Re-designing Distance Functions and Distance-Based Applications for High Dimensional Data

Charu C. Aggarwal
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
charu@watson.ibm.com

Abstract

In recent years, the detrimental effects of the curse of high dimensionality have been studied in great detail on several problems such as clustering, nearest neighbor search, and indexing. In high dimensional space the data becomes sparse, and traditional indexing and algorithmic techniques fail from the performance perspective. Recent research results show that in high dimensional space, the concept of proximity may not even be qualitatively meaningful [6]. In this paper, we try to outline the effects of generalizing low dimensional techniques to high dimensional applications and the natural effects of sparsity on distance based applications. We outline the guidelines required in order to re-design either the distance functions or the distance-based applications in a meaningful way for high dimensional domains. We provide novel perspectives and insights on some new lines of work for broadening application definitions in order to effectively deal with the dimensionality curse.

1 Introduction

In recent years, high dimensional search, retrieval and clustering have become very well studied problems because of the increased importance of high dimensional data mining applications. For such applications, the curse of high dimensionality tends to be a major obstacle in the development of effective algorithmic techniques in several ways. For example, the performance of similarity indexing structures in high dimensions degrades rapidly, so that each query requires the access of almost all the data. Clustering algorithms often behave in an unstable way, and sometimes fail to be performance efficient for high dimensional problems. Similar issues are encountered by classification problems in high dimensionality; it becomes difficult to model the proximity in the feature space onto proximity for the class variable. In high dimensional space, the data sparsity makes the concept of proximity difficult to visualize in an intuitively acceptable way. Often the direct application of distance-based methods (which were originally designed with an intuitive assumption of low dimensionality) to high dimensional problems result in unexpected performance and qualitative costs.

In order to understand the effects of the dimensionality curse on these problems more effectively, it helps to investigate the nature of underlying applications which make these problems important: it turns out that the same definitions of distance functions and distance-based applications which are used for low dimensions are no longer so relevant with increasing dimensionality. We discuss recent research results which deal with such problems in two ways:

- Problems such as similarity search and clustering may be redefined for high dimensional data in order to make them more effective and meaningful. Examples of such techniques are *projected clustering* and *projected nearest neighbor search* [2, 3, 10]. These methods are generalized definitions of the clustering and similarity search problem in which the problems are re-defined by using projections of the data which are locality specific. For example, a cluster is defined as a set of points which are closely related in some low dimensional projection; a different cluster may use some other low dimensional projection. These techniques have the additional advantage of providing interesting information about the data in terms of locality specific projections.

- A second approach is to change the underlying distance function in order to more accurately reflect the particular characteristics of the data set. This approach provides less information about the characteristics of the specific data set and requires greater understanding of the data; on the other hand this can be powerful in providing the ability to use well known algorithms for distance-based applications by only modifying the underlying distance function.

1.1 Theoretical Background

We will first establish certain notations and definitions which are very helpful for the purpose of explaining some recent results on high dimensional proximity.
d : Dimensionality of the data space.
N : Number of data points.
X_d : Data point from d-dimensional data distribution.
$dist_d(X_d)$: Distance of X_d from the origin $(0, \ldots, 0)$.
$Dmin_d, Dmax_d$: Nearest/Farthest distance of N points to origin.
$E[X], var[X]$: Expected value/variance of X.
$Y_d \rightarrow_p c$: Y_d converges in probability to c as $d \Rightarrow \infty$.

Theorem 1.1 Beyer et al. [6] *If:* $lim_{d\rightarrow\infty}\ var\left(\frac{dist_d(X_d)}{E[dist_d(X_d)]}\right) = 0$ *, then:* $\frac{Dmax_d - Dmin_d}{Dmin_d} \rightarrow_p 0$.

Proof: See [6] for proof of a more general version of this result. ■

Thus, this result shows that under certain pre-conditions on the data distribution and distance function, the difference between the maximum and minimum distances to a given target is small compared to the absolute distance to the nearest point in high dimensional space. This makes a proximity query unstable because a small relative change in a query point in a direction away from the nearest neighbor could change it into the furthest neighbor. In such a situation, it becomes quite questionable whether a nearest neighbor indeed has sufficient qualitative significance.

These results are also valuable from the performance perspective of indexing. Most indexing methods work by using some kind of partitioning (hierarchical or flat) of the data set. This partitioning is then used in order to perform pruning of the data set. The idea is that if it is already known that some neighbor X is close enough to the target, then one can prune away an entire partition by showing that the optimistic distance bound of the target to that partition is no better than the distance to X. If almost all points are equidistant to the target, then this optimistic bound is usually not sharp enough for effective pruning. This means that in high dimensional space, any indexing structure will access all the data. From this observation, it is clear that we not only need to create distance functions which are meaningful, but we also need a way to make it friendly to the performance needs of the application (which is indexing in this case).

