MultiCraft

International Journal of Engineering, Science and Technology
Vol. 2, No. 6, 2010, pp. 119-130

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY
www.ijest-ng.com

MySQL based selection of appropriate indexing technique in hospital system using multiclass SVM

Narendra Kohli[*] and Nishchal K. Verma

Department of Electrical Engineering, Indian Institute of Technology Kanpur, INDIA
[*]Corresponding Author: e-mail: nkohli@iitk.ac.in, Tel +91-512-2582426, Fax. +91-512-2533412

Abstract

This paper deals with selection of appropriate indexing technique applied on MySQL Database for a health care system and related performance issues using multiclass support vector machine (SVM). The patient database is generally huge and contains lot of variations. For the quick search or fast retrieval of the desired information from the database, it becomes exigent to select and implement the appropriate indexing technique. Multiclass SVM is proposed to be the optimal solution as SVM could be trained with patient datasets to select the appropriate indexing method in the database of MySQL. B-Tree index is directly applied in most of the cases but other indexing techniques such as Bitmap and Hash are also used as per requirement. Using SVM, corresponding to the given search parameter for information, retrieval of information from the patient database results in quick retrieval of required information in the minimum processing time. Various SVM based methods like one against one, one against all, and fuzzy decision function are implemented for classification on standard electrocardiogram (ECG) datasets chosen from the University of California at Irvine (UCI) Cardiac Arrhythmias database. Suitable SVM method to be used is highlighted in the result section. Depending upon different feature values of ECG dataset, the SVM is trained to identify which categories or classes the given data points belong to. In the similar way appropriate indexing techniques will be selected in the case of health care system application by using multiclass SVM for fast retrieval of patient data. If the findings are made an integral part of the hospital system it could facilitate quick and easy retrieval of patient information, doctor information, hospital information etc. from any hospital as per the requirement. Health care system applications will be developed as front end in .Net whereas, back end as database in MySQL.

Keywords: B-tree indexing, MySQL, Support vector machine, Smart card.

1. Introduction

Networking of the hospitals in today's world is essential and necessity of the society. It increases the accuracy and speed of patient information circulation which results in better services for patients. It also enhances the working efficiency of the hospital system (Hassol *et al*., 2004; Liu *et al*., 2006). The central idea behind developing such a application is to obtain, store, analyze, process and usage of patient information that concerns with the doctors, hospitals, laboratory tests etc. It requires administrator to generate a 10 digit unique patient-ID at registration counter in a hospital. The smart card to be issued to the patient containing his basic information like name, address, phone no. etc. with unique patient ID Information related to the patient like doctor diagnosis, test reports, MRI, CT-scan images etc will be stored in the databases of the hospital server as per the patient ID for future usage. For example in quick diagnosing the illness of the concerned patient. While visiting to the hospital for treatment, patient will need to carry only smart card and the administrator of the hospital or the doctor will use this smart card through card reader to extract the patient related needed information. The collected information need to be stored and retrieved through a database management tool. MySQL is preffered because it is widely used free, fast and reliable open source relational database management tool. It is an extensible, open storage database engine which allows multiple variations such as Berkeley DB, InnoDB, Heap and MyISAM. MySQL integrates seamlessly with a no of programming languages and other web based technologies. Clearly the system demands for fast retrieval of patient data and this highlights the importance of the selection of appropriate indexing techniques to be used in MySQL. Training of support vector machine is done based on existing patient database and its corresponding favoured indexing

technique to predict and thus, select the appropriate method of the indexing for the chosen search parameter in the given application. Multiclass SVM based methods basically assign labels to instances, where the labels are drawn from a finite set of several elements. This is achieved by reducing the single multiclass problem into multiple binary classification problems where each of the problems yields a binary classifier, which is expected to produce an output function that gives relatively high values for cases belonging to the positive class and relatively low values for cases belonging to the negative class. Three well-known SVM based methods to build such binary classifiers are one-against one (OAO), fuzzy decision function (FDF) and one-against-all (OAA), where each classifier distinguishes between one of the labels to the rest and between every pair of classes respectively. For the classification of new instances in cases of OAA, it uses winner-takes-all strategy, in which the class of the data is assigned by the classifier with the highest output function. Meanwhile the care is to be taken that the output functions are calibrated to produce comparable scores. In the classification by OAO approach, it makes use of max-wins voting strategy, in which each of the binary classification assigns the instance to one of the two classes, as such giving a vote to the assigned class, and finally the class with most votes determines the class of the given instance. To illustrate the selection of appropriate indexing technique in MySQL, classification has been done using support vector machine on ECG dataset and results have been proposed.

This paper is categorized into 9 Sections. An overview of the health care system is dealt in section 2. Section-3 discusses with the problem formulation of the health care system. The methodology to tackle with the existing health care problem is presented in section-4. Various findings observed are organized in various cases and mentioned in section-5. In section-6, we briefly review SVMs for multi classification. One-against-one, fuzzy decision function and one-against-all is briefly explained in section-7. Results based on SVM implementation on a standard ECG dataset are stated and discussed in section-8. Finally, in section-9, recommendations are suggested based on the observations.

