Kernels, Margins, and Low-dimensional Mappings*

Maria-Florina Balcan† Avrim Blum* Santosh Vempala‡

Abstract

Kernel functions are typically viewed as providing an implicit mapping of points into a high-dimensional space, with the ability to gain much of the power of that space without incurring a high cost if the result is linearly-separable by a large margin γ. However, the Johnson-Lindenstrauss lemma suggests that in the presence of a large margin, a kernel function can also be viewed as a mapping to a *low*-dimensional space, one of dimension only $\tilde{O}(1/\gamma^2)$. In this work, we explore the question of whether one can efficiently produce such low-dimensional mappings, using only black-box access to a kernel function. That is, given just a program that computes $K(x, y)$ on inputs x, y of our choosing, can we efficiently construct an explicit (small) set of features that effectively capture the power of the implicit high-dimensional space? We answer this question in the affirmative if our method is also allowed black-box access to the underlying data distribution (i.e., unlabeled examples). We also give a lower bound, showing that if we do not have access to the distribution, then this is not possible for an *arbitrary* black-box kernel function.

Our positive result can be viewed as saying that designing a good kernel function is much like designing a good feature space. Given a kernel, by running it in a black-box manner on random unlabeled examples, we can *efficiently* generate an explicit set of $\tilde{O}(1/\gamma^2)$ features, such that if the data was linearly separable with margin γ under the kernel, then it is approximately separable in this new feature space.

1 Extended Summary

Kernels functions have become a powerful tool in Machine Learning [9, 10, 11]. A kernel function can be viewed as allowing one to implicitly map data into a high-dimensional space and to perform certain operations there without paying a high price computationally. Furthermore, if the data has a large margin linear separator in that space, then one can avoid paying a high price in terms of sample size as well [4, 8].

The starting point for this work is the observation that if a learning problem indeed has the large margin property under some kernel $K(x, y) = \phi(x) \cdot \phi(y)$, then by the Johnson-Lindenstrauss lemma, a *random* linear projection of the "ϕ-space" down to a *low* dimensional space approximately preserves linear separability [1, 3, 5, 6]. Specifically, suppose data comes from some underlying distribution D over the input space X and is labeled by some target function c. If D is such that the target function has margin γ in the ϕ-space, then a random linear projection of the ϕ-space down to a space of dimension $d = O\left(\frac{1}{\gamma^2} \log \frac{1}{\varepsilon\delta}\right)$ will, with probability at least $1-\delta$, have a linear separator with error rate at most ε (see Arriaga and Vempala [3]). This means that for any kernel K and margin γ, we can, in principle, think of K as mapping the input space X into an $\tilde{O}(1/\gamma^2)$-dimensional space, in essence serving as a method for representing the data in a new (and not too large) feature space.

The question we consider in this work is whether, given kernel K, we can in fact produce such a mapping efficiently. The problem with the above observation is that it requires explicitly computing the function $\phi(x)$. In particular, the mapping of X into R^d that results from applying the Johnson-Lindenstrauss lemma is a function $F(x) = (r_1 \cdot \phi(x), \ldots, r_d \cdot \phi(x))$, where $r_1, \ldots, r_d$ are random vectors in the ϕ-space. Since for a given kernel K, the dimensionality of the ϕ-space might be quite large, this is not efficient. Instead, what we would like is an efficient procedure that given $K(.,.)$ as a black-box program, produces a mapping with the desired properties and with running time that depends (polynomially) only on $1/\gamma$ and the time to compute the kernel function K, with no dependence on the dimensionality of the ϕ-space.

*This work appears in Machine Learning Journal, 65(1):79 – 94, 2006 [2].

