their approximate functions aim to bound the change of the function values

$$L(\boldsymbol{w}+\boldsymbol{z})-L(\boldsymbol{w})=\sum_x \tilde{P}(x)\log\frac{T_{\boldsymbol{w}+\boldsymbol{z}}(x)}{T_{\boldsymbol{w}}(x)}+\sum_t Q_t(z_t), \tag{10}$$

where $T_{\boldsymbol{w}}(x)$ is defined in (1) and

$$Q_t(z_t)\equiv\frac{2w_tz_t+z_t^2}{2\sigma^2}-z_t\tilde{P}(f_t). \tag{11}$$

Then GIS, IIS and SCGIS use similar inequalities to get approximate functions. With

$$\begin{aligned}\frac{T_{\boldsymbol{w}+\boldsymbol{z}}(x)}{T_{\boldsymbol{w}}(x)}=&\frac{\sum_y S_{\boldsymbol{w}+\boldsymbol{z}}(x,y)}{T_{\boldsymbol{w}}(x)}=\frac{\sum_y S_{\boldsymbol{w}}(x,y)\left(e^{\sum_t z_tf_t(x,y)}\right)}{T_{\boldsymbol{w}}(x)}\\=&\sum_y P_{\boldsymbol{w}}(y|x)e^{\sum_t z_tf_t(x,y)},\end{aligned}$$

they apply $\log\alpha\le\alpha-1\ \forall\alpha>0$ and $\sum_y P_{\boldsymbol{w}}(y|x)=1$ to get

$$\begin{aligned}(10)\le&\sum_t Q_t(z_t)+\sum_x\tilde{P}(x)\left(\sum_y P_{\boldsymbol{w}}(y|x)e^{\sum_t z_tf_t(x,y)}-1\right)\\=&\sum_t Q_t(z_t)+\sum_{x,y}\tilde{P}(x)P_{\boldsymbol{w}}(y|x)\left(e^{\sum_t z_tf_t(x,y)}-1\right).\end{aligned} \tag{12}$$

GIS defines

$$f^\#\equiv\max_{x,y}f^\#(x,y),\qquad f^\#(x,y)\equiv\sum_t f_t(x,y),$$

and adds a feature $f_{n+1}(x,y)\equiv f^\#-f^\#(x,y)$ with $z_{n+1}=0$. Using Jensen's inequality and the assumption of non-negative feature values (5),

$$\begin{aligned}&e^{\sum_{t=1}^n z_tf_t(x,y)}=e^{\sum_{t=1}^{n+1}\frac{f_t(x,y)}{f^\#}z_tf^\#}\\\le&\sum_{t=1}^{n+1}\frac{f_t(x,y)}{f^\#}e^{z_tf^\#}=\sum_{t=1}^{n}\frac{f_t(x,y)}{f^\#}e^{z_tf^\#}+\frac{f^\#-f^\#(x,y)}{f^\#}=\sum_{t=1}^{n}\left(\frac{e^{z_tf^\#}-1}{f^\#}f_t(x,y)\right)+1.\end{aligned} \tag{13}$$

Substituting (13) into (12), the approximate function of GIS is

$$A^{\mathsf{GIS}}(\boldsymbol{z})=\sum_t Q_t(z_t)+\sum_{x,y}\tilde{P}(x)P_{\boldsymbol{w}}(y|x)\sum_t\left(\frac{e^{z_tf^\#}-1}{f^\#}f_t(x,y)\right).$$

Then we obtain n independent one-variable functions:

$$A_t^{\mathsf{GIS}}(z_t)=Q_t(z_t)+\frac{e^{z_tf^\#}-1}{f^\#}\sum_{x,y}\tilde{P}(x)P_{\boldsymbol{w}}(y|x)f_t(x,y).$$

IIS assumes $f_t(x,y) \geq 0$ and applies Jensen's inequality

$$e^{\sum_t z_t f_t(x,y)} = e^{\sum_t \frac{f_t(x,y)}{f^\#(x,y)} z_t f^\#(x,y)} \leq \sum_t \frac{f_t(x,y)}{f^\#(x,y)} e^{z_t f^\#(x,y)}$$

on (12) to get the approximate function

$$A_t^{\mathsf{IIS}}(z_t) = Q_t(z_t) + \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) \frac{e^{z_t f^\#(x,y)} - 1}{f^\#(x,y)}.$$

SCGIS is a sequential-update algorithm. It replaces $f^\#$ in GIS with

$$f_t^\# \equiv \max_{x,y} f_t(x,y). \tag{14}$$

Using $z_t \boldsymbol{e}_t$ as $\boldsymbol{z}$ in (10), a derivation similar to (13) gives

$$e^{z_t f_t(x,y)} \leq \frac{f_t(x,y)}{f_t^\#} e^{z_t f_t^\#} + \frac{f_t^\# - f_t(x,y)}{f_t^\#}. \tag{15}$$

The approximate function of SCGIS is

$$A_t^{\mathsf{SCGIS}}(z_t) = Q_t(z_t) + \frac{e^{z_t f_t^\#} - 1}{f_t^\#} \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y).$$

As a comparison, we expand $A_t^{\mathsf{CD}}(z_t)$ in (8) to the following form:

$$A_t^{\mathsf{CD}}(z_t) = Q_t(z_t) + \sum_x \tilde{P}(x) \log \frac{T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)}{T_{\boldsymbol{w}}(x)} \tag{16}$$

$$= Q_t(z_t) + \sum_x \tilde{P}(x) \log \left(1 + \sum_y P_{\boldsymbol{w}}(y|x)(e^{z_t f_t(x,y)} - 1)\right), \tag{17}$$

where (17) is from (1) and

$$S_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y) = S_{\boldsymbol{w}}(x,y) e^{z_t f_t(x,y)}, \tag{18}$$

$$T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x) = T_{\boldsymbol{w}}(x) + \sum_y S_{\boldsymbol{w}}(x,y)(e^{z_t f_t(x,y)} - 1). \tag{19}$$

A summary of approximate functions of IS and CD methods is in Table 1.

2.3 Convergence of Iterative Scaling and Coordinate Descent Methods

The convergence of CD methods has been well studied (e.g., Bertsekas, 1999; Luo and Tseng, 1992). However, for methods like IS which use only an approximate function to bound the function difference, the convergence is less studied. In this section, we generalize the linear convergence proof in Chang et al. (2008) to show the convergence of IS and CD methods. To begin, we consider any convex and differentiable function L: $R^n \to R$ satisfying the following conditions in the set

$$U = \{\boldsymbol{w} \mid L(\boldsymbol{w}) \leq L(\boldsymbol{w}^0)\}, \tag{20}$$

where $\boldsymbol{w}^0$ is the initial point of an IS/CD algorithm:

$$
\begin{aligned}
A_t^{\mathsf{GIS}}(z_t) &= Q_t(z_t) + \frac{e^{z_t f^\#} - 1}{f^\#} \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) \\
A_t^{\mathsf{IIS}}(z_t) &= Q_t(z_t) + \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) \frac{e^{z_t f^\#(x,y)} - 1}{f^\#(x,y)} \\
A_t^{\mathsf{SCGIS}}(z_t) &= Q_t(z_t) + \frac{e^{z_t f_t^\#} - 1}{f_t^\#} \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) \\
A_t^{\mathsf{CD}}(z_t) &= Q_t(z_t) + \sum_{x} \tilde{P}(x) \log\Big(1 + \sum_{y} P_{\boldsymbol{w}}(y|x)(e^{z_t f_t(x,y)} - 1)\Big)
\end{aligned}
$$

Table 1: Approximate functions of IS and CD methods.

1. ∇L is bi-Lipschitz: there are two positive constants $\tau_{\max}$ and $\tau_{\min}$ such that for any $\boldsymbol{u}, \boldsymbol{v} \in U$,
$$\tau_{\min}\|\boldsymbol{u} - \boldsymbol{v}\| \leq \|\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})\| \leq \tau_{\max}\|\boldsymbol{u} - \boldsymbol{v}\|. \tag{21}$$

2. Quadratic bound property: there is a constant $K > 0$ such that for any $\boldsymbol{u}, \boldsymbol{v} \in U$,
$$|L(\boldsymbol{u}) - L(\boldsymbol{v}) - \nabla L(\boldsymbol{v})^T(\boldsymbol{u} - \boldsymbol{v})| \leq K\|\boldsymbol{u} - \boldsymbol{v}\|^2. \tag{22}$$

The following theorem proves that (3) satisfies these two conditions.

Theorem 1 *$L(\boldsymbol{w})$ defined in* (3) *satisfies* (21) *and* (22).

The proof is in Section 7.1.

We denote $\boldsymbol{w}^k$ as the point after each iteration of the while loop in Algorithm 1 or 2. Hence from $\boldsymbol{w}^k$ to $\boldsymbol{w}^{k+1}$, n sub-problems are solved. The following theorem establishes our main linear convergence result for IS methods.

Theorem 2 *Consider Algorithm 1 or 2 to minimize a convex and twice differentiable function $L(\boldsymbol{w})$. Assume $L(\boldsymbol{w})$ attains a unique global minimum $\boldsymbol{w}^*$ and $L(\boldsymbol{w})$ satisfies* (21)*-*(22)*. If the algorithm satisfies*

$$
\begin{aligned}
\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\| &\geq \eta\|\nabla L(\boldsymbol{w}^k)\|, && (23) \\
L(\boldsymbol{w}^{k+1}) - L(\boldsymbol{w}^k) &\leq -\nu\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2, && (24)
\end{aligned}
$$

for some positive constants η and ν, then the sequence $\{\boldsymbol{w}^k\}$ generated by the algorithm linearly converges. That is, there is a constant $\mu \in (0,1)$ such that

$$L(\boldsymbol{w}^{k+1}) - L(\boldsymbol{w}^*) \leq (1-\mu)(L(\boldsymbol{w}^k) - L(\boldsymbol{w}^*)), \forall k.$$

The proof is in Section 7.2. Note that this theorem is not restricted to $L(\boldsymbol{w})$ in (3). Next, we show that IS/CD methods discussed in this paper satisfy (23)-(24), so they all possess the linear convergence property.

Theorem 3 *Consider $L(\boldsymbol{w})$ defined in* (3) *and assume $A_t(z_t)$ is exactly minimized in* GIS, IIS, SCGIS, *or* CD. *Then $\{\boldsymbol{w}^k\}$ satisfies* (23)-(24).

The proof is in Section 7.3.

2.4 Solving One-variable Sub-problems

After generating approximate functions, GIS, IIS, SCGIS and CD need to minimize one-variable sub-problems. In general, the approximate function possesses a unique global minimum. We do not discuss some rare situations where this property does not hold (for example, $\min_{z_t}\ A_t^{\mathsf{GIS}}(z_t)$ has an optimal solution $z_t = -\infty$ if $\tilde{P}(f_t) = 0$ and the regularization term is not considered).

Without the regularization term, by $A_t'(z_t) = 0$, GIS and SCGIS both have a simple closed-form solution of the sub-problem:

$$z_t = \frac{1}{f^s} \log \left(\frac{\tilde{P}(f_t)}{\sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y)} \right), \quad \text{where } f^s \equiv \begin{cases} f^{\#} & \text{if } s \text{ is } \mathsf{GIS}, \\ f_t^{\#} & \text{if } s \text{ is } \mathsf{SCGIS}. \end{cases} \tag{25}$$

For IIS, the term $e^{z_t f^{\#}(x,y)}$ in $A_t^{\mathsf{IIS}}(z_t)$ depends on x and y, so it does not have a closed-form solution. CD does not have a closed-form solution either.

With the regularization term, the sub-problems no longer have a closed-form solution. While many optimization methods can be applied, in this section we analyze the complexity of using the Newton method to solve one-variable sub-problems. The Newton method minimizes $A_t^s(z_t)$ by iteratively updating z_t:

$$z_t \leftarrow z_t - A_t^{s\prime}(z_t) / A_t^{s\prime\prime}(z_t), \tag{26}$$

where s indicates an IS or a CD method. This iterative procedure may diverge, so we often need a line search procedure to ensure the function value is decreasing (Fletcher, 1987, p. 47). Due to the many variants of line searches, here we discuss only the cost for finding the Newton direction. The Newton directions of GIS and SCGIS are similar:

$$-\frac{A_t^{s\prime}(z_t)}{A_t^{s\prime\prime}(z_t)} = -\frac{Q_t'(z_t) + e^{z_t f^s} \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y)}{Q_t''(z_t) + f^s e^{z_t f^s} \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y)}, \tag{27}$$

where f^s is defined in (25). For IIS, the Newton direction is:

$$-\frac{A_t^{\mathsf{IIS}\prime}(z_t)}{A_t^{\mathsf{IIS}\prime\prime}(z_t)} = -\frac{Q_t'(z_t) + \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) e^{z_t f^{\#}(x,y)}}{Q_t''(z_t) + \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) f^{\#}(x,y) e^{z_t f^{\#}(x,y)}}. \tag{28}$$

The Newton directions of CD is:

$$-\frac{A_t^{\mathsf{CD}\prime}(z_t)}{A_t^{\mathsf{CD}\prime\prime}(z_t)}, \tag{29}$$

where

$$\begin{aligned} A_t^{\mathsf{CD}\prime}(z_t) &= Q_t'(z_t) + \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x) f_t(x,y), \qquad (30) \\ A_t^{\mathsf{CD}\prime\prime}(z_t) &= Q_t''(z_t) + \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x) f_t(x,y)^2 - \\ & \sum_x \tilde{P}(x) \left(\sum_y P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x) f_t(x,y) \right)^2 . \qquad (31) \end{aligned}$$

Eqs. (27)-(28) can be easily obtained using formulas in Table 1. We show details of deriving (30)-(31) in Section 7.4.