2. Literature Review

The requirement for automation systems in hospitals and health centers are gaining more and more importance in the present scenario (Kohli and Verma, 2010). A patient might have his registration in different information systems. As per the requirement it is important to have the sharing of patient information possible between different parties without ignoring any of his privacy constraints and other related problems. The privacy constraints of patient-data were discussed in Peyret (1994). Marschollek and Demirbilek (2006) describe an application system that employs the new German Health Card to provide a technical solution for tracking patient pathways. The aim is to improve the flow and availability of health care information. Jha *et al*. (2009) surveyed all acute care hospitals of American Hospital Association. They determined the presence of specific electronic record functionalities and the proportion of hospitals that had such systems in their clinical areas. They also examined the relationship of adoption of electronic health records to specific hospital characteristics and factors that were reported to be barriers to or facilitators of adoption. A systematic review of the literature was performed to examine the impact of electronic health records (EHRs) on documentation time of physicians and nurses and to identify factors that may explain efficiency differences across studies (Poissant *et al*., 2005). Liu *et al*. (2006) proposed the usage of smart card for storing the important information of the patient. Nowadays varieties of smart cards are available in market. These cards have proven to be quite convenient tokens in terms of identification and authentication in day to day activities (Lockett, 2003). Depending upon the requirement, different types of smart cards can be used to store the patient information (Campbell and Stoupa, 1990; Scherrer, 1995). Patient access to their health care record (EHR) and web based communication between patients and providers can potentially improve the quality of health care (Hassol *et al*., 2004). Many Electronic Health Record (EHR) systems fail to provide user friendly interface due to the lack of systematic consideration of human centered computing issues. Such interfaces can be improved to provide easy to use, easy to learn and error-resistant EHR system to the users. To evaluate the usability of an EHR system and suggest area of improvement in the user interface was discussed in (Saitwal *et al*., 2010). Kim *et al*. (2001) and Weed (1991) came up with a client- server agent that allows access of a portal to the every permitted information system of the hospital that consists of PACS, RIS and HIS via the intranet and the internet. Query optimization in MySQL is discussed in (MySQL paper). Classification and hence selection of appropriate indexing technique is needed in MySQL for the smooth functioning of the healthcare system while requiring minimum time for processing, which could be implemented using multiclass support vector machine (Rifkin and Klautau, 2004) and (Bredensteniner and Bennett, 1999). This has been discussed and explained further in this paper through its implementation on ECG dataset. It is an emerging technique which is feebly implemented around the world.

3. Problem Formulation

3.1: If a patient is registered in one hospital and integration and retrieval of patient information is not possible in the hospital system then while taking the consultation with doctor, patient may easily forget to inform about his/her allergy to the medicine and thus, may not be able to explain the previous treatments properly which might result in incorrect prescriptions to the patient. Keeping this in mind a web based application i.e. smart card based online health care system is being proposed. The proposed system is shown in Figure 1, which can be found in Kohli and Verma (2010). The servers for the hospitals with very high technical configuration are required. All the servers of the hospitals should be connected with the centralized server of the hospital through internet. A very high bandwidth dedicated internet lease line must be used for the system. A smart card reader / writer unit needs to be attached to each computer of the hospital system. The proposed health care system would be loaded to all the servers of all the hospitals. These hospitals will be connected via intranet and internet. The patient smart card stores some important information

like unique patient ID, name, sex, date of birth, blood group etc. As per the patient-ID, patient details like treatment prescriptions, test reports, images like MRI, CT-scan etc. have been stored in the database of the hospital server. On the basis of stored details of the patient, doctor can prescribe the proper medicine. For fast retrieval of patient data different indexing techniques have been proposed in MySQL. Appropriate indexing technique has been selected using multiclass support vector machine. MySQL tuning has been used while designing the on line health care system.

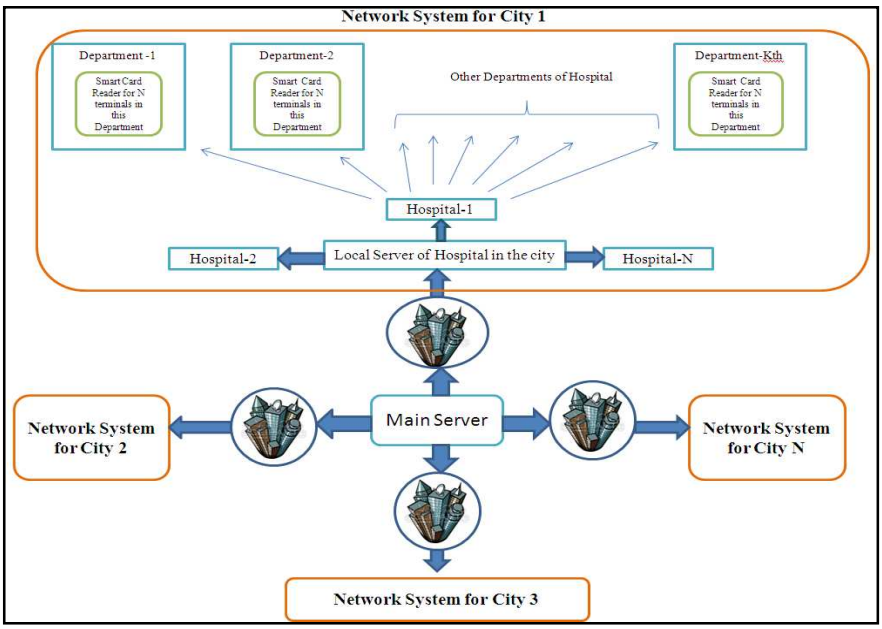

Figure 1. Smart card based on line health care system

3.2: An application, Smart card based online health care system has been developed that can serve maximum number of hospital and provide a platform for their interaction. As per the application, the system will generate the patient ID that is unique for maximum possible time interval. It will store patient information optimally and maximum possible time. To increase the performance of the application, we have considered the following technical aspects.

