Weighting Features to Recognize 3D Patterns of Electron Density in X-ray Protein Crystallography

Kreshna Gopal[1] Tod D. Romo[2] James C. Sacchettini[2] Thomas R. Ioerger[1]

[1]Department of Computer Science, Texas A&M University

[2]Department of Biochemistry & Biophysics, Texas A&M University

[1]{kgopal,ioerger}@cs.tamu.edu [2]{tromo,sacchett@tamu.edu}

Abstract

Feature selection and weighting are central problems in pattern recognition and instance-based learning. In this work, we discuss the challenges of constructing and weighting features to recognize 3D patterns of electron density to determine protein structures. We present SLIDER, a feature-weighting algorithm that adjusts weights iteratively such that patterns that match query instances are better ranked than mismatching ones. Moreover, SLIDER makes judicious choices of weight values to be considered in each iteration, by examining specific weights at which matching and mismatching patterns switch as nearest neighbors to query instances. This approach reduces the space of weight vectors to be searched. We make the following two main observations: (1) SLIDER efficiently generates weights that contribute significantly in the retrieval of matching electron density patterns; (2) the optimum weight vector is sensitive to the distance metric i.e. feature relevance can be, to a certain extent, sensitive to the underlying metric used to compare patterns.

1. Introduction

Defining a suitable measure of similarity is a fundamental requirement of pattern recognition [11], instance-based learning [2], case based reasoning [25, 27] and other machine learning approaches [29]. Cases or instances are typically compared by a similarity or distance function based on numeric features that characterize the relevant aspects of the instances. Potentially useful features are generally defined by an expert or extracted by automated techniques [28], and a subset of these features are automatically selected (or weighted), based on relevance to the task at hand [3]. The approaches to feature selection and weighting can be categorized into two major groups: *filter* methods try to build classifiers that take into account some properties of the features involved, such as correlations, dependencies and other information [23]; the features are considered independently of the induction algorithm. Another paradigm, dubbed *wrapper*, use part of the data sample to iteratively evaluate the subset of selected features by techniques such as cross-validation i.e. the features are selected based on the bias of the induction algorithm [22]. The challenges of defining and determining relevance are also addressed in [6], which stresses on the need for studies on difficult data sets, especially with large number of attributes, and where a large proportion of attributes are irrelevant.

In this work, we focus on SLIDER [17, 13], a filter approach that uses the following measure to evaluate the optimality of a set of weights: given a test instance, we look at how well the weighted features rank a known similar instance, relative to a set of known different ones. Another central idea in SLIDER is that evaluation is done for very specific weights where there is a switch in neighbors between a test region, its match and its mismatch. These "cross-over" weights are the ones which will influence accuracy of matching the most. Thus, by limiting the space of weights to be searched, and identifying the weights that make a significant difference, the efficiency and effectiveness of learning are largely ensured. In our empirical analysis, we compare different weighting schemes – uniform (all features are selected, and weighted equally), binary (feature weights can be either 0 or 1) and continuous weights, where the latter two are derived from the output of the SLIDER algorithm. We also analyze the sensitivity of the weighting methods to different distance measures. In this work, we look at the Minkowsky family of distance metrics (of order 1, 2 and 3) i.e. Manhattan (L_1), Euclidean (L_2) and L_3.

Our empirical investigation is in the domain of protein crystallography, where 3D patterns in an *electron density map* have to be recognized and fitted with molecular structural components (or amino acids) to determine the structure of a protein macromolecule (Figure 1). TEXTAL™ [21] is a system that automates this process of electron density map interpretation; it uses nearest neighbor learning [12] and case-based reasoning to recognize patterns of electron density (in small spherical regions in a density map) by comparing them to known solved patterns stored in a case-base. Good matches are retrieved and assembled together to create a protein model, guided by knowledge of the domain, either explicitly stated (like typical stereo-chemical constraints of proteins) or implicitly encoded in the solved cases.

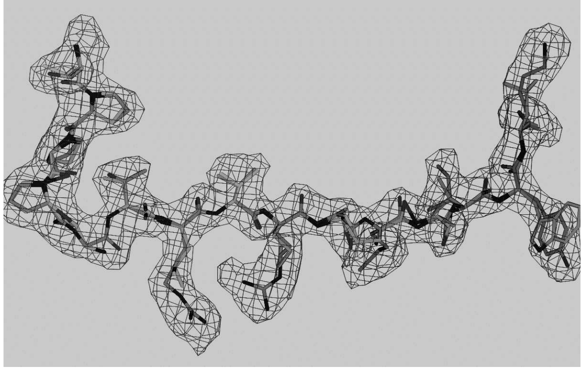


Figure 1. A portion of an electron density map which shows the contours of the intensity of electrons around that part of the protein. The correct structure (atoms and bonds) has been fitted in this density map.

The rest of this paper is organized in sections which discuss the following: (1) the significance and challenges of the protein crystallography domain, and an overview of the TEXTAL™ system; (2) the methods that TEXTAL™ uses to efficiently compare and retrieve matching cases from a (large) case-base; (3) the features that experts defined in this domain; (4) the SLIDER algorithm; (5) empirical results on the weights returned by SLIDER and their effectiveness in finding matching density patterns, for the three Minkowsky distance metrics; (6) analysis of the results, limitations and future work.