We separate the complexity analysis to two parts. One is on calculating of $P_{\boldsymbol{w}}(y|x)\ \forall x, y$, and the other is on the remaining operations.

For $P_{\boldsymbol{w}}(y|x) = S_{\boldsymbol{w}}(x,y)/T_w(x)$, parallel-update approaches calculate it once every n sub-problems. To get $S_{\boldsymbol{w}}(x,y)\ \forall x, y$, the operation

$$\sum_t w_t f_t(x,y) \quad \forall x, y$$

needs $O(\#\text{nz})$ time. If $XY \leq \#\text{nz}$, the cost for obtaining $P_{\boldsymbol{w}}(y|x),\ \forall x, y$ is $O(\#\text{nz})$, where X and Y are respectively the numbers of contexts and labels.[3] Therefore, on average each sub-problem shares $O(\#\text{nz}/n) = O(\bar{l})$ cost. For sequential-update methods, they expensively update $P_{\boldsymbol{w}}(y|x)$ after every sub-problem. A trick to trade memory for time is to store all $S_{\boldsymbol{w}}(x,y)$ and $T_{\boldsymbol{w}}(x)$, and use (18) and (19). Since $S_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y) = S_{\boldsymbol{w}}(x,y)$, if $f_t(x,y) = 0$, this procedure reduces the number of operations from the $O(\#\text{nz})$ operations to $O(\bar{l})$. However, it needs $O(XY)$ extra spaces to store all $S_{\boldsymbol{w}}(x,y)$ and $T_{\boldsymbol{w}}(x)$. This trick has been used in the SCGIS method (Goodman, 2002).

From (27) and (28), all remaining operations of GIS, IIS, and SCGIS involve the calculation of

$$\sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) (\text{a function of } z_t), \qquad (32)$$

which needs $O(\bar{l})$ under a fixed t. For GIS and SCGIS, since the function of z_t in (32) is independent of x, y, we can calculate and store $\sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y)$ in the first Newton iteration. Therefore, the overall cost (including calculating $P_{\boldsymbol{w}}(y|x)$) is $O(\bar{l})$ for the first Newton iteration and $O(1)$ for each subsequent iteration. For IIS, because $e^{z_t f^{\#}(x,y)}$ in (28) depends on x and y, we need $O(\bar{l})$ for every Newton direction. For CD, it calculates $P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)$ for every z_t, so the cost per Newton direction is $O(\bar{l})$. We summarize the cost for solving sub-problems of GIS, SCGIS, IIS and CD in Table 2.

2.5 Related Work

Our framework for IS methods includes two important components:

1. Approximate $L(\boldsymbol{w} + z_t\boldsymbol{e}_t) - L(\boldsymbol{w})$ or $L(\boldsymbol{w} + \boldsymbol{z}) - L(\boldsymbol{w})$ to obtain functions $A_t(z_t)$.

[3] If $XY > \#\text{nz}$, one can calculate $e^{w_t f_t(x,y)},\ \forall f_t(x,y) \neq 0$ and then the product $\prod_{t: f_t(x,y)\neq 0} e^{w_t f_t(x,y)}$. The complexity is still $O(\#\text{nz})$.

	CD	GIS	SCGIS	IIS
1st Newton direction	$O(\bar{l})$	$O(\bar{l})$	$O(\bar{l})$	$O(\bar{l})$
Each subsequent Newton direction	$O(\bar{l})$	$O(1)$	$O(1)$	$O(\bar{l})$

Table 2: Cost for finding Newton directions if the Newton method is used to minimize $A_t(z_t)$.

2. Sequentially or parallely minimize approximate functions.

Each component has been well discussed in many places. However, ours may be the first to investigate IS methods in detail. Below we discuss some related work.

The closest work to our framework might be Lange et al. (2000) from the statistics community. They discuss "optimization transfer" algorithms which construct $A_t(z_t)$ or $A(\boldsymbol{z})$ satisfying conditions similar to (6)-(7) or (9). However, they do not require one-variable sub-problems, so $A(\boldsymbol{z})$ of a parallel-update method may be non-separable. They discuss that "optimization transfer" algorithms can be traced back to EM (Expectation Maximization). In their paper, the function $A_t(z_t)$ or $A(\boldsymbol{z})$ is called a "surrogate" function or a "majorizing" function. Some also call it an "auxiliary" function. Lange et al. (2000) further discuss several ways to construct $A(\boldsymbol{z})$, where Jensen's inequality used in (13) is one of them. An extension along this line of research is by Zhang et al. (2007).

The concept of sequential- and parallel-update algorithms is well known in many subjects. For example, these algorithms are used in iterative methods for solving linear systems (Jacobi and Gauss-Seidel methods). Some recent machine learning works which mention them include, for example, Collins et al. (2002) and Dudík et al. (2004). Dudík et al. (2004) propose a variant of IS methods for L1-regularized maximum entropy. They consider both sequential- and parallel-update algorithms using certain approximate functions. Their sequential methods greedily choose coordinates minimizing $A_t(z_t)$, while ours in Section 2.1.1 chooses coordinates cyclicly.

Regarding the convergence, if the sub-problem has a closed-form solution like (25), it is easy to apply the result in Lange et al. (2000). However, the case with regularization is more complicated. For example, Dudík et al. (2004) point out that Goodman (2002) does not give a "complete proof of convergence." Note that the strict decrease of function values following conditions (6)-(7) or (9) does not imply the convergence to the optimal function value. In Section 2.3, we prove not only the global convergence but also the linear convergence for a general class of IS/CD methods.

3. Comparison and a New Coordinate Descent Method

Using the framework in Section 2, we compare CD and IS methods in this section. Based on the comparison, we propose a new and fast CD method.

3.1 Comparison of Iterative Scaling and Coordinate Descent Methods

An IS or CD method falls into a place between two extreme designs:

$$\begin{array}{ccc} A_t(z_t) \text{ a loose bound} & & A_t(z_t) \text{ a tight bound} \\ \text{Easy to minimize } A_t(z_t) & \Longleftrightarrow & \text{Hard to minimize } A_t(z_t) \end{array}$$

That is, there is a tradeoff between the tightness to bound the function difference and the hardness to solve the sub-problem. To check how IS and CD methods fit into this explanation, we obtain the following relationship of their approximate functions:

$$\begin{aligned} &A_t^{\mathsf{CD}}(z_t) \le A_t^{\mathsf{SCGIS}}(z_t) \le A_t^{\mathsf{GIS}}(z_t), \\ &A_t^{\mathsf{CD}}(z_t) \le A_t^{\mathsf{IIS}}(z_t) \le A_t^{\mathsf{GIS}}(z_t) \quad \forall\ z_t. \end{aligned} \tag{33}$$

The derivation is in Section 7.5. From (33), CD considers more accurate sub-problems than SCGIS and GIS. However, when solving the sub-problem, from Table 2, CD's each Newton step takes more time. The same situation occurs in comparing IIS and GIS.

The above discussion indicates that while a tight $A_t(z_t)$ can give faster convergence by handling fewer sub-problems, the total time may not be less due to the higher cost of each sub-problem.

3.2 A Fast Coordinate Descent Method

Based on the discussion in Section 3.1, we develop a CD method which is cheaper in solving each sub-problem but still enjoys fast final convergence. This method is modified from Chang et al. (2008), a CD approach for linear SVM. They approximately minimize $A_t^{\mathsf{CD}}(z_t)$ by applying only one Newton iteration. This approach is a truncated Newton method: In the early stage of the coordinate descent method, we roughly minimize $A_t^{\mathsf{CD}}(z_t)$ but in the final stage, one Newton update can quite accurately solve the sub-problem. The Newton direction at $z_t = 0$ is

$$d = -\frac{A_t^{\mathsf{CD}\prime}(0)}{A_t^{\mathsf{CD}\prime\prime}(0)}. \tag{34}$$

We discuss in Section 2.4 that the update rule (26) may not decrease the function value. Hence we need a line search procedure to find $\lambda \ge 0$ such that $z_t = \lambda d$ satisfies the following sufficient decrease condition:

$$A_t^{\mathsf{CD}}(z_t) - A_t^{\mathsf{CD}}(0) = A_t^{\mathsf{CD}}(z_t) \le \gamma z_t A_t^{\mathsf{CD}\prime}(0) \le 0, \tag{35}$$

where γ is a constant in $(0, 1/2)$. Note that $z_t A_t^{\mathsf{CD}\prime}(0)$ is negative under the definition of d in (34). Instead of (35), Grippo and Sciandrone (1999) and Chang et al. (2008) use

$$A_t^{\mathsf{CD}}(z_t) \le -\gamma z_t^2 \tag{36}$$

as the sufficient decrease condition. We prefer (35) as it is scale-invariant. That is, if A_t^{CD} is linearly scaled, then (35) holds under the same γ. In contrast, γ in (36) may need to be changed. To find λ for (35), a simple way is by sequentially checking $\lambda = 1, \beta, \beta^2, \ldots,$ where $\beta \in (0, 1)$. We choose β as 0.5 for experiments. The following theorem proves that the condition (35) can always be satisfied.

Theorem 4 *Given the Newton direction d as in* (34)*. There is $\bar{\lambda} > 0$ such that $z_t = \lambda d$ satisfies* (35) *for all $0 \le \lambda < \bar{\lambda}$.*

Algorithm 3 A fast coordinate descent method for Maxent

```
• Choose $\beta \in (0,1)$ and $\gamma \in (0,1/2)$. Give initial $\boldsymbol{w}$ and calculate $S_{\boldsymbol{w}}(x,y), T_{\boldsymbol{w}}(x), \forall x, y$.
• While $\boldsymbol{w}$ is not optimal
    – For $t = 1, \ldots, n$
        1. Calculate the Newton direction
           $d = -A_t^{\mathsf{CD}\prime}(0)/A_t^{\mathsf{CD}\prime\prime}(0)$
           $= \frac{-\left(\sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) + \frac{w_t}{\sigma^2} - \tilde{P}(f_t)\right)}{\sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y)^2 - \sum_x \tilde{P}(x) \left(\sum_y P_{\boldsymbol{w}}(y|x) f_t(x,y)\right)^2 + \frac{1}{\sigma^2}},$
           where
           $P_{\boldsymbol{w}}(y|x) = \frac{S_{\boldsymbol{w}}(x,y)}{T_{\boldsymbol{w}}(x)}.$
        2. While $\lambda = 1, \beta, \beta^2, \ldots$
           (a) Let $z_t = \lambda d$.
           (b) Calculate
               $A_t^{\mathsf{CD}}(z_t) = Q_t(z_t) + \sum_x \tilde{P}(x) \log \left(1 + \sum_y \frac{S_{\boldsymbol{w}}(x,y)}{T_{\boldsymbol{w}}(x)} (e^{z_t f_t(x,y)} - 1)\right).$
           (c) If $A_t^{\mathsf{CD}}(z_t) \leq \gamma z_t A_t^{\mathsf{CD}\prime}(0)$, then break.
        3. $w_t \leftarrow w_t + z_t$.
        4. Update $S_{\boldsymbol{w}}(x,y)$ and $T_{\boldsymbol{w}}(x)$ $\forall x, y$ by (18)-(19).
```

The proof is in Section 7.6. The new CD procedure is in Algorithm 3. In the rest of this paper, we refer to CD as this new algorithm.

In Section 2.3 we prove the linear convergence of IS/CD methods. In Section 7.7, we use the same framework to prove that Algorithm 3 linearly converges:

Theorem 5 *Algorithm 3 satisfies* (23)-(24) *and linearly converges to the global optimum of* (3).