- Normalization of tables.
- Maximum utilization of memory.
- Imposing necessary constraints on tables.
- Selecting best possible structure for queries.
- Tuning and optimizing queries.
- Optimal selection of join order and join algorithm.
- Managing insert operation to reduce load from server.
- Front end should be strong enough to reduce basic user faults and provide easy interaction.
- Speed up the DML (insert, update, delete) operations.

4. Methodology

The proposed application is implemented as follows.

The patient will provide basic information through smart card (having 10 digit patient ID and personal information) at registration counter where it will be uploaded to server. The smart card will store patient ID and personal information like name, address, DoB etc. of a person.. Patient's files may be in one of the two forms; image or PDF. Images are stored separately in image tables and PDFs in PDF table. These tables contain Image ID and PDF ID and their description. Image ID and PDF ID will be stored in their respective tables as foreign keys that may be in Patient-doctor or Patient Lab or in Patient-Room Tables. All the patient related information will be on a centralized database server, so the movement of files is not required. The required files for a particular patient & their lab test report & images, prescriptions, scan document & diagnosis reports can be retrieved from central database server. Accessing the information related to patients will be controlled by various levels of access control and it will prevent the unauthorized access from the data base. By normalizing the database all possible redundancies will be removed

Optimization is covered at application design level, database design level, memory utilization & optimization of queries for accessing the data base.

Few points about the optimization are as follows:

- Application is studied well and requirements are identified. These requirements are categorized on object level and respective tables are created to store that information. After that database is normalized up to BCNF (Boyce Codd Normal Form) level and 14 tables are created.
- Size of attributes (their respective data types) is further minimized in order to compact record size, so that maximum number of records can be stored in a single data page. Selected queries are designed to handle all possible search criteria.
- Indexes are created to speed up above read access. First data base IDs are minutely studied to find out what possible values different attributes can take and what could be the size of tables. Considering both the factors Hash, B-tree (clustered and non-clustered) and Bitmap indexes are proposed.
- Order of attributes in composite indexes are studied and justified.
- Every hospital should have day to day patient information locally to improve the performance particularly in cases of frequent insert and update operations.
- Due to monotonically increasing nature of patient ID there is always a problem of unbalancing of B-tree indexes. Index portioning & reverse key indexing is proposed to get rid-off this problem.
- In a multi-table join queries, we have studied the nature of the table & depending on the characteristics of the table, their order will be pre-estimated. MySQL queries hints can also be given to optimizer to follow that order to achieve the optimal performance.
- Properly joined algorithms are justified in respective cases and hints are given to optimizer.
- Selecting the most optimal execution path by the optimizer is the time consuming process. By giving the hints we can reduce the time taken by the optimizer up to some level.
- To improve the DML operation by the hospital we try to keep the data initially at the local server. This scheme reduces load from central server and problem of frequent unbalancing of index structure is also removed.
- Every day a particular time is selected to update central server by collecting information from local servers automatically. The operation to refresh the central server may be made more frequent if required.
- Transaction log is permanently off at central server as we don't need any log for recovery because all the information are at the local server and can be reproduced easily.
- All primary key, foreign key constraints are removed from central server as those constraints are already checked at local servers. So there is no need to revalidate the already validated information. It results in improved data loading performance at the central server.
- For updating central server we recommend to use bulk copy and bulk insert schemes.

5. Findings

In this paper we have discussed few cases on the database where out of 14 tables, 4 tables store patient related information, which are,

- Patient stores registration information of patient.
- PatientDoctor stores doctor related information like diseases, etc.
- PatientLab stores test report of patients.
- PatientRoom stores Room IDs, daily charges and other attributes of room where patient is admitted.

5.1 Case 1: Strategy for retrieving patient information based on patient ID attribute: B-Tree is a tree data structure that keeps data sorted and allows searches, sequential access, insertions and deletions in logarithmic amortized time. We have two kinds of B-Tree indexes: Clustered and Non-clustered. A table can contain only one clustered index and any number of Non-clustered indexes. Clustered index is better for columns with fixed size and unique values, PatID attribute satisfies these criteria but PatID attribute in PatientDoctor, PatientLab and PatientRoom is foreign key so these tables may contain more than one record for one PatID. For

these tables Non-clustered index is preferred. In this case B-Tree index gives better results than other indexing Techniques. It gives optimal results for insert, update and read operations.

5.2 Case 2: Strategy for retrieving patient information in original tables and day tables:We do not insert new records in original table but store it in some day tables. Those patients that got registered today will be treated by day tables. And rest by Original tables. Late night at a particular time day table records will be deleted and merged in original tables. The day table will be ready to hold new records from the next day morning. A ***hash*** index stores key value pairs based on a pseudo randomizing function called *hash function* so in such cases hash indexing is used in original table as it gives better result. And for day tables we further optimize read access over B-Tree index as it over come overflow situation. We perform all data insertion in original table at some fixed time and also maintain all index structures at that time.
B-tree indexing will be implemented on day tables.

Original tables and day tables will be used in following manner.

If (Date of registration== System date) then
Fetch records from Day Tables;
Else
Fetch records from Original Tables;

5.3 Case 3: Strategy for retrieving patient information based on multiple attributes as patient name and Date of Birth (DoB): We retrieve patient data using two attributes i.e. patient name and Date of Birth (DoB). Indexing also depend upon the order of attributes. For patient name we are using variable size as *varchar* and for DoB we are using *smalldatetime* of fixed size. When we are searching patient data by using this combination then first we have to search for the PatID from the Patient table and rest process remains the same as in the normal query. Returning back to the problem of order of attributes, an index works good with fixed size attributes and DoB attribute of 4 bytes in size suits for it. Hence, order should be DoB followed by patient name. But this order will not fit to our application due to following reasons.

