COMPSTAT'2004 Symposium

HIGH-DIMENSIONAL PROBABILISTIC CLASSIFICATION FOR DRUG DISCOVERY

Alexander Gray, Paul Komarek, Ting Liu, and Andrew Moore

Key words: virtual screening, high-throughput screening, classification, non-parametric, metric learning, nonisotropic, kernel density estimation.

COMPSTAT 2004 section: Biostatistics.

Abstract: Automated high-throughput drug screening constitutes a critical emerging approach in modern pharmaceutical research. The statistical task of interest is that of discriminating active versus inactive molecules given a target molecule, in order to rank potential drug candidates for further testing. Because the core problem is one of ranking, our approach concentrates on accurate estimation of unknown class probabilities, in contrast to popular non-probabilistic methods which simply estimate decision boundaries. While this motivates nonparametric density estimation, we are faced with the fact that the molecular descriptors used in practice typically contain thousands of binary features. In this paper we attempt to improve the extent to which kernel density estimation can work well in high-dimensional classification settings. We present a synthesis of techniques (SLAMDUNK: Sphere, Learn A Metric, Discriminate Using Nonisotropic Kernels) which yields favorable performance in comparison to previous published approaches to drug screening, as tested on a large proprietary pharmaceutical dataset.

1 Introduction: Classification for Drug Screening

Virtual screening refers to the use of statistical and computational methods for prioritizing candidate molecules for biological testing for their possible use as drugs. Because these assays are time-consuming and expensive, accurate "virtual" assays, or prioritization of molecules by computer, has direct impact in cost savings and more rapid drug development. Virtual screening, part of the more general enterprise of *high-throughput screening*, has thus become an increasingly pressing new component of modern drug development research.

The classification problem. In this paper we are concerned with the scenario of a large pharmaceutical research and development laboratory, which is as follows: We assume there is a single *target* molecule. There are multiple molecules which are known to interact in the desired fashion with the target molecule, *i.e.* are *active* with respect to the target, and a generally larger number of molecules known to be *inactive* with respect to the target. The task is to predict whether a previously unseen molecule will be active with respect to the target.

The features. The structure of a molecule determines its interaction with a target molecule – whether and how it will interlock, or "dock" with the target – but the interaction is itself a complex dynamic process whose complete characterization remains an oustanding problem of science. Thus, molecular descriptions used in virtual screening typically contain hundreds or thousands of binary (0/1) features, collecting all manner of both generic and target-specific properties which might be relevant to the classification task. Typical binary features record the absence or presence of a certain kind of atom or substructure, proximity relationship, and so on.

The goal. Our goal to design a classifier with the best possible prediction performance based on a proprietary commercial training set of 26,733 molecules, 6,348 binary features, and one output variable ("active" or not).

Recent work in virtual screening. Most of the well-known classification methods have been proposed for the virtual screening problem, including decision trees, neural networks, naive Bayes classifiers, and support vector machines (SVM) ([16]), which are currently considered to be one of the most empirically successful in general. Our work is strongly motivated by two of the most recently published comparisons of classification methods for virtual screening ([17],[10]), which reveal two slightly lesser-known winners. One is the 'binary kernel discriminator' (BKD) of [9], a simple kernel estimator for classification using a kernel based on the Hamming distance. (We note that the BKD is not formulated directly in terms of decision theory.) In [17], a fairly extensive comparison (by a different group of researchers than the ones who first proposed BKD's for this problem) between SVM's and BKD's was performed, demonstrating surprisingly clear superiority in the performance of BKD's over SVM's. In that work, molecule descriptions containing up to about 1,000 features were used. In [10], which performed experiments using the *same dataset* used in this paper, a conjugate gradient-based logistic regression (LR) method was demonstrated to have consistently favorable performance compared with several popular methods including SVM's with both linear and nonlinear (radial basis function) kernels, decision trees, naive Bayes classifiers, and k-nearest-neighbor classifiers. Our work ultimately contains aspects of both BKD and LR, achieving a method with performance superior to either one.

Ranking versus binary decision-making. To score the ranking performance of a classifier, we use the standard device of *receiver operating characteristic* (ROC) curves ([3]), which captures more information than simply the percentage of correctly-classified data.[1] The starting point for the approach of this paper is that the ranking problem is more difficult than the

[1] An ROC curve is constructed by sorting the data according to the predicted probability for the "active" class, *i.e.* $P(C_1|x)$. Starting at the origin and stepping through the data in order of decreasing "active" probability, a point on the curve is plotted by moving up one unit if the true label was actually "active" and moving right one unit if the prediction was incorrect. A summary of an ROC curve is the *area under the curve* (AUC), which is 0.5 for a classifier which guesses randomly and 1.0 for one which ranks perfectly.

standard classification problem because the quantity of interest is the posterior class probability rather than simply the error rate of making binary decisions. A classifier may estimate class probabilities with very large bias, but still perform well when scored in terms of accuracy in binary decision-making as long as the order relation between the class probabilities is maintained.

