Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. TR-113

Ranking on Data Manifolds

Dengyong Zhou,[1] Jason Weston,[1] Arthur Gretton,[1] Olivier Bousquet,[1] Bernhard Schölkopf[1]

June 2003

[1] Department of Schölkopf, Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany. Email: dengyong.zhou@tuebingen.mpg.de

Ranking on Data Manifolds

Dengyong Zhou, Jason Weston, Arthur Gretton
Olivier Bousquet and Bernhard Schölkopf
Max Planck Institute for Biological Cybernetics, 72076 Tuebingen, Germany
{*firstname.secondname* }@*tuebingen.mpg.de*

June 16, 2003

Abstract

The Google search engine has had a huge success with its PageRank web page ranking algorithm, which exploits global, rather than local, hyperlink structure of the World Wide Web using random walk. This algorithm can only be used for graph data, however. Here we propose a simple universal ranking algorithm for vectorial data, based on the exploration of the intrinsic global geometric structure revealed by a huge amount of data. Experimental results from image and text to bioinformatics illustrates the validity of our algorithm.

1 Introduction

The problem of ranking has become increasingly important since we entered the age of information explosion. To rank data, one must assign importance to objects given a query or queries. In information retrieval, one should return a list of documents ranked by their importance so that a user does not need to search the entire collection to find preferred documents. In bioinformatics, the protein ranking problem involves returning a ranked list of sequences that are likely to be evolutionarily related to a query sequence. This is of interest because two sequences that are descended from a common ancestral sequence are likely to fill similar functional roles in the cell. Many other instances also exist: ranking consumer products such as music, movies (sometimes called *recommender systems*), and other kinds of scientific phenomena (DNA sequences or binding molecules).

The Google search engine [2] accomplishes web page ranking by exploiting the global, rather than local, hyperlink structure of the Web [5]. Intuitively, it can be thought of as modelling the behavior of a random surfer on the graph of the Web, who simply keeps clicking on successive links at random and also periodically jumps to a random page. The web pages are ranked according to the probability distribution of the random walk. Empirical results show PageRank is superior to the naive ranking method, in which the web pages are simply ranked according to the sum of inbound hyperlinks and accordingly only the local structure of the Web is exploited. PageRank can only be used for graph data, however.

Here we propose a simple universal ranking algorithm, which exploits the intrinsic global geometric structure revealed by a huge amount of data [7]. We believe for many real world data types this should be superior to a local method, in which the vectorial data are simply ranked by pairwise Euclidean distances or inner products. Let us consider a toy problem to explain our motivation. We are given a set of points constructed in two moon patterns (Figure 1(a)). Given a query from the upper moon, the task is to rank the remaining points with respect to their similarities to the query. Intuitively, the degree of similarity of points in the upper moon to the query should decrease along the moon shape. This should also happen for the points in the lower moon. Furthermore, all of the points in the upper moon should be more similar to the query than the points in the lower moon. If we rank the points with respect to the query simply by Euclidean distance, then the left-most points in the lower moon will be more similar to the query than the right-most points in the upper moon (Figure 1(b)). Obviously this result is not consistent with our intuition (Figure 1(c)).

In general real-word vectorial data are often as highly structured as the above toy data [7]. To realize an algorithm that can learn such structure we take our inspiration from Google's Page Rank algorithm [2] and from work in semi-supervised learning [9, 10]. An intuitive description of our algorithm is as follows. We firstly define a weighted

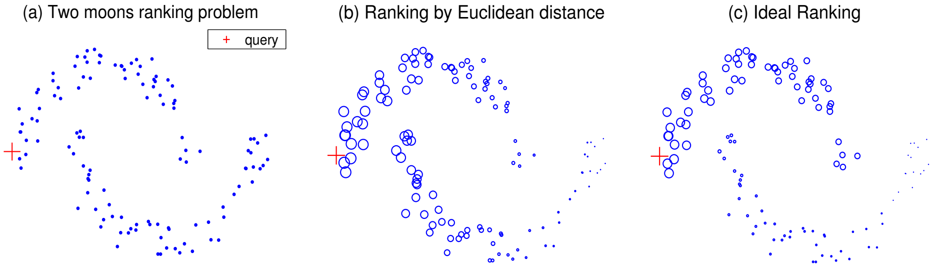


Figure 1: Ranking on two moons patterns. The marker size are proportional to the ranking in the last two figures. (a) toy data set with a single query; (b) ranking by the Euclidean distances; (c) ideal ranking result we hope to obtain.

network on the data and assign an authoritative score to every query. The query points act as source nodes that continually pump their authoritative scores to the remaining points via the weighted network, and the remaining points further spread the authoritative score they received to their neighbors. This spreading process is repeated until convergence, and the points are ranked according to the amounts of authoritative score they received. Note that our algorithm is given a query set and a database as input and is required to output a ranking: this is quite different from the domains of *preference relations* and *ordinal regression* which assume some objects should always be "preferred", independent of the query.

