Notes on Sparsemax

Lingpeng Kong
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA, 15213
lingpenk@cs.cmu.edu

Abstract

Sparsemax is an alternative to the *softmax* function. It encourages the output distribution to be sparse. In the notes, we summarize the ideas in (Martins and Astudillo, 2016). We first present the form of *softmax* and its corresponding loss function during training. Then, we discuss the form of *sparsemax*, the reason why it promotes sparsity and how to derive its corresponding loss function.

1 Softmax

We use the *softmax* function (a.k.a. *normalized exponential*) when we want to "squashes" a set of scores $\boldsymbol{z} = \{z_1, z_2, \ldots, z_n\}, \boldsymbol{z} \in \mathbb{R}^N$ into a probability distribution $\boldsymbol{p} = \{p_1, p_2, \ldots, p_n\}$ where we have two properties – (1) if $z_i > z_j$, then $p_i > p_j$ and (2) $\sum_i p_i = 1$, which guarantees a valid probability distribution. Component-wisely, we have

$$p_i = \frac{\exp z_i}{\sum_j \exp z_j} \tag{1}$$

In a multi-class classification problem, where we are given a training point $(\boldsymbol{x}^{(i)}, y^{(i)})$, where $\boldsymbol{x}^{(i)}$ is the *feature vector* and $y^{(i)}$ is the positive integer index for the class. During training, the loss function associated with *softmax* is what we usually call *neglogsoftmax* (i.e. negative log-likelihood or logistic loss), defined as follows:

$$L(\boldsymbol{x}^{(i)}, y^{(i)}) = L_{\text{softmax}}(\boldsymbol{z}; k) = -\log \text{softmax}_k(\boldsymbol{z}) = -z_k + \log \sum_j \exp z_j \tag{2}$$

where $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}^{(i)})$ is the score vector (z_i is the score for class i) and $k = y^{(i)}$ is the *gold* class index. The total loss of the training set is the sum of all the loss of the points in the set.

Note here we introduce two different things, the *softmax* funtion (i.e. $\text{softmax}(\boldsymbol{z})$) and the *softmax loss* (i.e. $L_{\text{softmax}(\boldsymbol{z};k)}$). During training, we need to compute $\nabla_{\boldsymbol{z}} L_{\text{softmax}(\boldsymbol{z};k)}$ as follows:

$$\nabla_{\boldsymbol{z}} L_{\text{softmax}}(\boldsymbol{z}; k) = -\boldsymbol{\delta}_k + \text{softmax}(\boldsymbol{z}) \tag{3}$$

$\boldsymbol{\delta}_k$ denotes the delta distribution on k, $[\delta_k]_j = 1$ if $j = k$, and 0 otherwise.

2 Sparsemax

Although *softmax* is appealing in many different ways, it is not *sparse*, which is an ideal *bias* we want the model to have when solving a lot of problems. The meaning of the word *sparse* here is similar as it is used in the context of *L1-regularization* (Tibshirani, 1996). Basically, we want the solution $\boldsymbol{p}$ to contain as many zeros as possible (i.e. a *sparse* distribution) unless there are enough evidences in the data indict the model should assign $p_i > 0$ to class i. As we can see in the *softmax* case (e.q. 1), we have $p_i > 0$ for all i. This is not a *sparse* solution at all.

To promote the sparsity, *sparsemax* is defined as the euclidean projection of the input vector $\boldsymbol{z}$ onto the probability simplex (Martins and Astudillo, 2016). Essentially this means, given a point $\boldsymbol{z}$ in $\mathbb{R}^N$ space, find the point $\boldsymbol{p}$ on the simplex (which is a hyperplane in this space), where the euclidean distance from $\boldsymbol{p}$ to $\boldsymbol{z}$ is minimized. The intuition behind this can be explained in two aspects. First, projecting $\boldsymbol{z}$ into a simplex guarantee the solution to sum up to 1 (hence a valid distribution)[1]. Second, because the projection usually falls on the boundary of the simplex, it promotes sparsity naturally[2].

Euclidean projection onto a simplex has a closed-form solution, therefore, the sparsemax function is defined as (componentwise)

$$\text{sparsemax}_i(\boldsymbol{z}) = [z_i - \tau(\boldsymbol{z})]_+ \tag{4}$$

$$\tau(\boldsymbol{z}) = \frac{(\sum_{j \in S(\boldsymbol{z})} z_j) - 1}{|S(\boldsymbol{z})|} \tag{5}$$

where $S(\boldsymbol{z}) := \{j \in [K] \mid \text{sparsemax}_j(\boldsymbol{z}) > 0\}$ is the support of $\text{sparsemax}(\boldsymbol{z})$.

The next problem we need to solve is to find an appropriate loss function for *sparsemax*. The most straight-forward solution of this is to modify the *softmax* in equation 2 to *sparsemax*.

$$L(\boldsymbol{x}^{(i)}, y^{(i)}) = L_{\text{sparsemax}}(\boldsymbol{z}; k) = -\log \text{sparsemax}_k(\boldsymbol{z}) \tag{6}$$

Unfortunately, this is not doable, because *sparsemax* assigns probability *exact* zero to some of the labels. Any model that assigns zero probability to a gold label would zero out the probability of the entire training set.

Therefore, the solution here is, instead of replacing the *softmax* to *sparsemax* in equation 2, do it in equation 3.

$$\nabla_{\boldsymbol{z}} L_{\text{sparsemax}}(\boldsymbol{z}; k) = -\boldsymbol{\delta}_k + \text{sparsemax}(\boldsymbol{z}) \tag{7}$$

From equation 7, we can compute that there exists a function $L_{\text{sparsemax}}$, whose differentiation follows this form. The function is

$$L_{\text{sparsemax}}(\boldsymbol{z}; k) = -z_k + \frac{1}{2} \sum_{j \in S(\boldsymbol{z})} (z_j^2 - \tau^2(\boldsymbol{z})) + \frac{1}{2} \tag{8}$$

$S(\boldsymbol{z})$ and τ are defined as in equation 5. That is the loss function associated with *sparsemax* we are looking for.

3 Discussion

For the gradient computation and implementation details, please refers to the paper and an implementation of this at `https://github.com/clab/cnn`.

[1] Using the projection to make the solution in the valid space is actually a very general idea in optimization (e.g. projected gradient descent).

[2] Think about the case when $\boldsymbol{z} \in \mathbb{R}^2$. The simplex here is a line segment between (0,1) and (1,0) in the 2-D space. In this space, only the points between the line $y = x + 1$ and $y = x - 1$ yield fractional solutions.

References

André FT Martins and Ramón F Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.