One-vs-Each Approximation to Softmax for Scalable Estimation of Probabilities

Michalis K. Titsias
Department of Informatics
Athens University of Economics and Business
mtitsias@aueb.gr

Abstract

The softmax representation of probabilities for categorical variables plays a prominent role in modern machine learning with numerous applications in areas such as large scale classification, neural language modeling and recommendation systems. However, softmax estimation is very expensive for large scale inference because of the high cost associated with computing the normalizing constant. Here, we introduce an efficient approximation to softmax probabilities which takes the form of a rigorous lower bound on the exact probability. This bound is expressed as a product over pairwise probabilities and it leads to scalable estimation based on stochastic optimization. It allows us to perform doubly stochastic estimation by subsampling both training instances and class labels. We show that the new bound has interesting theoretical properties and we demonstrate its use in classification problems.

1 Introduction

Based on the softmax representation, the probability of a variable y to take the value $k \in \{1, \dots, K\}$, where K is the number of categorical symbols or classes, is modeled by

$$p(y = k|\mathbf{x}) = \frac{e^{f_k(\mathbf{x};\mathbf{w})}}{\sum_{m=1}^{K} e^{f_m(\mathbf{x};\mathbf{w})}}, \tag{1}$$

where each $f_k(\mathbf{x}; \mathbf{w})$ is often referred to as *the score function* and it is a real-valued function indexed by an input vector $\mathbf{x}$ and parameterized by $\mathbf{w}$. The score function measures the compatibility of input $\mathbf{x}$ with symbol $y = k$ so that the higher the score is the more compatible $\mathbf{x}$ becomes with $y = k$. The most common application of softmax is multiclass classification where $\mathbf{x}$ is an observed input vector and $f_k(\mathbf{x}; \mathbf{w})$ is often chosen to be a linear function or more generally a non-linear function such as a neural network [3, 8]. Several other applications of softmax arise, for instance, in neural language modeling for learning word vector embeddings [15, 14, 18] and also in collaborating filtering for representing probabilities of $(user, item)$ pairs [17]. In such applications the number of symbols K could often be very large, e.g. of the order of tens of thousands or millions, which makes the computation of softmax probabilities very expensive due to the large sum in the normalizing constant of Eq. (1). Thus, exact training procedures based on maximum likelihood or Bayesian approaches are computationally prohibitive and approximations are needed. While some rigorous bound-based approximations to the softmax exists [5], they are not so accurate or scalable and therefore it would be highly desirable to develop accurate and computationally efficient approximations.

In this paper we introduce a new efficient approximation to softmax probabilities which takes the form of a lower bound on the probability of Eq. (1). This bound draws an interesting connection between the exact softmax probability and all its one-vs-each pairwise probabilities, and it has several desirable properties. Firstly, for the non-parametric estimation case it leads to an approximation of the

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

likelihood that shares the same global optimum with exact maximum likelihood, and thus estimation based on the approximation is a perfect surrogate for the initial estimation problem. Secondly, the bound allows for scalable learning through stochastic optimization where data subsampling can be combined with subsampling categorical symbols. Thirdly, whenever the initial exact softmax cost function is convex the bound remains also convex.

Regarding related work, there exist several other methods that try to deal with the high cost of softmax such as methods that attempt to perform the exact computations [9, 19], methods that change the model based on hierarchical or stick-breaking constructions [16, 13] and sampling-based methods [1, 14, 7, 11]. Our method is a lower bound based approach that follows the variational inference framework. Other rigorous variational lower bounds on the softmax have been used before [4, 5], however they are not easily scalable since they require optimizing data-specific variational parameters. In contrast, the bound we introduce in this paper does not contain any variational parameter, which greatly facilitates stochastic minibatch training. At the same time it can be much tighter than previous bounds [5] as we will demonstrate empirically in several classification datasets.

2 One-vs-each lower bound on the softmax

Here, we derive the new bound on the softmax (Section 2.1) and we prove its optimality property when performing approximate maximum likelihood estimation (Section 2.2). Such a property holds for the *non-parametric case*, where we estimate probabilities of the form $p(y = k)$, without conditioning on some $\mathbf{x}$, so that the score functions $f_k(\mathbf{x}; \mathbf{w})$ reduce to unrestricted parameters f_k; see Eq. (2) below. Finally, we also analyze the related bound derived by Bouchard [5] and we compare it with our approach (Section 2.3).