2. Protein crystallography & TEXTAL™

The *structural genomics* initiative is a worldwide effort to determine the structure of all proteins in a high-throughput mode [8,31]. This is motivated by the very rapid growth in the number of genomic sequences being discovered, since knowledge of the structure of proteins would shed light on the importance of genomic sequence regions, how the protein functions, and how drugs can be designed to effectively interact with proteins. Thus there has been a growing demand for high-throughput computational methods for protein structure determination, including rapid interpretation of electron density maps in X-ray crystallography. A density map is generated by the Fourier transformation of diffraction patterns obtained when X-rays are shone on the crystal of the protein. Interpreting a map essentially involves fitting known molecular structures known as amino acids into the density (Figures 1 and 2); there are 20 types of amino acids in nature, and proteins are essentially chains of typically 100-1000 amino acids that fold in complex 3D conformations. Maps are usually interpreted by crystallographers, with the help of visualization programs. The process can be laborious and challenging, especially since the map can be of poor quality (noisy and low resolution). There is also significant subjectivity in model building [30]; the difficulties in interpreting electron density are also discussed in [32].

One of the major difficulties is the fact that, to generate a density map, we need the intensity as well as phase information of diffracted patterns. But the phases cannot be experimentally determined, and have to be approximated by other means. This is known as the *phase problem*; thus the crystallographer has to go through various cycles of (1) interpret inaccurate maps, (2) improve phases from the model built, (3) use improved phases to generate better maps, which can be re-interpreted.

TEXTAL™ aims at automating this process of solving an electron density map, thereby saving up to weeks of effort required by an expert crystallography to interpret one map, especially if it is noisy and blurred. Given an unsolved map of a protein in XPLOR [7] format, TEXTAL™ first identifies the positions of special carbon atoms (called Cα's) which lie roughly at the center of each amino acid. This is achieved by a sub-system called CAPRA, or C-Alpha Pattern Recognition Algorithm [20]. Then spherical regions (of 5Å diameter) around each Cα are looked at; they are characterized by a set of numeric features, which are used to find matches from a case-base of solved patterns. These fragments of pre-determined structures are retrieved, and a macromolecular model is gradually built, by fitting the fragments together, subject to many constraints e.g. on bond lengths and angles. The model is also improved by aligning the sequence of amino acids obtained with the known sequence [34]; further refinement is done by moving the atoms slightly to improve the fit with the density – this process is known as real space refinement [9]. The

TEXTAL™ system is much larger is scope; for more details, refer to [21, 20, 17, 13] and http://textal.tamu.edu:12321. TEXTAL™ is also a component of PHENIX (http://www.phenix-online.org) [1], an integrated crystallographic computing environment.

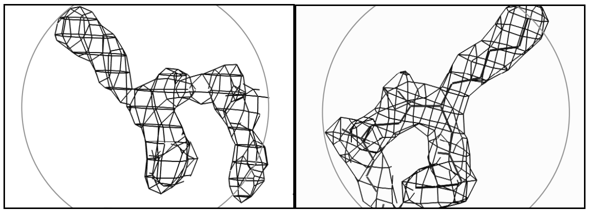


Figure 2. The pattern on the left represents a spherical region of unsolved electron density. An expert crystallographer would recognize the shape (with the help of a 3D visualization program) and model this region of the protein i.e. determine the positions of the atoms, and how they bond together (as shown in the pattern on the right). These two patterns are, in fact, identical but oriented differently in 3D space. Thus, rotation-invariant features are required to represent them.

In this paper, we focus on the choice of features to represent spherical regions of density patterns, and the determination of their weights for various distance functions that can be used to compare and retrieve the regions, based on k-nearest neighbor learning [12]. Before discussing the features and their weights, we first describe the general approach that TEXTAL™ uses to efficiently retrieve cases from a database.

3. Efficient case retrieval in TEXTAL™

Case-based reasoning systems typically need a large database of cases for wide problem coverage and high quality solutions. But large case-bases may cause degradation in efficiency, especially if the case matching function to determine similarity between two cases is expensive [35]. TEXTAL™ shares these challenges with many other case-based reasoning systems; in fact, given an unsolved spherical query pattern of electron density (q), the distance between q and each case c_i in the case-base can be determined, and the most similar (smallest distance) can be returned as the best match. One metric that can be used is the *density correlation* between q and c_i's, which computes the optimal superposition between two patterns. Since the number of possible 3D rotations is very large, this metric is too expensive, which we cannot afford to run over the whole case-base (of ~50,000 regions). Thus, we use an approximate, inexpensive, feature-based distance metric to filter a small number (say k = 500) of potential matches, and the density correlation procedure then makes the final ranking. In [15], we evaluate and compare various feature-based distance metrics for this approach, and argue in the favor of statistical and probabilistic measures as compared to geometric distances (like Euclidean).

It should be noted that a good match need not be the absolute best one according to the objective metric; it can be the top few matches (based on a tolerance on how high we wish the density correlation value to be to qualify for being a match). Given a query pattern, our aim is to try to get at least one good match (anywhere) in the top k, since the expensive objective will re-rank the top k matches, and identify the truly good ones. In [14], we discuss the effectiveness of this filtering scheme and how it depends on the level of tolerance of matching. We also discuss how the value of k is chosen, based on a loss function that represents the extent to which the fast feature-based distance measure approximates the objective density correlation metric.

