Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 03-038

Data Reduction in Support Vector Machines by a Kernelized Ionic
Interaction Model

Hyunsoo Kim and Haesun Park

September 22, 2003

Data Reduction in Support Vector Machines by a Kernelized Ionic Interaction Model

Hyunsoo Kim and Haesun Park*

Department of Computer Science and Engineering,
University of Minnesota,
200 Union Street S.E., 4-192 EE/CS Building,
Minneapolis, MN 55455, U.S.A.

September 19, 2003

Keywords: Data Reduction, Support Vector Machines, Ionic Interaction Model (IoI), Kernelized IB2 (KIB2), KIB2-IoI, Kernel Parameter Optimization

Abstract

A major drawback of support vector machines is that the computational complexity for finding an optimal solution scales as $O(n^3)$, where n is the number of training data points. In this paper, we introduce a novel ionic interaction model for data reduction in support vector machines. It is applied to select data points and exclude outliers in the kernel feature space and produce a data reduction algorithm with computational complexity of about $n^3/4$ floating point operations. The instance-based learning algorithm has been successfully applied for data reduction in high dimensional feature spaces obtained by kernel functions. We also present a data reduction method based on the kernelized instance based algorithm. We test the performances of our new methods which illustrate that the computation time can be significantly reduced without any significant decrease in the prediction accuracy.

1 Introduction

When handling a large amount of data in machine learning, it is important to reduce the computation complexity and memory requirement without degrading the prediction accuracy. There have been several approaches proposed to overcome the computational difficulty and memory problems. Recently, for support vector machines (SVMs) [8, 29, 30], incremental learning has been proposed in order to handle huge data [27]. This repeats training a subset of the entire training data set and use the support vectors found in the previous step and chunking [5] and merging these with what is found in iterative training.

Syes, *et al.* [28] selected a small subset of examples from the training set based on several selection strategies including the instance based IB2 selection method [1] and random sampling. They claim that the random sampling selection strategy is the most robust among the strategies they considered in terms of model independence i.e., with random sampling, training was successful with a wide range of different classifiers, such as multi-layer perceptron, nearest neighbor, C4.5, and SVMs. However, random selection method can sometimes cause a catastrophic decrease in accuracy.

Minimal support vector machines (MSVM) [12] remove redundant data and finds a minimal set of support vectors, which is based on linear programming instead of the standard quadratic programming formulation by representing the margin in terms of the 1-norm. The successive linearization algorithm leads to a locally minimal

*This work was supported in part by the National Science Foundation grants CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF). corresponding author: Prof. Haesun Park, E-mail: hpark@cs.umn.edu, Phone : 612-625-0041

Case 1	$\alpha_i = 0$	$y_i f(\mathbf{x}_i) \geq 1$	$\xi_i \geq 0$	$\mathbf{x}_i$ not a support vector and unstable
Case 2	$0 < \alpha_i < C$	$y_i f(\mathbf{x}_i) = 1$	$\xi_i = 0$	$\mathbf{x}_i$ a support vector and stable
Case 3	$\alpha_i = C$	$y_i f(\mathbf{x}_i) \leq 1$	$\xi_i > 0$	$\mathbf{x}_i$ a support vector and very stable

Table 1: KKT conditions and stability of the corresponding data points

number of support vectors. Then, the MSVM generates a linear separating hyperplane by using the minimum number of support vectors. The reduced support vector machines (RSVM), which uses a reduced size kernel, was also introduced [20].

There are also other results based on low-rank approximation of the kernel Gram matrix for reducing computational complexity of kernel-based machine learning methods [31, 11, 26].

In this paper, we propose three new methods for predicting support vectors for reducing computational complexity in SVMs. The first is based on an Ionic Interaction (IoI) model that utilizes concepts from Electrostatics for a binary decision. The second is a kernelized instance based (KIB2) method. The instance-based algorithm (IB2) [1] was applied to select points in the input space [28], and was further developed it into a kernelized-IB2, where IB2 is applied in the high dimensional feature space based on the kernel. The third is a hybrid KIB2-IoI method where IoI is used as a complementary method to select data points in addition to the data points selected by KIB2. The test results show that with our methods, computational complexity in SVMs can be reduced substantially without significant decrease in prediction accuracy.

2 Ionic Interaction (IoI) Model and Data Reduction Methods

The support vectors are the critical points near the boundary between the two classes, which determine an optimal separating hyperplane in SVM methods. Removing any training points that are not support vectors will have no effect on the hyperplane found [6]. If we can predict the support vectors before the training process, the computing time and the memory usage of SVMs will be reduced to an optimal separating hyperplane can be found by training a relatively smaller number of selected points instead of all of the given training points. For this purpose, we propose an ionic interaction (IoI) model for selecting the data points that are the candidates for the support vectors. The points in the (+) class and those in the (-) class are considered as plus ion ($\oplus$, *p-experon*) and minus ion ($\ominus$, *n-experon*), respectively, using a concept from Physics.