The rest of the paper is organized as follows. Section 2 describes our algorithm in detail. Section 3 presents experimental results on toy data, and image, text document and protein ranking problems. Section 4 concludes with a discussion and a description of further work.

2 Algorithm

Given a set of points $X = \{x_1, ..., x_s, x_{s+1} ..., x_m\}$ in $\mathbb{R}^n$, the first s points are the query points and the rest are the points that we want to rank with respect to the query points. Let $y_i(0) = 1$ if i is a query point, else $y_i(0) = 0$. The algorithm is:

1. Connect the two nearest points iteratively until a connected graph $G = (X, E)$ is obtained where E is the set of edges.
2. Form the affinity matrix K defined by K_{ij} =exp$(-\|x_i - x_j\|^2/2\sigma^2)$ if there is an edge $e(i, j) \in E$ or else $K_{ij} = 0$. Note that $K_{ii} = 0$ because there is no self-loop edge.
3. Compute the matrix L defined by $L = D^{-1/2}KD^{-1/2}$ in which D is the diagonal matrix with $D_{ii} = \Sigma_{j=1}^{m} K_{ij}$.
4. For all of the ranked points $(s < i \leq m)$ iterate $y_i(t + 1) = \Sigma_{j=1}^{s} L_{ij} y_j(0) + \alpha \Sigma_{j=s+1}^{m} L_{ij} y_j(t)$ where $\alpha \in [0, 1]$.
5. Let y_i^* denote the limit of the sequence $\{y_i(t)\}$. Then y_i^* is the ranking score of the i^{th} point (largest ranked first).

Now give detailed explanations about the steps in this ranking algorithm. First, a network on the data is created. This can be implemented by sorting the distances between the points in rising order and choosing a cut-off on the sorting list. We erase the self-spreading of the authoritative scores by setting the diagonal elements of K to zeros in the second step. The normalization in the third step is necessary to prove the algorithm's convergence. In the fourth step, the query points act as the source nodes to continually disseminate their authoritative scores to the ranked points and then these ranked points further spread the authoritative scores they received to their neighbors until a final globally stable state is arrived. The parameter $\alpha \in [0, 1]$ is used to control the authoritative scores received from the unlabeled neighbors. For $\alpha = 0$, no global structure is found, the algorithm ranks similarly to the original distance metric. Note that the authoritative scores of the query points are never changed, because they only spread their authoritative scores and don't receive the authoritative scores from their neighbors. Finally, all points are ranked by the accumulation of authoritative scores they have received.

See [9] for the convergence of this algorithm. Now compute the authoritative scores after this algorithm converges. Let $Y_U(t) = [y_{s+1}(t), ..., y_m(t)]^T$ and Y_U^* denote the limit of the sequence $\{Y_U(t)\}$. So $Y_U^* = [y_{s+1}^*, ..., y_m^*]^T$. By the fourth step of the above algorithm, we have

$$Y_U(t+1) = L_{US}Y_S + \alpha L_{UU}Y_U(t),$$

where L_{UU} and L_{US} are the sub-matrices of L with $S = \{1, ..., s\}$ and $U = \{s+1, ..., m\}$ and $Y_S = [y_1(0), ..., y_s(0)]^T$. Substituting the limit Y_U^* for $Y_U(t)$ and $Y_U(t+1)$, then

$$Y_U^* = L_{US}Y_S + \alpha L_{UU}Y_U^*,$$

which can be transformed into

$$(I - \alpha L_{UU})Y_U^* = L_{US}Y_S.$$

Since $I - \alpha L_{UU}$ is invertible [9], we have

$$Y_U^* = (I - \alpha L_{UU})^{-1}L_{US}Y_S.$$

This shows Y_U^* can be computed directly without iteration.