An interesting observation is that all the above dimensionality susceptible issues (both from the performance and meaningfulness perspective) can be traced back to the lack of contrast between the nearest and furthest neighbor. For example, consider a standard clustering algorithm in high dimensional data. If every pair of points is almost equi-distant, then what is meant by a cluster? How can we distinguish a cluster from the remaining data set? Would a randomized clustering algorithm provide stable results over multiple runs, and provide the same clusters for each run? It is indeed quite well known [3] that high dimensional clustering algorithms are unstable; different runs lead to significantly different clusters; furthermore none of these clusterings can be intuitively considered any better than the other because of the difficulty in defining proximity meaningfully. The criticality of the proximity problem tends to imply that by making a few changes to a small class of methods (such as the distance function), we may be able to create dimensionality resistant methods for a large number of problems.

1.2 Related Work on Distance Functions

The use of effective distance functions has been explored for data domains such as information retrieval and categorical data [7, 9, 14]. In both cases, the data shows certain typical domain-specific characterisitics. For example, the former domain is a zero-dominated (most attributes take on zero value) domain, whereas the latter is one in which there is no ordering of values for a given attribute. For such cases, specialized normalization techniques and statistical aggregate measures are used in order to identify the non-noisy aspects of the data and measure distances in a meaningful way.

We advocate the use of such statistical techniques even for arbitrary data sets in order to design effective high dimensional distance functions. The key is to be able to design meaningful distance functions which are also friendly to the performance needs of an application. For example, for an indexing application, we would like to be able to design a distance function which is *index-friendly* from a performance point of view; at the same time this performance gain should be obtained without sacrificing the quality of the distance function.

2 Redesigning Applications versus Redesigning Distance Functions

As we mentioned earlier, there are two ways of handling the meaninglessness issue of high dimensionality. One solution is to re-define problems such as clustering and similarity search in a more flexible way by examining them in the context of locality specific projections. Examples of such re-definitions are as follows:

- In projected clustering [2, 3], clusters are defined by partitioning the points such that each cluster exists within its own restricted set of dimensions. Even though the points are equi-distant from one another in full dimensionality, each cluster-specific

projection provides s subspace in which a particular set of points are close to one another.

- In projected nearest neighbor search [10], we search for interesting query-specific features which provide greatest discrimination in the neighborhood of the query point. These query specific features are used in order to define the most similar objects. This technique also provides interesting information about the dimensional selectivity in the neighborhood of a query-point.

In general, these techniques have the advantage of providing additional information about the data set in terms of locality-specific projections [2, 3, 10]. This information can be used for improving the performance of a host of methods which suffer from the dimensionality curse. One such example is an application of a generalized projected clustering technique [3] to Local Dimensionality Reduction (LDR) [13]. It has been shown in [13] that the LDR method provides a decomposition of the data which can be used to create an effective high dimensional index. The technical challenge in redefining applications is to do so in accordance with the needs of a given user; a task which requires considerable understanding of the application at hand.

A second approach is to re-design the distance function itself. Often the sparsity of high dimensional data has been understood in the context of particular distance norms such as the L_p-norm. An additional understanding of the nature of the L_p-norm may be found in [1]. For many high dimensional data mining methods, the choice of the distance function is not pre-defined, but is chosen heuristically. There is not much literature on how distance functions should be re-designed with increasing dimensionality for arbitrary applications. High dimensional index structures and algorithms use the euclidean distance metric as a natural extension of its use for spatial applications. The results in [6] show that such functions may be meaningless under many conditions for high dimensional applications.

On the other hand, for many other high dimensional domains of data such as Information Retrieval (IR), techniques have been proposed to measure similarity among objects based on the aggregate behavior of the data set. It is a difficult task to design such distance function for arbitrary applications, since designing distance function even for a specific domain of data such as IR has intrigued researchers over three decades [14]. Such a design in the IR community has been achieved by considerable testing and understanding of the particular characteristics of the data which are the most meaningful indicators of similarity. Unlike IR applications, we cannot use specific information about the "typical" nature of the data for arbitrary applications; therefore, for these cases, designing distance functions is an even more difficult task. The technical challenge of this general approach is to be able to design the distance function based on the *overall* behavior of the particular data set under consideration.

Thus, locality-specific projection methods [2, 3, 10, 13] *provide* insight about the behavior of a given data set; whereas designing effective distance functions for a given data set *requires* insight about its behavior. In the next section, we will provide some general insights into the design of meaningful high dimensional distance functions.