2.1 Derivation of the bound

Consider a discrete random variable $y \in \{1, \dots, K\}$ that takes the value k with probability,

$$p(y = k) = \text{Softmax}_k(f_1, \dots, f_K) = \frac{e^{f_k}}{\sum_{m=1}^{K} e^{f_m}}, \tag{2}$$

where each f_k is a free real-valued scalar parameter. We wish to express a lower bound on $p(y = k)$ and the key step of our derivation is to re-write $p(y = k)$ as

$$p(y = k) = \frac{1}{1 + \sum_{m \neq k} e^{-(f_k - f_m)}}. \tag{3}$$

Then, by exploiting the fact that for any non-negative numbers α_1 and α_2 it holds $1 + \alpha_1 + \alpha_2 \leq 1 + \alpha_1 + \alpha_2 + \alpha_1 \alpha_2 = (1 + \alpha_1)(1 + \alpha_2)$, and more generally it holds $(1 + \sum_i \alpha_i) \leq \prod_i (1 + \alpha_i)$ where each $\alpha_i \geq 0$, we obtain the following lower bound on the above probability,

$$p(y = k) \geq \prod_{m \neq k} \frac{1}{1 + e^{-(f_k - f_m)}} = \prod_{m \neq k} \frac{e^{f_k}}{e^{f_k} + e^{f_m}} = \prod_{m \neq k} \sigma(f_k - f_m). \tag{4}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Clearly, the terms in the product are pairwise probabilities each corresponding to the event $y = k$ conditional on the union of pairs of events, i.e. $y \in \{k, m\}$ where m is one of the remaining values. We will refer to this bound as one-vs-each bound on the softmax probability, since it involves $K - 1$ comparisons of a specific event $y = k$ versus each of the $K - 1$ remaining events. Furthermore, the above result can be stated more generally to define bounds on arbitrary probabilities as the following statement shows.

Proposition 1. *Assume a probability model with state space Ω and probability measure $P(\cdot)$. For any event $A \subset \Omega$ and an associated countable set of disjoint events $\{B_i\}$ such that $\cup_i B_i = \Omega \setminus A$, it holds*

$$P(A) \geq \prod_i P(A | A \cup B_i). \tag{5}$$

Proof. Given that $P(A) = \frac{P(A)}{P(\Omega)} = \frac{P(A)}{P(A) + \sum_i P(B_i)}$, the result follows by applying the inequality $(1 + \sum_i \alpha_i) \leq \prod_i (1 + \alpha_i)$ exactly as done above for the softmax parameterization. □

Remark. If the set $\{B_i\}$ consists of a single event B then by definition $B = \Omega \setminus A$ and the bound is exact since in such case $P(A|A \cup B) = P(A)$.

Furthermore, based on the above construction we can express a full class of hierarchically ordered bounds. For instance, if we merge two events B_i and B_j into a single one, then the term $P(A|A \cup B_i)P(A|A \cup B_j)$ in the initial bound is replaced with $P(A|A \cup B_i \cup B_j)$ and the associated new bound, obtained after this merge, can only become tighter. To see a more specific example in the softmax probabilistic model, assume a small subset of categorical symbols $\mathcal{C}_k$, that does not include k, and denote the remaining symbols excluding k as $\bar{\mathcal{C}}_k$ so that $k \cup \mathcal{C}_k \cup \bar{\mathcal{C}}_k = \{1, \ldots, K\}$. Then, a tighter bound, that exists higher in the hierarchy, than the one-vs-each bound (see Eq. 4) takes the form,

$$p(y = k) \geq \text{Softmax}_k(f_k, \mathbf{f}_{\mathcal{C}_k}) \times \text{Softmax}_k(f_k, \mathbf{f}_{\bar{\mathcal{C}}_k}) \geq \text{Softmax}_k(f_k, \mathbf{f}_{\mathcal{C}_k}) \times \prod_{m \in \bar{\mathcal{C}}_k} \sigma(f_k - f_m), \quad (6)$$

where $\text{Softmax}_k(f_k, \mathbf{f}_{\mathcal{C}_k}) = \frac{e^{f_k}}{e^{f_k} + \sum_{m \in \mathcal{C}_k} e^{f_m}}$ and $\text{Softmax}_k(f_k, \mathbf{f}_{\bar{\mathcal{C}}_k}) = \frac{e^{f_k}}{e^{f_k} + \sum_{m \in \bar{\mathcal{C}}_k} e^{f_m}}$. For simplicity of our presentation in the remaining of the paper we do not discuss further these more general bounds and we focus only on the one-vs-each bound.

