When is Single-Output Joint Inference Better?

Hal Daumé III

February 8, 2006

This is a very hand-wavy analysis, but it might still shed some light. I'm not going to consider structured prediction, since dependencies between labels confound the issue too much. Let's say we're trying to predict $f : \mathcal{X} \to -1, +1 \times -1, +1$. That is, just a simple four-way classification problem. I'll refer to the first class as a and the second as b. In the single-output case, I'll only care about the output on a.

I'm going to situate myself in the linear SVM model, with a handful of "separate" weight vectors. For the single-output case, we have a single weight vector $\boldsymbol{w}$. In the multi-output case, we have three: $\boldsymbol{w}^a$, $\boldsymbol{w}^b$ and $\boldsymbol{w}^{ab}$. The classification is as:

$$f(x) = \operatorname*{argmax}_a \max_b \boldsymbol{w}^{a\top}\boldsymbol{x} + b\boldsymbol{w}^{b\top}\boldsymbol{x} + ab\boldsymbol{w}^{ab\top}\boldsymbol{x} \tag{1}$$

The first step in my analysis is to make a solution for the joint output case to the single output case. Suppose we learn the three joint output weight vectors; we want to find a single-output weight vector that behaves identically to Eq (1). This actually isn't possible: we need to increase our feature space (essentially using a poly-2 kernel). To do this explicitly, suppose $|x| = I$ and define:

$$\tilde{x}_{i+I*j} = x_i x_j \tag{2}$$

Then, we can define:

$$\tilde{w}_{i+I*j} = k w_i^a \left[w_j^b + w_j^{ab} \right] \tag{3}$$

$$k \geq 2 \max_{\boldsymbol{x}} \left| \boldsymbol{w}^{a\top}\boldsymbol{x} + b\boldsymbol{w}^{b\top}\boldsymbol{x} + ab\boldsymbol{w}^{ab\top}\boldsymbol{x} \right| \tag{4}$$

You can verify that:

$$\operatorname*{argmax}_a \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{x}} = \operatorname*{argmax}_a \max_b \boldsymbol{w}^{a\top}\boldsymbol{x} + b\boldsymbol{w}^{b\top}\boldsymbol{x} + ab\boldsymbol{w}^{ab\top}\boldsymbol{x} \tag{5}$$

This is essentially because we're replacing the max over b with a sum over b. By multiplying the response for b by k, we ensure that none of the other terms (the a terms) will out-rank the b terms. This gives us the same overall decisions wrt a.

Note that going from $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ quadratically increases the norm of the input. From $\boldsymbol{w}$ to $\tilde{\boldsymbol{w}}$ quadratically increases the norm of the weights, *times* k.

Now, let's consider the bounds described by (McAllester, 2004). Based on the PAC-Bayesian argument, we have:

$$L(Q(w), D) \leq KL^{-1}\left(\frac{1}{M}\left[L_4(w,S) + \|w\|^2 R^2\right]\right. \tag{6}$$
$$\left., \frac{1}{M-1}\left[\|w\|^2 R^2 \ln\left(\frac{2lM}{\|w\|^2 R^2}\right) + \ln\left(\frac{M}{\delta}\right)\right]\right)$$

With $L_4(w, S)$ at least the training error, M the number of training points, w the weight vector, R the maximal norm of the examples and l is roughly the number of classes (when applied in the multiclass setting).

Now, let's consider what happens when we move from the joint multi output setting to the single output setting. The difference is that in the single output setting, the norms $\|w\|^2$ and R^2 are quadratically smaller, but the loss L_4 is larger. In particular, without further assumptions, in the worse case, L_4 will be quadratically larger (errors are anti-correlated). (Also, l drops from 4 to 2.)

Considering the first term in the KL bound (since this roughly corresponds to the "mean" expected error), we get something like:

$$\frac{1}{M}\left[L_{multi} + \|w\|^5 R^4\right] \text{ versus } \frac{1}{M}\left[L_{multi}^2 + \|w\|^2 R^2\right] \tag{7}$$

The first term is when we follow my advice and just do single-output learning; the second term is when we do joint learning. The question is when is the second term lower. The second term is lower precisely when:

$$L \leq \frac{1}{2}\sqrt{1 - 4\|w\|^2 R^2 (1 - \|w^3\| R^2)} \tag{8}$$

This basically means that in order for the single-output joint inference to be better, we had better be in a problem for which low errors are achievable. For instance, if $\|w\| = R = 2$, then our loss had better be at most 22, independent of the number of examples.

It should be noted that this is not at all a formal argument. The biggest problem is that I'm comparing two upper bounds, rather than an upper bound and a lower bound. The next biggest problem is that I'm ignoring a lot of lower order terms that might add up to something significant in my norm analysis. I'm also assuming the worst case scenarios: when mapping $\boldsymbol{w}$ to $\tilde{\boldsymbol{w}}$, I assume that we can't find a better weight vector than the obvious correspondant. Similarly, the quadratic bound on the increase in loss is pessimistic. These are both pessimistic in the same direction: as the tasks become more correlated, both will drop. But it's not clear which will drop more quickly. There are also many other problems, like ignoring the second term in the KL bound, etc.

Anyway, this isn't intended as a final answer, just a little peek at what might be going on. I don't think we have the technology right now to fully answer the question.

References

McAllester, D. (2004). Relating training algorithms to generalization bounds in structured classification. NIPS Workshop on Learning with Structured Outputs.