# Efficient Retrieval of Electron Density Patterns for Modeling Proteins by X-ray 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

*Inefficient case retrieval is a major problem in many case-based reasoning systems, especially when case matching is expensive and the case-base is large. In this paper, we present a two-phase approach where an inexpensive feature-based method is used to find a set of potential matches and a more expensive and accurate case matching method is used to make the final selection. This approach has been successfully employed in TEXTAL™, a system that retrieves previously solved 3D patterns of electron density from a database to determine the structure of proteins. Electron density patterns are characterized by numeric features and an appropriate distance measure is used to efficiently filter good matches through an exhaustive search of the database. These matches are then examined using a computationally expensive density correlation procedure based on finding an optimal superposition between 3D patterns. We provide an empirical and theoretical analysis of some of the keys issues related to this method. In particular, we define a model for estimating how approximate various feature-based similarity measures are (relative to an objective matching metric), and determine its relation to the number of cases that should be filtered from a given database to make the approach effective.*

## 1. Introduction

*Case-based reasoning* [14, 16] is a form of *instance-based learning* [1], which is model free or non-parametric [5] since prediction is done directly from the data without producing any explicit model of the problem domain. One of the essential steps in case-based reasoning algorithms is case retrieval, where most similar cases are recalled from a case-base by performing case matching. There are compelling reasons for having large case-bases: extensive problem coverage and better quality of solutions. But a large case-base generally induces a degradation in system efficiency, especially if the case matching is expensive [17].

The approach we use is based on employing a computationally fast, feature-based similarity measure (which we can afford to run over the whole case-base) to filter out a set of potential matches, given a query case. This similarity metric is expected to approximate a correct, objective, and usually expensive matching method; the latter can then make the final selection. This two-phase method for case retrieval has been previously proposed, in different flavors and application domains. For example, in [7] MAC/FAC (for "many are called but few are chosen") is proposed as a general strategy for efficient, similarity-based retrieval. In [3], similar stratified or hierarchical case-based reasoning methods are suggested in the planning domain. Other notable related work include [4, 20].

In our approach, a feature-based method is used to filter a reasonably small number of cases, say *k*, using the nearest neighbor rule [6]. Similarity between cases can be determined using a suitable, efficient feature-based distance metric e.g. geometric measures (like Hamming, Manhattan, Euclidean, etc.), or probabilistic measures [2, 15].

There are several questions related to this approach that have received considerable attention [18]: What should be the size and composition of the case-base? At what point does the case-base become "saturated," beyond which there is no gain in performance? How many cases should we filter? How good should features be in capturing relevant information about

cases? Should the features used to measure distance between cases be weighted equally? If not, how are the weights chosen? What is the most appropriate similarity metric?

Some of these issues have been addressed in the context of TEXTAL™, a case-base reasoning system that automatically determines protein structures using X-ray crystallography methods [13, 11, 9, 10]. In this paper, we empirically and theoretically discuss how the proposed approach helps in the efficient and effective retrieval of density patterns in TEXTAL™. In particular, we examine how the choice of *k* influences the effectiveness of retrieval for various similarity measures, given a database. We also provide a method for determining a suitable *k*, based on a loss function that quantifies the extent to which the inexpensive feature-based similarity measure approximates a correct, objective metric. More specifically we try to represent how well the approximate measure ranks patterns in the case-base according to similarity to query cases. This method bears some resemblance with PAC learning [19]; here we try to find out how many cases we should look at to obtain probably approximately correct matches.

The rest of the paper is organized as follows: the X-ray protein crystallography domain and TEXTAL™ are described in the next section. We then provide a general theoretical framework for the proposed case retrieval strategy. Next, we empirically analyze the effectiveness of the proposed approach in retrieving patterns of electron density in TEXTAL™, using various measures of similarity. The results are discussed, and compared to theoretically expected ones. We conclude by a general discussion, and describe current work to extend and complement the proposed approach.