As evaluating $A_t^{\mathsf{CD}}(z_t)$ via (17)-(19) needs $O(\bar{l})$ time, the line search procedure takes

$$O(\bar{l}) \times (\#\ \text{line search steps}).$$

This causes the cost of solving a sub-problem higher than that of GIS/SCGIS (see Table 2). Fortunately, we show that near the optimum, the line search procedure needs only one step:

Theorem 6 *In a neighborhood of the optimal solution, the Newton direction d defined in* (34) *satisfies the sufficient decrease condition* (35) *with $\lambda = 1$.*

The proof is in Section 7.8. If the line search procedure succeeds at $\lambda = 1$, then the cost for each sub-problem is similar to that of GIS and SCGIS.

Next we show that near the optimum, one Newton direction of CD's tight $A_t^{\mathsf{CD}}(z_t)$ already reduces the objective function $L(\boldsymbol{w})$ more rapidly than directions by exactly minimizing a loose $A_t(z_t)$ of GIS, IIS or SCGIS. Thus Algorithm 3 has faster final convergence than GIS, IIS, or SCGIS.

Theorem 7 *Assume $\boldsymbol{w}^*$ is the global optimum of* (3). *There is an $\epsilon > 0$ such that the following result holds. For any $\boldsymbol{w}$ satisfying $\|\boldsymbol{w} - \boldsymbol{w}^*\| \leq \epsilon$, if we select an index t and generate directions*

$$d = -A_t^{\mathsf{CD}\prime}(0)/A_t^{\mathsf{CD}\prime\prime}(0) \quad \textit{and} \quad d^s = \arg\min_{z_t} A_t^s(z_t), \quad s = \mathsf{GIS}, \mathsf{IIS} \textit{ or } \mathsf{SCGIS}, \tag{37}$$

then

$$\delta_t(d) < \min\left(\delta_t(d^{\mathsf{GIS}}), \delta_t(d^{\mathsf{IIS}}), \delta_t(d^{\mathsf{SCGIS}})\right),$$

where

$$\delta_t(z_t) \equiv L(\boldsymbol{w} + z_t \boldsymbol{e}_t) - L(\boldsymbol{w}).$$

The proof is in Section 7.9. Theorems 6 and 7 show that Algorithm 3 improves upon the traditional CD by approximately solving sub-problems, while still maintaining fast convergence. That is, it attempts to take both advantages of the two designs mentioned in Section 3.1.

3.2.1 Efficient Line Search

We propose a technique to speed up the line search procedure. We derive a function $\bar{A}_t^{\mathsf{CD}}(z_t)$ so that it is cheaper to calculate than $A_t^{\mathsf{CD}}(z_t)$ and satisfies $\bar{A}_t^{\mathsf{CD}}(z_t) \geq A_t^{\mathsf{CD}}(z_t)\ \forall z_t$. Then,

$$\bar{A}_t^{\mathsf{CD}}(z_t) \leq \gamma z_t A_t^{\mathsf{CD}\prime}(0) \tag{38}$$

implies (35), so we can save time by replacing step 2 of Algorithm 3 with

```
2'. While $\lambda = 1, \beta, \beta^2, \ldots$
    (a) Let $z_t = \lambda d$.
    (b) Calculate $\bar{A}_t^{\mathsf{CD}}(z_t)$.
    (c) If $\bar{A}_t^{\mathsf{CD}}(z_t) \leq \gamma z_t A_t^{\mathsf{CD}\prime}(0)$, then break.
    (d) Calculate $A_t^{\mathsf{CD}}(z_t)$.
    (e) If $A_t^{\mathsf{CD}}(z_t) \leq \gamma z_t A_t^{\mathsf{CD}\prime}(0)$, then break.
```

We assume non-negative feature values and obtain

$$\bar{A}_t^{\mathsf{CD}}(z_t) \equiv Q_t(z_t) + \tilde{P}_t \log\left(1 + \frac{e^{z_t f_t^{\#}} - 1}{f_t^{\#} \tilde{P}_t} \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y)\right), \tag{39}$$

where $f_t^{\#}$ is defined in (14),

$$\tilde{P}_t \equiv \sum_{\Omega_t} \tilde{P}(x), \quad \text{and} \quad \Omega_t \equiv \{x : \exists y \text{ such that } f_t(x,y) \neq 0\}. \tag{40}$$

The derivation is in Section 7.10. Because

$$\sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y), \ t = 1, \ldots, n \tag{41}$$

are available from finding $A_t^{\mathsf{CD}\prime}(0)$, getting $\bar{A}_t^{\mathsf{CD}}(z_t)$ and checking (38) take only $O(1)$, smaller than $O(\bar{l})$ for (35). Using $\log \alpha \leq \alpha - 1 \ \forall \alpha > 0$, it is easy to see that

$$\bar{A}_t^{\mathsf{CD}}(z_t) \leq A_t^{\mathsf{SCGIS}}(z_t), \ \forall z_t.$$

Therefore, we can simply replace $A_t^{\mathsf{SCGIS}}(z_t)$ of the SCGIS method with $\bar{A}_t^{\mathsf{CD}}(z_t)$ to have a new IS method.

4. Implementation Issues

In this section we analyze some implementation issues of IS and CD methods.

4.1 Row Versus Column Format

In many Maxent applications, data are sparse with few nonzero $f_t(x,y)$. We store such data by a sparse matrix. Among many ways to implement sparse matrices, two common ones are "row format" and "column format." For the row format, each (x,y) corresponds to a list of nonzero $f_t(x,y)$, while for the column format, each feature t is associated with a list of (x,y). The loop to access data in the row format is $(x,y) \rightarrow t$, while for the column format it is $t \rightarrow (x,y)$. By $(x,y) \rightarrow t$ we mean that the outer loop goes through (x,y) values and for each (x,y), there is an inner loop for a list of feature values. For sequential-update algorithms such as SCGIS and CD, as we need to maintain $S_{\boldsymbol{w}}(x,y) \ \forall x,y$ via (18) after solving the t-th sub-problem, an easy access of t's corresponding (x,y) elements is essential. Therefore, the column format is more suitable. In contrast, parallel-update methods can use either row or column formats. For GIS, we can store all n elements of (41) before solving n sub-problems by (25) or (27). The calculation of (41) can be done by using the row format and a loop of $(x,y) \rightarrow t$. For IIS, an implementation by the row format is more complicated due to the $e^{z_t f^{\#}(x,y)}$ term in $A_t^{\mathsf{IIS}}(z_t)$. Take the Newton method to solve the sub-problem as an example. We can calculate and store (28) for all $t = 1, \ldots, n$ by a loop of $(x,y) \rightarrow t$. That is, n Newton directions are obtained together before conducting n updates.

4.2 Memory Requirement

For sequential-update methods, to save the computational time of calculating $P_{\boldsymbol{w}}(y|x)$, we use (18)-(19), so $S_{\boldsymbol{w}}(x,y) \ \forall x,y$ must be stored. Therefore, $O(XY)$ storage is needed. For parallel-update methods, they also need $O(XY)$ spaces if using the column format: To calculate $e^{\sum_t w_t f_t(x,y)} \ \forall x,y$ via a loop of $t \rightarrow (x,y)$, we need $O(XY)$ positions to store $\sum_t w_t f_t(x,y) \ \forall x,y$. In contrast, if using the row format, the loop is $x \rightarrow y \rightarrow t$, so for each fixed x, we need only $O(Y)$ spaces to store $S(x,y) \ \forall y$ and then obtain $T_{\boldsymbol{w}}(x)$. This advantage makes the parallel update a viable approach if Y (the number of labels) is very large.

Data set	X	Y	n	#nz
CoNLL2000-P	197,979	44	168,674	48,030,163
CoNLL2000-C	197,252	22	273,680	53,396,844
BROWN	935,137	185	626,726	601,216,661

Table 3: Statistics of NLP data (0/1 features). X: number of contexts, Y: number of class labels, n: number of features, and #nz: number of total non-zero feature values; see (4).

4.3 Number of exp and log Operations

Many exp/log operations are needed in training a Maxent model. On most computers, exp/log operations are much more expensive than multiplications/divisions. It is important to analyze the number of exp/log operations in IS and CD methods.

We first discuss the number of exp operations. A simple check of (27)-(31) shows that the numbers are the same as those in Table 2. IIS and CD need $O(\bar{l})$ exp operations for every Newton direction because they calculate $e^{z_t f^{\#}(x,y)}$ in (28) and $e^{z_t f_t(x,y)}$ in (17), respectively. CD via Algorithm 3 takes only one Newton iteration, but each line search step also needs $O(\bar{l})$ exp operations. If feature values are binary, $e^{z_t f_t(x,y)}$ in (17) becomes e^{z_t}, a value independent of x, y. Thus the number of exp operations is significantly reduced from $O(\bar{l})$ to $O(1)$. This property implies that Algorithm 3 is more efficient if data are binary valued. In Section 5, we will confirm this result through experiments.

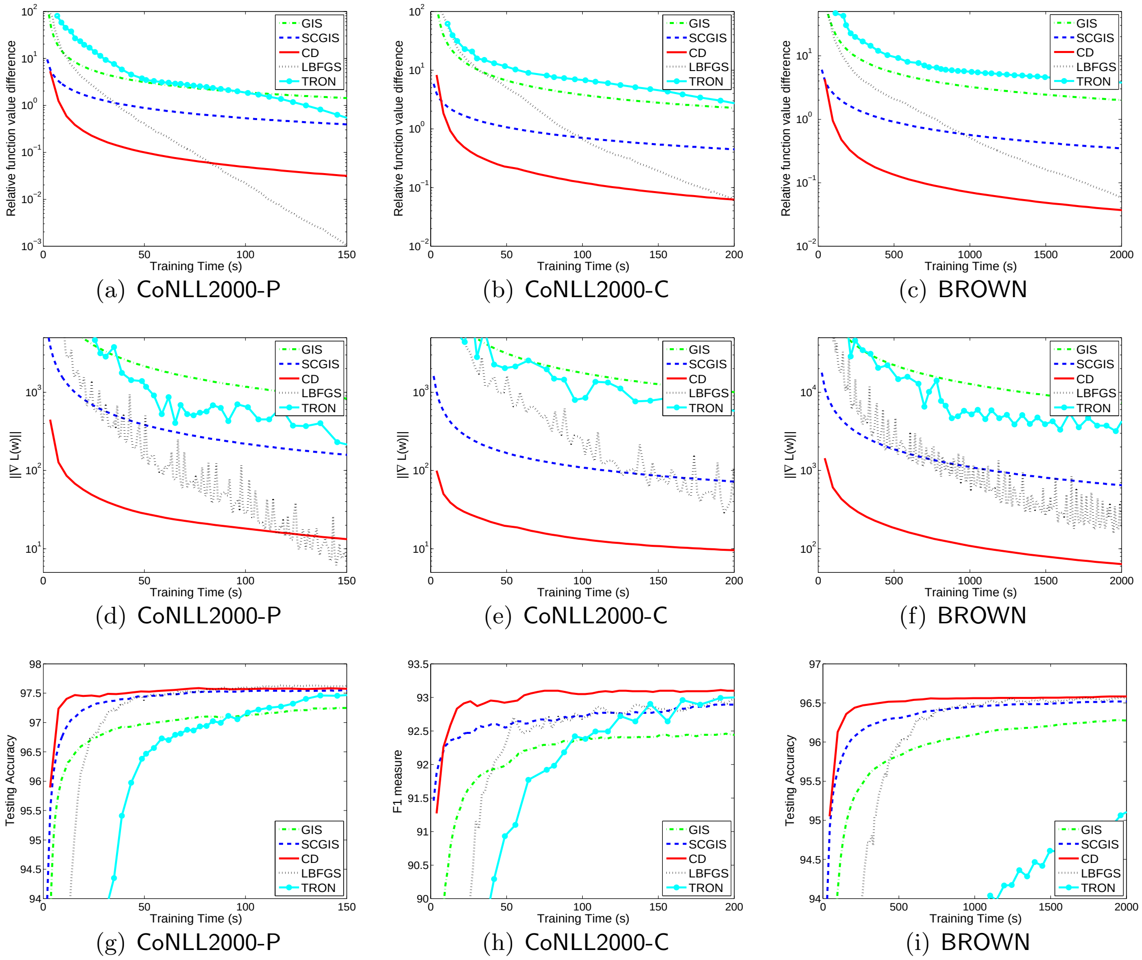
Regarding log operations, GIS, IIS and SCGIS need none as they remove the log function in $A_t(z_t)$. CD via Algorithm 3 keeps log in $A_t^{\mathsf{CD}}(z_t)$ and requires $O(|\Omega_t|)$ log operations at each line search step, where Ω_t is defined in (40).