This algorithm can be applied to arbitrary object sets $X = \{x_i\}$ endowed with pairwise distances $d(x_i, x_j)$ defined by a real valued function $d : X \times X \rightarrow \mathbb{R}$ which satisfies $d(x_i, x_j) \geq 0$ and $d(x_i, x_j) = d(x_j, x_i)$. All we need to do is to substitute $d(x_i, x_j)$ for $\|x_i - x_j\|$ in the Gaussian function in the second step. The triangle inequality $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$ is not necessary for this algorithm. In addition, we also do not require the condition $d(x_i, x_j) = 0$ if and only if $x = y$ because the diagonal elements of L are always set to zeros.

The remaining problem is how to choose the parameters σ and α. Intuitively, σ specifies the local affinities between the points, and α controls the extend of spreading. In real world applications these parameters σ, which controls the local affinity between points, should be chosen by the practicioner, who uses experience and/or a small labeled dataset, e.g. as for the PageRank algorithm in web-page ranking. Potentially, if one was given a small labeled set or a query set greater than size 1, one could use standard cross validation techniques.

3 Experiments

We validate our method using a toy problem and three real-world domains: image, text, and protein ranking. As a true labeling is known in these problems, i.e. the image and document categories and the protein families (which is not true in all ranking problems), we can compute the error using the Receiver Operator Characteristic (ROC) score[4]. The ROC score is the normalized area of true positives versus false positives over each possible classification threshold, where positives are the same class as the query set. The returned score is between 0 and 1, a score of 1 being given to a perfect ranking.

In our algorithm we have two free parameters. We fixed $\alpha = 0.95$ in our experiments which corresponds to almost maximal propagation: such a choice should work for datasets with very clear manifold structure. We investigate the effect of changes in α in the toy dataset. The parameter σ was chosen by a validation set of labeled data.

3.1 Toy Problem

In this experiment we considered the toy ranking problem mentioned in the introduction section. The two moons toy data is shown in Figure 2(a). The connected graph described in the first step of our algorithm is shown in Figure 2(b).The authoritative scores with different time steps: $t = 5, 10, 50, 200$ are shown in Figures 2(c)-(f).Note that the scores on each moon decrease along the moon shape away from the query, and the scores on the moon containing the query point are larger than on the other moon. The closed form solution doesn't work with $\alpha = 1$ (Figure 2(g)), but works well with $\alpha = 0.99$ (Figure 2(h)). Ranking by Euclidean distance is shown in Figure 2(i), which fails to capture the two moons structure.

It is interesting to note that that when $\alpha = 1$, the ranking improves with continued iterations and then deteriorates, but the same does not happen even when $\alpha = 0.99$. It appears there is a phase transition in the iterations if $\alpha = 1$.