These names were chosen since the data points can be considered as experience particles in a high dimensional parameter space of a human decision model based on experiences. The experience particles in the metaphysical space interact with each other like ions in the real physical space. The ions with the same sign repulse each other, and the repulsion makes them unstable. The ions with opposite sign attract each other, and the attraction makes them stable. The training data points which are adjacent to the boundary between the two classes tend to be stabilized by attraction between plus ions ($\oplus$) and minus ions ($\ominus$). The sample points which are far from the boundary are relatively unstable due to repulsion between adjacent ions with the same sign. The number of predicted sample points can be controlled by choosing a threshold of instability. Our data reduction method is derived based on the L_1 norm soft margin SVM which we first review briefly now.

The decision rule for binary classification given by $sign(f(\mathbf{x}))$ with

$$(2.1) \quad f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

is obtained from a 1-norm soft margin support vector machine (SVM). For the training data $(\mathbf{x}_i, y_i)$ with $y_i \in \{-1, +1\}$ for $1 \leq i \leq n$, $f(\mathbf{x})$ is obtained by solving an optimization problem

$$\begin{aligned} \min_{\mathbf{w}, \xi, b} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i \\ s.t. \quad & y_i [\mathbf{w}^T \mathbf{x}_i + b] \geq 1 - \xi_i, \quad \xi_i \geq 0, i = 1, \dots, n \end{aligned}$$

where $\xi = (\xi_1, \xi_2, \dots, \xi_n)^T$ is a vector of slack variables, and C is a parameter that controls the trade-off between a large margin to the separating hyperplane and small training error. Consider a nonlinear feature mapping $\phi(\mathbf{x})$ that maps the input data to a feature space (possibly an infinite dimensional space) where the mapped data may have a linearly separable structure. Without knowing the feature mapping $\phi(\mathbf{x})$ or the feature space explicitly, we can work on the feature space through a kernel function, as long as the problem formulation depends only on the inner products between data points in the feature space and not on the data points themselves. The linear SVMs

Algorithm 1 : IoI (Selective Data Point Support Vector Machines by IoI model)

Given m data points $\mathbf{x}_i$, $1 \leq i \leq m$, a kernel function K with kernel parameters, a desired selection percentage (ρ), and data selection threshold value (Φ_t), this algorithm finds a *total approximate support vectors set* ($ASV = ASV_p \cup ASV_n$, $|ASV| \leq INT(m * \rho/100)$) using the ionic interaction (IoI) model, where ASV_p and ASV_n are the *approximate support vectors sets* of the positive class and negative class, respectively.

1. calculate potential energy for all data points $\mathbf{x}_i$, $1 \leq i \leq m$, by

$$\Phi(i) = \sum_{j \in S_i} \frac{\delta(i, j)}{r_{ij}},$$

where S_i is the set of index j for all opposite sign ions to i th ion, $\delta(i, j)$ is always -1 since i th ion and j th ion have opposite sign here, and $r_{ij}^2 = \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j)$.

2. for each class do

- (a) sort all data points for a class by potential energy in ascending order.
represent the sorted data points as x_k for $1 \leq k \leq l$.
- (b) $ASV_c \leftarrow \{x_1\}$
- (c) for $k = 2$ to l
if $|\Phi(k) - \Phi(k-1)| > \Phi_t$ then $ASV_c \leftarrow ASV_c \cup \{x_k\}$
else x_k is not a support vector candidate
- (d) choose data points which have lower potential energy to select given selection percentage (ρ) of l data points for a class. ($|ASV_c| \leq INT(l * \rho/100)$).

3. $ASV = ASV_p \cup ASV_n$
-

can be extended to nonlinear SVMs by introducing the kernel function. The dual formulation with a kernel function K is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n \end{aligned}$$

where α_i , $1 \leq i \leq n$, denotes the optimal solution for the formulation. Then the decision rule is given by $sign(f(\mathbf{x}))$ with

$$(2.2) \quad f(\mathbf{x}) = \sum_{j=1}^n y_j \alpha_j K(\mathbf{x}, \mathbf{x}_j) + b,$$

where b is chosen so that $y_i f(\mathbf{x}_i) = 1$ for all i with $0 < \alpha_i < C$. According to the Karush-Kuhn-Tucker (KKT) conditions, optimal solutions α and $(\mathbf{w}, b)$ satisfy

$$\begin{aligned} \alpha_i [y_i (K(\mathbf{w}, \mathbf{x}_i) + b) - 1 + \xi_i] &= 0 \quad \text{and} \\ \xi_i (\alpha_i - C) &= 0, \quad i = 1, \dots, n. \end{aligned}$$

These conditions can be rewritten as

$$\text{Case 1: } y_i f(\mathbf{x}_i) \geq 1, \text{ if } \alpha_i = 0$$

$$\text{Case 2: } y_i f(\mathbf{x}_i) = 1, \text{ if } 0 < \alpha_i < C$$

$$\text{Case 3: } y_i f(\mathbf{x}_i) \leq 1, \text{ if } \alpha_i = C$$

The second and third cases occur when $\xi_i = 0$ and $\xi_i > 0$ respectively. The slack variable can have a non-zero value only when $\alpha_i = C$. If $\alpha_i = 0$ (Case 1), then $\mathbf{x}_i$ is not a support vector. If $0 < \alpha_i < C$ (Case 2), then $\mathbf{x}_i$ is the support vector with $\xi_i = 0$. If $\alpha_i = C$ (Case 3), then $\mathbf{x}_i$ is the support vector with $\xi_i > 0$.