4.4 Permutation of Indices in Solving Sub-problems

For sequential-update methods, one does not have to follow a cyclic way to update $w_1, \ldots, w_n$. Chang et al. (2008) report that in their CD method, a permutation of $\{1, \ldots, n\}$ as the order for solving n sub-problems leads to faster convergence. For sequential-update IS methods adopting this strategy, the linear convergence in Theorem 2 still holds.

5. Experiments

In this section, we compare IS/CD methods to reconfirm properties discussed in earlier sections. We consider two types of data for NLP (Natural Language Processing) applications. One is Maxent for 0/1-featured data and the other is Maxent (logistic regression) for document data with non-negative real-valued features. Programs used for experiments in this paper are online available at
`http://www.csie.ntu.edu.tw/~cjlin/liblinear/exp.html`.



(a) CoNLL2000-P (b) CoNLL2000-C (c) BROWN

(d) CoNLL2000-P (e) CoNLL2000-C (f) BROWN

(g) CoNLL2000-P (h) CoNLL2000-C (i) BROWN

Figure 2: Results on 0/1-featured data. The first row shows time versus the relative function difference (42). The second and third rows show $\|\nabla L(\boldsymbol{w})\|$ and testing performances along time, respectively. Time is in seconds.

5.1 Maxent for 0/1-featured Data in NLP

We apply Maxent models to part of speech (POS) tagging and chunking tasks. In POS tagging, we mark a POS tag to the word in a text based on both its definition and context. In a chunking task, we divide a text into syntactically correlated parts of words. That is, given words in a sentence annotated with POS tags, we label each word with a chunk tag. Other learning models such as CRF (Conditional Random Fields) may outperform Maxent for these NLP applications. However, we do not consider other learning models as the focus of this paper is to study IS methods for Maxent.

We use CoNLL2000 shared task data[4] for chunking and POS tagging, and BROWN corpus[5] for POS tagging. CoNLL2000-P indicates CoNLL2000 for POS tagging, and CoNLL2000-C means CoNLL2000 for chunking. CoNLL2000 data consist of Sections 15-18 of the Wall Street Journal corpus as training data and Section 20 as testing data. For the BROWN corpus, we randomly select four-fifth articles for training and use the rest for testing. We omit the stylistic tag modifiers "fw," "tl," "nc,"and "hl," so the number of labels is reduced from 472 to 185. Our implementation is built upon the OpenNLP package (Baldridge et al., 2001). We use the default setting of OpenNLP to extract binary features (0/1 values) suggested by Ratnaparkhi (1998). The OpenNLP implementation assumes that each feature index t corresponds to a unique label y. In prediction, we approximately maximize the probability of tag sequences to the word sequences by a beam search (Ratnaparkhi, 1998). Table 3 lists the statistics of data sets.

We implement the following methods for comparisons.

1. GIS and SCGIS: To minimize $A_t(z_t)$, we run Newton updates (without line search) until $|A_t'(z_t)| \leq 10^{-5}$. We can afford many Newton iterations because, according to Table 2, each Newton direction costs only $O(1)$ time.

2. CD: the coordinate descent method proposed in Section 3.2.

3. LBFGS: a limited memory quasi Newton method for general unconstrained optimization problems (Liu and Nocedal, 1989).

4. TRON: a trust region Newton method for logistic regression (Lin et al., 2008). We extend the method for Maxent.

We consider LBFGS as Malouf (2002) reports that it is better than other approaches including GIS and IIS. Lin et al. (2008) show that TRON is faster than LBFGS for document classification, so we include TRON for comparison. We exclude IIS because of its higher cost per Newton direction than GIS/SCGIS (see Table 2). Indeed Malouf (2002) reports that GIS outperforms IIS. Our implementation of all methods takes the property of 0/1 features. We use the regularization parameter $\sigma^2 = 10$ as under this value Maxent models achieve good testing performances. We set $\beta = 0.5$ and $\gamma = 0.001$ for the line search procedure (35) in CD. The initial $\boldsymbol{w}$ of all methods is $\mathbf{0}$.

We begin at checking time versus the relative difference of the function value to the optimum:

$$\frac{L(\boldsymbol{w}) - L(\boldsymbol{w}^*)}{L(\boldsymbol{w}^*)}, \tag{42}$$

where $\boldsymbol{w}^*$ is the optimal solution of (3). As $\boldsymbol{w}^*$ is not available, we obtain a reference point satisfying $\|\nabla L(\boldsymbol{w})\| \leq 0.01$. Results are in the first row of Figure 2. Next, we check these methods' gradient values. As $\|\nabla L(\boldsymbol{w})\| = 0$ implies that $\boldsymbol{w}$ is the global minimum, usually $\|\nabla L(\boldsymbol{w})\|$ is used in a stopping condition. The second row of Figure 2 shows time versus $\|\nabla L(\boldsymbol{w})\|$. We are also interested in the time needed to achieve a reasonable testing result. We measure the performance of POS tagging by accuracy and chunking by F1 measure.

[4] Data can be found at `http://www.cnts.ua.ac.be/conll2000/chunking`.

[5] Corpus can be found at `http://www.nltk.org`.

Problem	l	n	#nz	σ^2
astro-physic	62,369	99,757	4,834,550	$8l$
yahoo-japan	176,203	832,026	23,506,415	$4l$
rcv1	677,399	47,236	49,556,258	$8l$

Table 4: Statistics of document data (real-valued features). l: number of instances, n: number of features, #nz: number of total non-zero feature values, and σ^2: best regularization parameter from five-fold cross validation.

The third row of Figure 2 presents the testing accuracy/F1 versus training time. Note that (42) and $\|\nabla L(\boldsymbol{w})\|$ in Figure 2 are both log scaled.

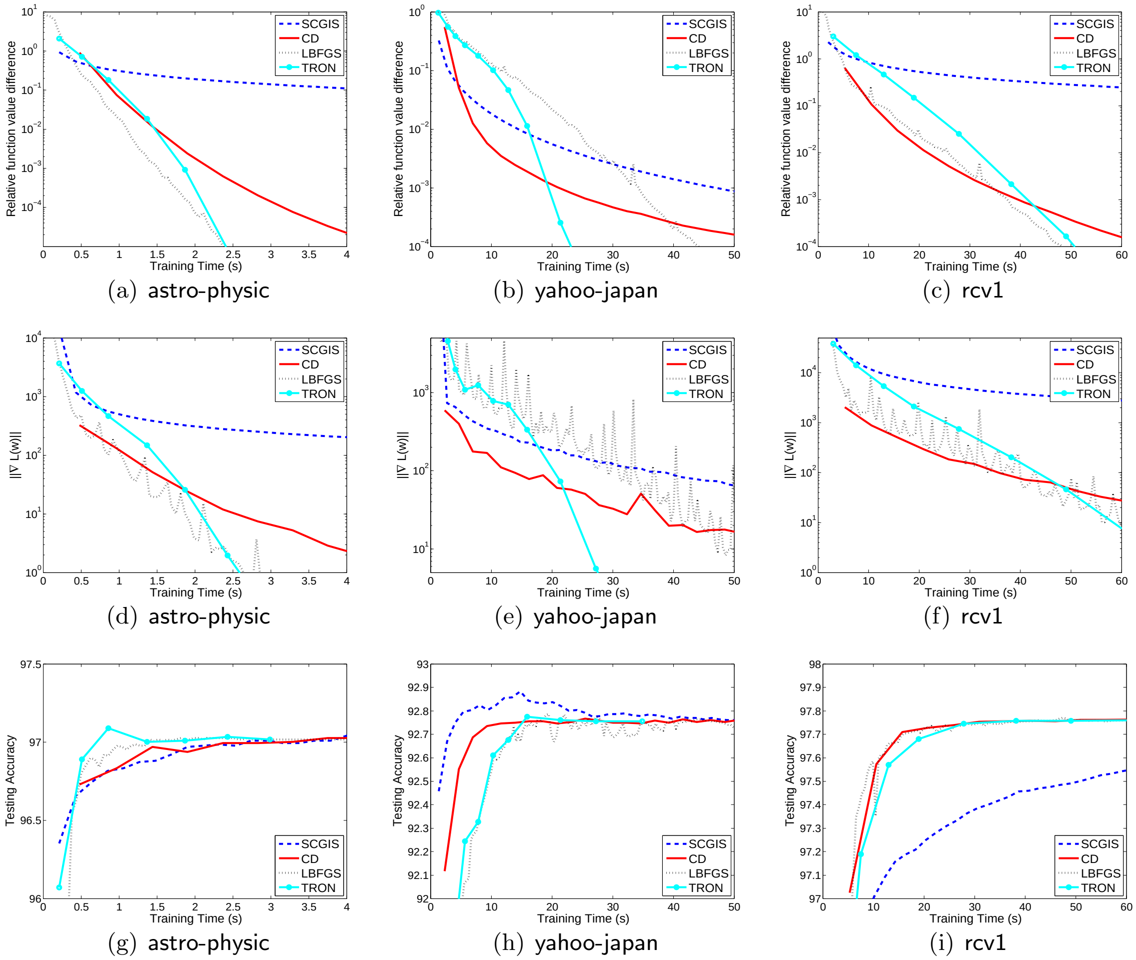
We give some observations from Figure 2. Among the three IS/CD methods compared, the new CD approach discussed in Section 3.2 is the fastest. SCGIS comes the second, while GIS is the last. This result is consistent with the tightness of their approximate functions; see (33). Regarding IS/CD methods versus LBFGS/TRON, the three IS/CD methods more quickly decrease the function value in the beginning, but LBFGS has faster final convergence. In fact, if we draw figures with longer training time, TRON's final convergence is the fastest. This result is reasonable as LBFGS and TRON respectively have superlinear and quadratic convergence, higher than the linear rate proved in Theorem 2 for IS methods. The choice of methods thus relies on whether one prefers getting a reasonable model quickly (IS/CD methods) or accurately minimizing the function (LBFGS/TRON). Practically CD/IS may be more useful as they reach the final testing accuracy rapidly. Finally, we compare LBFGS and TRON. Surprisingly, LBFGS outperforms TRON, a result opposite to that in Lin et al. (2008). We do not have a clear explanation yet. A difference is that Lin et al. (2008) deal with document data of real-valued features, but here we have 0/1-featured NLP applications. Therefore, one should always be careful that for the same approaches, observations made on one type of data may not extend to another.

In Section 4, we discussed a strategy of permuting n sub-problems to speed up the convergence of sequential-update IS methods. However, in training Maxent models for 0/1-featured NLP data, with/without permutation gives similar performances. We find that this strategy tends to work better if features are related. Hence we suspect that features used in POS tagging or chunking tasks are less correlated than those in documents and the order of sub-problems is not very important.

5.2 Maxent (Logistic Regression) for Document Classification

In this section, we experiment with logistic regression on document data with non-negative real-valued features. Chang et al. (2008) report that their CD method is very efficient for linear SVM, but is slightly less effective for logistic regression. They attribute the reason to that logistic regression requires expensive exp/log operations. In Section 4, we show that for 0/1 features, the number of IS methods' exp operations is smaller. Experiments here help to check if IS/CD methods are more suitable for 0/1 features than real values.

