in Proc. 9th Int. Symp. on Spatial and Temporal Databases (SSTD'05),
Angra dos Reis, Brazil, pp. 1 - 18, 2005

# Selectivity Estimation of High Dimensional Window Queries Via Clustering

Christian Böhm, Hans-Peter Kriegel, Peer Kröger, Petra Linhart

Institute for Computer Science, University of Munich, Germany
{boehm,kriegel,kroegerp,linhart}@dbs.ifi.lmu.de

**Abstract.** Query optimization is an important functionality of modern database systems and often based on estimating the selectivity of queries before actually executing them. Well-known techniques for estimating the result set size of a query are sampling and histogram-based solutions. Sampling-based approaches heavily depend on the size of the drawn sample which causes a trade-off between the quality of the estimation and the time in which the estimation can be executed for large data sets. Histogram-based techniques eliminate this problem but are limited to low-dimensional data sets. They either assume that all attributes are independent which is rarely true for real-world data or else get very inefficient for high-dimensional data. In this paper we present the first multivariate parametric method for estimating the selectivity of window queries for large and high-dimensional data sets. We use clustering to compress the data by generating a precise model of the data using multivariate Gaussian distributions. Additionally, we show efficient techniques to evaluate a window query against the Gaussian distributions we generated. Our experimental evaluation shows that this approach is significantly more efficient for multidimensional data than all previous approaches.

## 1 Introduction

The storage and management of vectors of a multidimensional feature space has become an important basic functionality of a database system. Advanced applications such as multimedia [1], CAD [2], molecular biology [3], etc. require efficient and effective methods for content based similarity search and data mining. Such methods are typically based on feature vectors of moderate or high dimensionality. Although a vast number of index structures [4, 5] and access methods [6] for vector data has been proposed, database management systems do not yet support the storage and retrieval of vector data in the same way as relational data from applications such as accounting and billing. In order to give full support to advanced applications the database system needs efficient and effective techniques for query optimization. One of the most important challenges in query optimization is the estimation of the selectivity of a query predicate. While a number of techniques to model the data distribution and thus to estimate the selectivity are known for one- and low-dimensional data spaces, this is still an unsolved problem for data spaces of medium to high dimensionality.

Three different paradigms of data modelling for selectivity estimation in general can be distinguished: Histograms, sampling, and parametric techniques. Of those three, only sampling can be directly applied without modification in higher dimensional data spaces. Many different sampling methods have been proposed. They share the common idea to evaluate the predicate on top of a small subset of the actual database objects and to extrapolate the observed selectivity. The well-known techniques differ in the way how the sample is drawn as well as in the determination of the suitable size of the sample. The general drawback of sampling techniques is that the accuracy of the result is strictly limited by the sample rate. To get an accurate estimation of the selectivity, a large sample (>10%) of the database is required. To evaluate the query on top of the large sample is not much cheaper than to evaluate it on the original data set which limits its usefulness for query optimization.

Histogram techniques, the most prevalent paradigm to model the data distribution in the one-dimensional case, have a different problem. This concept is very difficult to be carried over to the multidimensional case, even for low or moderate dimensional data. One way to adapt one-dimensional histograms to multidimensional data is to describe the distribution of the individual attributes of the vectors independently by usual histograms. These histograms are sometimes called marginal distributions. In this case, the selectivity of multidimensional queries can be determined easily provided that the attributes are statistically independent, i.e. neither correlated nor clustered. Real-world data sets, however, rarely fulfill this condition. Another approach is to partition the data space by a multidimensional grid and to assign a histogram bin to each grid cell. This approach may be possible for two- and three-dimensional spaces. However, for higher dimensional data this method becomes inefficient and ineffective since the number of grid cells is exponential in the dimensionality. Techniques of dimensionality reduction such as Fourier transformation, wavelets, principal component analysis or space-filling curves (Z-ordering, Hilbert) may reduce this problem to some extent. The possible problem reduction, however, is limited by the intrinsic dimensionality of the data set.

The idea of parametric techniques is to describe the data distribution by curves (functions) which have been fitted into the data set. In most cases Gaussian functions (normal distributions) are used. Instead of using one single Gaussian, a set of multivariate Gaussians can be fitted into the data set which makes the technique more accurate. Each Gaussian is then described by three parameters (mean, variance and the relative weight of the Gaussian in the ensemble). This approach can be transferred into the multidimensional case by two techniques. Like described above for histograms, the marginal distribution of each attribute can be modelled independently by a set of Gaussians. The multidimensional query selectivity can be estimated by combining the marginal distributions. This approach leads to similar problems like marginal histograms.

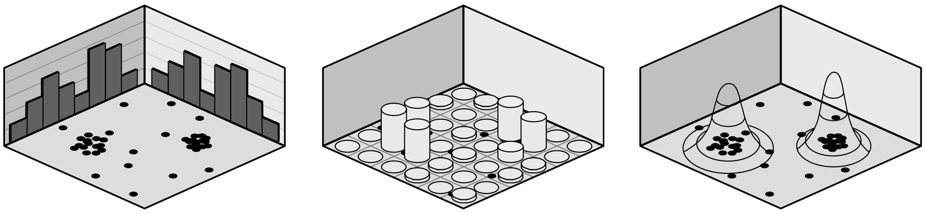
Therefore, our solution is different. Our technique directly models the multidimensional data distribution by a set of multivariate Gaussian functions. There are two options to use the Gaussian primitives: The Gaussians can either be used with a matrix containing both variances and covariances or with a vector of the multivariate variances only. As we will discuss later, both approaches have their advantages and disadvantages. When using Gaussians with covariance matrix, the data distribution can be described more accurately by a single primitive. On the other side, more storage is needed for the covariance matrices ($\mathrm{O}(d^2)$ for each Gaussian) compared to the variance vector approach ($\mathrm{O}(d)$ for each Gaussian). Moreover, the processing cost for reading the parameters and for the determination of the estimated selectivity is much higher when covariance matrices are used. Let us note that, unlike the approaches using marginal distributions, our Gaussian technique with no covariance matrix does not rely on the attribute independence assumption. This technique assumes attribute independence for each individual Gaussian primitive only, but places no constraints to the overall data distribution. We will discuss this issue in more detail in Section 4, an experimental validation is given in Section 5.