4. Features in TEXTAL™

In TEXTAL™, the features used to characterize spherical regions of electron density patterns have been manually designed by domain experts. One important restriction is that the features have to be rotation-invariant, since patterns to be compared can occur in any 3D orientation. Four classes of features have been defined (Table 1): (1) *statistical* features like mean, standard deviation, skewness and kurtosis of electron density distribution for a set of grid points in the spherical region; (2) features based on *moments of inertia*, where the inertia matrix is computed, and various ratios of eigenvalues for the three mutually perpendicular moments of inertia are defined as features; (3) a feature that captures how *symmetric* or balanced the region is, based on the distance of each grid point within the pattern to its center of mass; (4) features that reveal the *shape* of the pattern – typically an amino acid have three "spokes" emanating from its Cα; these spokes are identified, and various features are calculated based on the angles between these spokes.

Furthermore, for each region, we calculate these features at 4 different radii (3, 4, 5 and 6 Å); this is necessary since amino acids vary in shapes and sizes, and each feature captures slightly different information for different sizes. Thus, the total number of features that we use is 19*4 = 76.

Table 1. Definition of features used to describe spherical density patterns in TEXTAL™. The features are grouped into 4 classes; each feature has 4 versions for different radii of the sphere (3, 4, 5 and 6 Å, where 1 Å = 10^{-10}m).

Feature class	Description of feature	Method of computation (ρ_i is the electron density value at the i^{th} of n grid points in a region)
Statistical	Mean	$\rho = (1/n)\Sigma\ \rho_i$
	Standard deviation	$[(1/n)\Sigma\ (\rho_i\text{-}\rho)^2]^{1/2}$
	Skewness	$[(1/n)\Sigma\ (\rho_i\text{-}\rho)^3]^{1/3}$
	Kurtosis	$[(1/n)\Sigma\ (\rho_i\text{-}\rho)^4]^{1/4}$
Moments of Inertia	Magnitude of primary moment	Compute inertia matrix, diagolize & sort eigenvales.
	Magnitude of secondary moment	
	Magnitude of tertiary moment	
	Ratio of primary to secondary moment	
	Ratio of primary to tertiary moment	
	Ratio of secondary to tertiary moment	
Symmetry	Distance to center of mass	$\|<x_c,y_c,z_c>\|$, where $x_c = (1/n)\Sigma x_i\rho_i$, $y_c = (1/n)\Sigma y_i\rho_i$, $z_c = (1/n)\Sigma z_i\rho_i$,
Shape	Minimum angle between spokes	Find 3 “spokes” i.e. 3 distinct vectors with highest density summation, and compute angle min, max, median, sum, etc.
	Maximum angle between spokes	
	Median angle between spokes	
	Sum of spoke angles	
	Radial sum of first spoke	
	Radial sum of second spoke	
	Radial sum of third spoke	
	Spoke triangle area	

5. The SLIDER algorithm

In this section we describe the SLIDER algorithm to optimize weights for the Minkowsky family of distance metrics. We first focus on two-component mixtures (i.e. involving two features, where their weights sum up to 1) and then extend it to an arbitrary number of features. The weighted Minkowsky distance of order n between two patterns x and y, using two features i and j is defined as:

$$D_{i,j}(x,y) = (w_i|x_i - y_i|^n + w_j|x_j - y_j|^n)^{1/n}$$

If n = 1, we get the Manhattan distance; if n = 2, the metric is called Euclidean, and in general it is known as the Minkowsky distance of order n. We can drop the n^{th} root, since it is a monotonic transformation. Thus $D_{i,j}$ is re-defined as:

$$\begin{aligned} D_{i,j}(x,y) &= w_i|x_i - y_i|^n + w_j|x_j - y_j|^n \\ &= (1 - w)|x_i - y_i|^n + w|x_j - y_j|^n \end{aligned}$$

where w is set to w_j, the weight of feature j. One approach to approximate the optimal pair of weights is to use a test set to exhaustively evaluate accuracy for various pairs of weights defined over a grid, such as {0.0, 0.1, 0.2, …, 1.0}. This method is inefficient and is limited by the coarseness of the grid sampling. SLIDER proposes a more efficient approach.

Consider an instance x that has y as its closest neighbor according to f_i, and z as its closest neighbor according to f_j i.e. the nearest neighbor of x is y when w = 0, and it is z when w = 1 (w is the weight of feature j). The point at which the $D_{i,j}(x,y) = D_{i,j}(x,z)$ is given by:

$$\begin{aligned} &(1 - w)|x_i - y_i|^n + w|x_j - y_j|^n \\ &\qquad = (1 - w)|x_i - z_i|^n + w|x_j - z_j|^n \end{aligned}$$

Solving for w, and setting it to w_0, we get:

$$w_0 = \frac{|x_i - z_i|^n - |x_i - y_i|^n}{|x_j - y_j|^n - |x_i - y_i|^n + |x_i - z_i|^n - |x_j - z_j|^n} \tag{1}$$

In other words, if w is “slided” from 0 to 1, there is a weight w_0 at which $D_{i,j}(x,y) = D_{i,j}(x,z)$; this point is called a “cross-over”, which in fact, is a weight at which there is a net increase (or decrease) in accuracy, depending on which of y and z is truly closer to x

(Figure 3). When there is an increase in accuracy, the cross-over is referred to as positive, and negative otherwise. It should be noted that not all 3-tuple of instances will have a cross-over for a given pair of features.