## 2. Protein crystallography & TEXTAL™

Determining the 3D structure of a protein is a significant and challenging endeavor – it enables understanding of how the protein functions e.g. how protein enzymes work, which atoms are essential for catalysis, why one protein binds to a specific DNA sequence and not another. Furthermore, drug design can be based on the structure of proteins – for instance, if the active site of an enzyme is known, molecules can be designed to inhibit the enzyme. X-ray crystallography is the most commonly used method to determine the structure of proteins. One of the main steps in X-ray crystallography is to interpret an *electron density map*, which is obtained by the Fourier transformation of patterns that result from the diffraction of X-rays by the protein crystal. An electron density map shows how electrons are distributed over the macromolecule (Figure 1). Solving the structure essentially means fitting various known molecular structures (or amino acids) into the density (there are 20 types of amino acids; proteins are essentially unique linear sequences of typically 100-1000 amino acids, which have several degrees of freedom and thus can take various angular conformations). The way in which the protein "folds" will largely determine its properties and functions. Fitting amino acids into the density is done by crystallographers, with the help of molecular visualization programs, drawing from experience on how to visualize 3D density patterns and other knowledge of the domain. The process is usually tedious and time-consuming, especially if the electron density data is noisy. TEXTAL™ automates this process of structure determination by first finding the positions of central carbon atoms in amino acids called Cα's. This is done by a component of TEXTAL™ called CAPRA, or C-Alpha Pattern Recognition Algorithm [12]. TEXTAL™ then breaks down the electron density map into small spherical regions (with 5Å radius, where 1Å = $10^{-10}$m) around the Cα's determined by CAPRA, and for each region, searches a database for similar patterns of structures that are already solved (i.e. atoms and their coordinates are known). The fragments of solved structures are assembled together, followed by stereo-chemical refinements and alignment with the known sequence of amino acids to produce a final model. The TEXTAL™ system is much larger in scope; for more details, refer to [13, 12, 11, 8] and http://textal.tamu.edu:12321.

In this paper, we focus on one central problem in TEXTAL™: how to efficiently retrieve matching patterns of electron density from the case-base. One alternative is to use a metric called *density correlation*, which involves optimal 3D rotation and superposition between the two regions [10]. Since the number of possible rotations that need to be considered is very large, this method is expensive. The problem becomes practically intractable if we run this expensive metric on each of the ~50,000 regions of the database that we use. In fact, TEXTAL™ may take more than a day to solve a medium-sized protein structure. To speed up the process, we use an inexpensive, approximate metric to filter a relatively small number of cases (say $k$ = 500) on which we can afford to run the density correlation measure. This filtering enables reducing the computation time to a few hours. The inexpensive similarity methods that we use are based on finding differences between vectors of numeric features. These features are expected to characterize the relevant aspects of the spherical regions of density – example

features include statistics of local density distribution, moments of inertia, distance to center of mass, etc. In TEXTAL™, we use 76 features; [10] provides more details about how features were defined and weighted.

It should be noted that there are pragmatic benefits in trying to keep *k* as low as possible – the quicker structures can be solved, the more flexibility it gives the crystallographer to try "what if" situations, especially in an interactive setting.

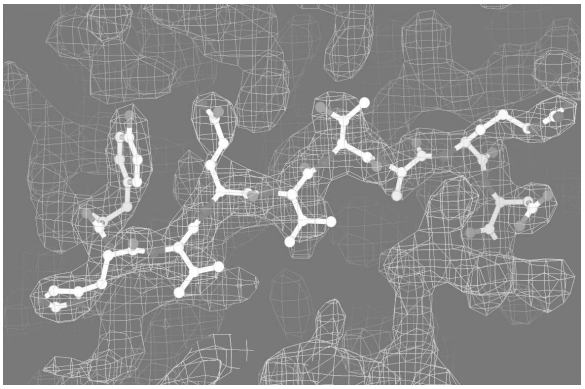

**Figure 1. Example of electron density around a fragment of a protein from Yeast called 1HQZ. The fragment consists of 9 amino acids. This stereo view has been made with Spock, a molecular graphics program written by Dr. J. Christopher (http://quorum.tamu.edu/spock).**

## 3. Framework for efficient case retrieval

The filtering approach employed by TEXTAL™ is general and potentially useful for many case-based reasoning applications, especially those characterized by large case-bases, expensive matching methods, and noisy data. The general strategy can be stated as follows: given a query case q, N stored cases, our goal of finding the best match can be met if we could use an objective matching metric (called *obj*) to rank all N cases according to similarity with q. Since this might be too expensive, we use an approximate similarity measure (called *sim*) to select *k* cases, and use *obj* for the final ranking of the *k* cases.