In this work we pursue the extent to which direct estimation of posterior class probabilities, as opposed to pure classification designed to minimize the binary error rate, might yield superior ranking performance. There are additional practical advantages to obtaining accurate class-conditional densities. Among them: imputation of missing data is naturally treated, outliers are more naturally identified, and difficult-to-classify data are easily isolated.

2 General Approach

Decision theory. The motivation above leads us naturally to the general framework of statistical decision theory. The posterior class probability $P(C_1|x)$, is expressed in terms of the class-conditional density $p(x|C_1)$:

$$P(C_1|x) = \frac{p(x|C_1)P(C_1)}{p(x|C_1)P(C_1) + p(x|C_2)P(C_2)} \tag{1}$$

where C_1 and C_2 are the two classes. If the class-conditional distributions on the right-hand side are known, the Bayes error rate is achived, *i.e.* the best possible performance is obtained.

Nonparametric density estimation. We consider the classifier obtained by estimating $p(x|C_1)$ and $p(x|C_2)$ with minimal assumptions, using the nonparametric *kernel density estimator* (KDE):

$$\hat{p}(x) = \frac{1}{N}\sum_{i}^{N} K_h(x, x_i) \tag{2}$$

where N is the number of data, $K()$ is called the kernel function and satisfies $\int_{-\infty}^{\infty} K_h(z)dz = 1$, and h is a scaling factor called the bandwidth. Kernel density estimation is the most widely-used and well-studied method for nonparametric density estimation, owing to both its simplicity and flexibility, and the many theorems establishing its consistency for near-arbitrary unknown densities and rates of convergence for its many variants ([14],[13]). We refer to the resulting classifier as a *nonparametric Bayes classifier* (NBC), for lack of a standard name. The standard form of kernel which is most often used is the *product kernel*, in which

$$K_h(x, x_i) = \prod_{d}^{D} K_d\left(\frac{\|x - x_i\|}{h}\right), \tag{3}$$

where D is the number of dimensions, *i.e.* the kernel function is a product of D univariate kernel functions, and all share the same bandwidth h. Though we could consider a setup in which separate bandwidths can be adjusted for each dimension, this creates a combinatorial problem which is intractable in our high-dimensional setting. If we ensure that the scales of the respective features are roughly the same, we need only adjust a single parameter h.

Thus, our classifier has two parameters, the bandwidth for each class. These are found by first estimating the optimal bandwidth for each density independently using least-squares cross-validation ([14]), then scoring bandwidth pairs nearby these values using the leave-one-out error score.

3 SLAMDUNK: Sphere, Learn A Metric, Discriminate Using Nonisotropic Kernels

The nonparametric Bayes classifier arises naturally when considering the best available method for accurate estimation of class probabilities with minimal assumptions. However, its power comes at potentially severe costs. The SLAMDUNK methodology consists of a set of procedures designed to mitigate the traditional limitations of nonparametric density estimation in the setting of high-dimensional classification, so that its distinct advantages may be exploited. We now treat in turn three significant roadblocks.

3.1 Fast Algorithms for Kernel Density Estimation

Estimation of the density at each of the N points, when performed in the straightforward manner, has $O(N^2)$ computational cost. Computational intractability impacts statistical inference quality directly – for example in [17] only 200 data were subsampled for each class to form the training set, due to the computational cost of BKD. In our experiments we use the entire set of 26,733 data. Any high-dimensional context demands the use of as much data as possible, forcing the computational issue. Fortunately, this problem has been largely mitigated in very recent work presenting a fast algorithm yielding simultaneously fast and accurate computation of kernel density estimates ([8]). The algorithm reduces the $O(N^2)$ cost to $O(N)$. Further, it is shown empirically in [8] that the algorithm's time complexity is not exponential in the dimension D, but instead appears to depend on the *intrinsic dimensionality*, the local dimensionality of the manifold upon which the data lies ([5]) (see below). However, the computational geometry methods employed by the algorithm require that the underlying distance be a true metric, which will constrain our methodology below.

3.2 Nonstationary and Nonisotropic Estimators

A major perceived obstacle is the statistical inefficiency of KDE in high dimensions. Theoretical bounds establish that in the worst case, the number of samples required for accurate kernel density estimation rises exponentially with the dimension ([14]). We now consider more realistic alternative choices for the kernel functions in KDE.