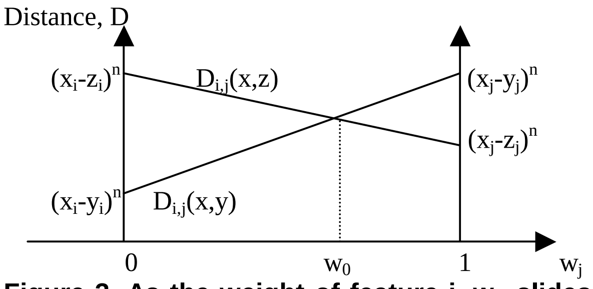


Figure 3. As the weight of feature j, w_j, slides from 0 to 1, the Minkowsky distance between x and y [$D_{i,j}(x,y)$] changes from less to greater than that between x and z [$D_{i,j}(x,z)$]. The "cross-over" occurs at w_0 i.e. there is a change in accuracy of prediction at w_0, depending on whether y or z is truly more similar to x.

Crossover points can also be determined by considering two subsets of features (instead of just two features). Consider two feature subsets A and B, with corresponding Minkowsky distances D_A and D_B respectively. A composite metric, D_{A+B}, can be defined as $D_{A+B}(x,y) = \lambda D_A(x,y) + (1 - \lambda) D_B(x,y)$. As λ is slided from 0 to 1, it may cause a switch of neighbors for three instances, as described earlier. Thus λ can be used to determine the new weight vector that increases accuracy, based on crossover points. Currently, SLIDER randomly chooses one feature (set A) and evaluates it against all remaining features (set B). The approach can be extended to compare feature sets of arbitrary size and composition.

The key idea behind SLIDER is to determine the cross-overs of many examples sliding over the weight of one feature at a time and determine the "optimum" weight value at which the overall accuracy increases the most. SLIDER uses a greedy approach [33] to iteratively choose a (random) feature, adjust its weight based on the above criterion, and stops when there is no net increase in accuracy.

Once all cross-over points are determined, we find the optimum weight (of the randomly chosen feature) which maximizes the difference between the number of positive cross-overs and negative cross-overs. This is done by first sorting the cross-over weights and initializing an accumulator to 0; we then sweep through the sorted list of weights, incrementing the accumulator when a positive cross-over is encountered, and decrementing the accumulator for every negative cross-over. The weight at which the accumulator reaches its peak is returned as the optimum weight.

The test procedure that we use to evaluate whether overall accuracy has improved by updating the weights is as follows: we define a test set S of pattern instances, and for each instance we find a match (high density correlation), and a set of mismatches (average or low density correlation). Given a weight vector and an instance, we compute the distance of that instance to the known match and mismatches; the rank of the match relative to the mismatches gives an estimate of the optimality of the weights. Given a weight vector w, a test set S of m cases, and for each case 1 match and n mismatches, we define the Ranking Consistency of w, RC(w) as follows:

$$RC(w) = 1/m \sum_i [n - rank(i)] \qquad (2)$$

where rank(i) is the rank of the match of i (relative to all n mismatches); note that lower rank implies more similar to the query pattern (i.e. the match should ideally have rank = 1).

The SLIDER algorithm is given in Figure 4. It should be noted that our objective function in this problem is a continuous metric (density correlation). But this approach can be extended to handle classification problems as well.

6. Results

SLIDER was used to optimize the weights independently for Manhattan (L_1), Euclidean (L_2) and Minkowsky distance of order 3 (L_3). 76 features were used, as described earlier. The weight vector in every iteration is evaluated through Ranking Consistency for a test set of various sizes (typically 500); for each query pattern in the test set, one match and 200 mismatches are pre-determined by the calculation of density correlation. The training examples were drawn from "ideal", artificially generated maps of proteins with known structures obtained from PDBSelect [16] (or http://www.cmbi.kun.nl/gv/pdbsel), a subset of the PDB database (http://www.rcsb.org/pdb) [5].

To evaluate the effectiveness of SLIDER in determining a final set of appropriate global weights for the filtering scheme described earlier, we use the following test procedure: we chose 200 regions that evenly cover the 20 different types of amino acids; the regions were obtained from a case-base generated from ~200 proteins from PDBSelect. For each test region, we exhaustively searched the case-base (of ~50,000 regions) to find their true, objective matches

```
Inputs: 1. Test set $S = \{S_1,\ldots,S_m\}$;
        2. For each $S_i$, a match $M_i$ & n mismatches $N_{i,j}$, $1 \le i \le m$, $1 \le j \le n$;
        3. F features.

Output: Optimized weight vector $w = <w_1, w_2, \ldots, w_F>$

for each feature $f_i$
    weight of feature i, $w_i \leftarrow 1/F$    // initialize $w_i$'s uniformly s.t. $\Sigma w_i = 1$

repeat
    select feature $f$ randomly
    Find all cross-over points for S, by solving linear equations (1)   // i.e. by sliding $w_f$ from 0 to 1
    Find the "optimum" weight of $f$, $w_f$*  // The weight that maximizes the difference between +ve & -ve
                                               cross-overs
    $w_f \leftarrow w_f$*
    for all features i, $i \neq f$,    $w_i \leftarrow w_i + (w_f - w_f^*)w_i/\sum w_k$, $k \neq f$   // Other weights are proportionally adjusted
                                                          (according to weight) s.t. all weights add up to 1
    Find Ranking Consistency of w, RC(w) using (2)
until  number of iterations exceeds a threshold and RC(w) does not improve

return w
```

Figure 4. The SLIDER algorithm.