The DoB may hold 365 different dates for a year. If we assume average life of a person as 75 years then DoB may contain 75*365 = 27375 set of unique values. If we cover 100 years scenario then this set will hold 36500 unique values. A patient DoB will take one value from this value set. If we take only first four characters of name attribute then it can generate 26*26*26*26 = 456976 unique values. This set is 12 times larger than DoB set. But most combinations of this set may be assumed as garbage names so the actual name set will always less than 27375. The another benefit of taking name attribute as the first column for indexing is that while comparing in *where* clause if its first byte does not match with the first byte in the corresponding database, then comparison of further bytes will be skipped and it will be taken over to the next value in the name attribute. But in case of DoB, all the four bytes will be checked every time. So searching of exact record set by name attribute will be easier and faster. More over if index will be generated on name attribute then it will be useful for other queries based on name.

Clearly, as {PatName, DoB} format is preferred over {DoB, PatName} format; BTree index becomes more useful than Hash index for the same. Since we have already created clustered index on PatID of Patient, the only option left for Index Architecture is to create a Non-clustered index.

5.4 Case 4: Strategy for retrieving patient information based on disease attribute: If we have to find number of patients of a particular disease. Disease information is stored with patient registration details in patient table. This attribute will hold a particular value from a list of disease that may contain few disease names. Since we have small number of distinct values, bitmap index must be used as bitmap indexes uses bit arrays and answer queries by performing bitwise logical operations on bitmaps.

5.5 Case 5: Strategy for retrieving patient information based on TestID attribute: If we have to find out number of enrolment for a particular test. This information can be found by the TestID attribute of TestBooking table. This attribute also hold less data that is some ID from a fixed set of TestID's. Here also Bitmap index should be used.

5.6 Case 6: Strategy for retrieving patient information based on DoctorID attribute: If we have to find out the number of patient under a particular doctor. This information can be found by DoctorID attribute of PatientDoctor table. This case is similar to test and disease forecast. So again bitmap index must be useful.

5.7 Case 7: Strategy for retrieving patient information based on HospitalID, DoctorID, LabID and TestID attributes: Index for other select queries

- Retrieve hospital details.
- Retrieve doctor details.
- Retrieve lab details.
- Retrieve test details.

Above information's will be retrieved from respective Hospital, Doctor, LabDetails and TestDetails tables. All the above tables have large number of records and new records will be updated time to time. Here Non-clustered B-Tree index should give most optimal result for select, insert and update operations.

5.8 Proposed indexing scheme: As per case1 to case 7, the proposed indexing scheme for health care system will be as per Table 1.

Table 1. Proposed indexing scheme

	Table name	Attribute	Indexing scheme
Original Tables (At Central Server)	Patient	PatID	Hash
		PatName, DoB	BTree
		Disease	Bitmap
	PatientDoctor	PatID	Hash
		DoctorID	Bitmap
	PatientLab	PatID	Hash
	PatientRoom	PatID	Hash
	TestInfo	TestID	Bitmap
	Hospital	HospitalID	BTree
	Doctor	DoctorID	BTree
	LabDetails	LabID	BTree
	TestDetails	TestID	BTree
Day Tables (At Local Server)	Patient	PatID	BTree
		PatName,DoB	BTree
	PatientDoctor	PatID	BTree
	PatientLab	PatID	BTree
	PatientRoom	PatID	BTree

6. Support Vector Machine

SVM are strong classifiers in the field of machine learning and we will be using SVM based one-against-all method for finding the best indexing technique in the given application. SVMs were designed for binary classifications and its algorithm can be better understood with a mathematical explanation and example as discussed in (Liu *et al.*, 2008). Let $S=\{(x_1,y_1),(x_2,y_2),...,(x_l,y_l)\}$ be a training set where x_i are m-dimensional attribute vectors, $y_i \varepsilon \{-1, +1\}$, $y_i = -1$, and $y_i = +1$ for class 1 and class 2, respectively. According to (Vapnik, 1995), the SVMs classifier is defined as follows:

$$D(x) = w^T \Phi(x) + b = 0 \quad , \tag{1}$$

where $\Phi(x)$ is a mapping function, w^T is a vector in the feature space, and b is a scalar.
To classify the data linearly separable in the feature space, the decision function satisfies the following condition:

$$y_i (w^T \Phi(x) + b) \geq 1 \, ,$$

$$\text{for} \quad i = 1,...,l \quad . \tag{2}$$

Among all the separating hyper planes, the optimal separating hyper plane with maximal margin between two classes can be formed as follows and as shown in Figure 2:

$$\min_{w,b} J(w,b) = \frac{1}{2} w^T w \tag{3}$$

subject to (2). If the training data are nonlinearly separable, the hard margin constraints are relaxed by introducing slack variables ξ_i in (2) as follows:

$$y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i$$

$$\text{for} \quad i = 1,...,l \quad , \tag{4}$$

$$\xi_i \geq 0$$

$$\text{for} \quad i = 1,...,l \quad . \tag{5}$$

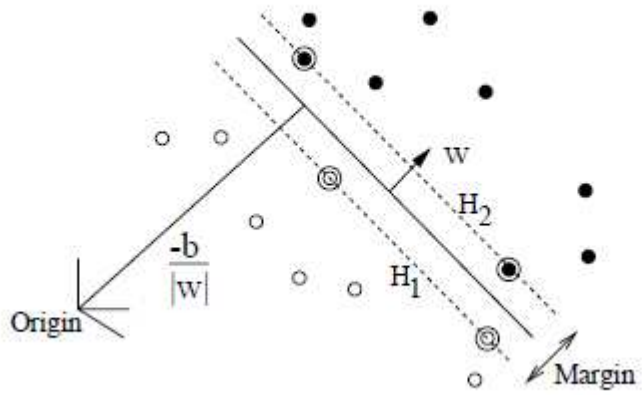

Figure 2: Linear separating hyper planes for the separable case. The support vectors are circled.