Logistic regression is a special case of maximum entropy with two labels +1 and −1. Consider training data $\{\bar{\boldsymbol{x}}_i, \bar{y}_i\}_{i=1}^l$, $\bar{\boldsymbol{x}}_i \in R^n$, $\bar{y}_i = \{1, -1\}$. Assume $\bar{x}_{it} \geq 0$, $\forall i, t$. We set



(a) astro-physic (b) yahoo-japan (c) rcv1

(d) astro-physic (e) yahoo-japan (f) rcv1

(g) astro-physic (h) yahoo-japan (i) rcv1

Figure 3: Results on real-valued document data. The first row shows time versus the relative function difference (42). The second and third rows show $\|\nabla L(\boldsymbol{w})\|$ and testing performances along time, respectively. Time is in seconds.

the feature $f_t(x_i, y)$ as

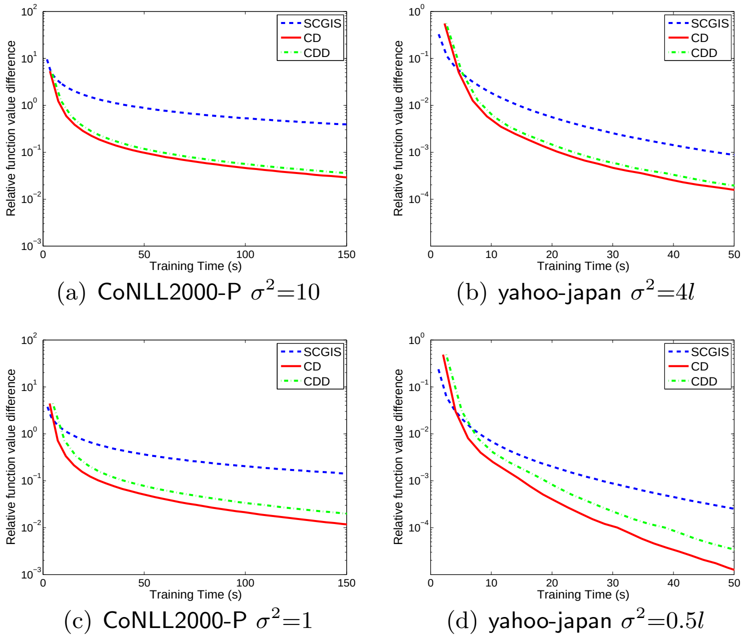
$$f_t(x_i, y) = \begin{cases} \bar{x}_{it} & \text{if } y = 1, \\ 0 & \text{if } y = -1, \end{cases}$$

where x_i denotes the index of the i-th training instance $\bar{\boldsymbol{x}}_i$. Then

$$S_{\boldsymbol{w}}(x_i, y) = e^{\sum_t w_t f_t(x_i, y)} = \begin{cases} e^{\boldsymbol{w}^T \bar{\boldsymbol{x}}_i} & \text{if } y = 1, \\ 1 & \text{if } y = -1, \end{cases}$$

and

$$P_{\boldsymbol{w}}(y|x_i) = \frac{S_{\boldsymbol{w}}(x_i, y)}{T_{\boldsymbol{w}}(x_i)} = \frac{1}{1 + e^{-y\boldsymbol{w}^T \bar{\boldsymbol{x}}_i}}. \tag{43}$$



(a) CoNLL2000-P σ^2=10

(b) yahoo-japan $\sigma^2=4l$

(c) CoNLL2000-P σ^2=1

(d) yahoo-japan $\sigma^2=0.5l$

Figure 4: This figure shows the effect of using (38) to do line search. The first and second rows show time versus the relative function difference with different σ^2. CDD indicates the CD method without using (38). Time is in seconds.

From (2) and (43),

$$L(\boldsymbol{w}) = \frac{1}{2\sigma^2}\sum_t w_t^2 + \frac{1}{l}\sum_i \log\left(1 + e^{-\bar{y}_i \boldsymbol{w}^T \bar{\boldsymbol{x}}_i}\right)$$

is the common form of regularized logistic regression. We give approximate functions of IS/CD methods in Section 7.11.

We compare the same methods: SCGIS, CD, LBFGS, and TRON. GIS is not included because of its slow convergence shown in Section 5.1. Our implementations are based on sources used in Chang et al. (2008).[6] We select three data sets considered in Chang et al. (2008). Each instance has been normalized to $\|\bar{\boldsymbol{x}}_i\| = 1$. Data statistics and σ^2 for each problem are in Table 4. We set $\beta = 0.5$ and $\gamma = 0.01$ for the line search procedure (35) in CD. Figure 3 shows the results of the relative function difference to the optimum, the gradient $\|\nabla L(\boldsymbol{w})\|$, and the testing accuracy.

From Figure 3, the relation between the two IS/CD methods is similar to that in Figure 2, where CD is faster than SCGIS. However, in contrast to Figure 2, here TRON/LBFGS may surpass IS/CD in an earlier stage. Some preliminary analysis on the cost per iteration seems to indicate that IS/CD methods are more efficient on 0/1-featured data due to a smaller number of exp operations, but more experiments/data are needed to draw definitive conclusions.

In Figure 3, TRON is only similarly to or moderately better than LBFGS, but Lin et al. (2008) show that TRON is much better. The only difference between their setting and ours

[6] Source can be found at `http://www.csie.ntu.edu.tw/~cjlin/liblinear/exp.html`.

is that Lin et al. (2008) add one feature to each data instance. That is, they modify $\bar{\boldsymbol{x}}_i$ to $\left[\begin{smallmatrix}\bar{\boldsymbol{x}}_i \\ 1\end{smallmatrix}\right]$, so weights of Maxent become $\left[\begin{smallmatrix}\boldsymbol{w} \\ b\end{smallmatrix}\right]$, where b is called the bias term. It is surprising that this difference affects LBFGS' performance that much.

6. Discussion and Conclusions

In (38), we propose a way to speed up the line search procedure of Algorithm 3. Figure 4 shows how effective this trick is by varying the value of σ^2. Clearly, the trick is more useful if σ^2 is small. In this situation, the function $L(\boldsymbol{w})$ is well conditioned (as it is closer to a quadratic function $\sum_t w_t^2$). Hence (38) more easily holds at $\lambda = 1$. Then the line search procedure costs only $O(1)$ time. However, a too small σ^2 may downgrade the testing accuracy. For example, the final accuracy for yahoo-japan is 92.75% with $\sigma^2 = 4l$, but is 92.31% with $\sigma^2 = 0.5l$.

Some work has concluded that approaches like LBFGS or nonlinear conjugate gradient are better than IS methods for training Maxent (e.g., Malouf, 2002; Daumé, 2004). However, experiments in this paper show that comparison results may vary under different circumstances. For example, comparison results can be affected by:

1. Data of the target application. IS/CD methods seem to perform better if features are 0/1 and if implementations have taken this property.

2. The IS method being compared. Our experiments indicate that GIS is inferior to many methods, but other IS/CD methods like SCGIS or CD (Algorithm 3) are more competitive.

In summary, we create a general framework for iterative scaling and coordinate descent methods for maximum entropy. Based on this framework, we discuss the convergence, computational complexity, and other properties of IS/CD methods. We further develop a new coordinate decent method for Maxent. It is more efficient than existing iterative scaling methods.

7. Proofs and Derivations

We define 1-norm and 2-norm of a vector $\boldsymbol{w} \in R^n$:

$$\|\boldsymbol{w}\|_1 \equiv \sum_{t=1}^{n} |w_t|, \quad \|\boldsymbol{w}\|_2 \equiv \sqrt{\sum_{t=1}^{n} w_t^2}.$$

The following inequality is useful in our proofs.

$$\|\boldsymbol{w}\|_2 \leq \|\boldsymbol{w}\|_1 \leq \sqrt{n}\|\boldsymbol{w}\|_2, \quad \forall \boldsymbol{w} \in R^n. \tag{44}$$

Subsequently we simplify $\|\boldsymbol{w}\|_2$ to $\|\boldsymbol{w}\|$.

7.1 Proof of Theorem 1

Due to the regularization term $\frac{1}{2\sigma^2}\boldsymbol{w}^T\boldsymbol{w}$, one can prove that the set U defined in (20) is bounded; see, for example, Theorem 1 of Lin et al. (2008). As $\nabla^2 L(\boldsymbol{w})$ is continuous in the bounded set U, the following $\tau_{\max}$ and $\tau_{\min}$ exist:

$$\tau_{\max} \equiv \max_{\boldsymbol{w} \in U} \lambda_{\max}(\nabla^2 L(\boldsymbol{w})) \quad \text{and} \quad \tau_{\min} \equiv \min_{\boldsymbol{w} \in U} \lambda_{\min}(\nabla^2 L(\boldsymbol{w})), \tag{45}$$

where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ mean the largest and the smallest eigenvalues of a matrix, respectively. To show that $\tau_{\max}$ and $\tau_{\min}$ are positive, it is sufficient to prove $\tau_{\min} > 0$. As $\nabla^2 L(\boldsymbol{w})$ is I/σ^2 plus a positive semi-definite matrix, it is easy to see $\tau_{\min} \geq 1/(\sigma^2) > 0$.

To prove (21), we apply the multi-dimensional Mean-Value Theorem (Apostol, 1974, Theorem 12.9) to $\nabla L(\boldsymbol{w})$. If $\boldsymbol{u}, \boldsymbol{v} \in R^n$, then for any $\boldsymbol{a} \in R^n$, there is a $\boldsymbol{c} = \alpha \boldsymbol{u} + (1-\alpha)\boldsymbol{v}$ with $0 \leq \alpha \leq 1$ such that

$$\boldsymbol{a}^T(\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})) = \boldsymbol{a}^T \nabla^2 L(\boldsymbol{c})(\boldsymbol{u} - \boldsymbol{v}). \tag{46}$$

Set

$$\boldsymbol{a} = \boldsymbol{u} - \boldsymbol{v}.$$

Then for any $\boldsymbol{u}, \boldsymbol{v} \in U$, there is a point $\boldsymbol{c}$ such that

$$(\boldsymbol{u} - \boldsymbol{v})^T(\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})) = (\boldsymbol{u} - \boldsymbol{v})^T \nabla^2 L(\boldsymbol{c})(\boldsymbol{u} - \boldsymbol{v}). \tag{47}$$

Since U is a convex set from the convexity of $L(\boldsymbol{w})$, $\boldsymbol{c} \in U$. With (45) and (47),

$$\|\boldsymbol{u} - \boldsymbol{v}\| \|\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})\| \geq (\boldsymbol{u} - \boldsymbol{v})^T(\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})) \geq \tau_{\min} \|\boldsymbol{u} - \boldsymbol{v}\|^2.$$

Hence

$$\|\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})\| \geq \tau_{\min} \|\boldsymbol{u} - \boldsymbol{v}\|. \tag{48}$$

By applying (46) again with $\boldsymbol{a} = \nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})$,

$$\begin{aligned} \|\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})\|^2 \leq& \|\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})\| \|\nabla^2 L(\boldsymbol{c})(\boldsymbol{u} - \boldsymbol{v})\| \\ \leq& \|\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})\| \|\boldsymbol{u} - \boldsymbol{v}\| \tau_{\max}. \end{aligned}$$

Therefore,

$$\|\nabla L(\boldsymbol{u}) - \nabla L(\boldsymbol{v})\| \leq \tau_{\max} \|\boldsymbol{u} - \boldsymbol{v}\|. \tag{49}$$

Then (21) follows from (48) and (49)

To prove the second property (22), we write the Taylor expansion of $L(\boldsymbol{u})$:

$$L(\boldsymbol{u}) = L(\boldsymbol{v}) + \nabla L(\boldsymbol{v})^T(\boldsymbol{u} - \boldsymbol{v}) + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{v})^T \nabla^2 L(\boldsymbol{c})(\boldsymbol{u} - \boldsymbol{v}),$$

where $\boldsymbol{c} \in U$. With (45), we have

$$\frac{\tau_{\min}}{2}\|\boldsymbol{u} - \boldsymbol{v}\|^2 \leq L(\boldsymbol{u}) - L(\boldsymbol{v}) - \nabla L(\boldsymbol{v})^T(\boldsymbol{u} - \boldsymbol{v}) \leq \frac{\tau_{\max}}{2}\|\boldsymbol{u} - \boldsymbol{v}\|^2.$$

Since $\tau_{\max} \geq \tau_{\min} > 0$, L satisfies (22) by choosing $K = \tau_{\max}/2$.

7.2 Proof of Theorem 2

The following proof is modified from Chang et al. (2008). Since $L(\boldsymbol{w})$ is convex and $\boldsymbol{w}^*$ is the unique solution, the optimality condition shows that

$$\nabla L(\boldsymbol{w}^*) = \boldsymbol{0}. \tag{50}$$

From (21) and (50),

$$\|\nabla L(\boldsymbol{w}^k)\| \geq \tau_{\min} \|\boldsymbol{w}^k - \boldsymbol{w}^*\|. \tag{51}$$

With (23) and (51),

$$\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\| \geq \eta\tau_{\min} \|\boldsymbol{w}^k - \boldsymbol{w}^*\|. \tag{52}$$

From (24) and (52),

$$L(\boldsymbol{w}^k) - L(\boldsymbol{w}^{k+1}) \geq \nu\eta^2\tau_{\min}^2 \|\boldsymbol{w}^k - \boldsymbol{w}^*\|^2. \tag{53}$$

Combining (22) and (50),

$$L(\boldsymbol{w}^k) - L(\boldsymbol{w}^*) \leq K \|\boldsymbol{w}^k - \boldsymbol{w}^*\|^2. \tag{54}$$

Using (53) and (54),

$$L(\boldsymbol{w}^k) - L(\boldsymbol{w}^{k+1}) \geq \frac{\nu\eta^2\tau_{\min}^2}{K} \Big(L(\boldsymbol{w}^k) - L(\boldsymbol{w}^*) \Big).$$

This is equivalent to

$$\Big(L(\boldsymbol{w}^k) - L(\boldsymbol{w}^*) \Big) + \Big(L(\boldsymbol{w}^*) - L(\boldsymbol{w}^{k+1}) \Big) \geq \frac{\nu\eta^2\tau_{\min}^2}{K} \Big(L(\boldsymbol{w}^k) - L(\boldsymbol{w}^*) \Big).$$

Finally, we have

$$L(\boldsymbol{w}^{k+1}) - L(\boldsymbol{w}^*) \leq \left(1 - \frac{\nu\eta^2\tau_{\min}^2}{K}\right) \Big(L(\boldsymbol{w}^k) - L(\boldsymbol{w}^*) \Big). \tag{55}$$

Let $\mu \equiv \nu\eta^2\tau_{\min}^2/K$. As all constants are positive, $\mu > 0$. If $\mu > 1$, $L(\boldsymbol{w}^k) > L(\boldsymbol{w}^*)$ implies that $L(\boldsymbol{w}^{k+1}) < L(\boldsymbol{w}^*)$, a contradiction to the definition of $L(\boldsymbol{w}^*)$. Thus we have either $\mu \in (0,1)$ or $\mu = 1$, which suggests we get the optimum in finite steps.