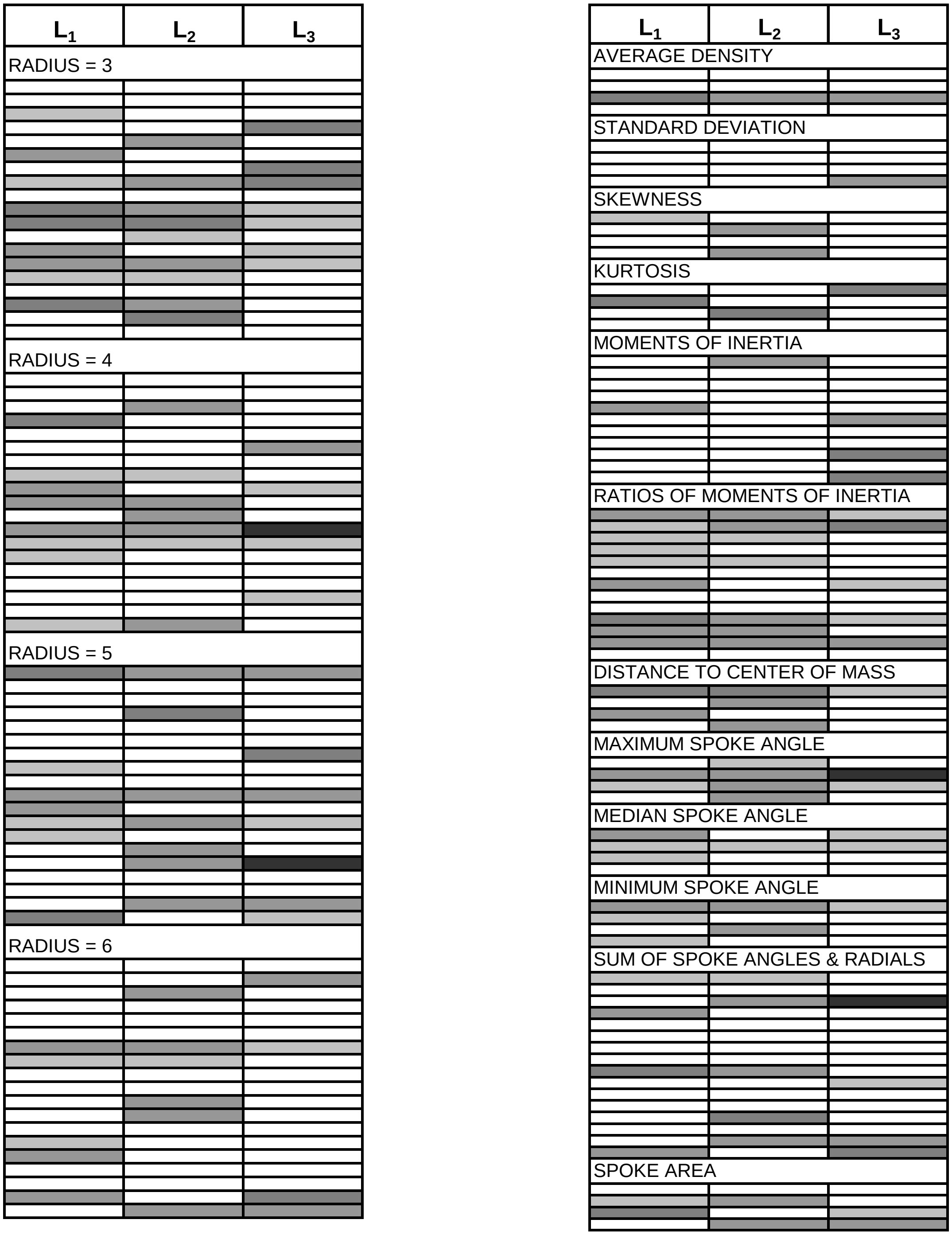
(based on density correlation); then we use the weighted feature-based distance metrics to rank all the ~50,000 regions according to similarity, and find out if the feature-based metric manages to "catch" a good match in the top k

Figures 5 and 6 compare the weights returned by SLIDER for the three Minkowsky metrics. For all three, between 25 and 30 features (out of 76) are selected i.e. those features with weights non-negligibly greater than 0. Moreover, there is strong tendency to choose the same features, and even weigh them similarly. There are 28 features for which all three metrics have yielded zero weight. It should be noted that when different features are chosen, they often are very similar, in two ways: (1) they are closely related e.g. standard deviation, skewness and kurtosis; (2) it may be the same feature (like mean density) but at different radii.

Figure 5 tries to capture this concordance in returned weights, by first sorting the features based on radius and then grouping them on identity (such that related features are as contiguous as possible). Figure 6 groups features the other way round i.e. for each feature, the four radii at which they are calculated are shown, sorted in ascending order. The weights have been linearly graded on a 5-level scale, where the darker the shade, the higher the weight.