To obtain a collection of Gaussians distributions we apply a clustering algorithm. Clustering is the task of grouping vectors into different subsets (the clusters) such that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized. That means points belonging to the same cluster are close together whereas points of different clusters are far away from each other. Many different algorithms have been proposed such as k-means [7], single-link [8], density-based clustering [9, 10] and many others. Most of these algorithms use a point as a representative of each cluster. In contrast, the EM clustering algorithm (expectation maximization) [11] uses a multivariate Gaussian function as a cluster representative. We will discuss the suitability of different variants of the EM algorithm for our problem of getting a good approximation of the actual data distribution.

To summarize our contribution, we propose in this paper a new cost model for estimating the selectivity of multidimensional queries on top of vector data of medium to high dimensionality. The data distribution is represented by a set of multivariate Gaussian functions that have been determined using the EM clustering algorithm. We develop two methods for estimating the selectivity of window queries and range queries using the multivariate Gaussians. We demonstrate experimentally the superiority of our approach over competitive cost models based on histograms and sampling. The remainder of our paper is organized as follows: In Section 2 we discuss related work on selectivity estimation and point out our contribution. Section 3 and 4 describes in detail our proposed methods to find a representation of the data distribution by an ensemble of multivariate Gaussian functions using EM clustering and to estimate the selectivity on top of this model. Section 5 contains the experimental evaluation, and section 6 concludes our paper.

## 2 Related Work

In this chapter, we review current approaches for selectivity estimation and discuss their potentials.



(a) One dimensional histograms

(b) Multi dimensional histograms

(c) Selectivity estimation via clustering

**Fig. 1.** Visualization of different concepts for selectivity estimation.

### 2.1 Review

Recent work on selectivity estimation can be categorized into three classes, namely histogram-based methods, sampling-based methods, and parametric methods. In the following, we review and discuss the most important representatives of each class briefly.

**Histogram-based Methods.** The most widespread approach for selectivity estimation in practice is the use of histograms. In general, the data space is partitioned into buckets, and the frequency of points inside each bucket is computed. We can distinguish between one-dimensional and multi-dimensional histograms.

Selectivity estimation using one-dimensional histograms is based on the assumption that the attributes of the data set are independent, i.e. there is no correlation between different dimensions of the feature space. For each dimension, a histogram is built and the selectivity of a window query $q$ is estimated in each dimension separately. The selectivity of $q$ in the full-dimensional space is evaluated by multiplying the selectivity estimations for each attribute. Equi-width histograms [12] compute buckets of fixed size and variable point frequency, whereas equi-depth histograms [13] compute buckets of variable size and fixed point frequency.

With growing dimensionality of the feature space, the recombination of one-dimensional buckets becomes costly. Thus, in recent years, multi-dimensional histograms have been investigated. Multi-dimensional equi-depth histograms [14] partition the feature space into multi-dimensional buckets with variable size and fixed point frequency. In [14] an algorithm to construct multi-dimensional equi-depth histograms is presented that iteratively partitions the data space along each attribute into a fixed number of buckets, where the order of attributes is fixed. The selectivity of a window query $q$ is estimated analogously to one-dimensional histograms taking the buckets into account that intersect with $q$. The algorithm MHIST [15] partitions the data space along the single attributes in a similar way, but decides in each step which attribute is partitioned rather than processing the attributes in a fixed order.

STHoles [16] is a recent approach that proposes hierarchically organized multi-dimensional histograms. A histogram may contain another histogram completely, or may be completely covered by part of another histogram. The is-part of hierarchy of the histograms is represented as a tree where each node represents a bucket. Using this hierarchical concept, a non-uniform distribution inside a bucket $b$ can be adopted more accurately by several smaller buckets inside (that are part of) $b$. STHoles histograms are constructed using a set of sample queries as reference. Regions in the data space, that are queried more frequently, can thus be represented in more detail through a larger number of buckets. The histograms are refined after each query. However, the refinement procedure takes care that no more than a fixed upper bound of buckets is generated. If this upper bound is violated temporarily during reorganization, some buckets are merged.

In [17] the authors propose another strategy of computing multidimensional histograms using Wavelet transformation. In particular, the authors show how to apply a Wavelet transformation to one dimensional data sets. The data space is split evenly in a recursive fashion. The Wavelet coefficients are computed for each bucket. The resulting grid can be more fine grained than for traditional histograms because using the Wavelet coefficients the data is compressed more efficiently. For higher dimensional data, the authors in [17] suggest to split each attribute recursively in a given order.

**Sampling-based Methods.** A second approach for estimating the selectivity of queries is based on sampling. Usually, the selectivity of a query $q$ is estimated on a small sample of the database and is then extrapolated onto the entire database. The simplest way of computing a sample is the well-known random sampling method. A more data driven variant is adaptive sampling [18, 19].

A similar approach called 'double sampling' is proposed in [20]. The main difference to the adaptive sampling method is a different estimation of the sample size. In fact, the sample size is reduced using a two-way sampling procedure. However, there is no hint on how to choose the size of the first sample.

**Parametric Methods.** In [21], a method called Adaptive Selectivity Estimator (ASE) is proposed that tries to approximate the distribution of the data objects along one attribute using an appropriate polynomial function. This function is adopted and refined taking predefined queries into account, and minimizes the squared error between the real and the estimated selectivity. ASE is evaluated in [21] using one- and two-dimensional data sets only.