In order to obtain the optimal separating hyper plane, we should minimize

$$\min_{w,b,\xi_i} J(w,b,\xi_i) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{l} \xi_i \quad . \tag{6}$$

As per (4) and (5), parameter γ determines the tradeoff between the maximum margin and the minimum classification error. The optimization problem of (6) is a convex quadratic program and can be solved using Lagrange multiplier method. By using Lagrange multipliers α_i and β_i (i = 1, 2…..l), the Lagrangian function can be constructed as follows:

$$\begin{aligned} L(w,b,\alpha_i,\xi_i,\beta_i,) &= J(w,b,\xi_i) \\ &- \sum_{i=1}^{l} \alpha_i \{ y_i [w^T \phi(x_i) + b] - 1 + \xi_i \} - \sum_{i=1}^{l} \beta_i \xi_i \end{aligned} \tag{7}$$

According to the Kuhn–Tucker theorem, the solution of the optimization problem using Largarngian function is as follows:

$$w = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i) \quad . \tag{8}$$

The training examples (x_i, y_i) with nonzero Lagrangian coefficients α_i are called support vectors. By solving the following convex quadratic programming problem, we can find the α_i coefficients.

$$\max[-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j (\phi(x_i)^T \cdot \phi(x_j)) \alpha_i \alpha_j + \sum_{i=1}^{l} \alpha_i] \tag{9}$$

subject to

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \qquad \text{for} \quad i = 1, ..., l \quad , \tag{10}$$

$$0 \le \alpha_i \le \gamma, \qquad \text{for} \quad i = 1, ..., l \quad . \tag{11}$$

By substituting (8) into (1), the classifier can be obtained. Given new input x, $f(x)$, can be estimated by using (12). If $f(x) > 0$, the sample belongs to class 1; otherwise class 2 as,

$$f(x) = \text{sgn}\{\sum_{i=1}^{l} \alpha_i y_i \cdot (\phi(x_i)^T \cdot \phi(x)) + b\} \tag{12}$$

where

$$\text{sgn}(x) = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases} .$$

In (12), the pair wise inner product in the feature space can be computed from the original data items using a kernel function (Aizeman *et al*, 1964), (Saitoh, 1988) and the kernel function can be denoted by

$$K(x, x_i) = \phi(x)^T \cdot \phi(x_i) \quad . \tag{13}$$

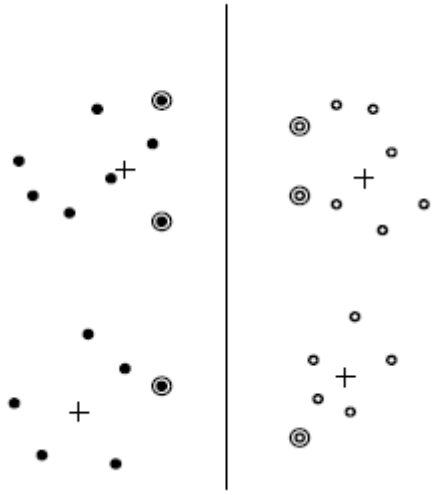

Figure 3: A classical RBF network finds the centers of RBF nodes by k-means clustering (marked by crosses).

A classical RBF network and an SVM with RBF kernels uses RBF nodes centered on the support vectors (circled), i.e., the data points closest to the separating hyper plane (the vertical line illustrated) is shown in Figure 3. The typical kernel functions include polynomial kernel functions and radial basis functions (RBFs). In this way, $f(x)$ can be rewritten as follows:

$$f(x) = \text{sgn}\{\sum_{i=1}^{l} \alpha_i y_i \cdot K(x_i, x) + b\} \quad . \tag{14}$$

7. Multiclass Classification Algorithm

The multiclass classification problem refers to assigning each of the observations into one of *k* classes. In this section, we have introduced the one-against-one, fuzzy decision function and one-against-all methods. The figures depicting the methods discussed in this section can be found in (Abe, 2006). At first, we introduce the one-against-one method.

As discussed in Liu *et al.* (2008), let us assume $S=\{(x_1,y_1),(x_2,y_2),...,(x_l,y_l)\}$ is a training set, where $x_i \in R^m$ and $y_i \in (1,2,.....k)$. For the one against-one method (Kreßel, 1998), one needs to determine $k(k-1)/2$ classifiers for the k - classes problems. The optimal hyperplane with SVMs for class i against class j is

$$D_{ij}(x) = w_{ij}^T \phi(x) + b_{ij} = 0, \qquad i < j, 1 < j \le k, 1 \le i < k \quad ,$$

where $w^T{}_{ij}$ is a vector in the feature space, $\Phi(x)$ is a mapping function, and b_{ij} is a scalar.
Here the orientation of the optimal hyperplane is defined as per the following equation:

$$D_{ij}(x) = -D_{ji}(x) \tag{15}$$

7.1 One-against-One method

For the input vector, one computes

$$D_i(x) = \sum_{j \ne i,\, j=1}^{k} \text{sgn}(D_{ij}(x)) \tag{16}$$

and classifies x into the class

$$\arg\max_{i=1,...,k}(D_i(x)) \tag{17}$$

7.2 Fuzzy Decision Function Method

To overcome the unclassifiable region, the FDF method based on the one-against-one scheme was introduced (Vapnik, 1995) as shown in Figure 4. For the input vector x , the 1-D membership function $m_{ij}(x)(i, j = 1,2,...,k)$ in the directions orthogonal to the optimal separating hyperplanes $D_{ij}(x) = 0$ is defined as follows:

$$m_{ij}(x) = \begin{cases} 1, & 1 \le D_{ij}(x) \\ D_{ij}(x), & otherwise \end{cases} \tag{18}$$