We make the following remarks on the feature weights computed by SLIDER:

- The consistency in features selected (and weighted) across the three metrics shows that the algorithm converges. But the risk of local minima still exists; this is partially addressed by the randomized choice of feature in each iteration.
- Table 2 shows the sum of weights for each radius; we can again observe significant similarity of weights for the three metrics. Furthermore, we can note that the total weights for radius 3 Å is the maximum, and total weights for radius 6 Å is the minimum; these observations are intuitive – the 3D spherical patterns are expected to cover amino acids of various shapes and sizes, which justifies the choice of feature values at different radii. At 3 Å radius, we expect that the pattern to be significantly characterized, although inadequately since some large amino acids may not be totally encapsulated in the sphere. But at 6 Å, we face the problem of having noise due to density of neighboring residues; this trend is captured by our weight optimization algorithm. Furthermore, many features seem to be particularly relevant at 5 Å, including average density, ratios of moments of inertia and sum of spoke angles.



Figures 5 and 6. The relative weights of 76 features returned by SLIDER for L_1 (Manhattan), L_2 (Euclidean), and L_3 are shown. In Figure 5 (left), the features are first sorted on radius (in Å), and then on identity, such that related weights are contiguous. In Figure 6 (right), the features are sorted on identity and then on radius (in ascending order, from top to bottom). Darker the shade, higher is the weight. The white cells represent features with zero weight.

Table 2. Sum of all 19 feature weights, independently for 4 different radii.

Radius in Å (1Å = 10^{-10}m)	Sum of weights for three Minkowsky metrics		
	Manhattan	Euclidean	L_3
3	.36	.35	.36
4	.22	.20	.16
5	.28	.27	.34
6	.14	.18	.14

- The individual moments of inertia seem irrelevant; but their ratios provide more information related to the shape of the density pattern (e.g. spherical, ellipsoidal, etc.). This exemplifies the feature interaction problem [19], where several features may not appear relevant on an individual basis, but when looked at in combination, they contribute significantly to the description of the pattern.

The strong similarity of weights across the three metric is largely expected. Some weights are relevant, irrespective of the underlying metric. Nonetheless, there are differences, and interestingly, these differences do capture the sensitivity of "optimum" weights to the metric being used. Figures 7-9 show the percentage of times the three Minkowsky metrics manage to catch at least one good match within the top k, for various values of k. We can observe that both feature selection (binary weights) and feature weighting (continuous weights) improve over uniform weights (all 76 features equally weighted). Furthermore, continuous weights improve on binary weights for all three metrics. The differences among the three weighting schemes are more marked for Manhattan, a little less so for Euclidean, and even lesser for L_3. In [15], we observed that, regardless of the weights, Manhattan distance is a better metric than Euclidean, which is better than L_3. This seems to suggest that the better the metric, the more sensitive it is to the weights. An important point to note is that, given a distance metric, using continuous weights does not improve pattern matching (as compared to binary weights) if the weights used are those optimized for another metric e.g. the weights determined by SLIDER for Euclidean will not make continuous weights outperform binary weights for Manhattan.

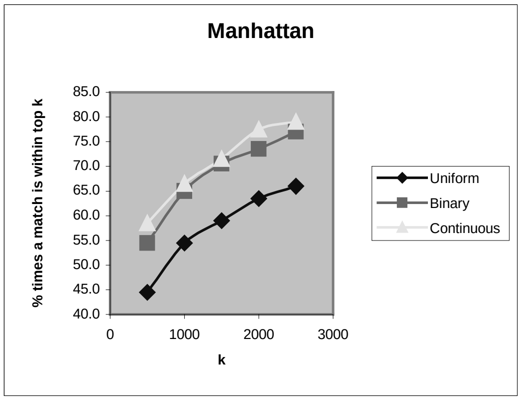


Figure 7. The % of test cases where a match is "caught" in top k for Manhattan.

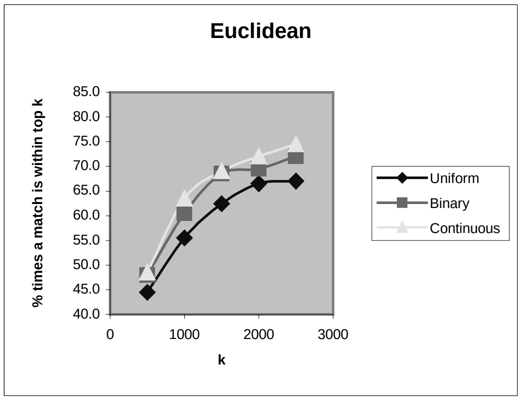


Figure 8. The % of test cases where a match is "caught" in top k for Euclidean.

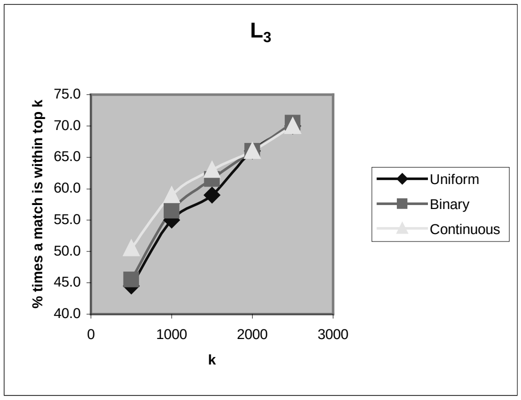


Figure 9. The % of test cases where a match is "caught" for L_3: the improvements with weighting is less for L_3.