7.3 Proof of Theorem 3

We prove the result for GIS and IIS first. Let $\bar{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} A^s(\boldsymbol{z})$, where s indicates GIS or IIS method.[7] From the definition of $A^s(\boldsymbol{z})$ and its convexity,[8]

$$\nabla A^s(\boldsymbol{0}) = \nabla L(\boldsymbol{w}^k) \text{ and } \nabla A^s(\bar{\boldsymbol{z}}) = \boldsymbol{0}.^9 \tag{56}$$

[7] The existence of $\bar{z}$ follows from that U is bounded. See the explanation in the beginning of Section 7.1.

[8] It is easy to see that all $A_t(z_t)$ in Table 1 are strictly convex.

[9] Because $\nabla_t L(\boldsymbol{w}^k) = A_t^{\mathsf{CD}\prime}(0)$, we can easily obtain $\nabla A^s(\boldsymbol{0}) = \nabla L(\boldsymbol{w}^k)$ by checking $A_t^{s\prime}(0) = A_t^{\mathsf{CD}\prime}(0)$, where s is GIS or IIS. See formulas in Table 1.

Note that $\nabla A^s(\boldsymbol{z})$ is the gradient with respect to $\boldsymbol{z}$, but $\nabla L(\boldsymbol{w})$ is the gradient with respect to $\boldsymbol{w}$. Since U is bounded, the set $\{(\boldsymbol{w}, \boldsymbol{z}) \mid \boldsymbol{w} \in U \text{ and } \boldsymbol{w} + \boldsymbol{z} \in U\}$ is also bounded. Thus we have that

$$\max_{\boldsymbol{w} \in U} \max_{\boldsymbol{z}: \boldsymbol{w}+\boldsymbol{z} \in U} \lambda_{\max}\left(\nabla^2 A^s(\boldsymbol{z})\right)$$

is bounded by a constant K. Here $\lambda_{\max}(\cdot)$ means the largest eigenvalue of a matrix. To prove (23), we use

$$\begin{aligned} &\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\| = \|\bar{\boldsymbol{z}} - \boldsymbol{0}\| \\ \geq &\frac{1}{K}\|\nabla A^s(\bar{\boldsymbol{z}}) - \nabla A^s(\boldsymbol{0})\| = \frac{1}{K}\|\nabla A^s(\boldsymbol{0})\| = \frac{1}{K}\|\nabla L(\boldsymbol{w}^k)\|, \end{aligned} \tag{57}$$

where the inequality is from the same derivation for (49) in Theorem 1. The last two equalities follow from (56).

Next, we prove (24). By (56) and the fact that the minimal eigenvalue of $\nabla^2 A^s(\boldsymbol{z})$ is greater than or equal to $1/(\sigma^2)$, we have

$$A^s(\boldsymbol{0}) \geq A^s(\bar{\boldsymbol{z}}) - \nabla A^s(\bar{\boldsymbol{z}})^T \bar{\boldsymbol{z}} + \frac{1}{2\sigma^2}\bar{\boldsymbol{z}}^T\bar{\boldsymbol{z}} = A^s(\bar{\boldsymbol{z}}) + \frac{1}{2\sigma^2}\bar{\boldsymbol{z}}^T\bar{\boldsymbol{z}}. \tag{58}$$

From (9) and (58),

$$L(\boldsymbol{w}^k) - L(\boldsymbol{w}^{k+1}) = L(\boldsymbol{w}^k) - L(\boldsymbol{w}^k + \bar{\boldsymbol{z}}) \geq A^s(\boldsymbol{0}) - A^s(\bar{\boldsymbol{z}}) \geq \frac{1}{2\sigma^2}\bar{\boldsymbol{z}}^T\bar{\boldsymbol{z}} = \frac{1}{2\sigma^2}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2.$$

Let $\nu = 1/(2\sigma^2)$ and we obtain (24).

We then prove results for SCGIS and CD. For the convenience, we define some notation. A sequential algorithm starts from an initial point $\boldsymbol{w}^0$, and produces a sequence $\{\boldsymbol{w}^k\}_{k=0}^{\infty}$. At each iteration, $\boldsymbol{w}^{k+1}$ is constructed by sequentially updating each component of $\boldsymbol{w}^k$. This process generates vectors $\boldsymbol{w}^{k,t} \in R^n$, $t = 1, \ldots, n$, such that $\boldsymbol{w}^{k,1} = \boldsymbol{w}^k$, $\boldsymbol{w}^{k,n+1} = \boldsymbol{w}^{k+1}$, and

$$\boldsymbol{w}^{k,t} = [w_1^{k+1}, \ldots, w_{t-1}^{k+1}, w_t^k, \ldots, w_n^k]^T \text{ for } t = 2, \ldots, n.$$

By an argument similar to (57) and (58), we can prove that the one-variable function $A_t^s(z_t)$, where s is SCGIS or CD, satisfies

$$|w_t^{k,t+1} - w_t^{k,t}| \geq \bar{\eta}|A_t^{s\prime}(0)| = \bar{\eta}|\nabla L(\boldsymbol{w}^{k,t})_t| \text{ and} \tag{59}$$

$$L(\boldsymbol{w}^{k,t}) - L(\boldsymbol{w}^{k,t+1}) \geq \frac{1}{2\sigma^2}|w_t^{k,t+1} - w_t^{k,t}|^2. \tag{60}$$

Note that $\bar{\eta} > 0$ is a positive constant. To prove (23), taking the summation of (59) from $t = 1$ to n,

$$\begin{aligned} \|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|_1 &\geq \bar{\eta}\sum_{t=1}^n |\nabla L(\boldsymbol{w}^{k,t})_t| \geq \bar{\eta}\sum_{t=1}^n \left(|\nabla L(\boldsymbol{w}^{k,1})_t| - |\nabla L(\boldsymbol{w}^{k,t})_t - \nabla L(\boldsymbol{w}^{k,1})_t|\right) \\ &= \bar{\eta}\left(\|\nabla L(\boldsymbol{w}^{k,1})\|_1 - \sum_{t=1}^n |\nabla L(\boldsymbol{w}^{k,t})_t - \nabla L(\boldsymbol{w}^{k,1})_t|\right). \end{aligned} \tag{61}$$

Since $L(\boldsymbol{w})$ satisfies (21), using (44),

$$\begin{aligned}&\sum_{t=1}^{n} |\nabla L(\boldsymbol{w}^{k,t})_t - \nabla L(\boldsymbol{w}^{k,1})_t| \leq \sum_{t=1}^{n} \|\nabla L(\boldsymbol{w}^{k,t}) - \nabla L(\boldsymbol{w}^{k,1})\|_1 \\ \leq &\sum_{t=1}^{n} \sqrt{n}\tau_{\max}\|\boldsymbol{w}^{k,t} - \boldsymbol{w}^{k,1}\|_1 \leq n\sqrt{n}\tau_{\max}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^{k}\|_1.\end{aligned} \tag{62}$$

From (61) and (62), we have

$$\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|_1 \geq \frac{\bar{\eta}}{1 + \bar{\eta}n\sqrt{n}\tau_{\max}} \|\nabla L(\boldsymbol{w}^{k,1})\|_1.$$

This inequality and (44) imply

$$\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\| \geq \tfrac{1}{\sqrt{n}}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|_1 \geq \tfrac{\bar{\eta}}{\sqrt{n}+\bar{\eta}n^2\tau_{\max}}\|\nabla L(\boldsymbol{w}^k)\|.$$

Let $\eta = \bar{\eta}/(\sqrt{n} + \bar{\eta}n^2\tau_{\max})$. We then have (23).

Taking the summation of (60) from $t = 1$ to n, we get (24):

$$L(\boldsymbol{w}^k) - L(\boldsymbol{w}^{k+1}) \geq \frac{1}{2\sigma^2}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2.$$

7.4 Derivation of (30)-(31)

Using (18)-(19), we have

$$\frac{dS_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y)}{dz_t} = S_{\boldsymbol{w}}(x,y)e^{z_t f_t(x,y)} f_t(x,y) = S_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y)f_t(x,y)$$

and

$$\frac{dT_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)}{dz_t} = \sum_y S_{\boldsymbol{w}}(x,y)e^{z_t f_t(x,y)} f_t(x,y) = \sum_y S_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y)f_t(x,y).$$

Then (30) can be obtained from (16), the definition of $T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)$ in (1), and the following calculation:

$$\frac{d\log T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)}{dz_t} = \frac{\sum_y S_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y)f_t(x,y)}{T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)} = \sum_y P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)f_t(x,y).$$

For (31), we use

$$\begin{aligned}
&\frac{d\sum_y P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)f_t(x,y)}{dz_t}\\
=&\sum_y f_t(x,y)\frac{T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)\frac{dS_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y)}{dz_t}-\frac{dT_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)}{dz_t}S_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y)}{T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)^2}\\
=&\sum_y f_t(x,y)\left(f_t(x,y)P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)-\frac{S_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x,y)}{T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)}\sum_{y'}P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y'|x)f_t(x,y')\right)\\
=&\sum_y P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)f_t(x,y)^2-\sum_y\left(P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)f_t(x,y)\sum_{y'}P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y'|x)f_t(x,y')\right)\\
=&\sum_y P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)f_t(x,y)^2-\left(\sum_y P_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(y|x)f_t(x,y)\right)^2.
\end{aligned}$$

7.5 Derivation of (33)

From (6) and (9), we immediately have $A_t^{\mathsf{SCGIS}}(z_t) \geq A_t^{\mathsf{CD}}(z_t)$ and $A_t^{\mathsf{IIS}}(z_t) \geq A_t^{\mathsf{CD}}(z_t)$. Next we prove that $A_t^{\mathsf{GIS}}(z_t) \geq A_t^{\mathsf{SCGIS}}(z_t)$. Assume $D(z_t) \equiv A_t^{\mathsf{GIS}}(z_t) - A_t^{\mathsf{SCGIS}}(z_t)$. Then

$$D'(z_t) = \left(e^{z_t f^\#} - e^{z_t f_t^\#}\right)\sum_{x,y}\tilde{P}(x)P_{\boldsymbol{w}}(y|x)f_t(x,y).$$

Since $f^\# \geq f_t^\# \geq 0$,

$$\begin{aligned}
D'(z_t) &\geq 0 \text{ if } z_t > 0,\\
D'(z_t) &\leq 0 \text{ if } z_t < 0.
\end{aligned}\tag{63}$$

From Taylor expansion, there exists h between 0 and z_t such that

$$D(z_t) = D(0) + z_t D'(h).$$

From $A_t^{\mathsf{SCGIS}}(0) = A_t^{\mathsf{GIS}}(0) = 0$, we have $D(0) = 0$. By (63), $z_t D'(h) \geq 0$, so

$$A_t^{\mathsf{GIS}}(z_t) - A_t^{\mathsf{SCGIS}}(z_t) = D(z_t) \geq D(0) = 0.$$

We can use a similar method to prove $A_t^{\mathsf{GIS}}(z_t) \geq A_t^{\mathsf{IIS}}(z_t)$.