In most applications, we may not be adamant about retrieving the very best batch, but content with something close enough. This notion of a "reasonably good" match is formalized by specifying a tolerance $\delta$, based on which any one of the $\lambda$ top matches of q is deemed good enough if:

$$[obj(q, m_1) - obj(q,m_\lambda)]/ obj(q, m_1) < \delta \qquad (1)$$

where $m_i$ is the $i^{th}$ best match according to the objective metric *obj* (we assume that *obj* and *sim* are positive and increase with similarity).

Our aim can now be stated as follows: given a good enough match m for a query case q, we have to ensure that *sim* "catches" m within the *k* (say 500) filtered cases with probability $\varphi$ (say 0.95) or higher (here we make the pragmatic choice of having a single value of *k*, independent of q) i.e.

$$P(rank(q,m,sim) < k \mid rank(q,m,obj) < \lambda) > \varphi \qquad (2)$$

where rank(*query*, *case*, *metric*) is the rank of *case* according to similarity with *query* using similarity measure *metric* (rank decreases with similarity i.e. the best match has rank = 1).

We now use the following loss function *Q* as an estimate of the error related to the approximate measure *sim* in the ranking of the good enough match m of q.

$$Q(q,m,sim) = \frac{1}{(1 + e^{[k - rank(q,m,sim)]/\tau})} \qquad (3)$$

This function is chosen because it attributes loss close to zero if *rank* < *k*, and loss close to 1 if *rank* > *k*, with $\tau > 0$ (Figure 2). In the limit (as $\tau$ approaches zero), it becomes the Heaviside step function (Figure 2), which maximally penalizes any ranking (of a match) above *k*, and exonerates any ranking below *k*. But to realistically capture the loss (on the average) in ranking effectiveness of a similarity measure, the loss function is smoothened out, with a more gradual increase or decrease in loss as *rank* departs from *k*.

We can use this loss function to represent the probability of getting a match m of an arbitrary q within the top *k* as follows:

$$P(rank(q,m,sim) < k \mid rank(q,m,obj) < \lambda)$$

$$= 1 - Q(q,m,sim) \qquad (4)$$

Since rank(q,m,sim) for a match m (i.e. rank(q,m,obj) < $\lambda$) is not known (unless we compute *obj* exhaustively), we substitute it by r*, the average rank(q,m,sim) for a large test set of diverse q's and their matches m's. It should be noted that the computation of r* is expensive since it involves running the objective matching function on all members of the case-base, for each test case.

Substituting rank(q,m,sim) by r*, we derive the following inequality from equations 2, 3 and 4:

$$1 - \frac{1}{(1 + e^{[k - r^*]/\tau})} > \varphi$$

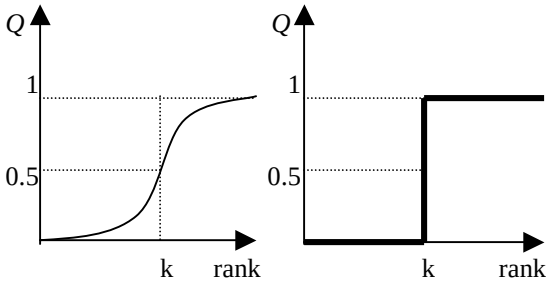


**Figure 2. Loss function *Q* to assess a similarity measure. If the latter ranks a match above *k*, the loss is high (i.e. the similarity measure is not very good), whereas a ranking below *k* implies lower loss. In the limit (as $\tau$ approaches zero), we obtain the Heaviside step function, as shown on the right.**

The value of $\tau$ reflects how lenient we are in assessing the ability of the approximate method to appropriately rank a match (i.e. below *k*). The limiting cases are: (1) $\tau = 0$, which attributes maximum (minimum) loss to any ranking immediately above (below) *k* (2) when $\tau$ approaches infinity which gives constant loss $Q = 0.5$; this corresponds to the fact that all ranks are equally acceptable, or all cases in the database are equally good matches, which occur when $\lambda = N$ i.e. any value of $k \leq N$ will catch the "match". Solving for *k*, we get:

$$k > r^* + \tau \ln(\varphi/(1-\varphi)) \quad (5)$$

## 4. Results

We now empirically evaluate the effectiveness of the filtering approach in retrieving electron density patterns in TEXTAL™, using different approximate similarity measures. We also empirically determine r*, the average rank of matches (against all other cases in the database, based on the objective metric) for a set of query instances. r* is computed for various levels of tolerance, and we use (5) to estimate the appropriate minimum value of *k* for each similarity measure for various values of $\delta$ and $\varphi$, and analyze consistency with empirical results.