### 2.2 Discussion

As noticed above, current approaches for selectivity estimation have severe drawbacks. Sampling techniques suffer from the fact, that the accuracy of the result is strictly constrained by the sample rate. High sample rates on the other hand are quite inefficient and limit the usefulness of sampling techniques for query optimization. One-dimensional histograms (cf. Figure 1(a)) rely on the attribute independence assumption, i.e. on the assumption that the attributes are neither correlated nor clustered. This is quite unrealistic in real-world data sets which rarely fulfill this condition. Multi-dimensional histograms (cf. 1(b)) become inefficient and ineffective for higher dimensionalities since the number of grid cells is exponential in the dimensionality. Techniques of dimensionality reduction are (at least) limited by the intrinsic dimensionality of the data set. Similar problems are prevalent using parametric methods.

In this paper, we propose the use of clustering to get an accurate characterization of the data by means of a collection of multivariate Gaussians (cf. 1(c)). Our two methods are called SEC (Selectivity Estimation via Clustering) and SEC+ and both use different variants of the EM clustering algorithm to extract a collection of Gaussian distributions. For SEC each Gaussian is represented by the mean value, the variances and the covariances and the relative weight of the Gaussian in the ensemble. SEC+ uses the same representation but leaves out the covariances. Based on these representations, SEC and SEC+ efficiently and effectively estimate the selectivity of window queries in spatial data. We empirically show that especially SEC+ yields significantly more accurate results than comparative methods, especially when applied to higher dimensional data.

## 3 SEC: Selectivity Estimation Via Clustering

The overall goal of representing a given data set for selectivity estimation is to find a model of the data that is as compact as possible (low amount of storage necessary) and as accurate as possible (for accurate selectivity estimations). The key idea of our new approach is to use a clustering algorithm to gain an accurate description of the data and then use this description for selectivity estimation. In this section, we describe both the clustering process and the method for selectivity estimation in detail.

### 3.1 Describing the Data Via Clustering

Clustering has gained a lot of attention from the data mining research community over the past decades. In particular, clustering is the task of grouping objects of a data set into classes (clusters), while maximizing intra-cluster similarity and minimizing inter-cluster similarity. An overview over recent work on clustering can be found e.g. in [22]. Often clustering algorithms can also be used to obtain a compact representation of a data set. An efficient way to represent a data set for selectivity estimation, is to use a mixture of different distribution functions. The most prominent algorithm that tries to describe the data by multiple distribution functions is the EM algorithm [11]. In the following, we describe a variant of this algorithm which is used by our selectivity estimation method SEC.

Let $\mathcal{D}$ be a set of $d$-dimensional points, i.e. $\mathcal{D} \subseteq \mathbb{R}^d$. The general idea of the EM algorithm is to describe the data by a mixture $M$ of $k$ Gaussian distributions. Note that the EM algorithm can also be seen as a variant of k-means clustering. Instead of assigning each object to a cluster as is the case for k-means-based clustering algorithms, it assigns each object to a cluster according to a weight representing the probability of membership.

Each cluster $C \in M$ is a tuple $C = (\mu_C, \Sigma_C)$, where

- $\mu_C$ is the mean value of all points in $C$ and
- $\Sigma_C$ is the $d \times d$ covariance matrix of all points in $C$.

To compute the probability distributions, we need the following concepts.

The probability with which a point $x \in \mathcal{D}$ belongs to a Gaussian distribution $C = (\mu_C, \Sigma_C)$ can be computed by:

$$P(x|C) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_C|}} e^{-\frac{1}{2}(x-\mu_C)^{\mathbf{T}}(\Sigma_C)^{-1}(x-\mu_C)}.$$

The combined probability for $k$ clusters can then be computed by:

$$P(x) = \sum_{i=1}^{k} w_{C_i} P(x|C_i),$$

where $w_{C_i}$ is the fraction of points that belongs to cluster $C_i = (\mu_{C_i}, \Sigma_{C_i})$, i.e. $w_{C_i}$ is the weight of $C_i$.

Then the probability that a point $x \in \mathcal{D}$ belongs to a cluster $C$ can be computed by the rule of Bayes:

$$P(C|x) = w_C \frac{P(x|C_i)}{P(x)}.$$

The accuracy of a mixture $M = (C_1, \ldots, C_k)$ of $k$ Gaussian distributions which describes how good the model approximates the actual data set can be computed by:

$$E(M) = \sum_{x \in \mathcal{D}} \log(P(x)).$$

The higher the value of $E(M)$, the more likely it is that the data set $\mathcal{D}$ has been generated by the mixture $M$ of $k$ Gaussian distributions. Thus, the aim of the EM algorithm is to optimize the parameters of $M$, i.e. the parameters of the $k$ Gaussian distributions $C_1, \ldots, C_k$, such that $E(M)$ is maximized. For that purpose, the algorithm proceeds in four steps:

1. *Initialization*
   Since the clusters, i.e. Gaussian distributions $C_1, \ldots, C_k$, are unknown at the beginning, a set of $k$ initial clusters are built randomly. For that purpose, each point $x \in \mathcal{D}$ is randomly assigned to one cluster $C_i$. An initial model is produced by computing $\mu_C$ and $\Sigma_C$ for each cluster $C \in M$
2. *Expectation*
   Based on the current model, the parameters $\mu_C$ and $\Sigma_C$ can be computed for each cluster $C \in M$ and the accuracy $E(M)$ of this mixture $M$ is obtained.
3. *Maximization*
   In this step the accuracy of the mixture is improved via a recomputation of the parameters of each of the $k$ clusters. Given a mixture $M$ of $k$ Gaussians, the parameters $\mu_C$, $\Sigma_C$, and $w_C$ of each cluster $C \in M$ are recomputed.

