Extended classification with modified Perceptron

Gunnar Rätsch and Jyrki Kivinen

We use a Bioinformatics problem of classifying splice sites of the nematode C. elegans to motivate extensions of the standard Perceptron algorithm. The task is essentially to solve a two class classification problem, but there are several issues to be taken into account. First the data set is rather large (200 000–300 000 training examples) and there are about twelve times more negative examples than positive. The Perceptron or similar iterative – for instance SMO-like – algorithms seem to be best suited to approach this kind of problem. However, the standard formulations do not take the uneven class distributions into account and may not have a bias term (as for instance the standard Perceptron).

In a first approach we consider a Perceptron algorithm with different sized (normalized) margins for the positive and negative class, γ^+ and γ^-, respectively. We have proven a mistake bound of the following type

$$a^2 M^+ + (1-a)^2 M^- \leq \frac{\max_n \|\boldsymbol{x}_n\|_2^2}{4\gamma^2},$$

where γ is the (normalized) margin of the problem and the a is a parameter of the algorithm which controls the trade-off between the number of mistakes M^+ and M^- of the positive and the negative class, respectively. This algorithm achieves good results in the splice site recognition problem – comparable to SVMs. However, we find that the number of updates is quite large (as the number of SVs in SVMs as well). For our application problem at hand we find that it is actually not necessary to classify a positive example correctly against all negative examples, but only against a small set of about twelve negative examples (the other alternatives near the site of interest).

To solve this problem we propose an extension of the Perceptron algorithm to solve the so-called *ordinal regression* problem, where one is given relations between certain examples. In our case we know that the *biochemical activity* of the true site has to be larger than the activities of all (local) alternatives for this site (so far ignoring alternative splicing). The idea of the algorithm is that one always receives two examples and has to predict the relationship between the these examples. If there is a mistake, then both examples are updated. On the theoretical side, we have shown that the number of mistakes M of the *online ordinal regression algorithm* is bounded by

$$M \leq \frac{\max_n \|\boldsymbol{x}_n\|_2^2}{\gamma^2},$$

where γ is the normalized "margin" for the problem:

$$\gamma = \max_{\|\mathbf{w}\|_2=1} \quad \min_{\mathbf{x}^+ \in \mathcal{C}^+, \mathbf{x}^- \in \mathcal{C}^-} \langle \mathbf{w}, \mathbf{x}^+ - \mathbf{x}^- \rangle$$

and $\mathcal{C}^+$ and $\mathcal{C}^-$ are the sets of positive and negative examples, respectively. Hence, we get the same number of updates as for the usual Perceptron algorithm. Empirically, we find that the resulting algorithm generates solutions with much less active examples (SVs) than the standard formulation and SVMs and is therefore considerably faster in classifying new data – without loss of accuracy. Preliminary experiments show that the proposed algorithm performs better than the Perceptron algorithm, but not as good as the SVM with appropriately tuned regularization parameter. This suggests that a regularized version of our proposed algorithm might lead to further improvements.

The standard Perceptron does not include a bias and the usual way to implement a bias is to augment the vectors by a dimension containing a constant. However, this leads to an additional factor of at least 4 in the mistake bound. Exploiting the fact that a special case of the online ordinal regression approach is equivalent to the standard classification case with bias, we show that the number of updates U of a perceptron with *implicit bias* is bounded as for the perceptron without bias:

$$U \leq \frac{\max_n \|\boldsymbol{x}_n\|_2^2}{\gamma^2},$$

where γ is the normalized margin of the problem. Note, however, that the number of mistakes might be larger. This algorithm is particularly useful when using the online algorithm in a batch setting.

So far we have explored only a small subset of possible algorithms related to ordinal regression: there are many combinations thinkable: online vs. batch; 2-class vs. ranking vs. ordinal regression; regularized (or not); single vs. ensemble versions, etc. The extension to regularized online ordinal regression seems to be most promising for our problem (work in progress).