3 Desiderata for Dimensionality Resistant Distance Functions

One of the helpful observations from the previous section is that the reasons for the failure of high dimensional algorithms both from the performance and meaningfulness perspective are rooted in the same reason of poor discrimination between the furthest and nearest neighbor in high dimensional space. This also means that by designing dimensionality resistant distance functions we may be able to design methods which not only provide superior performance but are also able to provide results which are superior from a qualitative perspective. One of the fatal flaws in the design of high dimensional index structures and algorithms is that they have generally relied on the use of the L_p-norm as the default method for building index structures and algorithms. This assumption is perhaps rooted in the initial development of index structures for spatial applications in which the L_2 norm has special interpretability in 2 or 3-dimensions. However, this interpretability is not really relevant for high dimensional applications. For many carefully studied domains of data such as information retrieval and categorical data, the distance functions are usually based on the statistical aspects of the corresponding feature vectors.

The design of distance functions for well studied data domains such as Information Retrieval provides some useful hints for the high dimensional case. We list some of the practical desiderata for effective dimensionality resistant distance functions below.

(1) Contrasting: Straight-forward extensions of the L_p-norm are disadvantageous from a practical perspective because they lead to the non-contrasting behavior of the distances. This non-contrasting behavior is because two high dimensional objects are unlikely to be very similar in all the dimensions. The averaging effects over the different dimensions may lead to a lack of contrast. A different way of developing contrasting distance functions is to make them sensitive to the number of dimensions on which two records are similar. This is not necessary to implement for low dimensional ap-

plications; for such cases, the L_p-norm is a reasonable solution. We will see that it is possible to design dimensionality sensitive distance functions which automatically adjust the similarity calculation mechanism in a way so as to continue to be contrasting with increasing dimensionality.

(2) Statistically Sensitive: It is very rare that the data is uniformly distributed along any given dimension. Some of the values may be more sparsely populated than others. This behavior is commonly noted in Information Retrieval applications in which most attribute values (corresponding to frequency of presence of words from a large statistical collection) are zero. This is one domain in which the design of effective distance functions has been very well studied [14]. Distance functions such as dice, cosine or jaccard coefficient [14] do not treat the attribute values uniformly. For example, for a given pair of documents only the attributes on which both documents have non-zero values (words which are present rather than absent) are relevant. This is because the sparsely present attributes have much smaller frequency of co-occurrance, and are therefore statistically more relevant to the recognition of similarity. Furthermore, even among these attributes, some are weighted more highly than others based on the relative presence of the values in the entire data set [14]. In fact, the problem of term-frequency and inverse-document-frequency normalization has been well studied and documented as an important problem with considerable qualitative effects on distance functions [14] in the IR domain.

There is no reason why such methods may not be generalized to arbitrary data sets. The use of statistical properties of distance functions masks out the noise effects in high dimensionality and is able to magnify the contrast by only using the statistically significant values.

(3) Skew Magnification: In high dimensional space, many of the attributes are correlated with one another. These correlations may be used in order to magnify the effect of high dimensional skews in measuring similarity. The use of inter-attribute correlations has been used for designing distance functions in categorical domains where there is no natural ordering of attribute values. In such cases, the use of inter-attribute summary information provides considerable insight into similarity of objects by examining whether commonly co-occurring inter-attribute values are present in the two objects [9].

In this paper, our motivation for incorporating the concept of inter-attribute similarity to arbitrary data sets is slightly different; we would like to use the inter-attribute similarity in order to increase the level of discrimination of the proximity calculation. The use of aggregate behavior of the data in terms of inter-attribute correlations becomes more important for high dimensional data, where there may be considerable redundancies, dependencies and relationships among the large number of attributes. The use of straightforward linearly separable distance functions such as the L_p-norm may be a very poor representation of proximity in high dimensional space, as most of the proximity information may be hidden in the aggregate summary behavior of the data.

(4) Compactness: An important practical requirement for distance function calculation is that of compactness. This refers to the fact that a distance function can be calculated efficiently in terms of time and space requirements. This is quite important, since we advocate the use of statistical information about the data set. Thus, such information needs to be maintainable and usable in a compact way. Also, the distance function calculation may be a bottleneck operation in many algorithms. Therefore, efficiency in such calculations is paramount to the success of such methods from a performance perspective.

In order to create a distance function which satisfies the needs discussed above, we will illustrate a function which provides greater weightage to proximity on a given dimension rather than lack of it. Even though this way of measuring similarity leads to a loss of information, (because many dimensions are not used in the similarity calculations) it is very effective at masking out the noise effects in high dimensionality. Furthermore, this reduction of noise effects leads to qualitative improvements which are sufficient to offset the corresponding loss of information.