The resulting mixture $M'$ has an equal or higher accuracy than $M$, i.e. $E(M) \leq E(M')$. For improving the mixture, the parameters are recomputed as follows:

$$w_C = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} P(C|x),$$

$$\mu_C = \frac{\sum_{x \in \mathcal{D}} x \cdot P(C|x)}{\sum_{x \in \mathcal{D}} P(C|x)},$$

$$\Sigma_C = \frac{\sum_{x \in \mathcal{D}} P(C|x)(x - \mu_C)(x - \mu_C)^{\mathbf{T}}}{\sum_{x \in \mathcal{D}} P(C|x)}.$$

4. *Iteration*
   Step 2 and 3 are iterated until the accuracy of the improved mixture $M'$ differs from the accuracy of the previous mixture $M$ by a smaller value than a user specified threshold $\varepsilon$, i.e. until $|E(M) - E(M')| < \varepsilon$.

The result of the EM algorithm is a set of $k$ $d$-dimensional Gaussian distributions, each represented by the mean value $\mu$ and the covariance matrix $\Sigma$. The assignment of a point $x \in \mathcal{D}$ to a cluster $C$ is given by the probability $P(C|x)$. We thus can compute how likely a point is assigned to each of the $k$ clusters.

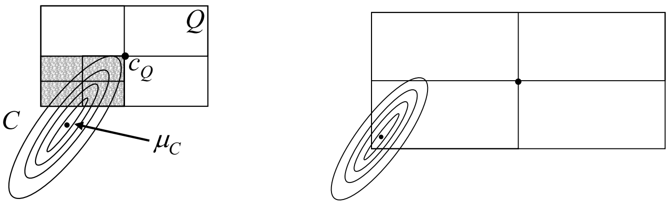
The accuracy of the result of the EM algorithm, i.e. the accuracy of the resulting mixture, depends on the initial mixture, i.e. on step 1 of the algorithm, and on the choice of $k$. In [23] a method for producing a good initial mixture is presented which is based on multiple sampling. It is empirically shown that using this method the EM algorithm achieves accurate clustering results. The authors further propose a method for determining a suitable number of clusters, i.e. a suitable value for $k$.

### 3.2 Selectivity Estimation of Window Queries

As discussed in the previous subsection, we describe the data distribution using $k$ Gaussian distributions each represented by a mean value and a covariance matrix. Let us note that this representation does not rely on the unrealistic attribute independence assumption nor has it problems in higher dimensions such as exponential storage cost that must be compensated by less accuracy.

In the following, we assume that $M$ is a mixture of $k$ Gaussian distributions computed by the EM algorithm applied on the database $\mathcal{D}$ as described above. We will also call $M$ a model that describes the distribution of the objects in $\mathcal{D}$ and we will use the two notions *Gaussian distribution* and *cluster* interchangeably for a given $C \in M$. A window query $Q$ is a list of $d$ pairs $(l^i, u^i)$, where $l^i$ and $u^i$ are the lower and upper bounds, respectively, of $Q$ in the $i$-th dimension, where $1 \leq i \leq d$. The center of $Q$ is denoted by $c_Q$.

Intuitively, a good estimation of the selectivity of a query $Q$ is the integral $\mathcal{I}(Q, C)$ of the intersection of $Q$ and each $C \in M$. A straightforward idea to estimate the selectivity of a query $Q$ using the model $M$ is the following. For each cluster $C \in M$ we can compute the probability that the center $c_Q$ is in



(a) Illustration

(b) Problem

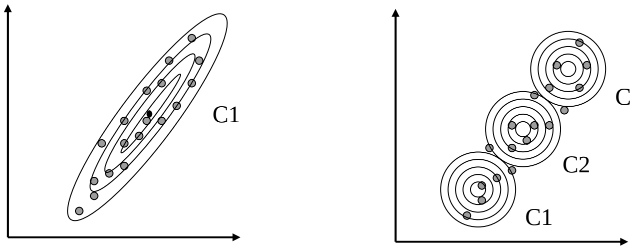
**Fig. 2.** The naive approach to selectivity estimation.

$C$, i.e. $P(c_Q|C)$. This probability can then be multiplied with the volume of the query. The resulting integral of the intersection of $Q$ and cluster $C$ is a first approximation of the selectivity of $Q$. If the integral is above a threshold $\varepsilon$, it may be interesting to further improve the estimation. We can achieve this by decomposing $Q$ into $2^d$ rectangles $Q_i$ of equal size and computing the integrals of the intersection of $C$ with each resulting rectangle $Q_i$. This can be iteratively continued until all decomposed $Q_i$ have an integral above $\varepsilon$. Then the selectivity of $Q$ w.r.t. $C$ can be computed by the sum of the integrals of the decomposed windows $Q_i$ having an integral above $\varepsilon$, multiplied by the weight of the cluster $w_C$. The overall selectivity of $Q$ is then simply the sum over all $C \in M$. This approach is illustrated in Figure 2(a). The query $Q$ is decomposed into four smaller windows. One of them (marked in gray) is further decomposed. The selectivity of $Q$ w.r.t. $C$ is the sum of the integrals of the intersection of each gray window with $C$.

We called this approach SEC (Selectivity Estimation via Clustering). The next chapter will present an approach called SEC+ that proposes certain improvements over the basic version SEC.

## 4 SEC+: Improved Selectivity Estimation via Clustering

Unfortunately, the simple idea of decomposing the query window rises several problems. First of all, the iterative decomposition of $Q$ into $2^d$ rectangles is quite inefficient and requires high storage cost. For an accurate estimation, however, we probably need multiple decompositions, i.e. several iterations of the decomposition process. Secondly, representing a window query only by its center has drawbacks, too. Especially in higher dimensional spaces, the center of $Q$ may be far away from any $C \in M$ even if $Q$ contains $C$. This is illustrated in Figure 2(b). Although query $Q$ contains a large part of $C$, the center of $Q$ is too far away from $\mu_C$ and the probability $P(c_Q|C)$ is far too small. Thus, multiplication of $P(c_Q|C)$ with the volume of $Q$ will yield a very small value, most likely below a reasonable threshold $\varepsilon$. A third problem is that the storage cost in SEC for a single cluster is relatively high ($d^2 + d$ values per cluster). Therefore, we modify