As discussed earlier, the objective similarity measure *obj* used is *density correlation* between spherical electron density patterns. We experiment with three approximate similarity measures, given here in decreasing order of accuracy, relative to the objective metric:

(1) a probabilistic metric, where, given a query instance vector q, we compute the similarity likelihood ratio $r(d_i)$ for each case $c_i$ in the database, where $d_i = c_i - q$. The higher the ratio, the more similar is the pattern $c_i$ to q. The similarity likelihood ratio $r(d_i)$ is given by:

$$r(d_i) = (d_i-\mu_D)^T \Sigma_D^{-1}(d_i-\mu_D) - (d_i-\mu_S)^T \Sigma_S^{-1}(d_i-\mu_S)$$

where S and D are classes of known pairs of similar and different regions respectively, with mean feature difference $\mu_S$ and $\mu_D$, and covariance matrices $\Sigma_S$ and $\Sigma_D$ respectively. For more details, refer to [2, 9].

(2) weighted Euclidean distance, $L_2$, given by:

$$L_2 = [\Sigma w_j(x_j - y_j)^2]^{1/2}$$

where x and y are feature vectors, and $w_j$ is a measure of the relevance of feature j. The weights $w_j$'s are determined by the SLIDER algorithm, described in [10].

(3) the cosine distance, where the distance between vectors x and y is given by $1 - x.y/|x||y|$.

A test set of 200 query regions was generated in a way to evenly cover all types of amino acids, and for each query case, a database of ~50,000 density pattern regions was exhaustively searched and ranked according to *obj*. The tolerable $\lambda$ matches were determined using (1) with four values of tolerance $\delta$. The mean $\lambda$ (over the 200 test cases) is plotted against $\delta$ in Figure 3.

The three similarity measures were then used to rank all cases in the database for each query case, and various statistics were computed. The value of *k* that would assure retrieval of a good match with probability $\varphi = 0.95$ and $\varphi = 0.80$ are computed using (5), by setting $\tau$ to 600. The results are given in Tables 1 and 2 respectively.

The predicted values of *k* can be observed to be reasonably consistent with what was empirically obtained for the probabilistic measure (Figure 4). We can also note that theoretically expected ranks are more conservative as compared to what are actually observed i.e. good matches are generally obtained with much lower *k* than theoretically predicted. More consistent results can probably be obtained by choosing a more appropriate loss function.

Figure 5 shows the best ranked good enough match for various measures and tolerances. Figures 6 and 7 show the probability of getting a match in the top *k* for different *k*'s and δ's respectively. The probability values shown are, in fact, empirically observed P(rank(q,m,sim) < k | rank(q,m,obj) < λ). Figure 8 shows the probability that a case retrieved in the top k is actually a good enough match i.e. P(rank(q,m,obj) < λ) | rank(q,m,sim) < k) for varying δ.

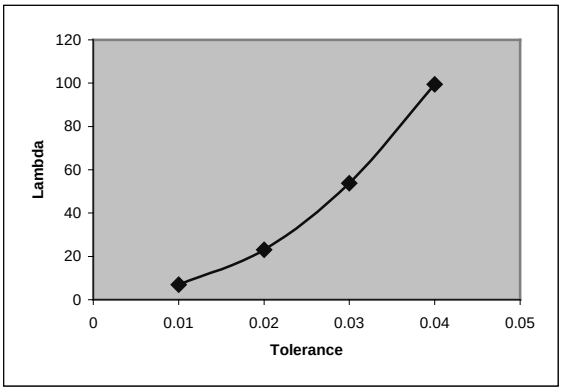


**Figure 3. λ is the number of true matches of a case, for a given tolerance δ. This graph shows how the average λ (over a test set) varies with δ. See equation 1.**

We make the following two main observations: (1) The different similarity measures are significantly effective in filtering out good cases. The probabilistic measure is particularly successful, and outperforms the commonly used Euclidean measure. In [9] we compare more distance measures, including Manhattan and other Minkowsky metrics, and like [1,7,6], we argue that probabilistic and statistical distance measures tend to outperform geometric measures. Geometric measures are too parametric and constrained, whereas the probabilistic measure defined earlier captures more information about pattern variations through the values of mean, variance and density estimates derived from objectively defined similar and different patterns. (2) There is reasonable consistency between the empirically determined values of *k* and theoretically expected ones (based on the loss function). That is, the theoretical model provides an informed way of setting *k*, if we wish to retrieve, on the average, approximate matches (as defined by δ and λ) with probability φ, or higher.