4 Sample Distance Function

In this section, we will discuss a simple distance function for high dimensional objects. We do not necessarily claim that this technique is the optimum one from any perspective; however our intention is to show how a very naive application of the above-mentioned techniques and principles can substantially reduce the effect of the dimensionality curse for high dimensional applications.

One of the reasons for the lack of discrimination between the nearest and furthest neighbor is the fact that for every pair of points there are dimensions with varying distances to the corresponding values in the target. The dominant components of distance functions such as the Euclidean metric are the dimensions on which the points are farthest apart; for the particular case of high dimensional data, this results in very poor measurement of similarity. This is because when the dimensionality is high, even the most similar records are likely to have a few feature values which are well separated because of noise effects and sparseness of the data; the exact degree of dissimilarity on these few noisy

dimensions will determine the nearest neighbor to the target. In general, for a given feature, we expect the values for two randomly picked records to be reasonably well separated (average separation along that range); there is no interesting statistical information in this fact. For distance functions such as the L_p-norm, the results of [6] show that the averaging effects of the different dimensions (many of which are noisy) start predominating with increasing dimensionality.

A different and complementary view of similarity would be one in which a predefined proximity threshold is defined for each dimension, and the overall similarity is defined *both* by the number and quality of similarity along the dimensions on which the two records are more proximate than this threshold. Thus, the similarity function is directly affected by the number of dimensions which have this interestingly high level of proximity, and beyond a certain quality threshold, the exact degree of dissimilarity on a given dimension is not considered relevant. Since the meaningfulness problem is sensitive to the data dimensionality, the criterion for picking this proximity threshold is also dependent on the data dimensionality.

4.1 Dimensionality Resistant Distance Functions by Proximity Thresholding

In order to perform the proximity thresholding, we discretize the data into several ranges. Specifically, we assume that each dimension is divided into k_d *equi-depth*[1] ranges. The reason for picking equi-depth ranges is that it provides better normalization in terms of distinguishing the records with respect to the aggregate data set behavior. Each of these is a contiguous range of values, such that a given range contains a fraction $1/k_d$ of the total number of records. Specifically, we denote the jth range for dimension i by $\mathcal{R}[i, j]$. In order to emphasize the sensitivity of k_d on the data dimensionality, we have used the dimensionality d in the subscript.

Let $X = (x_1, \ldots x_d)$ and $Y = (y_1, \ldots y_d)$ be two records. Then the set of dimensions on which the two records are similar are those which share the same ranges. Thus, for dimension i, if both x_i and y_i belong to the same range $\mathcal{R}[i, j]$, then the two records are said to be in *proximity* on dimension i. The entire set of dimensions on which the two records lie in the same range is referred to as the *proximity set*. Let $\mathcal{S}[X, Y, k_d]$ be the proximity set for two records X and Y for a given level of discretization. Furthermore, for each dimension $i \in \mathcal{S}[X, Y, k_d]$, let m_i and n_i be the upper and lower bounds for the corresponding range in the dimension i in whi ch the records X and Y are in proximity to one another. Then, for a given pair of records X and Y and a level of discretization k_d, the similarity between the records is given by:

$$PIDist(X, Y, k_d) = \left[\sum_{i \in \mathcal{S}[X,Y,k_d]} \left(1 - \frac{|x_i - y_i|}{m_i - n_i}\right)^p \right]^{1/p} \quad (1)$$

Note that the value of the above expression will vary between 0 and $|\mathcal{S}[X, Y, k_d]|$, since each individual expression in the summation lies between 0 and 1.

The above use of the similarity function guarantees a non-zero similarity compon ent only for those dimensions, in which the two records are proximate enough. The use of equi-depth partitions ensures that the probability that two records have a component in the same partitions given by $1/k_d$. Thus, on the average the above summation is likely to have d/k_d components. For more similar records, the number of such dimensions will be greater, and each such individual component is also likely to contribute more to the similarity value. The above function leads to the ignoring of the exact degree of dissimilarity on the distant dimensions: we see from the empirical tests in [4], that for the case of high dimensional data this creates a sparsity/noise reduction which outweighs the effects of information loss.