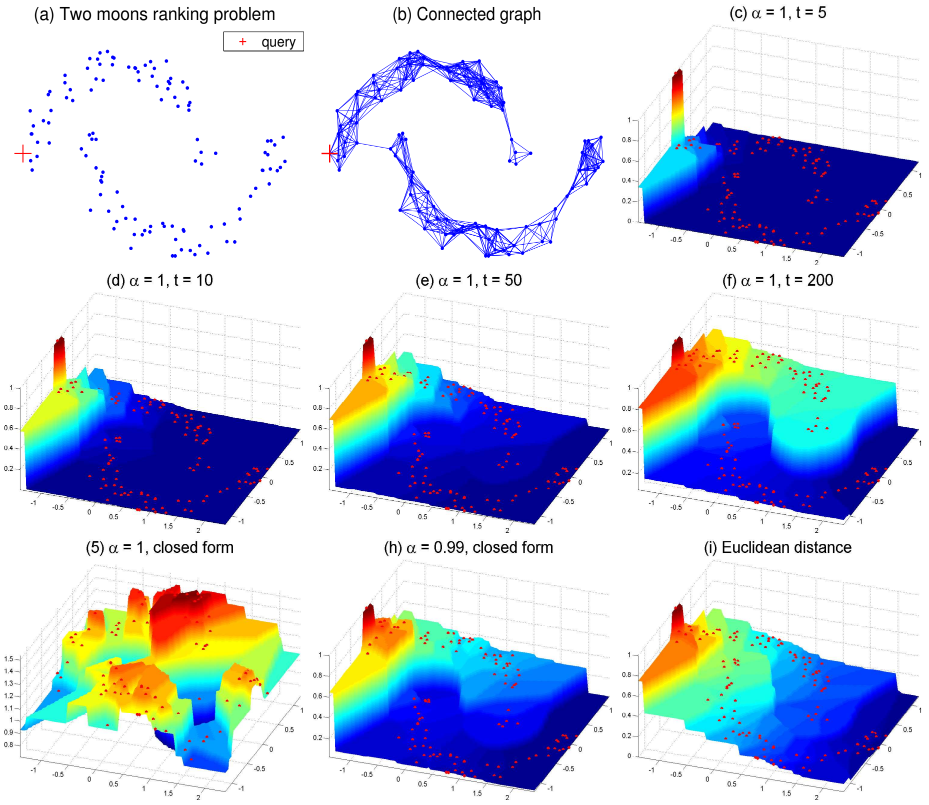


Figure 2: Ranking on two moons patterns. (a) toy data set with one query point; (b) connected graph; (c)-(f) ranking with different time steps: $t = 5, 10, 50, 200$; (g) closed form with $\alpha = 1$; (h) closed form with $\alpha = 0.99$; (i) ranking by simple Euclidean distances. Note that two moons structure was discovered in (h) but not in (i).

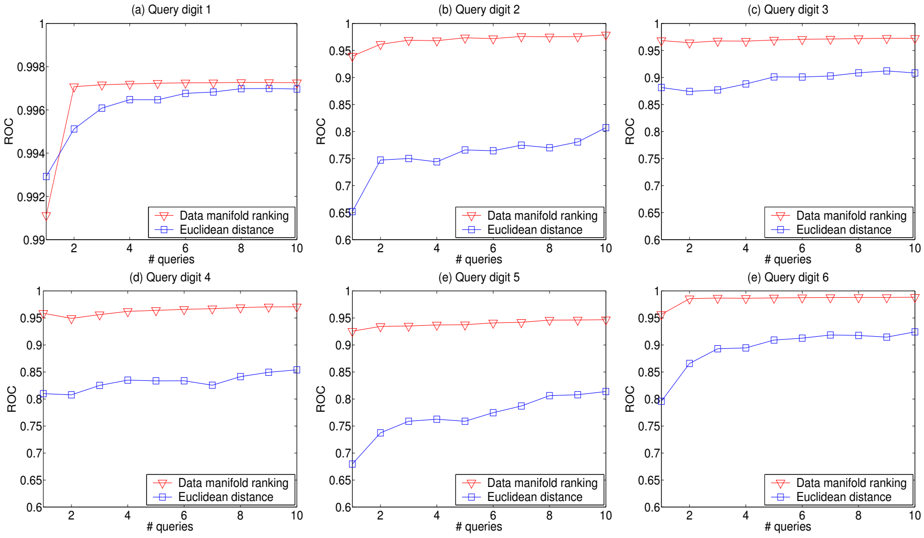
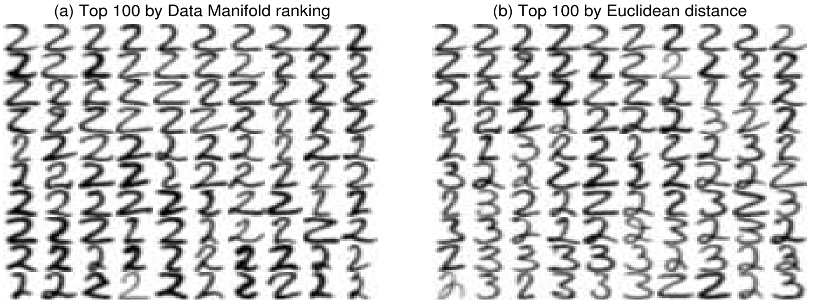


Figure 3: ROC on USPS for queries from digits 1 to 6. Note that this experimental results also provide indirect proof of the intrinsic manifold structure in USPS.



(a) Top 100 by Data Manifold ranking

(b) Top 100 by Euclidean distance

Figure 4: Ranking digits on USPS. The top-left digit in each figure is the query. (a) top 100 by Data Manifold ranking; (b) top 100 by Euclidean distance ranking. Note that there are much more *2*s with knots in the right figure.