(a) SEC

(b) SEC+

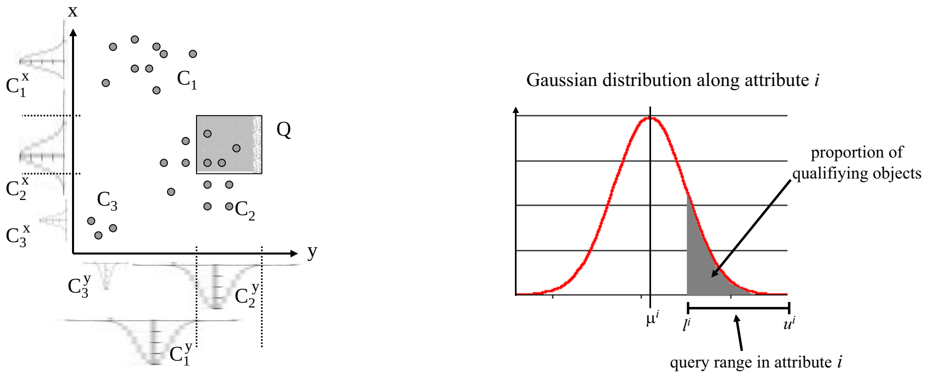
**Fig. 3.** Visualization of the model created for SEC (a) and SEC+ (b).

in SEC+ our data model, the clustering algorithm, as well as our method of selectivity estimation.
To reduce the storage cost per cluster our idea is to store only the diagonal elements of the covariance matrix, i.e. we store only the variance values of single attributes but no covariances between different attributes. This means that the Gaussian functions are oriented in an axis-parallel way rather than arbitrarily. Note that this does not mean we assume attribute independence for the complete data space (which would be unacceptable as discussed before). We assume only that the points which are associated to a common cluster observe the attribute independence assumption. This is much easier to motivate than demanding global independence for the complete data set because (1) the individual clusters contain data which is locally selected and (2) we can modify the EM algorithm to determine preferably clusters in which the assumption is fulfilled. (3) Due to saved storage cost we can maintain considerably more individual clusters in our model which generally allows a better adaptation to the real data distribution. In our SEC+-model a cluster is represented by its $d$-dimensional mean vector $\mu_C$ and a $d$-dimensional variance vector $var_C = (var_C^1, ..., var_C^d)$.

To guarantee that the EM-algorithm determines a good approximation of the real data distribution, we adapt the probability density function $P(x|C)$ for the clusters in order to use diagonal matrices only:

$$P(x|C) = \frac{1}{\sqrt{(2\pi)^d \prod_{1 \leq j \leq d} var_C^j}} e^{-\frac{1}{2} \sum_{1 \leq j \leq d} (x_j - \mu_{C,j})^2 var_C^j}.$$

This additionally makes the clustering algorithm more efficient as the variance vector is easier to determine and to invert (for computing the determinant) than a quadratic covariance matrix. Moreover, since our new model with axis-parallel Gaussians is now reflected in the algorithm and in the accuracy measure $E(M)$ the EM algorithm searches for an optimal model according to the new demands.


x
y
$C_1^x$
$C_2^x$
$C_3^x$
$C_1$
$C_2$
$C_3$
Q
$C_3^y$
$C_2^y$
$C_1^y$
Gaussian distribution along attribute *i*
proportion of qualifiying objects
$\mu^i$
$l^i$
$u^i$
query range in attribute *i*

(a) One-dimensional Gaussian distributions for three clusters.

(b) Intersection of a query and a Gaussian distribution in attribute $i$.

**Fig. 4.** Illustration of selectivity estimation using SEC+.

Figure 3 shows that our new model is not constraining the accuracy we reach for selectivity estimation. In case of strong correlations in the data set, the algorithm simply assigns more Gaussian functions to the data set. Due to the dramatically reduced storage cost for an individual Gaussian function, the overall storage requirement for the complete model is still much lower. Note that the algorithm may assign up to $d$ times more clusters without any extra storage cost. We will evaluate this issue experimentally in Section 5 and show that a higher number of axis-parallel Gaussians can even represent data distributions exhibiting non axis-parallel clusters more accurate than a lower number of non axis-parallel Gaussians.

Due to our new model, the method of selectivity estimation given a window query can also be improved with respect to efficiency and effectiveness in SEC+. Still, the integrals of the intersections of the query window and the Gaussian distributions are computed. However, each multi-dimensional Gaussian distribution $C$ is split into all one-dimensional distributions and the selectivity is determined using these one-dimensional distributions. This methods avoids the problems of the first approach. We will highlight the procedure of SEC+ in the following.

Instead of measuring the selectivity of a query $Q$ by the probability $P(c_Q|C)$ of a cluster $C \in M$, we split the $d$-dimensional Gaussian distribution into $d$ corresponding one-dimensional Gaussian distributions $C^i$. This is visualized in Figure 4(a). The integral $\mathcal{I}(Q, C)$ is the product of the integrals of all one-dimensional distributions, i.e. $\mathcal{I}(Q, C) = \prod_{i=1}^{d} \mathcal{I}(Q, C^i)$. Let us note, that this requires the assumption that the attributes are independent for the points belonging to that cluster. However, as discussed above this is no serious constraint. Figure 4(b) illustrates the integral of the intersection of a one-dimensional Gaussian and the query $Q$ for an attribute $i$. The query in that attribute is given by the interval $[l^i, u^i]$. The integral $\mathcal{I}(Q, C^i)$ measures the proportion of qualifying points, i.e. points of $\mathcal{D}$ that match the query $Q$.