We now further verify the consistency of empirical results with what are expected, based on rules of probability. We can see that the prior probabilities are:

$$P(\text{rank}(q,m,sim) < k) = k/N, \text{ and}$$
$$P(\text{rank}(q,m,obj) < \lambda) = \lambda/N$$

Using Bayes' rule, we obtain the following:

$$\lambda[P(\text{rank}(q,m,sim) < k \mid \text{rank}(q,m,obj) < \lambda)] = k[P(\text{rank}(q,m,obj) < \lambda \mid \text{rank}(q,m,sim) < k)]$$

This can be re-written as:

$$\lambda[P(\text{rank}(q,m,sim) < k \mid \text{rank}(q,m,obj) < \lambda)]/ k[P(\text{rank}(q,m,obj) < \lambda \mid \text{rank}(q,m,sim) < k)] = 1 \quad (6)$$

The LHS of (6) was computed for all combinations of δ and *k* shown in Figures 5-8, using mean of the λ's. The values ranged from roughly 1 to 3 (Figure 9). They are all fairly close to 1, as should ideally be the case. Nonetheless, the departure from 1 can be attributed to the high variance in λ.

**Table 1. Theoretically predicted values of k for catching a match with probability > .95**

| **Similarity measure** | **δ=.01, λ=6.9** | **δ=.02, λ=23.1** | **δ=.03, λ=53.8** | **δ=.04, λ=99.5** |
|---|---|---|---|---|
| **Probabilistic** | 2865 | 2140 | 1941 | 1860 |
| **Euclidean** | 5797 | 3175 | 2258 | 2048 |
| **Cosine** | 6883 | 3822 | 2424 | 2135 |

**Table 2. Theoretically predicted values of k for catching a match with probability > .80**

| **Similarity measure** | **δ=.01, λ=6.9** | **δ=.02, λ=23.1** | **δ=.03, λ=53.8** | **δ=.04, λ=99.5** |
|---|---|---|---|---|
| **Probabilistic** | 1930 | 1205 | 1006 | 925 |
| **Euclidean** | 4862 | 2240 | 1323 | 1113 |
| **Cosine** | 5948 | 2887 | 1489 | 1200 |

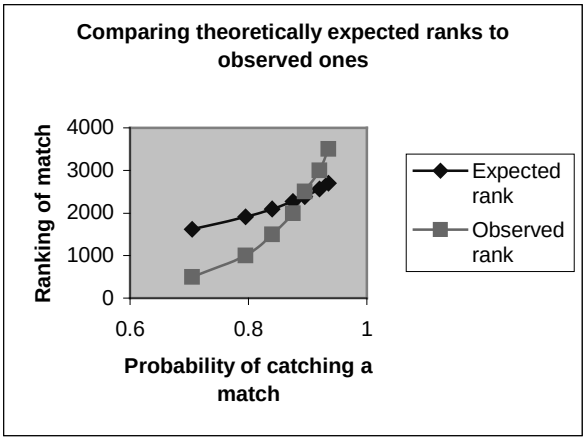


**Figure 4. Theoretically expected vs. observed ranks for the probabilistic distance measure ($\delta = 0.01$, $\tau = 600$).**

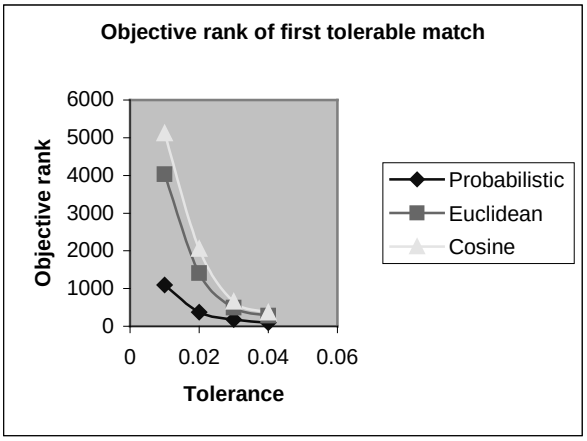


**Figure 5. The rank (according to the objective metric) of the first match retrieved (for k = 500). Rank decreases with similarity.**