The computationally useful aspect of the bound in Eq. (4) is that it factorizes into a product, where each factor depends only on a pair of parameters (f_k, f_m). Crucially, this avoids the evaluation of the normalizing constant associated with the global probability in Eq. (2) and, as discussed in Section 3, it leads to scalable training using stochastic optimization that can deal with very large K. Furthermore, approximate maximum likelihood estimation based on the bound can be very accurate and, as shown in the next section, it is exact for the non-parametric estimation case.

The fact that the one-vs-each bound in (4) is a product of pairwise probabilities suggests that there is a connection with Bradley-Terry (BT) models [6, 10] for learning individual skills from paired comparisons and the associated multiclass classification systems obtained by combining binary classifiers, such as one-vs-rest and one-vs-one approaches [10]. Our method differs from BT models, since we do not combine binary probabilistic models to a posteriori form a multiclass model. Instead, we wish to develop scalable approximate algorithms that can surrogate the training of multiclass softmax-based models by maximizing lower bounds on the exact likelihoods of these models.

2.2 Optimality of the bound for maximum likelihood estimation

Assume a set of observation $(y_1, \ldots, y_N)$ where each $y_i \in \{1, \ldots, K\}$. The log likelihood of the data takes the form,

$$\mathcal{L}(\mathbf{f}) = \log \prod_{i=1}^{N} p(y_i) = \log \prod_{k=1}^{K} p(y = k)^{N_k}, \quad (7)$$

where $\mathbf{f} = (f_1, \ldots, f_K)$ and N_k denotes the number of data points with value k. By substituting $p(y = k)$ from Eq. (2) and then taking derivatives with respect to $\mathbf{f}$ we arrive at the standard stationary conditions of the maximum likelihood solution,

$$\frac{e^{f_k}}{\sum_{m=1}^{K} e^{f_m}} = \frac{N_k}{N}, \; k = 1, \ldots, K. \quad (8)$$

These stationary conditions are satisfied for $f_k = \log N_k + c$ where $c \in \mathbb{R}$ is an arbitrary constant. What is rather surprising is that the same solutions $f_k = \log N_k + c$ satisfy also the stationary conditions when maximizing a lower bound on the exact log likelihood obtained from the product of one-vs-each probabilities.

More precisely, by replacing $p(y = k)$ with the bound from Eq. (4) we obtain a lower bound on the exact log likelihood,

$$\mathcal{F}(\mathbf{f}) = \log \prod_{k=1}^{K} \left[\prod_{m \neq k} \frac{e^{f_k}}{e^{f_k} + e^{f_m}} \right]^{N_k} = \sum_{k > m} \log P(f_k, f_m), \quad (9)$$

where $P(f_k, f_m) = \left[\frac{e^{f_k}}{e^{f_k} + e^{f_m}} \right]^{N_k} \left[\frac{e^{f_m}}{e^{f_k} + e^{f_m}} \right]^{N_m}$ is a likelihood involving only the data of the pair of states (k, m), while there exist $K(K-1)/2$ possible such pairs. If instead of maximizing the exact log likelihood from Eq. (7) we maximize the lower bound we obtain the same parameter estimates.

Proposition 2. *The maximum likelihood parameter estimates* $f_k = \log N_k + c, k = 1, \ldots, K$ *for the exact log likelihood from Eq. (7) globally also maximize the lower bound from Eq. (9).*

Proof. By computing the derivatives of $\mathcal{F}(\mathbf{f})$ we obtain the following stationary conditions

$$K - 1 = \sum_{m \neq k} \frac{N_k + N_m}{N_k} \frac{e^{f_k}}{e^{f_k} + e^{f_m}}, \; k = 1, \ldots, K, \tag{10}$$

which form a system of K non-linear equations over the unknowns $(f_1, \ldots, f_K)$. By substituting the values $f_k = \log N_k + c$ we can observe that all K equations are simultaneously satisfied which means that these values are solutions. Furthermore, since $\mathcal{F}(\mathbf{f})$ is a concave function of $\mathbf{f}$ we can conclude that the solutions $f_k = \log N_k + c$ globally maximize $\mathcal{F}(\mathbf{f})$. $\square$

Remark. Not only is $\mathcal{F}(\mathbf{f})$ globally maximized by setting $f_k = \log N_k + c$, but also each pairwise likelihood $P(f_k, f_m)$ in Eq. (9) is separately maximized by the same setting of parameters.