Given the $d$-dimensional Gaussian distribution $C = (\mu_C, var_C)$, the $d$ corresponding one-dimensional Gaussian distributions can easily be obtained. These one-dimensional Gaussian distributions are represented by the mean $\mu_C^i$ and by the standard deviation $s_C^i$, i.e. $C^i = (\mu_C^i, s_C^i)$. The mean value $\mu_C^i$ is simply the $i$-th component of $\mu_C$. The standard deviation $s_C^i$ can be computed as follows:

$$s_C^i = \sqrt{var_C^i},$$

where $var_C^i \in var_C$ is the variance of attribute $i$. Obviously, at this point, we do not need the covariance matrix $\Sigma_C$, but only the variances $var_C^i$ which has the above discussed advantages. The integral $\mathcal{I}(Q, C^i)$ can then be computed rather straightforward. We simply materialize the standard Gaussian distribution $\Phi$ with $\mu = 0$ and $\sigma = 1$ in a table. The integral can then be computed as follows:

$$\mathcal{I}(Q, C^i) = |\frac{\Phi(u_i) - \mu_C^i}{\sigma_C^i} - \frac{\Phi(l_i) - \mu_C^i}{\sigma_C^i}|.$$

The selectivity of $Q$ w.r.t. a cluster $C$ is then the product of all attribute-wise integrals, i.e.

$$\mathcal{I}(Q, C) = \prod_{i=1}^{d} \mathcal{I}(Q, C^i).$$

The overall selectivity of a query $Q$ is estimated as the weighted sum of selectivities of $Q$ w.r.t. all $C_i \in M$, formally

$$\text{SEC+}(Q, M) = \sum_{c \in M} w_C \cdot \mathcal{I}(Q, C).$$

The pseudo code of our method SEC+ is given in Figure 5. In the next section we will show experimentally that SEC+ is superior to SEC and to other comparative methods.

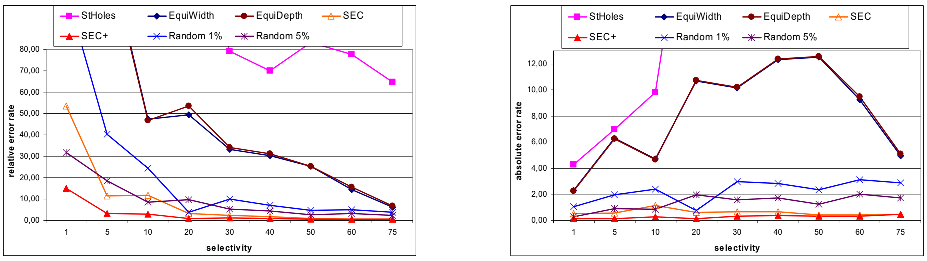
## 5 Experimental Evaluation

In this section, we present a broad experimental evaluation of SEC, SEC+ and comparative methods on synthetic and real-world data sets. We used randomly generated window queries throughout all our experiments. To judge the accuracy of each selectivity estimation method we used two measurements to compute the error rate of each method, the relative error rate and the absolute error rate. Let $S_Q$ be the true selectivity and $S'_Q$ the estimated selectivity of a query $Q$. Let $n$ be the number of tuples in the considered data set. The relative error rate $E_r(q)$ measures the error of the estimation w.r.t. the true selectivity, formally

$$E_r(Q) = \frac{|S_Q - S'_Q|}{S_Q}$$

```
SEC+ (SetOfObjects $\mathcal{D}$, Query $Q$)
  Compute model $M$ of $\mathcal{D}$ by EM($\mathcal{D}$);
  for each cluster $C_i \in M$ do
    for each dimension $j$ of $\mathcal{D}$ do
      Compute $\mathcal{I}(Q, C_i^j)$
    end for
      Compute $\mathcal{I}(Q, C_j) = \prod_{i=1}^{d} \mathcal{I}(Q, C_j^i)$
  end for
  Compute $\textsc{Sec}{+}(Q, M) = \sum_{i=1}^{k} w_{C_i} \cdot \mathcal{I}(Q, C_i)$.
```

**Fig. 5.** Pseudocode of SEC+.



(a) Relative error rate

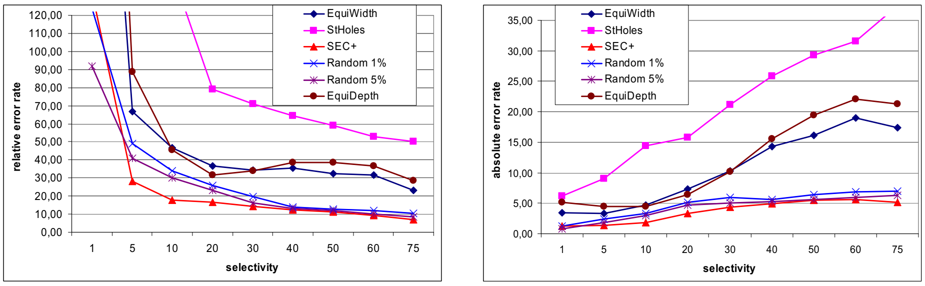
(b) Absolute error rate

**Fig. 6.** Results on synthetic data set with Gaussian non-axis parallel ellipsoid clusters.

The absolute error rate $E_a(Q)$ measures the error of the estimation w.r.t. the size of the database, formally

$$E_a(Q) = \frac{|S_Q - S'_Q|}{n}$$

We compared SEC+ to several competitive methods, including random sampling using 1% of the database as sampling rate (indicated in the diagrams by "Random 1%"), random sampling using 5% of the database as sampling rate (indicated in the diagrams by "Random 5%"), one-dimensional equi-width histograms using 30 intervals per dimension (indicated in the diagrams by "EquiWidth"), one-dimensional equi-depth histograms using an interval capacity of 5% of the data set (indicated in the diagrams by "EquiDepth"), and multi-dimensional histograms (STHoles) using 1,000 randomly generated sample queries to establish the histogram as proposed in [16]. In [23] a method to choose the parameter $k$ and the intial clustering for any variant of the EM algorithm is



(a) Relative error rate

(b) Absolute error rate

**Fig. 7.** Results on the "Abalone" data set.

described. We used this method to determine $k$ and the intial clustering for SEC and SEC+.