4.2 Picking the Proximity Threshold

We would like to pick the proximity threshold in a way that the meaningfulness of the problem is retained; yet the amount of information loss is minimal. Thus, k_d should be picked "just large enough" in order to retain meaningfulness with increasing dimensionality. An interesting mathematical analysis is provided in [4], which shows that for the worst kind of (uniformly distributed) data, it suffices to pick $k_d = \lceil \theta d \rceil$. This means that the distance function is increasingly stringent in discarding dimensions with increasing dimensionality. The idea is to provide a measure of the number of dimensions on which the records are close enough to make statistical sense based on a certain discretization threshold; higher thresholds provide better quality bounds for each dimension, but fewer percentage of dimensions. The theoretical results of [4] seem to indicate that in high dimensional space it is better to aim for higher quality bounds for each dimension; but a smaller *percentage* (not number) of retained dimensions. This results in more meaningful similarity computations for high dimensional problems.

An interesting aspect of this distance function is the nature of its sensitivity to data dimensionality; the choice of k_d ensures that for low dimensional applications it is somewhat similar to the L_p-norm; whereas for high dimensional applications, it behaves somewhat similar to IR-like distance functions in giving

[1] In equi-depth ranges, each range contains an equal number of records. In equiwidth ranges, each range contains a similar length of values covered.

greater weightage to those dimensions on which the records are most similar.

4.3 Use of Inter-Attribute Correlations

The use of inter-attribute correlations in order to measure similarity of categorical attributes has been discussed in [9]. This technique is also relevant for high dimensional data where considerable understanding of the trends in the data may be used for determining similarity. For example, the technique of generalized projected clustering [3] (which is an application-redefinition method) uses local correlations in a very direct way in order to determine similarity. Such local correlations can also be hidden in the distance function. An example of such a function is provided in [4].

5 Merits of Distance Function and Application Re-definitions

It has been shown in [4] that the above distance function can be used in conjunction with an index representation effectively so that the performance of the technique improves (in terms of fraction of data accessed) with dimensionality. As evident from detailed empirical results in the same paper; the nearest neighbors are qualitatively more meaningful as well. This method proposes a significant shift from most of the known indexing structures such as the X-Tree, VA-File, SR-Tree, and TV-Tree [5, 8, 11, 12], all of which are dependent on pre-existing distance functions such as the L_p-norm.

Distance function re-definition is very useful in cases when an application or algorithm is very dependent on the use of a similarity calculation subroutine; in other cases, application re-definition techniques similar to those discussed in [2, 3, 10, 13] may provide deeper insights into the localized behavior of the data. The relative merits of these two techniques are dependent on specific situations and needs; however, the general approach of re-designing distance functions and distance based applications for high dimensional problems is a promising line of future research.

References

[1] Aggarwal C. C., Hinneburg A., Keim D. On the Surprising Behavior of Distance Metrics in High Dimensional Space. *ICDT Conference*, 2001.

[2] Aggarwal C. C. et al. Fast Algorithms for Projected Clustering. *ACM SIGMOD Conference*, 1999.

[3] Aggarwal C. C., Yu P. S. Finding Generalized Projected Clusters in High Dimensional Spaces. *ACM SIGMOD Conference*, 2000.

[4] Aggarwal C. C., Yu P. S. The IGrid Index: Reversing the Dimensionality Curse for Similarity Indexing in High Dimensional Space. *ACM SIGKDD Conference*, 2000.

[5] Berchtold S., Keim D. A., Kriegel H.-P.: The X-Tree: An Index Structure for High-Dimensional Data. *VLDB Conference*, 1996.

[6] Beyer K., Goldstein J., Ramakrishnan R., Shaft U. When is Nearest Neighbors Meaningful? *ICDT Conference*, 1999.

[7] Das G., Mannila H., Ronkainen P. Similarity of Attributes by External Probes. *KDD Conference*, 1998.

[8] Weber R., Scheck H. J., Blott S. A Quantitative Analysis and Performance Study for Similarity Search Methods in High Dimensional Spaces. *VLDB Conference*, 1998.

[9] Ganti V., Gehrke J. Ramakrishnan R. CACTUS: Clustering Categorical Data Using Summaries. *ACM SIGKDD Conference*, 1999.

[10] Hinneburg A., Aggarwal C. C., Keim D. What is the nearest neighbor in high dimensional spaces? *VLDB Conference*, 2000.

[11] Katayama N., Satoh S. The SR-Tree: An Index Structure for High Dimensional Nearest Neighbor Queries. *ACM SIGMOD Conference*, 1997.

[12] Lin K.-I., Jagadish H. V., Faloutsos C. The TV-tree: An Index Structure for High Dimensional Data. *VLDB Journal*, 3(4): pages 517–542, 1992.

[13] Chakrabarti K., Mehrotra S. Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces. *VLDB Conference*, 2000.

[14] Salton G., McGill M. J. Introduction to Modern Information Retrieval. *Mc Graw Hill*, New York, 1983.