2.3 Comparison with Bouchard's bound

Bouchard [5] proposed a related bound that next we analyze in terms of its ability to approximate the exact maximum likelihood training in the non-parametric case, and then we compare it against our method. Bouchard [5] was motivated by the problem of applying variational Bayesian inference to multiclass classification and he derived the following upper bound on the log-sum-exp function,

$$\log \sum_{m=1}^{K} e^{f_m} \leq \alpha + \sum_{m=1}^{K} \log \left(1 + e^{f_m - \alpha}\right), \tag{11}$$

where $\alpha \in \mathbb{R}$ is a variational parameter that needs to be optimized in order for the bound to become as tight as possible. The above induces a lower bound on the softmax probability $p(y = k)$ from Eq. (2) that takes the form

$$p(y = k) \geq \frac{e^{f_k - \alpha}}{\prod_{m=1}^{K} (1 + e^{f_m - \alpha})}. \tag{12}$$

This is not the same as Eq. (4), since there is not a value for α for which the above bound will reduce to our proposed one. For instance, if we set $\alpha = f_k$, then Bouchard's bound becomes half the one in Eq. (4) due to the extra term $1 + e^{f_k - f_k} = 2$ in the product in the denominator.[1] Furthermore, such a value for α may not be the optimal one and in practice α must be chosen by minimizing the upper bound in Eq. (11). While such an optimization is a convex problem, it requires iterative optimization since there is not in general an analytical solution for α. However, for the simple case where $K = 2$ we can analytically find the optimal α and the optimal $\mathbf{f}$ parameters. The following proposition carries out this analysis and provides a clear understanding of how Bouchard's bound behaves when applied for approximate maximum likelihood estimation.

Proposition 3. *Assume that* $K = 2$ *and we approximate the probabilities* $p(y = 1)$ *and* $p(y = 2)$ *from (2) with the corresponding Bouchard's bounds given by* $\frac{e^{f_1 - \alpha}}{(1+e^{f_1-\alpha})(1+e^{f_2-\alpha})}$ *and* $\frac{e^{f_2 - \alpha}}{(1+e^{f_1-\alpha})(1+e^{f_2-\alpha})}$. *These bounds are used to approximate the maximum likelihood solution by maximizing a bound* $\mathcal{F}(f_1, f_2, \alpha)$ *which is globally maximized for*

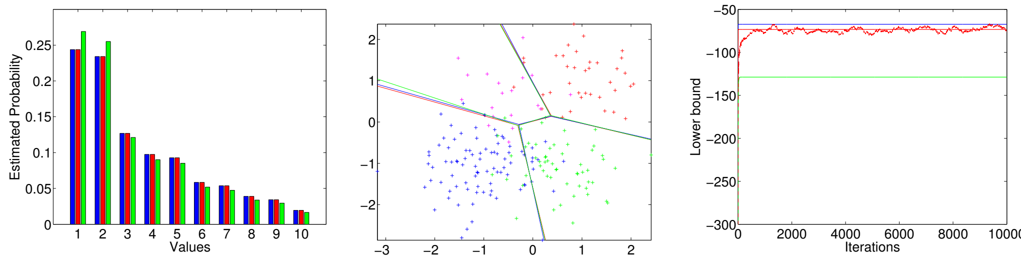
$$\alpha = \frac{f_1 + f_2}{2}, \;\; f_k = 2 \log N_k + c, \;\; k = 1, 2. \tag{13}$$

The proof of the above is given in the Supplementary material. Notice that the above estimates are biased so that the probability of the most populated class (say the $y = 1$ for which $N_1 > N_2$) is overestimated while the other probability is underestimated. This is due to the factor 2 that multiplies $\log N_1$ and $\log N_2$ in (13).

Also notice that the solution $\alpha = \frac{f_1 + f_2}{2}$ is not a general trend, i.e. for $K > 2$ the optimal α is not the mean of f_ks. In such cases approximate maximum likelihood estimation based on Bouchard's bound requires iterative optimization. Figure 1a shows some estimated softmax probabilities, using a dataset

[1] Notice that the product in Eq. (4) excludes the value k, while Bouchard's bound includes it.

of 200 points each taking one out of ten values, where $\mathbf{f}$ is found by exact maximum likelihood, the proposed one-vs-each bound and Bouchard's method. As expected estimation based on the bound in Eq. (4) gives the exact probabilities, while Bouchard's bound tends to overestimate large probabilities and underestimate small ones.



(a) (b) (c)

Figure 1: (a) shows the probabilities estimated by exact softmax (blue bar), one-vs-each approximation (red bar) and Bouchard's method (green bar). (b) shows the 5-class artificial data together with the decision boundaries found by exact softmax (blue line), one-vs-each (red line) and Bouchard's bound (green line). (c) shows the maximized (approximate) log likelihoods for the different approaches when applied to the data of panel (b) (see Section 3). Notice that the blue line in (c) is the exact maximized log likelihood while the remaining lines correspond to lower bounds.