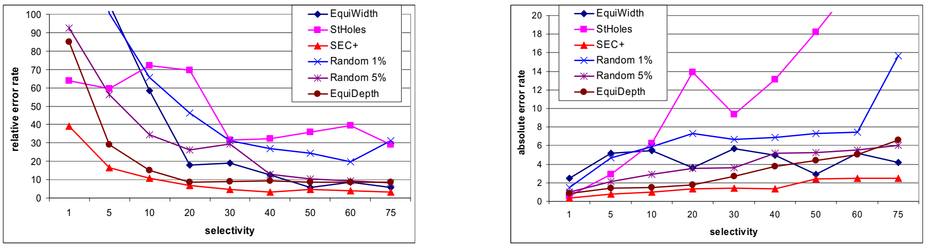
## 5.1 Comparison of SEC and SEC+.

In Figure 6, we compared SEC and SEC+ with its competitors. SEC is the variant that uses covariances throughout the EM-clustering process, whereas SEC+ only uses variances. All our experiments show that SEC+ performs as good or even better than SEC. For a comparison of SEC, SEC+ and other techniques, we applied all methods on a data set of 10,000 5-dimensional tuples containing several non-axis parallel Gaussian clusters. This data set was chosen because it seems to favor SEC which uses covariances over SEC+. SEC outperforms all competitive methods besides SEC+. However, as illustrated in Figure 6, the use of the covariances during clustering does not achieve a gain in accuracy compared to the improved SEC+ algorithm. In case of queries with lower selectivity, SEC+ even outperforms SEC which uses covariances during clustering. Let us note that we needed less storage for SEC+ than for SEC in our experiments but achieved better results when using SEC+ rather than SEC. This result was repeated in all other experiments, justifying the use of SEC+. Thus, throughout the rest of our evaluation, we will only show the results of SEC+.

## 5.2 Accuracy of SEC+.

*Comparison with other methods.* We applied SEC+ and the comparative methods on several real-world data sets. Due to space limitations, we focus on two data sets. The first one is the "Abalone" benchmark data set from the UCI Machine Learning Database Repository[1]. It contains approximately 4,200 objects in a 8-dimensional feature space. The second data set is a gene expression

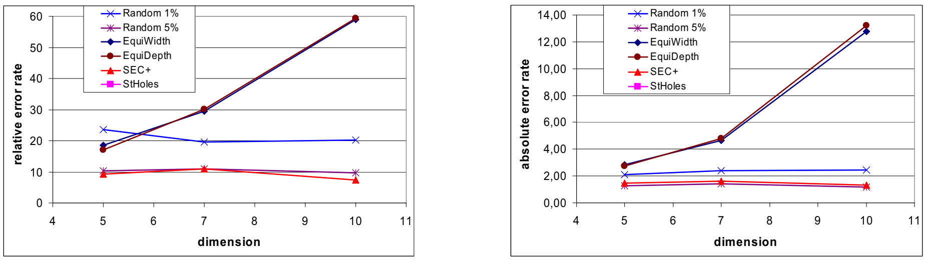
[1] `http://www.ics.uci.edu/~mlearn/MLRepository.html`



(a) Relative error rate

(b) Absolute error rate

**Fig. 8.** Results on the gene expression data set.



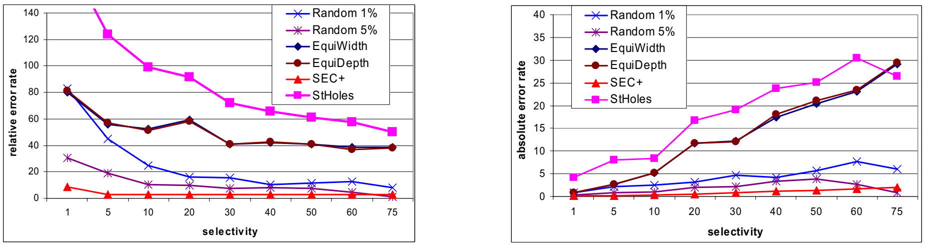
(a) Relative error rate

(b) Absolute error rate

**Fig. 9.** Comparison of the accuracy w.r.t. dimensionality of the data set.

data set from our project partners[2] and contains approximately 1500 objects in a 5-dimensional feature space. We evaluated the error rates of SEC+ and the comparative methods w.r.t. the selectivity (in %) of the queries. The results on the "Abalone" data set are depicted in Figure 7. SEC+ outperforms all other methods regarding relative and absolute error rates. Only for very selective queries (<5%), random sampling with 5% sampling rate is slightly better. However, a sampling rate of 5% is rather high for large databases. A similar observation can be made from Figure 8 illustrating the results on the gene expression data set. Again, SEC+ outperforms all other comparative methods w.r.t. both the relative error rate and the absolute error rate, even for very selective queries. The histogram based approaches perform slightly better than on the "Abalone" data set, especially compared to the sampling based approaches. We guess that this can be explained with the lower dimensionality of the gene expression data set. Both experiments show, that SEC+ outperforms competitive approaches in terms of accuracy especially in high dimensional spatial data.

[2] Genomatix SW GmbH: `http://www.genomatix.de/`



(a) Relative error rate

(b) Absolute error rate

**Fig. 10.** Results on the synthetic data set with 20% noise.

*Accuracy w.r.t. data dimensionality.* We evaluated the accuracy of SEC+ and the five comparative methods w.r.t. the dimensionality of the data set using synthetic data of 10,000 tuples and a sample query $q$ having a selectivity $S_q = 10\%$. The results are visualized in Figure 9. As expected, we can observe that the accuracy of random sampling methods are independent of the dimensionality of the data set, whereas the accuracy of histogram-based methods detoriates with increasing dimensionality. Let us note that StHoles is not shown in the charts because its error rates are far above the interval shown here. It can also be seen that the accuracy of SEC+ is independent of the data dimensionality. In a 5-, 7- and 10-dimensional feature spaces, SEC+ performs better than all other techniques besides 5% random sampling. But even 5% random sampling which is already very inefficient for large data sets, is only as good as SEC+.