†Computer Science Department, Carnegie Mellon University. {ninamf,avrim}@cs.cmu.edu

‡College of Computing, Georgia Institute of Technology. vempala@cc.gatech.edu

Our main result is a positive answer to this question, if our procedure for computing the mapping is also given black-box access to the distribution D (i.e., unlabeled data). Specifically, given black-box access to a kernel function $K(x, y)$, a margin value γ, access to unlabeled examples from distribution D, and parameters ε and δ, we can in polynomial time construct a mapping $F : X \to R^d$ (i.e., to a set of d real-valued features) where $d = O\left(\frac{1}{\gamma^2} \log \frac{1}{\varepsilon\delta}\right)$ with the following property. If the target concept indeed has margin γ in the ϕ-space, then with probability $1 - \delta$ (over randomization in our choice of mapping function), the induced distribution in R^d is separable with error $\leq \varepsilon$. In fact, not only will the data in R^d be separable, but it will be separable with margin $\Omega(\gamma)$. Note that the logarithmic dependence on ε implies that if the learning problem has a perfect separator of margin γ in the ϕ-space, we can set ε small enough so that with high probability a set S of $O(d \log d)$ labeled examples would be perfectly separable in the mapped space. This means we could apply an arbitrary zero-noise linear-separator learning algorithm in the mapped space, such as a highly-optimized linear-programming package. However, while the dimension d has a logarithmic dependence on $1/\varepsilon$, the number of (unlabeled) examples we use to produce our mapping is $\tilde{O}(1/(\gamma^2\varepsilon))$. Our procedure uses two types of "random" mappings. The first is a mapping based on random examples drawn from D that is used to construct the intermediate space, and the second is a mapping based on Rademacher/binary (or Gaussian) random vectors in the intermediate space as in the Johnson-Lindenstrauss lemma.

Relation to Support Vector Machines and Margin Bounds: Given a set S of n training examples, the kernel matrix defined over S can be viewed as placing S into an n-dimensional space, and the weight-vector found by an SVM will lie in this space and maximize the margin with respect to the training data. Our goal is to define a mapping over the entire distribution, with guarantees with respect to the distribution itself. In addition, the construction of our mapping requires only unlabeled examples, and so could be performed before seeing any labeled training data if unlabeled examples are freely available.

Interpretation: Kernel functions are often viewed as providing much of the power of an implicit high-dimensional space without having to pay for it. Our results suggest that an alternative view of kernels is as a (distribution-dependent) mapping into a low-dimensional space. In this view, designing a good kernel function is much like designing a good feature space. Given a kernel, by running it in a black-box manner on random unlabeled examples, one can efficiently generate an explicit set of $\tilde{O}(1/\gamma^2)$ features, such that if the data was linearly separable with margin γ under the kernel, then it is approximately separable using these new features.

References

[1] D. Achlioptas, "Database-friendly Random Projections", Journal of Computer and System Sciences, Volume 66, Issue 4, pp. 671–687, 2003.

[2] M.-F. Balcan, A. Blum and S. Vempala "Kernels as Features: On Kernels, Margins, and Low-dimensional Mappings", Machine Learning Journal, 65(1):79 – 94, 2006.

[3] R. I. Arriaga, S. Vempala, "An Algorithmic Theory of Learning, Robust Concepts and Random Projection", Proceedings of the 40th Foundations of Computer Science, pp. 616–623, 1999. Journal version to appear in Machine Learning.

[4] P. Bartlett, J. Shawe-Taylor, "Generalization Performance of Support Vector Machines and Other Pattern Classifiers", Advances in Kernel Methods: Support Vector Learning, pp. 43–54, MIT Press, 1999.

[5] S. Dasgupta, A. Gupta, "An Elementary Proof of the Johnson-Lindenstrauss Lemma", Tech Report, UC Berkeley, 1999.

[6] W. B. Johnson, J. Lindenstrauss, " Extensions of Lipschitz Mappings into a Hilbert Space", Contemporary Mathematics, Volume 26, pp. 189–206, 1984.

[7] B. Scholkopf, C. J. C. Burges, S. Mika, "Advances in Kernel Methods: Support Vector Learning", MIT Press, 1999.

[8] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, M. Anthony, "Structural Risk Minimization over Data-Dependent Hierarchies", IEEE Transactions on Information Theory, Volume 44(5), pp. 1926–1940, 1998.

[9] J. Shawe-Taylor, N. Cristianini, "Kernel Methods for Pattern Analysis", Cambridge University Press, 2004.

[10] B. Scholkopf, K. Tsuda, J.-P. Vert, "Kernel Methods in Computational Biology", MIT Press, 2004.

[11] B. Scholkopf, A. J. Smola , "Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond", MIT University Press, Cambridge, 2002.