3.2 Image Ranking

In this experiment we considered a task of ranking on USPS handwritten 16x16 digits dataset. We ranked digits from 1 to 6 in our experiments. There are 1269, 929, 824, 852, 716 and 834 examples for each class, with a total of 5424 examples. We randomly selected examples from one class of digits to be the query set over 30 trials, and then ranked the remaining digits with respect to these sets. We chosed $\sigma = 1.25$ produced by the validation method. We used the Euclidean distance based ranking method as a baseline: given a query set $\{x_s\}(s \in S)$, the points x are ranked, lowest value first, according to $\min_{s \in S} \|x - x_s\|$.

The results, measured as ROC scores, are summarized in Figure 3; each plot corresponds to a different query class, from digit one to six respectively. Our algorithm is comparable to the baseline when a digit *1* is the query. For the other digits, however, our algorithm significantly outperforms the baseline. This experimental result also provides indirect proof of the underlying manifold structure in the USPS digit dataset [7].

The top ranked 100 images obtained by our algorithm and the Euclidean distance method with a random digit *2* as the query are shown in Figure 4. The top-left point is the query. Note that there are some threes in the right figure. Furthermore, the *2*s in the left figure are more similar to the query than the twos in the right figure; and there are a large number of curly twos in the right figure, which do not match well with the query.

3.3 Text Ranking

In this experiment, we addressed a simple task of text document ranking. We used the Mac and Win subsets from the 20 newsgroups data set [10]. There are 961 and 985 examples in the two classes.

We selected one class of points as the query set, and then ranked the remaining points with respect to these seed points. We left out 25 positive examples for each task as a validation set for selecting σ, yielding $\sigma = 0.15$ and $\sigma = 0.25$ for Win and Mac queries respectively. We used the ranking method based on inner products as a baseline [3]. We also used normalized inner products to induce the pairwise distances, $d(x_i, x_j) = 1 - \langle x_i, x_j \rangle / \|x_i\| \|x_j\|$, to construct the affinity matrix used in our algorithm. The ranking accuracies for 100 randomly selected queries for each class are given in Figure 5. A Wilcoxon signed rank test of equality of medians shows that our method is significantly better than inner products with a significance level of $\alpha = 0.05$, returning a p-value of 1e-16 for Mac queries, and 1e-5 for Win queries.

3.4 Protein Ranking

In this experiment, we investigated the task of ranking a protein database relative to a given query, which is one of the most important applications in bioinformatics. A biologist supplies the amino acid sequence of a protein and expects highly ranked database sequences to be evolutionary related to the query, and hence to have similar structure and function. This is thus one of the key tools in protein function discovery. The most successful method so far is a web-based tool called PSI-BLAST [1] which returns a score for each protein in the database, calculated through heuristic alignment. One of its main advantages is that it exploits the structure of unlabeled data to

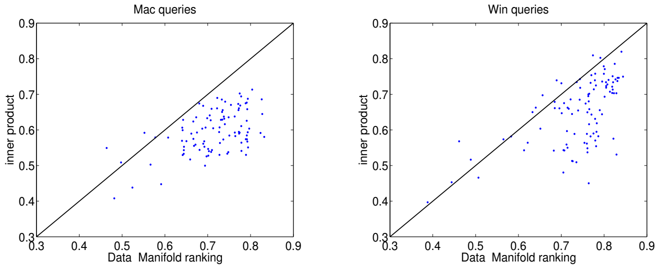


Figure 5: ROC score scatter plots of 100 random queries from the category Mac (left) and Win (right) comparing our method with inner products on the Mac vs. Win text categorization dataset.

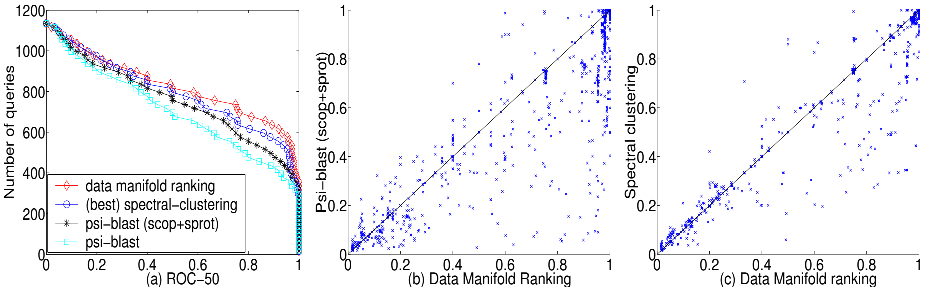


Figure 6: (a) plots the total number of queries for which a given method exceeds an ROC-50 score threshold, comparing our method with the original PSI-BLAST scores and an optimal spectral clustering induced representation; (b) & (c) show scatter plots of ROC-50 scores comparing these methods to our approach.

improve the ranking by iteratively building a more accurate profile of the query given its nearest neighbors. In this experiment we investigate if our method can improve on the ranking given by PSI-BLAST by capturing further global information. We do this by using the PSI-BLAST scores as a pairwise distance representation that serves as input to our algorithm[1]. This work builds on the approaches developed in [8].