In Vapnik (1995), the membership functions $m_i(x)$ are given by

$$m_i(x) = \min_{j=1,\ldots,k}(m_{ij}(x)) \tag{19}$$

Using (18), sample x is classified into the class

$$\arg\max_{i=1,\ldots,k}(D_i(x)). \tag{20}$$

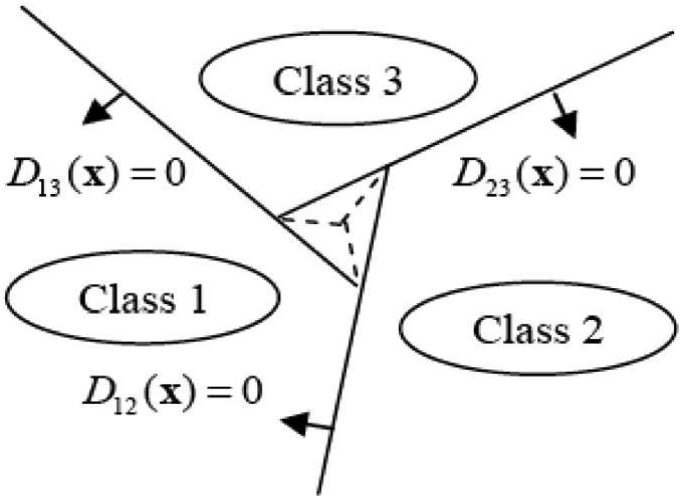

Figure 4: Fuzzy decision function method based on one against one scheme

7.3 One-against-all method

For a *k* class problem, the one-against-all method constructs *k* SVM models. The *i*th SVM is trained with all of the training examples in the *i*th class with positive labels and all other examples with negative labels. The final output of the one-against-all method is the class that corresponds to the SVM with the highest output value (Debnath, 2004). Thus, by solving the optimization problem in (3)-(5) using all the training samples in the dataset, the decision function of the *i*th SVM is

$$D_i(x) = w_i^T \phi(x) + b_i \tag{21}$$

The input vector **x** will be assigned to the class that corresponds to the largest value of the decision functions.

The class of **x** = $\arg\max_{i=1,..,k}(D_i(x))$

8. Results and Discussion

Usage and implementation of SVM for the selection of appropriate indexing technique in hospital system can be further elucidated through an example. Here, for the implementation of SVM based methods we have taken the standard multivariate ECG dataset which is chosen from the University of California at Irvine (UCI) Cardiac Arrhythmias database (Kohli *et al,* 2010) and thereby used results (Murphy and Aha, 1998) to conclude and validate our proposed theory. The experiments were conducted on a personal computer with 1.5 GHz CPU and 1 GB of RAM. Initially this database contained 452 instances and 279 attributes, of which 206 attributes are linear valued and the rest are nominal. But owing to the presence of many missing values and also zero valued columns, it became imperative to preprocess and resize the dataset while maintaining the reliability and relevance of the dataset. Thus, columns containing all zeroes or all missing values were removed first from the dataset, followed by the removal of rows having missing values in the dataset and removal of classes having insignificant number of instances. This resized the dataset to total 377 instances and 166 attributes, distributed into 6 classes with class 1 referring to 'normal' ECG, class 2 to 5 referring to different classes of arrhythmia and class 6 referring to the rest of unclassified ones. Table Table 1 in Kohli *et al* (2010) shows different arrhythmia classes with number of instances belonging to each of the classes in the dataset.

All the SVM based methods taken here were trained by half of the total dataset chosen fairly from the main dataset ensuring representation of all classes present in the required percentage. The remaining half of the main dataset was used for testing and analysis purpose. Table 2 in Kohli *et al* (2010) shows the representation of each of the classes in the training and the testing datasets.

We applied the SVM based methods using Gaussian kernel. The kernel parameter σ and the regularization parameter lambda λ in the Gaussian kernel were empirically optimized in Table 2 (Kohli *et al,* 2010) and Table 3 (Kohli *et al,* 2010) respectively,

by minimizing the error rate on the validation dataset in order to obtain the best accuracy rate in terms of percentage and thus ensuring better classification of the arrhythmia dataset.