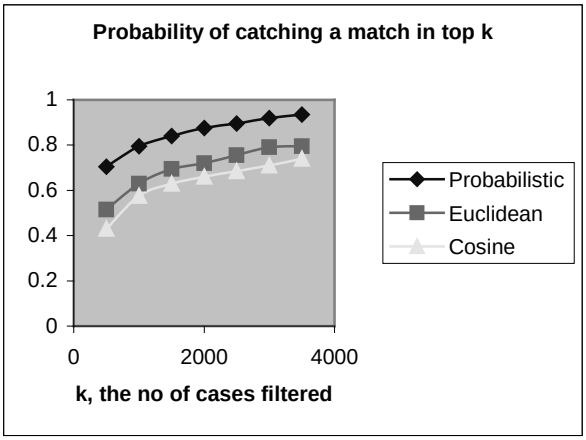


**Figure 6. P(rank(q,m,sim) < k | rank(q,m,obj) < λ) vs. k (δ = 0.01).**

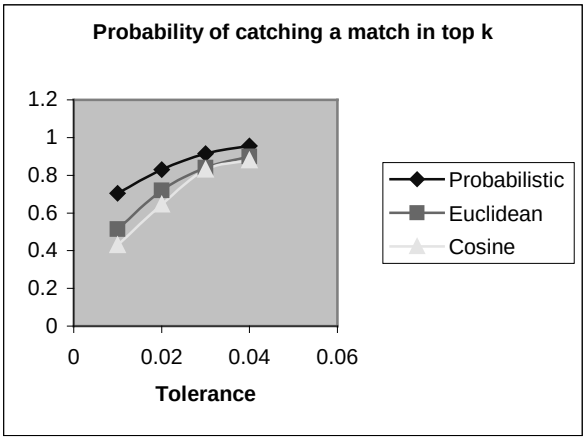


**Figure 7. P(rank(q,m,sim) < k | rank(q,m,obj) < λ) vs. δ (k = 500).**

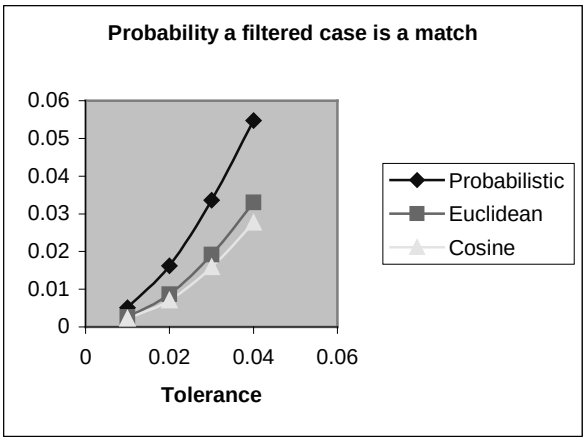


**Figure 8. P(rank(q,m,obj) < λ | rank(q,m,sim) < k ) vs. δ (k = 500).**

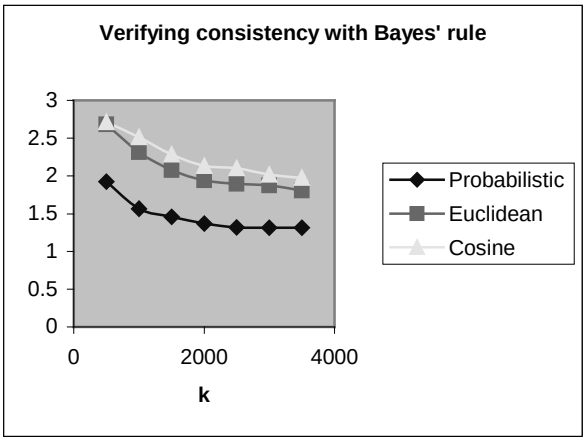


**Figure 9. Plot of λ[P(rank(q,m,sim) < k | rank(q,m,obj)]/ k[P(rank(q,m,obj) < λ | rank(q,m,sim) < k)]; should ideally be 1 for all k, as per equation 6.**

