Adaptive Gaussian Kernel SVMs

Nathan Srebro and Sam Roweis

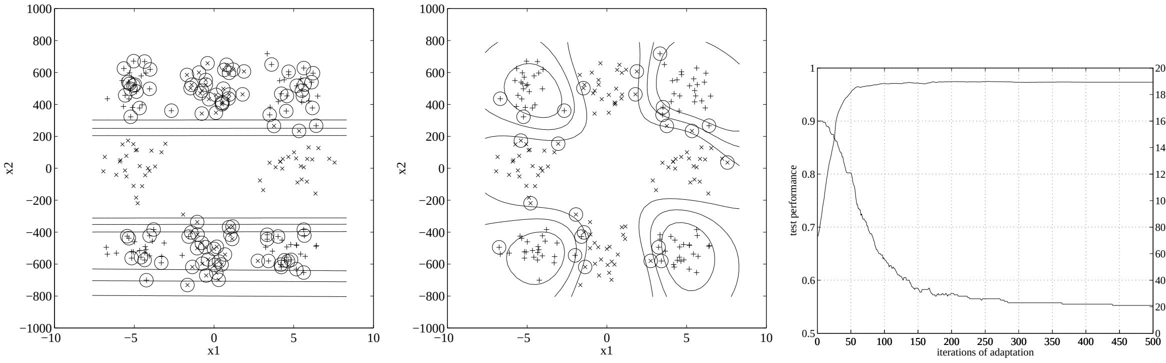
University of Toronto, Department of Computer Science

nati,roweis@cs.toronto.edu

We consider binary classification using Support Vector Machines with Gaussian kernels: $K_\Sigma(x_i, x_j) = e^{-(x_i-x_j)'\Sigma^{-1}(x_i-x_j)}$ and address the problem of selecting a covariance matrix Σ which gives good classification performance. As with other nonparametric classification methods based on distances between data points (such as nearest neighbor and Parzen methods), the choice of distance function (Σ) can be crucial to the success of learning. A good choice of Σ accounts for differences in the scales of different features (coordinates of the input vectors x), can remove global correlations between features and can adapt to the fact that some features may be much more informative about the class labels than others. We suggest an approach which ties training the SVM and choosing the distance function, as opposed to the more standard practice of feature normalization as a separate pre-processing step.

Several previous attempts have been made to "learn the kernel matrix", ranging from simply adapting the scalar "bandwidth" (size) of a spherical Gaussian kernel, through a general approach fitting several of kernel parameters [1], to learning a kernel which is a linear combination of a pre-specified set of basis or dictionary kernels [2] or learning kernels regularized through Hyperkernels [3].

In this work, we consider a simple but powerful and flexible family of Gaussian kernels by representing the covariance through its inverse square root: $\Sigma^{-1} = A^T A$ and directly learning a feature transformation matrix A. This is directly analogous to the approach taken by the Neighborhood Components Analysis algorithm [4] which learns a similar transformation for use with nearest neighbor classification. Instead of fixing the covariance matrix Σ and searching for large-margin linear classifier in the feature space induced by K_Σ, we suggest learning Σ *and* a large margin classifier jointly. We search for a kernel and classifier pair achieving large margin and small misclassification cost. We do so by



(a) (b) (c)

Figure 1: Data from two classes, each a mixture of four Gaussians. Note the different scale of the two axes. (a) SVM classification (decision boundary, margin, support vectors) with the best (in terms of test error) possible spherical Gaussian kernel $K_{\sigma I}$ and a slack penalty of $C = 10$. (b) SVM classification with the the kernel found using our alternating optimization method, with the same fixed slack penalty. (c) Fraction of test points correctly classified, and number of support vectors, for intermediate solutions of the alternating optimization.

alternately optimizing the classifier fixing the covariance Σ (standard SVM training), and optimizing Σ fixing the Lagrange multipliers α determining the classifier. The latter optimization is done by direct gradient descent of a smooth analogue to the SVM objective which replaces the hinge loss with a differentiable approximation. Although learning the maximum-margin classifier for fixed K_Σ is a convex optimization problem, the joint optimization problem, over the classifier *and* Σ, is not convex. Nevertheless, our initial experiments suggest the alternating optimization procedure described can successfully recover a matrix $A = \Sigma^{-1/2}$ yielding good SVM classifier performance (Figure 1).

Note that for Gaussian kernels, $K_\Sigma(x, x) = 1$ for all x and Σ, i.e. the implied transformation into feature space maps the data into the unit sphere, and so the geometric margin is directly comparable across different kernels K_Σ. The misclassification cost, as measured by a hinge loss (or similar cost) is also comparable across different Gaussian kernels. Thus, changes in the kernel which both increase the margin *and* lower the misclassification cost are always good. However, as with training a Support Vector Machine with a fixed kernel, it is not clear how to balance the margin and the misclassification error. Considering the entire set of attainable (margin,cost) pairs, the true object of interest is the exterior “path” of this set, i.e. the set of kernels K_Σ and classifiers for which it is not possible to increase the margin without increasing the misclassification cost, or reduce the misclassification cost without reducing the margin (where we are allowed to vary both the kernel and the classifier). This path is a one-dimensional manifold of kernel-classifier pairs, and our method is in effect a local search which tries to improve this path around a particular operating point. A more ambitious goal is to obtain the entire path, as can be done for a single kernel [5].

References

[1] O. Chapelle, V. Vapnik, O. Bousquet, and S. Makhuerjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

[2] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[3] Chen Soon Ong and Alexander Smola. Machine learning using hyperkernels. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

[4] Jacob Goldberger, Sam Roweis, Geoffrey Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*. 2005.

[5] Trevor Hastie, Saharon Rosset, Rob Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.