3 Stochastic optimization for extreme classification

Here, we return to the general form of the softmax probabilities as defined by Eq. (1) where the score functions are indexed by input $\mathbf{x}$ and parameterized by $\mathbf{w}$. We consider a classification task where given a training set $\{\mathbf{x}_n, y_n\}_{n=1}^N$, where $y_n \in \{1, \ldots, K\}$, we wish to fit the parameters $\mathbf{w}$ by maximizing the log likelihood,

$$\mathcal{L} = \log \prod_{n=1}^{N} \frac{e^{f_{y_n}(\mathbf{x}_n;\mathbf{w})}}{\sum_{m=1}^{K} e^{f_m(\mathbf{x}_n;\mathbf{w})}}. \tag{14}$$

When the number of training instances is very large, the above maximization can be carried out by applying stochastic gradient descent (by minimizing $-\mathcal{L}$) where we cycle over minibatches. However, this stochastic optimization procedure cannot deal with large values of K because the normalizing constant in the softmax couples all scores functions so that the log likelihood cannot be expressed as a sum across class labels. To overcome this, we can use the one-vs-each lower bound on the softmax probability from Eq. (4) and obtain the following lower bound on the previous log likelihood,

$$\mathcal{F} = \log \prod_{n=1}^{N} \prod_{m \neq y_n} \frac{1}{1 + e^{-[f_{y_n}(\mathbf{x}_n;\mathbf{w}) - f_m(\mathbf{x}_n;\mathbf{w})]}} = -\sum_{n=1}^{N} \sum_{m \neq y_n} \log\left(1 + e^{-[f_{y_n}(\mathbf{x}_n;\mathbf{w}) - f_m(\mathbf{x}_n;\mathbf{w})]}\right) \tag{15}$$

which now consists of a sum over both data points and labels. Interestingly, the sum over the labels, $\sum_{m \neq y_n}$, runs over all remaining classes that are different from the label y_n assigned to $\mathbf{x}_n$. Each term in the sum is a logistic regression cost, that depends on the pairwise score difference $f_{y_n}(\mathbf{x}_n;\mathbf{w}) - f_m(\mathbf{x}_n;\mathbf{w})$, and encourages the n-th data point to get separated from the m-th remaining class. The above lower bound can be optimized by stochastic gradient descent by subsampling terms in the double sum in Eq. (15), thus resulting in a doubly stochastic approximation scheme. Next we further discuss the stochasticity associated with subsampling remaining classes.

The gradient for the cost associated with a single training instance $(\mathbf{x}_n, y_n)$ is

$$\nabla \mathcal{F}_n = \sum_{m \neq y_n} \sigma\left(f_m(\mathbf{x}_n;\mathbf{w}) - f_{y_n}(\mathbf{x}_n;\mathbf{w})\right)\left[\nabla_{\mathbf{w}} f_{y_n}(\mathbf{x}_n;\mathbf{w}) - \nabla_{\mathbf{w}} f_m(\mathbf{x}_n;\mathbf{w})\right]. \tag{16}$$

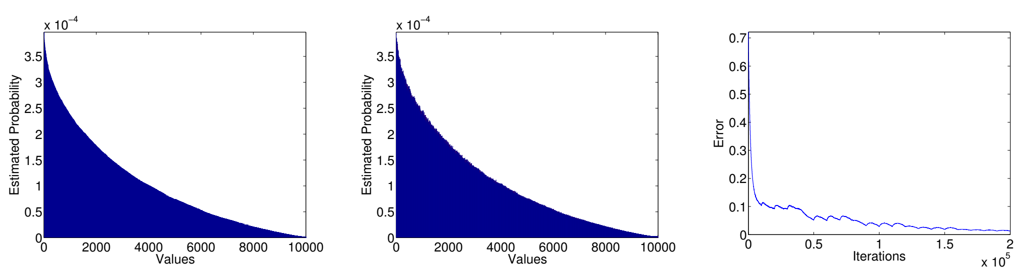
This gradient consists of a weighted sum where the sigmoidal weights $\sigma\left(f_m(\mathbf{x}_n;\mathbf{w}) - f_{y_n}(\mathbf{x}_n;\mathbf{w})\right)$ quantify the contribution of the remaining classes to the whole gradient; the more a remaining class overlaps with y_n (given $\mathbf{x}_n$) the higher its contribution is. A simple way to get an unbiased stochastic estimate of (16) is to randomly subsample a small subset of remaining classes from the set $\{m | m \neq y_n\}$. More advanced schemes could be based on importance sampling where we introduce a proposal distribution $p_n(m)$ defined on the set $\{m|m \neq y_n\}$ that could favor selecting classes with large sigmoidal weights. While such more advanced schemes could reduce variance, they require prior knowledge (or on-the-fly learning) about how classes overlap with one another. Thus, in Section 4 we shall experiment only with the simple random subsampling approach and leave the above advanced schemes for future work.