Nonstationary estimators. It has long been noted that the assumption of spatial *stationarity*, or a single scale h holding across the entire space is deficient. Visually it is clear that smoothing with a fixed bandwidth is unappealing when the dataset contains regions of differing density, which is inevitable in practice. Adaptive (or variable-kernel) kernel density estimators have been studied and shown to be more effective than fixed-width kernel density estimators in experimental studies, *e.g.* [15]. In these estimators, the variable bandwidth h_i for each point x_i is obtained by scaling the single global bandwidth h by a factor

$$\lambda_i \propto \{\tilde{p}(x_i)\}^{-1/2} \tag{4}$$

where $\tilde{p}()$ is a pilot estimate of the density, to which the overall estimator is largely insensitive ([1]). Many simple choices can be used for this pilot estimate, including adaptive Gaussian mixture models or variable kernel estimators based on nearest-neighbor distances ([2]).

Nonisotropic estimators. It has been noted by many authors (particularly in the field of machine learning, in which high-dimensional data classification and clustering is routinely performed) that in practice it is virtually never the case that a dataset's intrinsic dimensionality is equal to its explicit dimensionality D, *e.g.* [4]. With the assumption that the data lie on a linear manifold, the dimension of the subspace can be estimated using the eigenspectrum from a principal components analysis ([5]). However in general the data may lie on a nonlinear manifold ([12]). A common way estimator of the intrinsic dimension with minimal assumptions has been called, among other things, the correlation dimension ([7]), but amounts to the 2-point correlation function used in spatial statistics. Very often in practice the intrinsic dimension $D' << D$, regardless of which variant of its definition is used.

With this in mind, the standard product kernel, which is *isotropic*, *i.e.* has equal extent in all directions, is a poor match to realistic high-dimensional data. Further, as noted earlier, the behavior of volumes in high dimensionalities, rising exponentially in D, is disastrous when D is large. Instead we use an estimator in which the univariate bandwidths h_i are replaced by matrices H_i, resulting in a multivariate kernel such as the multivariate Gaussian

$$K_{H_i}(x, x_i) = \frac{1}{(2\pi)^{D/2}|H_i|^{1/2}} \exp\left\{-\frac{1}{2}(x - x_i)^T {H_i}^{-1}(x - x_i)\right\} \tag{5}$$

where $H_i = h\lambda_i\Sigma_k$, with Σ_k the covariance matrix estimated from x_i and its k nearest neighbors of x_i. Such estimators have received relatively little study, though one example showing their consistency is [6]. By allowing increased sensitivity to the local manifold of the data, or correlations in the feature space, we deflate the worst-case curse of dimensionality in KDE, relative to the naive product kernel estimator.

3.3 Coordinate Transformation and Metric Learning

One of Vapnik's central arguments for the non-probabilistic approach [16] underlying the SVM is that if the error rate is the desired quantity to be minimized, estimation of entire densities rather than simpler decision boundaries is unnecessary and wasteful of modeling capacity ([16]). We now introduce a methodology for essentially focusing less modeling effort on directions that are less relevant with respect to the decision boundary.

Metric learning. An implicit part of the kernel estimator is the underlying metric used to obtain the distances. The standard Euclidean distance is used by default. It can be seen as a special case of a more general weighted Euclidean distance

$$d(x,y) = \|x-y\| = \sqrt{(x-y)^T W (x-y)} \tag{6}$$

in which the matrix W is diagonal containing all 1's. We instead consider the metric weight matrix W as a free parameter to adjust to maximize the performance of our estimator. We refer to this as "learning the metric".[2] This functional form retains the metric properties, for the purpose of using the fast algorithm described above.

The linear discriminant metric. We propose a form of W which is diagonal, and relates the metric to the decision boundary. We obtain the vector w (the diagonal of W) which is the result of a linear classifier such as logistic regression or a linear support vector machine (we use logistic regression based on the favorable experimental results described earlier). The weight vector w describes a classifier where the class prediction for x is obtained by computing wx and comparing it to a threshold w_0. Thus if two points x and y lie on the decision boundary of the classifier, we have that $w^T(x-y) = 0$, *i.e.* the vector w is orthogonal to the decision boundary. By taking the metric formed by the norm

$$d(x,y) = \|w^T(x-y)\| \tag{7}$$

we obtain a metric which measures distance along w, or between the class means (with the appropriate Gaussian assumptions). This can be interpreted as measuring the extent to which the linear discriminant prefers class 1 or class 2. It can be regarded as an implicit form of dimensionality reduction, by realizing that values of w tending to zero will cause the metric to assign negligible weight in those directions, which is tantamount to removing the corresponding features.

Sphering. Our diagonal restriction on W motivates the removal of correlation between the features in advance. Normalizing each feature so that they all have roughly the same scale is also important for KDE as noted earlier.

[2] Although asymptotic results imply that the choice of metric does not affect performance, finite-sample experiments show that marked improvements can be made by adjusting the metric to the task at hand [4, 11].

For this reason we perform these operations (*sphering* the data) as the first step of our methodology using principal component analysis (PCA). We also take the opportunity at this stage to examine the resulting eigenspectrum and remove low-eigenvalue features. Our overall dimension reduction scheme thus includes two kinds of steps: this PCA-based explicit feature removal, which aims to 'denoise' the data, and the use of discriminant information to replace isotropy in the metric.