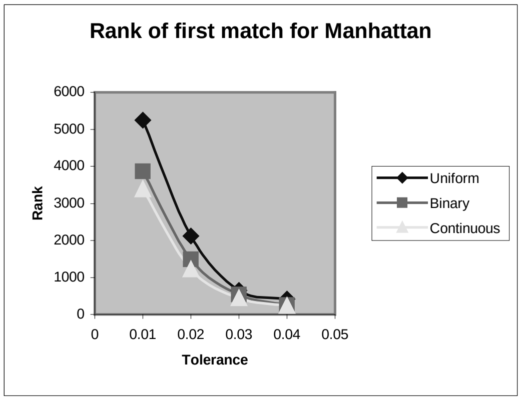


Figure 10. The rank of first "good match" for various levels of tolerance.

There are other possible ways of assessing the effectiveness of retrieval. For instance, Figure 10 shows how the Manhattan metrics rank a true "match" – a case is a match to a query if their density correlation is within a *tolerance* of the correlation between the query and its absolute best match. Figure 10 shows the rank for 4 levels of tolerance between .01 and .04, where density correlation lies between 0 and 1, the latter corresponding to a perfect match. Similar results are obtained for Euclidean and L_3 (not shown).

7. Discussion

The SLIDER system has been successfully applied to determine the weights of features for the complex problem of recognizing patterns of electron density for finding the structure of proteins. But the techniques employed are general and potentially useful in other domains, especially those with high-dimensional, noisy data. The salient aspects of our approach are:

- SLIDER is a filter method which avoids searching a large space of possible weight vectors. Instead, the evaluation is performed at weight values which matter i.e. where there is a marked change in accuracy of matching. Furthermore, locating these weight values can be done efficiently, since it involves solving linear equations applicable to many metrics (like the Euclidean distance). The benefits of restricting the number of weights searched and used for nearest neighbor classification are emphasized in the DIET system [24]; the latter also argue that there are probably no benefits in using weights beyond two possible values (0 and 1) - but the SILDER algorithm does manage in computing finer weight values that improve matching and case retrieval.
- SLIDER was used to optimize weights for three Minkowsky distance metrics, and proved to be successful in improving pattern matching and retrieval, in the context of a case-based reasoning and nearest-neighbor strategy to efficiently retrieve matches. The weights as determined by SLIDER were largely similar for the various metrics; nonetheless, the slight differences were significant in capturing the sensitivity of relevance to the distance metric being used. We argue that the relevance of features in describing a pattern is not absolute; it depends on how the features are used to determine similarity, especially since similarity itself is often a fuzzy concept, with imprecise ways of determining it.

As future work, we are currently looking at the following issues, where there is considerable scope for improvement and investigation:

- SLIDER is currently limited to distance metrics for which cross-overs weights can be calculated by solving simple linear equations. This may not be possible for other metrics, like those based on probabilistic and statistical methods [4, 26, 15]. We are currently investigating methods where cross-over points for such metrics can be efficiently determined (by binary search over the space of weights, for instance).
- One aspect which probably necessitates closer scrutiny is the definition of matches and mismatches to assess if the updated weights improve accuracy. We use a simple strategy where two patterns are said to match/mismatch if their density correlation in above/below a threshold. We observed that final set weights returned by SLIDER is sensitive to this threshold. What would be an appropriate threshold, and how can it be determined? Or is there a better way of assessing similarity in this context? Should we use "perfect" matches/mismatches in our training set, or do we need to allow for near-matches/mismatches as well, which will enable us capture the nuances in the information that is required to confidently say how different two instances are?
- More generally, we are also working on other strategies to weight features, including analyzing the sensitivity of feature relevance to the context [10,18] and methods based on Single Value Decomposition (SVD) and Principal Component Analysis (PCA).

8. References

[1] Adams, P.D., Gopal, K., Grosse-Kunstleve, R.W., Hung, L.W., Ioerger, T.R., McCoy, A.J., Moriarty, N.W., Pai, R.K., Read, R.J., Romo, T.D., Sacchettini, J.C., Sauter, N.K., Storoni, L.C., and Terwilliger, T.C. 2004. "Recent developments in the PHENIX software for automated crystallographic structure determination", *Journal of Synchrotron Rad. 11*: 53-55.

[2] Aha, D.W. 1990. "A Study of Instance-Based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, and Psychological Observations", Ph.D. diss., University of California, Irvine.

[3] Aha, D.W. 1998. "Feature Weighting for Lazy Learning Algorithms", Liu H., and Motoda, H. eds. *Feature Extraction, Construction and Selection: A Data Mining Perspective.* Boston, MA: Kluwer.

[4] Aksoy, S. and Haralick, R.M. 2001. "Probabilistic vs. Geometric Similarity Measures for Image Retrieval", *Proceedings of Computer Vision and Pattern Recognition (CPRV)*, 112-128. IEEE Computer Society Press.

[5] Berman H.M., Westbrook J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. 1992. "The Protein Data Bank", *Nucleic Acids Research* 28: 235-242.

[6] Blum, A.L. and Langley, P. 1997. "Selection of relevant features and examples in machine learning", *Artificial Intelligence* 97:245-271.

[7] Brünger, A.T. 1992. "XPLOR manual, version 3.1", Yale University, New Haven, CT.

[8] Burley, S.K., Almo, S.C., Bonanno, J.B., Capel, M., Chance, M.R., Gaasterland, T., Lin, D., Sali, A., Studier, W., and Swaminathian, S. 1999. "Structural genomics: Beyond the Human Genome Project", *Nature Genetics* 232:151-157.