To illustrate the above stochastic gradient descent algorithm we simulated a two-dimensional data set of 200 instances, shown in Figure 1b, that belong to five classes. We consider a linear classification model where the score functions take the form $f_k(\mathbf{x}_n, \mathbf{w}) = \mathbf{w}_k^T \mathbf{x}_n$ and where the full set of parameters is $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_K)$. We consider minibatches of size ten to approximate the sum $\sum_n$ and subsets of remaining classes of size one to approximate $\sum_{m \neq y_n}$. Figure 1c shows the stochastic evolution of the approximate log likelihood (dashed red line), i.e. the unbiased subsampling based approximation of (15), together with the maximized exact softmax log likelihood (blue line), the non-stochastically maximized approximate lower bound from (15) (red solid line) and Bouchard's method (green line). To apply Bouchard's method we construct a lower bound on the log likelihood by replacing each softmax probability with the bound from (12) where we also need to optimize a separate variational parameter α_n for each data point. As shown in Figure 1c our method provides a tighter lower bound than Bouchard's method despite the fact that it does not contain any variational parameters. Also, Bouchard's method can become very slow when combined with stochastic gradient descent since it requires tuning a separate variational parameter α_n for each training instance. Figure 1b also shows the decision boundaries discovered by the exact softmax, one-vs-each bound and Bouchard's bound. Finally, the actual parameters values found by maximizing the one-vs-each bound were remarkably close (although not identical) to the parameters found by the exact softmax.

4 Experiments

4.1 Toy example in large scale non-parametric estimation

Here, we illustrate the ability to stochastically maximize the bound in Eq. (9) for the simple non-parametric estimation case. In such case, we can also maximize the bound based on the analytic formulas and therefore we will be able to test how well the stochastic algorithm can approximate the optimal/known solution. We consider a data set of $N = 10^6$ instances each taking one out of $K = 10^4$ possible categorical values. The data were generated from a distribution $p(k) \propto u_k^2$, where each u_k was randomly chosen in $[0, 1]$. The probabilities estimated based on the analytic formulas are shown in Figure 2a. To stochastically estimate these probabilities we follow the doubly stochastic framework of Section 3 so that we subsample data instances of minibatch size $b = 100$ and for each instance we subsample 10 remaining categorical values. We use a learning rate initialized to $0.5/b$ (and then decrease it by a factor of 0.9 after each epoch) and performed 2×10^5 iterations. Figure 2b shows the final values for the estimated probabilities, while Figure 2c shows the evolution of the estimation error during the optimization iterations. We can observe that the algorithm performs well and exhibits a typical stochastic approximation convergence.



(a) (b) (c)

Figure 2: (a) shows the optimally estimated probabilities which have been sorted for visualizations purposes. (b) shows the corresponding probabilities estimated by stochastic optimization. (c) shows the absolute norm for the vector of differences between exact estimates and stochastic estimates.

4.2 Classification

Small scale classification comparisons. Here, we wish to investigate whether the proposed lower bound on the softmax is a good surrogate for exact softmax training in classification. More precisely, we wish to compare the parameter estimates obtained by the one-vs-each bound with the estimates obtained by exact softmax training. To quantify closeness we use the normalized absolute norm

$$\text{norm} = \frac{|\mathbf{w}_{\text{softmax}} - \mathbf{w}_*|}{|\mathbf{w}_{\text{softmax}}|}, \tag{17}$$

where $\mathbf{w}_{\text{softmax}}$ denotes the parameters obtained by exact softmax training and $\mathbf{w}_*$ denotes estimates obtained by approximate training. Further, we will also report predictive performance measured by classification error and negative log predictive density (nlpd) averaged across test data,

$$\text{error} = (1/N_{test}) \sum_{i=1}^{N_{test}} I(y_i \neq t_i), \quad \text{nlpd} = (1/N_{test}) \sum_{i=1}^{N_{test}} -\log p(t_i|\mathbf{x}_i), \tag{18}$$

where t_i denotes the true label of a test point and y_i the predicted one. We trained the linear multiclass model of Section 3 with the following alternative methods: exact softmax training (SOFT), the one-vs-each bound (OVE), the stochastically optimized one-vs-each bound (OVE-SGD) and Bouchard's bound (BOUCHARD). For all approaches, the associated cost function was maximized together with an added regularization penalty term, $-\frac{1}{2}\lambda||\mathbf{w}||^2$, which ensures that the global maximum of the cost function is achieved for finite $\mathbf{w}$. Since we want to investigate how well we surrogate exact softmax training, we used the same fixed value $\lambda = 1$ in all experiments.