*Influence of noisy data.* Next, we tested the influence of noisy data on the accuracy of SEC+ and the competitive methods. Since SEC+ relies on clustering, noisy data may cause problems in generating an accurate compression of the data distribution and thus may influence the selectivity estimation. Figure 10 illustrates the error rates of SEC+ and the comparative methods w.r.t. the selectivity of the query on a synthetic data set of 10,000 tuples of 5 dimensions with 80% of the data belonging to clusters and 20% noise objects. As it can be seen, SEC+ is quite robust against noisy data. For a broad range of query selectivity, SEC+ outperforms its competitors w.r.t. the relative and absolute error rates. Again, the sampling based methods are ranked second followed by one-dimensional histograms. Equi-depth and equi-width histograms produce nearly the same results in that experiment.

## 6 Conclusions

Advanced database applications rely on accurate and efficient query optimization. One key step for query optimization is the estimation of the selectivity of a given query. Recent approaches for selectivity estimation have problems with medium to high dimensional data spaces and/or usually require a high sampling rate to achieve accurate results.

In this paper, we proposed two new methods for selectivity estimation of spatial window queries called SEC (Selectivity Estimation via Clustering) and SEC+. Our solutions are based on modelling the data through a set of multivariate Gaussian functions which are computed using different variants of the EM clustering algorithm. Two techniques to derive an accurate estimation of the query size using the generated models are discussed in detail. A broad experimental evaluation illustrates that SEC+ outperforms existing approaches in terms of accuracy. In particular, SEC+ is robust against the dimensionality of the data space and can handle noisy data effectively.

## References

1. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., Equitz, W.: "Efficient and Effective Querying by Image Content". Journal of Intelligent Information Systems **3** (1994) 231–262
2. Mehrotra, R., Gary, J.: "Feature-Based Retrieval of Similar Shapes". In: Proc. 9th Int. Conf. on Data Engineering, Vienna, Austria,. (1993) 108–115
3. Shoichet, B.K., Bodian, D.L., Kuntz, I.D.: "Molecular Docking Using Shape Descriptors". Journal of Computational Chemistry **13** (1992) 380–397
4. Berchtold, S., Keim, D.A., Kriegel, H.P.: "The X-Tree: An Index Structure for High-Dimensional Data". In: Proc. 22nd Int. Conf. on Very Large Databases (VLDB'96). (1996)
5. Lin, K.I., Jagadish, H.V., Faloutsos, C.: "The TV-tree an index structure for high-dimensional data". VLDB Journal: Very Large Data Bases **3** (1994) 517–542
6. Weber, R., Schek, H.J., Blott, S.: "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces". In: Proc. 24th Int. Conf. on Very Large Databases (VLDB'98). (1998) 194–205
7. McQueen, J.: "Some Methods for Classification and Analysis of Multivariate Observations". In: 5th Berkeley Symp. Math. Statist. Prob. Volume 1. (1967) 281–297
8. Sibson, R.: "SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method". The Computer Journal **16** (1973) 30–34
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), Portland, OR. (1996) 291–316
10. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: "OPTICS: Ordering Points to Identify the Clustering Structure". In: Proc. ACM Int. Conf. on Management of Data (SIGMOD'99). (1999)
11. Dempster, A.P., Laird, N.M., Rubin, D.B.: "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society, Series B **39** (1977) 1–31
12. Selinger, P.G., Astrahan, M.M., Chamberlin, D.D., Lorie, R.A., Price, T.G.: "Access Path Selection in a Relational Database Management System". In: Proc. ACM Int. Conf. on Management of Data (SIGMOD'79). (1979)

13. Piatetsky-Shapiro, G., Connell, C.: “Accurate estimation of the number of tuples satisfying a condition”. In: Proc. ACM Int. Conf. on Management of Data (SIGMOD’84). (1984)
14. Muralikrishna, M., De Witt, D.J.: “Equi-Depth Histograms For Estimating Selectivity Factors For Muli-Dimensional Queries”. In: Proc. ACM Int. Conf. on Management of Data (SIGMOD’88). (1988)
15. Poosala, V., Ioannidis, Y.E.: “Selectivity Estimation without the Attribute Value Independence Assumption”. In: Proc. 23rd Int. Conf. on Very Large Databases (VLDB’97). (1997)
16. Bruno, N., Chaudhuri, S., Gravan, L.: “STHoles: a Multidimensional Workload-aware Histogram”. In: Proc. ACM Int. Conf. on Management of Data (SIGMOD’01). (2001)
17. Matias, Y., Vitter, J.S., Wang, M.: ”Wavelet-Based Histograms for Selectivity Estimation”. In: Proc. ACM Int. Conf. on Management of Data (SIGMOD’98). (1998) 448–459
18. Lipton, R., Naughton, J.: “Query size estimation by adaptive sampling”. In: Proc. ACM Symp. on Principles of Database Systems (PODS’90). (1990)
19. Lipton, R., Naughton, J., Schneider, D.: “Practical selectivity estimation through adaptive sampling”. In: Proc. ACM Int. Conf. on Management of Data (SIGMOD’90). (1990)
20. Hou, W.C., Ozsoyoglu, G., Dodgu, E.: “Error-constrained Count Query: Evaluation in Relational Databases”. In: Proc. ACM Int. Conf. on Management of Data (SIGMOD’91). (1991)
21. Chen, C.M., Roussopoulos, N.: “Adaptive Selectivity Estimation Using Query Feedback”. In: Proc. ACM Int. Conf. on Management of Data (SIGMOD’94). (1994)
22. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Academic Press (2001)
23. Fayyad, U., Reina, C., Bradley, P.: ”Initialization of Iterative Refinement Clustering Algorithms”. In: Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD’98). (1998)