7.6 Proof of Theorem 4

From (31), we can define

$$H = \max_t\left(\frac{1}{\sigma^2} + \sum_{x,y}\tilde{P}(x)f_t(x,y)^2\right) \geq A_t^{\mathsf{CD}''}(z_t),\ \forall z_t.\tag{64}$$

From the Taylor expansion of $A_t^{\mathsf{CD}}(z_t)$ at $z_t = 0$, there exists h between 0 and d such that $z_t = \lambda d$ satisfies

$$\begin{aligned}
& A_t^{\mathsf{CD}}(\lambda d) - \gamma \lambda d A_t^{\mathsf{CD}\prime}(0) \\
= \;& A_t^{\mathsf{CD}}(0) + A_t^{\mathsf{CD}\prime}(0)\lambda d + \frac{1}{2} A_t^{\mathsf{CD}\prime\prime}(h)\lambda^2 d^2 - \gamma \lambda d A_t^{\mathsf{CD}\prime}(0) \\
\leq \;& A_t^{\mathsf{CD}\prime}(0)\lambda d + \frac{1}{2} H \lambda^2 d^2 - \gamma \lambda d A_t^{\mathsf{CD}\prime}(0) \\
= \;& -\lambda \frac{A_t^{\mathsf{CD}\prime}(0)^2}{A_t^{\mathsf{CD}\prime\prime}(0)} + \frac{1}{2} H \lambda^2 \frac{A_t^{\mathsf{CD}\prime}(0)^2}{A_t^{\mathsf{CD}\prime\prime}(0)^2} + \gamma \lambda \frac{A_t^{\mathsf{CD}\prime}(0)^2}{A_t^{\mathsf{CD}\prime\prime}(0)} \\
= \;& \lambda \frac{A_t^{\mathsf{CD}\prime}(0)^2}{A_t^{\mathsf{CD}\prime\prime}(0)} \left(\lambda \left(\frac{H}{2A_t^{\mathsf{CD}\prime\prime}(0)} \right) - 1 + \gamma \right). \qquad (65)
\end{aligned}$$

If we choose

$$\bar{\lambda} = \frac{2A_t^{\mathsf{CD}\prime\prime}(0)(1-\gamma)}{H}, \qquad (66)$$

then for $\lambda \leq \bar{\lambda}$, (65) is non-positive. Therefore, (35) is satisfied for all $0 \leq \lambda \leq \bar{\lambda}$.

7.7 Proof of Theorem 5

Following the proof in Section 7.3, it is sufficient to prove inequalities in the same form as (59) and (60). By Theorem 4, any $\lambda \in [\beta\bar{\lambda}, \bar{\lambda}]$, where $\beta \in (0,1)$ and $\bar{\lambda}$ is defined in (66), satisfies the sufficient decrease condition (35). Since Algorithm 3 selects λ by trying $\{1, \beta, \beta^2, \dots\}$, with (64), the selected value λ satisfies

$$\lambda \geq \beta\bar{\lambda} = \beta \frac{2A_t^{\mathsf{CD}\prime\prime}(0)(1-\gamma)}{H}.$$

This and (34) suggest that the step size $z_t = \lambda d$ in Algorithm 3 satisfies

$$|z_t| = \lambda \left| \frac{-A_t^{\mathsf{CD}\prime}(0)}{A_t^{\mathsf{CD}\prime\prime}(0)} \right| \geq \frac{2\beta(1-\gamma)}{H} \left| A_t^{\mathsf{CD}\prime}(0) \right|. \qquad (67)$$

From (34), (35), $z_t = \lambda d$, $A_t^{\mathsf{CD}\prime\prime}(0) \geq 1/\sigma^2$ and $\lambda \leq 1$, we have

$$A_t^{\mathsf{CD}}(z_t) - A_t^{\mathsf{CD}}(0) \leq \gamma z_t A_t^{\mathsf{CD}\prime}(0) = -\gamma z_t d A_t^{\mathsf{CD}\prime\prime}(0) \leq -\frac{\gamma}{\lambda\sigma^2} z_t^2 \leq -\frac{\gamma}{\sigma^2} z_t^2. \qquad (68)$$

Note that z_t is the step taken for updating $w_t^{k,t}$ to $w_t^{k,t+1}$. With $A_t^{\mathsf{CD}}(z_t) = L(\boldsymbol{w}^{k,t+1}) - L(\boldsymbol{w}^{k,t})$, (67)-(68) are in the same form as (59)-(60). In Section 7.3, (59)-(60) are sufficient to prove the desired conditions (23)-(24) for the linear convergence (Theorem 2). Therefore, Algorithm 3 linearly converges.

7.8 Proof of Theorem 6

A direct calculation of $A_t^{\mathsf{CD}\prime\prime\prime}(z_t)$ shows that it is bounded for all z_t and w_t^k. We assume that a bound is M. Using $A_t^{\mathsf{CD}}(0) = 0$, (34) and Taylor expansion, there exists h between

0 and d such that

$$\begin{aligned} A_t^{\mathsf{CD}}(d) &= A_t^{\mathsf{CD}\prime}(0)d + \frac{1}{2}A_t^{\mathsf{CD}\prime\prime}(0)d^2 + \frac{1}{6}A_t^{\mathsf{CD}\prime\prime\prime}(h)d^3 \\ &= -\frac{1}{2}\frac{A_t^{\mathsf{CD}\prime}(0)^2}{A_t^{\mathsf{CD}\prime\prime}(0)} + \frac{1}{6}A_t^{\mathsf{CD}\prime\prime\prime}(h)d^3 \\ &\leq -\frac{1}{2}A_t^{\mathsf{CD}\prime\prime}(0)d^2 + \frac{1}{6}M\left|d^3\right| \qquad (69) \\ &= -\gamma d^2 A_t^{\mathsf{CD}\prime\prime}(0) + \left(\gamma A_t^{\mathsf{CD}\prime\prime}(0) - \frac{1}{2}A_t^{\mathsf{CD}\prime\prime}(0) + \frac{1}{6}M|d|\right)d^2 \\ &= \gamma d A_t^{\mathsf{CD}\prime}(0) + \left(\gamma A_t^{\mathsf{CD}\prime\prime}(0) - \frac{1}{2}A_t^{\mathsf{CD}\prime\prime}(0) + \frac{1}{6}M|d|\right)d^2. \end{aligned}$$

Note that $\gamma < 1/2$. As $A_t^{\mathsf{CD}\prime\prime}(0) \geq 1/\sigma^2$ and $|A_t^{\mathsf{CD}\prime}(0)| \to 0$ when $\boldsymbol{w}$ converges to the optimal solution $\boldsymbol{w}^*$, near the optimum, d is small enough so that

$$0 \leq |d| \leq \frac{6}{M}\left(\frac{1}{2} - \gamma\right)A_t^{\mathsf{CD}\prime\prime}(0).$$

Then we obtain $A_t^{\mathsf{CD}}(d) \leq \gamma d A_t^{\mathsf{CD}\prime}(0)$ and (35) is satisfied.

7.9 Proof of Theorem 7

The following lemma, needed for proving Theorem 7, shows that the direction taken by CD is bigger than that of GIS, IIS, or SCGIS.

Lemma 8 *There exists a positive constant λ such that in a neighborhood of $\boldsymbol{w}^*$,*

$$|d^s|(1+\lambda) \leq |d| = \left|\frac{\delta_t'(0)}{\delta_t''(0)}\right|, \qquad (70)$$

where d and d^s are defined in (37)*.*

Proof. Since $d^s = \arg\min_{z_t} A_t^s(z_t)$ and $A_t^s(z_t)$ is strictly convex,

$$A_t^{s\prime}(d^s) = 0. \qquad (71)$$

We separate the proof to two cases: $A_t^{s\prime}(0) > 0$ and $A_t^{s\prime}(0) < 0$. If $A_t^{s\prime}(0) = 0$, then $d^s = 0$, so (70) immediately holds.

If $A_t^{s\prime}(0) > 0$, from the strict convexity of $A_t^s(z_t)$ and (71), $d^s < 0$. It is sufficient to prove that there is λ such that $A_t^{s\prime}(d/(1+\lambda)) \leq 0$. This result implies $d/(1+\lambda) \leq d^s$, so we obtain (70) .

Using Taylor expansion, if $z_t f^s(x,y) < 0$, then

$$e^{z_t f^s(x,y)} \leq 1 + z_t f^s(x,y) + \frac{1}{2}z_t^2(f^s(x,y))^2, \qquad (72)$$

where

$$f^s(x,y) \equiv \begin{cases} f^{\#} \text{ if } s \text{ is } \mathsf{GIS}, \\ f_t^{\#} \text{ if } s \text{ is } \mathsf{SCGIS}, \\ f^{\#}(x,y) \text{ if } s \text{ is } \mathsf{IIS}. \end{cases}$$

From Table 1 and (72),

$$\begin{aligned}
A_t^{s\prime}(z_t) &= \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) e^{z_t f^s(x,y)} + Q_t'(z_t) \qquad (73)\\
&\le \left(\sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) + \frac{w_t}{\sigma^2} - \tilde{P}(f_t) \right) + \left(R_1(\boldsymbol{w}) + \frac{1}{\sigma^2} \right) z_t + \frac{1}{2} z_t^2 R_2(\boldsymbol{w})\\
&= A_t^{s\prime}(0) + \left(R_1(\boldsymbol{w}) + \frac{1}{\sigma^2} - \frac{1}{2}|z_t| R_2(\boldsymbol{w}) \right) z_t,
\end{aligned}$$

where

$$\begin{aligned}
R_1(\boldsymbol{w}) &\equiv \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) f^s(x,y) \quad \text{and}\\
R_2(\boldsymbol{w}) &\equiv \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y) f^s(x,y)^2.
\end{aligned}$$

Now the Newton direction is

$$d = -\frac{A_t^{\mathsf{CD}\prime}(0)}{A_t^{\mathsf{CD}\prime\prime}(0)} = -\frac{\delta_t'(0)}{\delta_t''(0)} = -\frac{A_t^{s\prime}(0)}{\delta_t''(0)} < 0. \qquad (74)$$

From (31),

$$\begin{aligned}
\delta_t''(0) &= \sum_{x,y} \tilde{P}(x) P_{\boldsymbol{w}}(y|x) f_t(x,y)^2 - \sum_{x} \tilde{P}(x) \left(\sum_{y} P_{\boldsymbol{w}}(y|x) f_t(x,y) \right)^2 + \frac{1}{\sigma^2}\\
&\le R_1(\boldsymbol{w}) - R_3(\boldsymbol{w}) + \frac{1}{\sigma^2},
\end{aligned} \qquad (75)$$

where

$$R_3(\boldsymbol{w}) \equiv \sum_{x} \tilde{P}(x) \left(\sum_{y} P_{\boldsymbol{w}}(y|x) f_t(x,y) \right)^2.$$

When $\boldsymbol{w} \to \boldsymbol{w}^*$, $R_1(\boldsymbol{w})$, $R_2(\boldsymbol{w})$, $R_3(\boldsymbol{w})$ respectively converge to $R_1(\boldsymbol{w}^*)$, $R_2(\boldsymbol{w}^*)$, $R_3(\boldsymbol{w}^*)$. Moreover, as $\boldsymbol{w} + d^s \boldsymbol{e}_t \in \{\bar{\boldsymbol{w}} \mid L(\bar{\boldsymbol{w}}) \le L(\boldsymbol{w})\}$, $d^s \to 0$ when $\boldsymbol{w} \to \boldsymbol{w}^*$. Therefore,

$$\lim_{\boldsymbol{w} \to \boldsymbol{w}^*} \frac{\delta_t''(0)}{R_1(\boldsymbol{w}) + \frac{1}{\sigma^2} - \frac{1}{2}|d^s| R_2(\boldsymbol{w})} = 1 - \frac{R_3(\boldsymbol{w}^*)}{R_1(\boldsymbol{w}^*) + \frac{1}{\sigma^2}}. \qquad (76)$$

Here we can assume $R_3(\boldsymbol{w}^*) > 0$. If not, $f_t(x,y) = 0$ for all x, y. Then $w_t^* = 0$ is obtained in just one iteration. From (76), we can choose a positive λ such that

$$\frac{1}{1+\lambda} > 1 - \frac{R_3(\boldsymbol{w}^*)}{R_1(\boldsymbol{w}^*) + \frac{1}{\sigma^2}}. \qquad (77)$$

From (76) and (77), for any $\boldsymbol{w}$ in a neighborhood of $\boldsymbol{w}^*$,

$$\delta_t''(0) \le \frac{R_1(\boldsymbol{w}) + \frac{1}{\sigma^2} - \frac{1}{2}|d^s| R_2(\boldsymbol{w})}{1+\lambda}.$$

From (74),

$$\frac{d}{1+\lambda} \leq -\frac{A_t^{s\prime}(0)}{R_1(\boldsymbol{w}) + \frac{1}{\sigma^2} - \frac{1}{2}|d^s|R_2(\boldsymbol{w})}. \tag{78}$$

From (73) with $z_t = d/(1+\lambda)$ and (78),

$$A_t^{s\prime}\left(\frac{d}{1+\lambda}\right) \leq A_t^{s\prime}(0) - A_t^{s\prime}(0) = 0.$$

Therefore, $d/(1+\lambda) \leq d^s < 0$.