For the classification of a data through SVM, there are various methods available by which SVM can be implemented. We have chosen three well-known and widely used SVM based methods, one-against-one, fuzzy decision function and one-against-all methods for comparison purpose. It can be observed from the classification results of arrhythmia in ECG dataset that SVM based one-against-all algorithm shows the higher percentage of accuracy rate and thereby better tendency to classify data more accurately. The SVM was trained and optimized by classifying the dataset at various σ values. That particular configuration of the system is chosen or that σ value is fixed for computation in testing at which the highest percentage of accuracy rate is obtained; and next, the system was optimally converged by classifying it again at various λ values. The range of values taken for σ and λ values are, $\sigma = [2^{-4}, 2^{-3}, 2^{-2}, ..., 2^{7}, 2^{8}]$ and $\lambda = [2^{-4}, 2^{-3}, 2^{-2}, ..., 2^{7}, 2^{8}]$. It is found that though one-against-one and fuzzy decision function are quite popular method for SVM classification but in terms of performance in healthcare system dataset, one-against-all dominates over all other methods used for comparison in ECG classification. Clearly, OAA method results in the highest accuracy rate but in general, the very high accuracy rate is difficult to obtain. This could be due to the presence of a particular class sweeping maximum share of number of instances in the total dataset. This reflects the further potential of OAA to give even higher results in cases of ECG datasets with more uniform distribution, thereby ensuring better training of the system.

Table 2. The accuracy rate (in %) wrt σ value

No.	σ	OAO	FDF	OAA
1	2^{-4}	62.77	62.77	62.77
2	2^{-3}	62.77	62.77	62.77
3	2^{-2}	62.77	62.77	62.77
4	2^{-1}	62.77	62.77	62.77
5	2^{0}	62.77	62.77	62.77
6	2^{1}	62.77	62.77	62.77
7	2^{2}	62.77	62.77	62.77
8	2^{3}	62.77	62.77	69.15
9	2^{4}	**69.15**	62.77	**72.87**
10	2^{5}	64.89	62.77	68.61
11	2^{6}	62.77	62.77	63.30
12	2^{7}	62.77	62.77	62.77
13	2^{8}	62.77	62.77	62.77

Table 3. The accuracy rate (in %) wrt λ value at best σ value

No.	λ	OAO	FDF	OAA
1	2^{-4}	3.72	62.77	68.09
2	2^{-3}	3.72	62.77	68.09
3	2^{-2}	3.72	62.77	68.09
4	2^{-1}	54.26	62.77	**73.40**
5	2^{0}	**69.68**	62.77	72.34
6	2^{1}	69.15	62.77	72.87
7	2^{2}	69.15	62.77	72.87
8	2^{3}	69.15	62.77	72.87
9	2^{4}	69.15	62.77	72.87
10	2^{5}	69.15	62.77	72.87
11	2^{6}	69.15	62.77	72.87
12	2^{7}	69.15	62.77	72.87
13	2^{8}	69.15	62.77	72.87

In the example discussed here, the SVM is trained based on ECG dataset having data of different kinds of arrhythmias with different features and the new test data is predicted to belong to a particular arrhythmia category based on its classification results obtained by the SVM based method. In the same way we are proposing SVM technique as appropriate indexing technique for the considered database. Like various arrhythmias in the ECG dataset taken here, the patient dataset also consists of various different categories based on their characteristics. There are generally various search parameters based on which retrieval of patient data could be required. The retrieval may be needed on the basis of patient name, patient DOB or may be on the basis of particular disease name, etc. In the existing technique for the information retrieval from the patient database in MySQL, the indexing technique B-Tree is fixed for all cases. Thus, if we use SVM classification to categorize the MySQL database and thus select the

most suitable indexing technique corresponding to the given search parameter and thereby use that particular indexing technique for information retrieval from the patient database, instead of directly applying B-Tree index for any case of indexing it results in quick retrieval of required information in the minimum processing time. .

9. Recommendations and Future Work

This paper presents selection of appropriate indexing technique in MySQL for the proposed hospital system and its performance issues are discussed using SVM technique. Here, we have proposed the implementation of different indexing techniques in accordance to the Table 1 for quick and easy retrieval of patient information. In the beginning, SVM is supposed to be trained with data pertaining to different indexing techniques. The corresponding appropriate indexing is obtained using multiclass SVM and thus selected for further processing and required information retrieval. The outcomes in the case of ECG dataset taken above clearly indicate that among various SVM based methods, one-against-all could be preferred for implementation purpose to obtain results with higher accuracy. Retrieval of patient data is faster after implementation of suitable indexing techniques, i.e. Hash indexes, Bitmap indexes and B-Tree indexes in the corresponding cases instead of directly using only B-Tree indexing in the MySQL. In the future, the modified SVM based method will be proposed for selection of appropriate indexing technique in MySQL to make the information retrieval process even faster.

Acknowledgement

We acknowledge Mr. Abhishek Roy, student of NIT Surathkal for his help in the preparation of paper.

References

Abe S., 2006. Support vector machines for pattern classification. *Springer-Verlag*, London, U.K.

Aizeman M., Braverman E. and Rozonoer L., 1964. Theoretical foundations of potential function method in pattern recognition learning. *Autom. Remote Control*, Vol. 25, pp. 821–837.

Benscart R. and Paradinas P., 1991. Smart card for health care, in telematics in medicine. *Elsevier Science Publishers* B.V., North, Holland.

Bredensteniner E.J. and Bennett K.P., 1999. Multicategory classification by support vector machines. *Comput. Optim. Appl.*, Vol. 12, pp. 53-79.

Campbell J.R. and Stoupa R., 1990. The patient, the provider, the processor: information management in ambulatory care. *Proceeding SCAMC*, IEEE Computer Society Press 1990, pp. 930-940.

Charegaonkar Vishal, Nair Kiran Hariharan, and Gautam Gaurav, Smart Card Transaction Processing-Case study: Applying best practices to improve performance, IBM, web address: www.ibm.com/websphere/developer/zones/hipods Date: 1 February 2008

Debnath R., Takahide N. and Takahashi H., 2004. A decision based one-against-one method for multi-class support vector machine. *Pattern Anal. Appl.*, Vol. 7, pp. 164-175.