We randomly chose 200 classes (protein *superfamilies*) from the SCOP 1.59 database resulting in 1070 proteins in our dataset. We took each protein in turn as the query (seed) and measured a method's ability to rank highly other members of the query class. This is usually measured using ROC-50 rather than ROC, which is the normalized area of false positives versus true positives for varying classification thresholds up to the first 50 false positives, a standard method in bioinformatics. The latter captures the fact that biologists are only interested in looking at the first few highest ranked hits, not the whole ranked list. We compared two variants of PSI-BLAST: PSI-BLAST(SCOP), which had access to the whole of the SCOP database (7329 examples) and PSI-BLAST(SCOP+SPROT) which also had access to a further 94,074 proteins from the Swiss-Prot 40 database, thus allowing the algorithm to better capture global structure. For our method we used the latter scores as the distance measure to form the matrix K, and we chose $\sigma = 1000$ which was the only power of 10 not producing a degenerate matrix (of all ones or all zeros). Note that we only trained our algorithm with the matrix of 1070 database proteins, thus handicapping our method compared to both PSI-BLAST methods.

Finally, methods such as spectral clustering [6] can also be used for ranking problems as they induce a distance measure from global cluster information: one simply ranks with the Euclidean distance using the new representation. We also tested spectral clustering using the same input matrix as our algorithm, for this algorithm we report the best parameters found by looking at the test error which were $\sigma = 1000$, $k = 100$, where k is the number of dimensions. The complete results are given in Figure 6. A Wilcoxon signed rank test of equality of medians shows that our method is significantly better than PSI-BLAST(SCOP+SPROT) and spectral clustering with a significance level of $\alpha = 0.05$, returning a p-value smaller than 1e-21. We expect further performance improvements when we

[1] Technically, PSI-BLAST scores do not give rise to a metric as the triangle inequality is sometimes violated, however our algorithm does not require this in any case.

train on the complete matrix of SCOP and Swiss-Prot sequences. The full results of this study will be presented in a forthcoming paper.

4 Conclusion

We proposed a simple universal ranking algorithm for vectorial data, the key idea of which is to exploit the intrinsic global geometric structure revealed by a huge amount of data. Experimental results from image and text to bioinformatics reveal the validity of this algorithm. We believe our method can give a significant improvement whenever the chosen distance measure gives local rather than global information and that there is global structure in the problem to be found in the given representation. We conjecture such a situation is often the case in real-world problems [7].

Future research should adress model selection. If labeled data is not available then it may be possible to look to the theory of stability of algorithms or other model selection approaches for clustering to choose appropriate hyperparameters. There are also a number of possible extensions to the approach. For example one could implement an *iterative feedback* framework: as the user specifies positive feedback this can be used to extend the query set and improve the ranking output. A second example is to perform constrained ranking, where one is also given some label knowledge. Finally, and most importantly, we are interested in applying this algorithm to a wide-ranging real-word problems.

Acknowledgments

We would like to thank Andre Elisseeff, Christina Leslie and William Stafford Noble for their assistance on the protein ranking task. We would also like to thank Andrew Ng for helpful discussions about spectral clustering.

References

[1] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3398–2402, 1997.

[2] S. Brin and L. Page. The anatomy of a large scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, pages 107–117, 1998.

[3] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, second edition, 2001.

[5] J. Kleinberg and S. Lawrence. The structure of the web. *Science*, 294:1849–1850, 2001.

[6] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[7] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[8] J. Weston, A. Elisseff, C. Leslie, and W. Noble. Machine learning approaches to protein ranking: discriminative, semi-supervised, scalable algorithms. Technical report, Max-Planck Institute for Biological Cybernetics, 2003. (in preparation).

[9] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schoelkopf. Learning with local and global consistency. Technical report, Max Planck Institute for Biological Cybernetics, 2003. (in preparation).

[10] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. 20th International Joint Conf. on Machine Learning*, 2003.