If $A_t^{s\prime}(0) < 0$, then $d^s > 0$. Using Taylor expansion, if $z_t f^s(x,y) > 0$, we have

$$e^{z_t f^s(x,y)} \geq 1 + z_t f^s(x,y).$$

Then (73) becomes

$$\begin{aligned} A_t^{s\prime}(z_t) &\geq \left(\sum_{x,y} \tilde{P}(x)P_{\boldsymbol{w}}(y|x)f_t(x,y) + \frac{w_t}{\sigma^2} - \tilde{P}(f_t)\right) + \left(R_1(\boldsymbol{w}) + \frac{1}{\sigma^2}\right) z_t \\ &= A_t^{s\prime}(0) + \left(R_1(\boldsymbol{w}) + \frac{1}{\sigma^2}\right) z_t. \end{aligned} \tag{79}$$

From (75) and a derivation similar to (76), there is a $\lambda > 0$ such that

$$\delta_t''(0)(1+\lambda) \leq R_1(\boldsymbol{w}) + \frac{1}{\sigma^2}.$$

Let $z_t = d/(1+\lambda)$ in (79). With (74),

$$A_t^{s\prime}\left(\frac{d}{1+\lambda}\right) \geq A_t^{s\prime}(0) - A_t^{s\prime}(0) = 0.$$

Therefore, $0 < d^s \leq d/(1+\lambda)$.

Proof of Theorem 7 We prove this theorem by calculating a lower bound of $\delta_t(d^s) - \delta_t(d)$. From (69),

$$\delta_t(d) \leq -\frac{1}{2}\frac{\delta_t'(0)^2}{\delta_t''(0)} + \frac{1}{6}Md^3, \tag{80}$$

where M is an upper bound of $\delta_t'''(z_t)$. If $\boldsymbol{w}$ is sufficiently close to $\boldsymbol{w}^*$,

$$\begin{aligned} \delta_t(d^s) &= \delta_t'(0)d^s + \frac{1}{2}\delta_t''(0)(d^s)^2 + \frac{1}{6}\delta_t'''(h)(d^s)^3 \\ &= \frac{1}{2}\delta_t''(0)\left(\frac{\delta_t'(0)}{\delta_t''(0)} - d^s\right)^2 - \frac{1}{2}\frac{\delta_t'(0)^2}{\delta_t''(0)} + \frac{1}{6}\delta_t'''(h)(d^s)^3 \\ &\geq \frac{1}{2}\delta_t''(0)\left(\frac{\lambda}{1+\lambda}\right)\frac{\delta_t'(0)^2}{\delta_t''(0)^2} - \frac{1}{2}\frac{\delta_t'(0)^2}{\delta_t''(0)} - \frac{1}{6}M|d^3| \\ &= \frac{1}{2}\left(\frac{-1}{1+\lambda}\right)\frac{\delta_t'(0)^2}{\delta_t''(0)} - \frac{1}{6}M|d^3|, \end{aligned} \tag{81}$$

where h is between 0 and d^s and the inequality is from Lemma 8. Combining (80) and (81),

$$\begin{aligned}\delta_t(d^s) - \delta_t(d) &\geq \frac{1}{2}\left(1 - \frac{1}{1+\lambda}\right)\frac{\delta_t'(0)^2}{\delta_t''(0)} - \frac{1}{3}M|d^3| \\ &= \left(\frac{1}{2}\left(\frac{\lambda}{1+\lambda}\right) - \frac{1}{3}M\frac{|\delta_t'(0)|}{\delta_t''(0)^2}\right)\frac{\delta_t'(0)^2}{\delta_t''(0)}.\end{aligned}$$

Since $\delta_t''(0) \geq 1/(\sigma^2)$ and $\delta_t'(0) \to \nabla_t L(\boldsymbol{w}^*) = 0$, there is a neighborhood of $\boldsymbol{w}^*$ so that $\delta_t'(0)$ is small enough and

$$\frac{1}{2}\left(\frac{\lambda}{1+\lambda}\right) > \frac{1}{3}M\frac{|\delta_t'(0)|}{\delta_t''(0)^2}.$$

Therefore, $\delta_t(d^s) > \delta_t(d)$ in a neighborhood of $\boldsymbol{w}^*$.

7.10 Derivation of (39)

From Jensen's inequality and the fact that $\log(x)$ is a concave function,

$$\frac{\sum_{\Omega_t}\tilde{P}(x)\log\frac{T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)}{T_{\boldsymbol{w}}(x)}}{\sum_{\Omega_t}\tilde{P}(x)} \leq \log\left(\frac{\sum_{\Omega_t}\tilde{P}(x)\frac{T_{\boldsymbol{w}+z_t\boldsymbol{e}_t}(x)}{T_{\boldsymbol{w}}(x)}}{\sum_{\Omega_t}\tilde{P}(x)}\right).$$

With (16), (19) and (40), we have

$$A_t^{\mathsf{CD}}(z_t) \leq Q_t(z_t) + \tilde{P}_t\log\left(1 + \frac{\sum_{\Omega_t}\tilde{P}(x)\sum_y P_{\boldsymbol{w}}(y|x)(e^{z_t f_t(x,y)} - 1)}{\tilde{P}_t}\right).$$

By the inequality (15),

$$\begin{aligned}A_t^{\mathsf{CD}}(z_t) &\leq Q_t(z_t) + \tilde{P}_t\log\left(1 + \frac{\sum_{\Omega_t}\tilde{P}(x)\sum_y P_{\boldsymbol{w}}(y|x)\left(\frac{f_t(x,y)e^{z_t f_t^\#}}{f_t^\#} + \frac{f_t^\# - f_t(x,y)}{f_t^\#} - 1\right)}{\tilde{P}_t}\right) \\ &= Q_t(z_t) + \tilde{P}_t\log\left(1 + \frac{\left(e^{z_t f_t^\#} - 1\right)\sum_{\Omega_t}\tilde{P}(x)\sum_y P_{\boldsymbol{w}}(y|x)f_t(x,y)}{f_t^\#\tilde{P}_t}\right) \\ &= \bar{A}_t^{\mathsf{CD}}(z_t).\end{aligned}$$

Note that replacing $\tilde{P}_t$ with $\sum_x \tilde{P}(x)$ leads to another upper bound of $A_t^{\mathsf{CD}}(z_t)$. It is, however, looser than $\bar{A}_t^{\mathsf{CD}}(z_t)$.

7.11 Logistic Regression

We list approximate functions of IS/CD methods for logistic regression. Note that

$$\tilde{P}(x_i, y) = \begin{cases}\frac{1}{l} & \text{if } y = \bar{y}_i, \\ 0 & \text{otherwise,}\end{cases} \quad \text{and } \tilde{P}(f_t) = \sum_{i:\bar{y}_i=1}\frac{1}{l}\bar{x}_{it}. \tag{82}$$

For GIS, using the formula in Table 1 and (43),

$$A_t^{\mathsf{GIS}}(z_t) = Q_t(z_t) + \frac{1}{l}\left(\frac{e^{z_t f^{\#}} - 1}{f^{\#}} \sum_i \frac{\bar{x}_{it}}{1 + e^{-\boldsymbol{w}^T \bar{\boldsymbol{x}}_i}}\right),$$

where from (11) and (82),

$$Q_t(z_t) = \frac{2w_t z_t + z_t^2}{2\sigma^2} - \frac{z_t}{l} \sum_{i:\bar{y}_i = 1} \bar{x}_{it} \quad \text{and} \quad f^{\#} \equiv \max_j f^{\#}(i).$$

Similarly, IIS and SCGIS respectively solve

$$A_t^{\mathsf{IIS}}(z_t) = Q_t(z_t) + \frac{1}{l}\left(\sum_i \frac{\bar{x}_{it}}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}} \frac{e^{z_t f^{\#}(i)} - 1}{f^{\#}(i)}\right),$$

$$A_t^{\mathsf{SCGIS}}(z_t) = Q_t(z_t) + \frac{1}{l}\left(\frac{e^{z_t f_t^{\#}} - 1}{f_t^{\#}} \sum_i \frac{\bar{x}_{it}}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}}\right),$$

where $f_t^{\#} = \max_i \bar{x}_{it}$ and $f^{\#}(i) = \sum_t \bar{x}_{it}$. Finally, from (17), (30), and (31),

$$A_t^{\mathsf{CD}}(z_t) = Q_t(z_t) + \frac{1}{l} \sum_i \log\left(1 + \frac{e^{z_t \bar{x}_{it}} - 1}{1 + e^{-\boldsymbol{w}^T \bar{\boldsymbol{x}}_i}}\right),$$

$$A_t^{\mathsf{CD}\prime}(0) = \frac{w_t}{\sigma^2} + \frac{1}{l}\left(\sum_i \frac{\bar{x}_{it}}{1 + e^{-\boldsymbol{w}^T \bar{\boldsymbol{x}}_i}} - \sum_{i:\bar{y}_i = 1} \bar{x}_{it}\right),$$

$$A_t^{\mathsf{CD}\prime\prime}(0) = \frac{1}{\sigma^2} + \frac{1}{l}\left(\sum_i \frac{e^{-\boldsymbol{w}^T \bar{\boldsymbol{x}}_i} \bar{x}_{it}^2}{(1 + e^{-\boldsymbol{w}^T \bar{\boldsymbol{x}}_i})^2}\right).$$

Acknowledgments

The authors thank anonymous reviewers and associate editor for helpful comments.

References

Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In *Proceedings of the Twenty Fourth International Conference on Machine Learning (ICML)*, 2007.

Tom M. Apostol. *Mathematical Analysis*. Addison-Wesley, second edition, 1974.

Jason Baldridge, Tom Morton, and Gann Bierner. OpenNLP package, 2001. URL `http://opennlp.sourceforge.net/`.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA 02178-9998, second edition, 1999.

Léon Bottou. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, 2004.

Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale L2-loss linear SVM. *Journal of Machine Learning Research*, 9:1369–1398, 2008. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/cdl2.pdf`.

Stanley F. Chen and Ronald Rosenfeld. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50, January 2000.

Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1–3):253–285, 2002.

Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *Journal of Machine Learning Research*, 9:1775–1822, 2008.

John N. Darroch and Douglas Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

Hal Daumé, III. Notes on CG and LM-BFGS optimization of logistic regression. 2004. URL `http://www.cs.utah.edu/~hal/megam/`.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

Miroslav Dudík, Steven J. Phillips, and Robert E. Schapire. Performance guarantees for regularized maximum entropy density estimation. In *Proceedings of the 17th Annual Conference on Computational Learning Theory*, pages 655–662, New York, 2004. ACM press.

Roger Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, 1987.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33, 2010.

Jianfeng Gao, Galen Andrew, Mark Johnson, and Kristina Toutanova. A comparative study of parameter estimation methods statistical natural language processing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 824–831, 2007.

Alexandar Genkin, David D. Lewis, and David Madigan. Large-scale Bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007.

Joshua Goodman. Sequential conditional generalized iterative scaling. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 9–16, 2002.

Luigi Grippo and Marco Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization Methods and Software*, 10:587–637, 1999.

Fang-Lan Huang, Cho-Jui Hsieh, Kai-Wei Chang, and Chih-Jen Lin. Iterative scaling and coordinate descent methods for maximum entropy. In *Proceedings of the 47th Annual Meeting of the Association of Computational Linguistics (ACL)*, 2009. Short paper.

Rong Jin, Rong Yan, Jian Zhang, and Alex G. Hauptmann. A faster iterative scaling algorithm for conditional exponential model. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 2003.

S. Sathiya Keerthi, Kaibo Duan, Shirish Shevade, and Aun Neow Poo. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61:151–165, 2005.

Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research*, 8:1519–1555, 2007. URL `http://www.stanford.edu/~boyd/l1_logistic_reg.html`.

Kenneth Lange, David R. Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, March 2000.

Chih-Jen Lin, Ruby C. Weng, and S. Sathiya Keerthi. Trust region Newton method for large-scale logistic regression. *Journal of Machine Learning Research*, 9:627–650, 2008. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/logistic.pdf`.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

Zhi-Quan Luo and Paul Tseng. On the convergence of coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72 (1):7–35, 1992.

Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th conference on Natural language learning*, pages 1–7. Association for Computational Linguistics, 2002.

Ryan McDonald and Fernando Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 81–88, 2006.

Thomas P. Minka. A comparison of numerical optimizers for logistic regression, 2003. URL `http://research.microsoft.com/~minka/papers/logreg/`.

Adwait Ratnaparkhi. *Maximum Entropy Models For Natural Language Ambiguity Resolution.* PhD thesis, University of Pennsylvania, 1998.

Nicol N. Schraudolph, Jin Yu, and Simon Gunter. A stochastic quasi-Newton method for online convex optimization. In *Proceedings of the 11th International Conference Artificial Intelligence and Statistics (AISTATS)*, pages 433–440, 2007.

S.V.N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 969–976, 2006.

Zhihua Zhang, James T. Kwok, and Dit-Yan Yeung. Surrogate maximization/minimization algorithms and extensions. *Machine Learning*, 69(1):1–33, October 2007.