We considered three small scale multiclass classification datasets: MNIST[2], 20NEWS[3] and BIBTEX [12]; see Table 1 for details. Notice that BIBTEX is originally a multi-label classification dataset [2]. where each example may have more than one labels. Here, we maintained only a single label for each data point in order to apply standard multiclass classification. The maintained label was the first label appearing in each data entry in the repository files[4] from which we obtained the data.

Figure 3 displays convergence of the lower bounds (and for the exact softmax cost) for all methods. Recall, that the methods SOFT, OVE and BOUCHARD are non-stochastic and therefore their optimization can be carried out by standard gradient descent. Notice that in all three datasets the one-vs-each bound gets much closer to the exact softmax cost compared to Bouchard's bound. Thus, OVE tends to give a tighter bound despite that it does not contain any variational parameters, while BOUCHARD has N extra variational parameters, i.e. as many as the training instances. The application of OVE-SGD method (the stochastic version of OVE) is based on a doubly stochastic scheme where we subsample minibatches of size 200 and subsample remaining classes of size one. We can observe that OVE-SGD is able to stochastically approach its maximum value which corresponds to OVE.

Table 2 shows the parameter closeness score from Eq. (17) as well as the classification predictive scores. We can observe that OVE and OVE-SGD provide parameters closer to those of SOFT than the parameters provided by BOUCHARD. Also, the predictive scores for OVE and OVE-SGD are similar to SOFT, although they tend to be slightly worse. Interestingly, BOUCHARD gives the best classification error, even better than the exact softmax training, but at the same time it always gives the worst nlpd which suggests sensitivity to overfitting. However, recall that the regularization parameter λ was fixed to the value one and it was not optimized separately for each method using cross validation. Also notice that BOUCHARD cannot be easily scaled up (with stochastic optimization) to massive datasets since it introduces an extra variational parameter for each training instance.

Large scale classification. Here, we consider AMAZONCAT-13K (see footnote 4) which is a large scale classification dataset. This dataset is originally multi-labelled [2] and here we maintained only a single label, as done for the BIBTEX dataset, in order to apply standard multiclass classification. This dataset is also highly imbalanced since there are about 15 classes having the half of the training instances while they are many classes having very few (or just a single) training instances.

Further, notice that in this large dataset the number of parameters we need to estimate for the linear classification model is very large: $K \times (D+1) = 2919 \times 203883$ parameters where the plus one accounts for the biases. All methods apart from OVE-SGD are practically very slow in this massive dataset, and therefore we consider OVE-SGD which is scalable.

We applied OVE-SGD where at each stochastic gradient update we consider a single training instance (i.e. the minibatch size was one) and for that instance we randomly select five remaining classes. This

[2] `http://yann.lecun.com/exdb/mnist`

[3] `http://qwone.com/~jason/20Newsgroups/`

[4] `http://research.microsoft.com/en-us/um/people/manik/downloads/XC/XMLRepository.html`

Table 1: Summaries of the classification datasets.

Name	Dimensionality	Classes	Training examples	Test examples
MNIST	784	10	60000	10000
20NEWS	61188	20	11269	7505
BIBTEX	1836	148	4880	2515
AMAZONCAT-13K	203882	2919	1186239	306759

Table 2: Score measures for the small scale classification datasets.

	SOFT (error, nlpd)	BOUCHARD (norm, error, nlpd)	OVE (norm, error, nlpd)	OVE-SGD (norm, error, nlpd)
MNIST	(0.074, 0.271)	(0.64, 0.073, 0.333)	(0.50, 0.082, 0.287)	(0.53, 0.080, 0.278)
20NEWS	(0.272, 1.263)	(0.65, 0.249, 1.337)	(0.05, 0.276, 1.297)	(0.14, 0.276, 1.312)
BIBTEX	(0.622, 2.793)	(0.25, 0.621, 2.955)	(0.09, 0.636, 2.888)	(0.10, 0.633, 2.875)