Hassol A, James M., Walker, Kidder D, Rokita K, Young D, 2004. Patient experiences and attitudes about access to a patient electronic health care record and linked web messaging. *Journal of the American Medical Informatics Association*, Vol. 11, No. 6, pp. 505-513.

Huser, V.,Narus, S.P.,Rocha, R.A., 2010. Evaluation of a flowchart-based EHR query system: A case study of RetroGuide, *Journal of Biomedical Informatics*, Vol. 43, No. 1, pp.41-50.

Jha Ashish K., DesRoches Catherine M., Campbell Eric G., Donelan Karen, Rao Sowmya R., Ferris Timothy G., Shields Alexandra, Rosenbaum Sara and Blumenthal David, 2009. Use of electronic health records in U.S. hospitals, *N Engl J Med*, Vol. 360, pp. 1628-1638.

Kardas Geylani,Turhan Tunali E, 2006. Design and implementation of a smart card based healthcare information system, *Computer Methods and Programs in Biomedicine*, Vol. 81, No. 1, pp. 66-78,

Kim J., Feng D.D., Cai T.W. and Eberl S., 2001. Integrated multimedia medical data agent in e-health. *Pan-Sydney Area Workshop on visual Information Processing*, Sydney, Australia.

Kohli N. and Verma N.K., 2010. Performance issues of health care system with audio and Video Facilities. *Proceedings of the 3rd international conference on Data Management, IMT Ghaziabad*, India.

Kohli N. and Verma N.K., 2010. Performance issues of health care system using MySQL. *3rd IEEE International Conference on Computer Science and Information Technology*, Chengdu, China.

Kohli N., Verma N.K. and Roy A., 2010. SVM based methods for arrhythmia classification in ECG. *2010. 1st IEEE International Conference ICCCT-2010, MNNIT,Allahabad.*

Kreßel U. H. G., 1998. Pairwise classification and support vector machines. *In Advances in Kernel Methods-Support Vector Learning*, Schölkopf B., Burges C. and Smola A., Eds., MIT Press, Cambridge, MA, pp. 255–268.

Liu B., Hao Z. and Tsang E.C.C, 2008. Nesting one-against-one algorithm based on SVM's for pattern classification. *IEEE Transactions on Neural Networks*, Vol. 19, No. 12, pp. 2044-2052.

Liu Chien-Tsai, Yanga Pei-Tun, Yeha Yu-Ting and Wangb Bin-Long, 2006. The impacts of smart cards on hospital information Systems: an investigation of the first phase of the national health insurance smart card project in Taiwan, *International Journal of Medical Informatics*, Vol. 75, No. 2, pp. 173-181

Lockett E., Park S., Jiang G.C., Riddle M., 2003. Security aspects of smart cards. Term project CS 574 Fall 2003 San Diego state University.

Marschollek Michael, and Demirbilek Edip, 2006. Providing longitudinal health care information with the new German health card-a pilot system to track patient pathways, *Computer Methods and Programs in Biomedicine*, Vol.81, No. 3, pp. 266-271.

Murphy P.M. and Aha D.W., 1998. UCI Machine Learning Repository Database Online. http://archive.ics.uci.edu/ml], *University of California, School of Information and Computer Science*, Irvine, CA,

MySQL: www.MySQL.com.

Rifkin R. and Klautau A., 2004. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, Vol. 5, pp. 101-141,

Saitoh S., 1988. Theory of Reproducing Kernels and Its Applications. *Longman*, Harlow, U.K.,

Saitwal, H., Feng, X., Walji, M. , Patel, V., ,Zhang, J, 2010. Assessing performance of an electronic health record (EHR) using cognitive task analysis, *International Journal of Medical Informatics*, Vol. 79, No 7, pp. 501-506.

Scherrer J.R., 1995. The hospital information system-integrated patient records. *Elsevier Science Ireland Ltd*., Vol. 48.

Vapnik V.N., 1995. The nature of statistical learning theory. *Springer-Verlag*, London, U.K.

Weed L.L., 1991. Knowledge coupling: new premises and new tools for medical care and education. *Springer Verlog*, New York.

Biographical notes

Narendra Kohli was born in India on April 13, 1963. He received M. Sc. Engg. from Kiev University, Kiev in 1988 and perusing Ph.D. from Indian Institute of Technology Kanpur, India since 2004. He is currently an Associate Professor and Head with the Department of Computer Science & Engineering, Harcourt Butler Technological Institute Kanpur, India. He is a Member of IE (India), Fellow of IETE (India), and senior member of ISE.

Dr. Nishchal K.Verma was born in India on September 9, 1973. He received the B.E. degree from the Faculty of Engineering, Dayalbagh Educational Institute, Agra, India, in 1996, the M.Tech. degree from the Indian Institute of Technology (IIT) Roorkee, Roorkee, India, in 2003, and the Ph.D. degree from IIT Delhi, New Delhi, India, in 2007, all in electrical engineering. He is currently an Assistant Professor with the Department of Electrical Engineering, IIT Kanpur, India. His research interests include fuzzy systems, neural networks, data mining, fault diagnosis, bioinformatics, color segmentation, video clip or image sequence modeling, machine learning, and computational intelligence. Dr. Verma is a reviewer for several reputed national and international journals and conferences, including the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, and CYBERNETICS: PARTS A, B, AND C, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and *Pattern Recognition*.

Received July 2010
Accepted September 2010
Final acceptance in revised form September 2010