4 Experimental Results

Our dataset contains 26,733 rows and 6,348 attributes, and is sparse, containing 3,732,607 non-zero input values. It has 804 positive output values ("active" class). A pre-analysis of the data, however, reveals that 2290 columns are empty. Furthermore, 388 out of 8,235,711 pairs of columns are identical. These are also removed. Among the remaining columns, a column reduction scheme also reveals linear dependencies. Removal of 406 columns from the remaining 3,871 columns is performed. This leaves about half of the original dimensions. We then perform PCA, keeping only 100 of these dimensions. This value was chosen to correspond roughly to the inflection point of the eigenspectrum, as per common practice, and captured 97% of the variance in this case. All experiments were performed using 10-fold cross-validation, in which the data is broken into 10 equally-sized disjoint subsets, and testing (evaluation) is performed on one of them while training is performed on the other 9 put together.

Method	AUC
k-nearest neighbors	0.862 ± 0.017
Bayes classifier	0.891 ± 0.012
decision tree	0.893 ± 0.011
linear support vector machine	0.918 ± 0.010
RBF support vector machine	0.927 ± 0.013
logistic regression	0.931 ± 0.012
SLAMDUNK fixed isotropic kernel	0.933 ± 0.017
SLAMDUNK fixed isotropic + metric learning	0.937 ± 0.012
SLAMDUNK variable nonisotropic + metric learning	0.940 ± 0.012

The first part of the table lists the results of the experimental evaluation of [10] performed on the same data. Each method was tested both with and without the use of PCA projecting to 100 dimensions. The table shows only the better of the two procedures, for each method. The second part of the table shows the SLAMDUNK results on this data. We see a progression in the AUC score when metric learning is performed, and when variable and nonisotropic kernels are used, showing that these techniques each contribute to increased prediction quality in a non-conflicting and additive manner. [3]

[3]Test results are not shown for each of the possible combinations of the sub-techniques

5 Conclusion

We have presented a methodology called SLAMDUNK which we have designed to have favorable properties for the problem of virtual screening. We have demonstrated its favorable performance on a real pharmaceutical dataset in current use for drug discovery, providing evidence that this line of thinking may hold promise for this important contemporary problem. More generally, this work explores the extent to which fully probabilistic methods can be successful in high-dimensional problems.

References

[1] I. Abramson. On Bandwidth Variation in Kernel Estimation – A Square Root Law. *Annals of Statistics*, 10:1217–1223, 1982.

[2] L. Breiman, W. Meisel, and E. Purcell. Variable Kernel Estimates of Multivariate Densities. *Technometrics*, 19:135–144, 1977.

[3] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[4] J. H. Friedman. Flexible Metric Nearest Neighbor Classification. Technical report, Stanford University, 1994.

[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition, 2nd ed.* Academic Press, 1990.

[6] G. H. Givens. Consistency of the Local Kernel Density Estimator. Technical report, Colorado State University, 1994.

[7] P. Grassberger and I. Procaccia. Measuring the Strangeness of Strange Attractors. *Physica D*, pages 189–208, 1983.

[8] A. G. Gray and A. W. Moore. Very Fast Multivariate Kernel Density Estimation via Computational Geometry. In *Joint Statistical Meeting 2003*, 2003. to be submitted to JASA.

[9] G. Harper, J. Bradshaw, J. C. Gittins, and D. V. S. Green. Prediction of Biological Activity for High-Throughput Screening Using Binary Kernel Discrimination. *J. Chem. Inf. Comput. Sci.*, 41:1295–1300, 2001.

[10] P. Komarek and A. W. Moore. Fast Robust Logistic Regression for Large Sparse Datasets with Binary Outputs. In *Workshop on AI and Statistics*, 2003.

[11] T. P. Minka. Distance Measures as Prior Probabilities. Technical report, Massachusetts Institute of Technology, 2000.

[12] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500), December 2000.

[13] D. W. Scott. *Multivariate Density Estimation*. Wiley, 1992.

[14] B. W. Silverman. *Density Estimation*. Chapman and Hall, New York, 1986.

[15] G. Terrell and D. W. Scott. Variable Kernel Density Estimation. *Annals of Statistics*, 20(3):1236–1265, 1992.

[16] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[17] D. Wilton and P. Willett. Comparison of Ranking Methods for Virtual Screening in Lead-Discovery Programs. *J. Chem. Inf. Comput. Sci.*, 43:469–474, 2003.

Address: Carnegie Mellon University, Robotics Institute, 5000 Forbes Avenue, Pittsburgh PA 15221 *E-mail*: `agray@cs.cmu.edu`

of SLAMDUNK for lack of space.