## 5. Conclusion

We proposed a general strategy for efficient case retrieval by approximating an objective, expensive similarity metric with a fast, feature-based similarity measure, and using the latter as a filter of probably good matches, based on k-nearest neighbor learning. With this approach, case-based reasoning systems can afford large case-bases as well as improve on time performance. We empirically and theoretically analyzed the issue of the number of cases that need to be filtered. We proposed a test procedure and a theoretical model based on a loss function to represent how approximate different measures of similarity are, and to predict the choice of *k*, based on stringency of expected results and an estimation of the degree of inaccuracy of ranking by the approximate measures. One of the limitations of the proposed test procedure is the computational cost related to the exhaustive search of the database for determining truly best matches of test cases. We are currently investigating various approaches to statistically model the relationship between the approximate and objective measures of similarity, and derive the expected objective ranks of matches, in lieu of explicitly computing these objective ranks through exhaustive search. An important closely related issue not discussed in this paper is the choice of the size and composition of the database. We are currently developing methods that would eliminate redundancy and ensure more consistency between the objective and approximate similarity measures. We are considering two basic approaches: (1) *a priori* pre-processing of the entire case-base to eliminate noise and redundancies, and (2) dynamic screening of the case-base where we determine (at run-time) inconsistencies and dubious matches (because of noise in the form of incorrect data or irrelevant features) in the local region of the feature space under consideration. Finally, we recognize that a global value of *k* has its limitations. For some cases that have a large number of good matches, a relatively low *k* should catch a match with high probability, whereas more difficult cases may require comparison with more potential matches for effective retrieval. Context-sensitive determination of *k* is yet another worthwhile future investigation.

## 6. References


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

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

[3] L.K. Branting and D.W. Aha, "Stratified Case-Based Reasoning: Reusing Hierarchical Problem Solving Episodes", *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995, pp. 384-390.

[4] E. Chavez, G. Navarro, R. Baeza-Yates and J. Marroquin, "Proximity Searching in Metric Spaces", *ACM Computing Surveys*, 2001, 33(3):273-321.

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

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

[7] K. Forbus, D. Gentner and K. Law, "MAC/FAC: A Model of Similarity-based Retrieval", *Cognitive Science*, 19(2), April-June 2001, pp. 141-205.

[8] K. Gopal, R. Pai, T.R. Ioerger, T.D. Romo and J.C. Sacchettini, "TEXTAL: Artificial Intelligence Techniques for Automated Protein Structure Determination", *Proceedings of the Fifteenth Conference on Innovative Applications of Artificial Intelligence*, AAAI Press, Menlo Park, CA, 2003, pp. 93-100.

[9] K. Gopal, T.D. Romo, J.C. Sacchettini and T.R. Ioerger, "Evaluation of Geometric & Probabilistic Distance Measures to Retrieve Electron Density Patterns for Protein Structure Determination", *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, NV, 2004, pp. 427-432.

[10] K. Gopal, T.D. Romo, J.C. Sacchettini and T.R. Ioerger, "Weighting Features to Recognize 3D Patterns of Electron Density in X-ray Protein Crystallography", *Proceedings of Computational Systems Bioinformatics*, Stanford, CA, 2004, pp. 255-265.

[11] T.R. Holton, J.A. Christopher, T.R. Ioerger and J.C. Sacchettini, "Determining Protein Structure from

Electron Density Map Using Pattern Recognition", *Acta. Cryst.* D46, 2000, pp. 722-734.

[12] T.R. Ioerger and J.C. Sacchettini, "Automatic Modeling of Protein Backbones in Electron Density Maps Via Prediction of C-alpha Coordinates", *Acta Cryst.* D5, 2002, pp. 2043-2054.

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

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

[15] P. Kontkanen, P. Myllymaki, T. Silander and H. Tirri, "A Bayesian Approach for Retrieving Relevant Cases", in P. Smith, ed., *Artificial Intelligence Applications*, Sunderland, UK, 1997, pp. 67-72.

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

[17] B. Smyth and P. Cunningham, "The Utility Problem Analyzed: A Case-Based Reasoning Perspective", *Advances in Case-Based Reasoning, Lecture Notes in Computer Science*, Springer, Heidelberg, 1996, pp. 392-399.

[18] G. Toussaint, "Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress", *Computing Science and Statistics*, 34, 2002.

[19] L. Valiant, "A Theory of the Learnable", *Communications of the ACM*, 27(11), 1984, pp. 1134-1142.

[20] M. Veloso, *Learning by Analogical Reasoning*, Springer Verlag, Berlin, 1994.