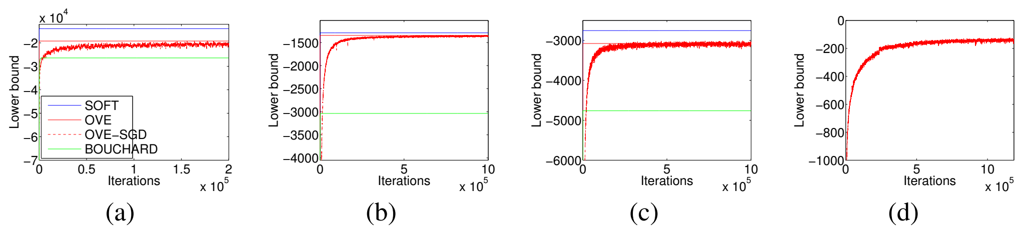


Figure 3: (a) shows the evolution of the lower bound values for MNIST, (b) for 20NEWS and (c) for BIBTEX. For more clear visualization the bounds of the stochastic OVE-SGD have been smoothed using a rolling window of 400 previous values. (d) shows the evolution of the OVE-SGD lower bound (scaled to correspond to a single data point) in the large scale AMAZONCAT-13K dataset. Here, the plotted values have been also smoothed using a rolling window of size 4000 and then thinned by a factor of 5.

leads to sparse parameter updates, where the score function parameters of only six classes (the class of the current training instance plus the remaining five ones) are updated at each iteration. We used a very small learning rate having value 10^{-8} and we performed five epochs across the full dataset, that is we performed in total 5×1186239 stochastic gradient updates. After each epoch we halve the value of the learning rate before next epoch starts. By taking into account also the sparsity of the input vectors each iteration is very fast and full training is completed in just 26 minutes in a stand-alone PC. The evolution of the variational lower bound that indicates convergence is shown in Figure 3d. Finally, the classification error in test data was 53.11% which is significantly better than random guessing or by a method that decides always the most populated class (where in AMAZONCAT-13K the most populated class occupies the 19% of the data so the error of that method is around 79%).

5 Discussion

We have presented the one-vs-each lower bound on softmax probabilities and we have analyzed its theoretical properties. This bound is just the most extreme case of a full family of hierarchically ordered bounds. We have explored the ability of the bound to perform parameter estimation through stochastic optimization in models having large number of categorical symbols, and we have demonstrated this ability to classification problems.

There are several directions for future research. Firstly, it is worth investigating the usefulness of the bound in different applications from classification, such as for learning word embeddings in natural language processing and for training recommendation systems. Another interesting direction is to consider the bound not for point estimation, as done in this paper, but for Bayesian estimation using variational inference.

Acknowledgments

We thank the reviewers for insightful comments. We would like also to thank Francisco J. R. Ruiz for useful discussions and David Blei for suggesting the name *one-vs-each* for the proposed method.

References

[1] Yoshua Bengio and Jean-Sébastien Sénécal. Quick training of probabilistic neural nets by importance sampling. In *Proceedings of the conference on Artificial Intelligence and Statistics (AISTATS)*, 2003.

[2] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 730–738. Curran Associates, Inc., 2015.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] D. Bohning. Multinomial logistic regression algorithm. *Annals of the Inst. of Statistical Math*, 44:197–200, 1992.

[5] Guillaume Bouchard. Efficient bounds for the softmax function and applications to approximate inference in hybrid models. Technical report, 2007.

[6] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[7] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.

[9] Siddharth Gopal and Yiming Yang. Distributed training of large-scale logistic models. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 289–297. JMLR Workshop and Conference Proceedings, 2013.

[10] Tzu-Kuo Huang, Ruby C. Weng, and Chih-Jen Lin. Generalized Bradley-Terry models and multi-class probability estimates. *J. Mach. Learn. Res.*, 7:85–115, December 2006.

[11] Shihao Ji, S. V. N. Vishwanathan, Nadathur Satish, Michael J. Anderson, and Pradeep Dubey. Blackout: Speeding up recurrent neural network language models with very large vocabularies. 2015.

[12] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. In *In: Proceedings of the ECML/PKDD-08 Workshop on Discovery Challenge*, 2008.

[13] Mohammad Emtiyaz Khan, Shakir Mohamed, Benjamin M. Marlin, and Kevin P. Murphy. A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012*, pages 610–618, 2012.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[15] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1751–1758, 2012.

[16] F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 246–252. Citeseer, 2005.

[17] Ulrich Paquet, Noam Koenigstein, and Ole Winther. Scalable Bayesian modelling of paired symbols. *CoRR*, abs/1409.2824, 2012.

[18] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[19] Sudheendra Vijayanarasimhan, Jonathon Shlens, Rajat Monga, and Jay Yagnik. Deep networks with large output spaces. *CoRR*, abs/1412.7479, 2014.