On multiple kernel learning

O. Chapelle

Motivation

Multiple kernel learning has recently been a topic of interest [3, 4]. The setting is the following: given p kernel functions $K_1, \ldots, K_p$ that are potentially well suited for a given problem, find a linear combination of these kernels such that the resutling kernel $K = \sum \lambda_p K_p$ is "optimal" in some sense.

The aim of this presentation is to revisit some of the proposed approaches and to give both a well founded theoretical justification as well as a en efficient algorithm to learn this linear combination of kernels.

Theoretical justification

Margin has been argued to be a good quantity to maximize and that is the reason why the objective function that (hard margin) SVMs minimize is the invert squared margin. Let us define $M(K)$ as the minimum of this objective function for a kernel K. Based on this motivation, it has often been suggested to find the kernel matrix by minimizing M. We would like to point out that one has to be cautious with this approach. Indeed, the SVM objective function has been derived to find the hyperplane *given* a kernel, but there is no guarantee that this is sensible quantity to optimize for learning the kernel matrix. Actually, one can obtain arbitrary large margins by multiplying the kernel matrix by a large constant.

A well funded framework is to consider the λ_i as hyperparameters and to learn them using a *model selection* criterion [1]. Based on generalization error bounds for SVMs, [1] suggests for instance to use $\mathrm{tr}(K)M(K)$. This is equivalent to minimize $M(K)$ under constraint $\mathrm{tr}(K)$=constant. Since the SVM is invariant under translation, one can also use the recentered (in feature space) kernel matrix $\tilde{K}$ and the constraint become $\sum \lambda_i \mathrm{tr}(\tilde{K}_i)$ = constant, which is almost the same as the formulation of [3] but with a slightly different linear constraint.

Efficient optimization

The objective function $M(\sum \lambda_i K_i)$ is convex in λ. One can also compute in closed form its gradient and Hessian. We thus propose to find the coefficient λ by a Newton-type optimization, which is much more efficient than the SDP formulation of [3]. In our experiments only couple of steps are necessary to reach convergence. The most expensive part of the algorithm is the evaluation of M which requires an SVM training.

Linear case and feature selection

One can consider a special case where each kernel is the outer product between the training data on a given dimension: $[K_p]_{ij} = x_{ip}x_{jp}$. The multiple kernel learning algorithm will effectively do feature selection in this case [2]: at the end of optimization, the λ_i corresponding to non important features are zero (or have small values).

Another interesting advantage of the linear kernel is that one can train the SVM by a primal minimization. Since this is a min-min problem (as opposed to a min-max problem if the SVM is trained in the dual), the weight vector w and the parameters λ can be optimized simultaneously. Again, we propose an efficient Newton-type method for this purpose.

This algorithm was applied in a multiclass text classification scenario. The multiclass training is done in a 1-vs-the-rest setting, but by having the same λ_i for all classifiers, we were able to achieve *simultaneous* feature selection (i.e. all classifiers use the same set of features).

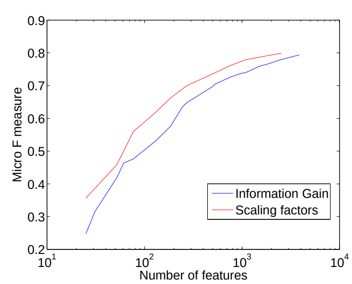

Figure 1: Accuracy of an SVM on the 20 Newsgroup dataset. The features have either been selected by Information Gain or according to the scaling factors λ_i.

References

[1] O. Bousquet and D. Herrmann. On the complexity of learning the kernel matrix. In *Advances in Neural Information Processing Systems*, 2003.

[2] Y. Grandvalet and S. Canu. Adaptive scaling for feature selection in svms. In *Advances in Neural Information Processing Systems*, 2002.

[3] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5, 2004.

[4] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7, 2006.