[9] Diamond, R. 1971. "A real-space refinement procedure for proteins", *Acta Cryst*. A27:436-452.

[10] Domingos, P. 1997. "Context-Sensitive Feature Selection for Lazy Learners", *Artificial Intelligence Review* 11:227-253.

[11] Duda, R.O., Hart, P.E., and Stork, D.G. 2001. *Pattern Classification*. New York, NY: John Wiley and Sons Inc.

[12] Fix, E. and Hodges, J. 1951. "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties", Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.

[13] Gopal, K., Pai, R., Ioerger, T.R., Romo, T.D., and Sacchettini, J.C. 2003. "TEXTAL™: Artificial intelligence techniques for uutomated protein structure determination", *Proceedings of the Fifteenth Conference on Innovative Applications of Artificial Intelligence Conference*, 93-100. Menlo Park, CA: AAAI Press.

[14] Gopal, K., Romo, T.D, Sacchettini, J.C, and Ioerger, T.R. 2004. "Efficient retrieval of electron density patterns for modeling proteins by X-ray crystallography", *submitted*.

[15] Gopal, K., Romo, T.D, Sacchettini, J.C, and Ioerger, T.R. 2004. "Evaluation of geometric & probabilistic measures of similarity to retrieve electron density patterns for protein structure determination", to appear in *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, NV.

[16] Hobohm, U., Scharf, M., Schneider, R., and Sander, C. 1992. "Selection of a represetative set of structures from the Brookhaven Protein Data Bank", *Protein Science* 1: 409-417.

[17] Holton, T.R., Christopher, J.A., Ioerger, T.R., and Sacchettini, J.C. 2000. "Determining protein structure from electron density maps using pattern matching", *Acta Cryst*. D46:722-734.

[18] Howe, N. and Cardie, C. 1997. "Examining locally varying weights for Nearest Neighbor algorithms", *Lecture Notes in Artificial Intelligence* 455-466. Springer.

[19] Ioerger, T.R. 1999. "Detecting feature interactions from accuracies of random feature subsets", *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 49-54. Menlo Park, CA: AAAI Press.

[20] Ioerger, T.R. and Sacchettini, J.C. 2002. "Automatic modeling of protein backbones in electron-density maps via prediction of C-alpha coordinates", *Acta Cryst.* D5:2043-2054.

[21] Ioerger, T.R and Sacchettini, J.C. 2003. "The TEXTAL system: Artificial Intelligence Techniques for Automated Protein Model-Building", Sweet, R.M. and Carter, C.W., eds., *Methods in Enzymology* 374: 244-270.

[22] John, G., Kohavi, R. and Pfleger, K. 1994. "Irrelevant features and the subset selection problem", *Proceedings of the Eleventh International Conference on Machine Learning*, 121-129. San Mateo, CA: Morgan Kaufmann.

[23] Kira, K. and Rendell, L.A. 1992. “A practical approach to feature selection”, *Proceedings of the Ninth Conference on Machine Learning*, 249-256. San Mateo, CA: Morgan Kaufmann.

[24] Kohavi, R., Langley, P., and Yun, Y. 1997. “The utility of feature weighting in nearest-neighbor algorithms”, *Proceedings of the European Conference on Machine Learning*. Prague: Springer-Verlag.

[25] Kolodner, J. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

[26] Kontkanen, P., Myllymaki, P., Silander, T., and Tirri H. 1997. “A Bayesian Approach for Retrieving Relevant Cases”, P. Smith, ed. *Artificial Intelligence Applications* (Proceedings of the EXPERSYS-97 Conference), Sunderland, UK, 67-72, IITT International.

[27] Leake, D.B. ed. 1996. *Case-Based Reasoning – Experiences, Lessons and Future Directions*. Cambridge, MA: MIT Press.

[28] Liu, H. and Motoda, H. eds. 1998. *Feature Extraction, Construction, and Selection: A Data Mining Perspective*. Boston, MA: Kluwer.

[29] Mitchell, T. 1997. *Machine Learning*. Boston, MA: McGraw-Hill.

[30] Mowbray, S.L., Helgstrand, C., Sigrell, J.A., Cameron, A.D., and Jones, T.A. 1999. “Errors and reproducibility in electron-density map interpretation”, *Acta Cryst*. D55: 1309-1319.

[31] Orengo, C.A., Pearl, F.M., Bray, J.E., Todd, A.E., Matin, A.C., Lo Conte, L., and Thornton, J.M. 1999. “The CATH Database provides insights into protein structure/function relationships”, *Nucleic Acids Res.* 27: 275-279.

[32] Richardson, J.S. and Richardson, D.C. 1985. “Interpretation of electron density maps”, *Methods in Enzymology* 115:189-206.

[33] Russel, S. and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.

[34] Smith, T.F. and Waterman, M.S. 1981. “Identification of common molecular subsequences”, *J. Mol. Biol*. 147:195-197.

[35] Smyth, B. and Cunningham, P. 1996. “The Utility Problem Analyzed: A Case-Based Reasoning Perspective”, *Third European Workshop*, EWCBR-96, Lausanne, Switzerland. Advances in Case-Based Reasoning, *Lecture Notes in Computer Science*, 392-399. Heildelberg: Springer.