Now, we would like to eliminate data points that are not support vectors by dropping unstable ions which have relatively higher potential energy. In Case 2, the ions face the ions with the opposite sign over the separating hyperplane, so they are relatively stable. In Case 3, the ions are soaked into the opposite ionic pool, so they are very stable and will not be eliminated when unstable ions are dropped. In Case 1, the ions are surrounded by many with the same sign, so they are eliminated by the dropping process since they are relatively unstable.

In Table 1, we summarize the information for the three cases based on which ones are used to design our Ionic Interaction model. After choosing training points which are considered as approximate support vectors by the Ionic Interaction model, the separating hyperplane can be obtained by any variation of support vector machines. The approximate support vectors depend on the kernel type and kernel parameters. They are optimized to maximize prediction correctness of the test set for the iterative process.

3 Computation of Potential Energy and Data Reduction Algorithms

The Coulomb potential energy between the i th and j th ions is

$$(3.3) \quad V(i, j) = \frac{q_i q_j}{D \cdot r_{ij}}$$

where q_i and q_j are the charges of the ions, r_{ij} is the Euclidean distance between the i th ion and the j th ion, and D is a dielectric constant. Assuming the homogeneous dielectric medium with $D = 1$, the potential energy between the i th ion and other ions can be defined as

$$(3.4) \quad \Phi_o(i) = \sum_{j \neq i}^n \frac{q_i q_j}{r_{ij}}$$

where n is the total number of ions. The analogous potential energy between the i th data point and the rest which have been mapped to a higher dimensional feature space by a feature mapping $\phi(\mathbf{x})$ is

$$(3.5) \quad \Phi(i) = \sum_{j \in S_i} \frac{\delta(i, j)}{\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|}$$

where S_i is the set of index j of the ions with the opposite sign, and

$$(3.6) \quad \delta(i, j) = \begin{cases} +1, & \text{sign}(q_i) = \text{sign}(q_j) \\ -1, & \text{otherwise} \end{cases}$$

assuming the absolute charge of the ions is 1. In our model, $\delta(i, j)$ is always -1 since S_i in Eqn. (3.5) involves only the ions with the opposite sign of the i th ion. Only the attraction is considered to select the points near the class boundary even though the i th ion may be surrounded by many same sign ions. Considering all other points in calculating the potential energy of the i th ion does not work well for finding the points near the class boundary since an ion can be surrounded by ions of the same sign so that it has high potential energy.

The distance r_{ij} between two data points in the feature space can be calculated by

$$(3.7) \quad \begin{aligned} r_{ij}^2 &= \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 \\ &= K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j), \end{aligned}$$

assuming that the kernel K which gives the inner products between the data points mapped by the feature mapping ϕ

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$$

is given. This process requires about $n^2/2$ floating point operations (flops) [13] to obtain all pairwise distances r_{ij} between any two data points for $i > j$. Note that $r_{ij} = 0$ when $i = j$ and $r_{ij} = r_{ji}$ when $i < j$. When using the radial basis function (RBF) kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

the distance between any two data points in the feature space becomes even more simple to compute as

$$(3.8) \quad r_{ij}^2 = 2 - 2K(\mathbf{x}_i, \mathbf{x}_j)$$

since $K(\mathbf{x}_i, \mathbf{x}_i) = 1$ and $K(\mathbf{x}_j, \mathbf{x}_j) = 1$. We can avoid any redundant calculations by storing the value of each computed value $\delta(i, j)/(\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|)$. After calculating all pairwise distances, the potential energies for all ions can be calculated with computational complexity of about $n^3/4$ flops assuming the number of positive and negative ions are about the same. This is a cheaper process compared to solving a quadratic programming problem for which the complexity is over $O(n^3)$ for data reduction. In addition, various ways to approximate the potential energy can be applied.

In order to determine the local environment of the i th ion, we may consider only opposite sign ions within an adjustable parameter the ‘*cutoff radius*’, R_c , from the i th ion. Significant performance improvement can be achieved by taking advantage of range searching with kd-trees [3, 2] to find all other points within the cut-off radius R_c of the specific point. If the potential energy $\Phi(i)$ is less than the *threshold potential value* Φ_s , then the i th ion is considered an approximate support vector since the i th ion is relatively stable compared to others. The reasonable range of Φ_s values can be found by numerical experiments considering various data sets.

It is possible to approximate the potential energy by counting the number of points with the opposite sign within the cut-off radius. The approximate potential energy for the i th data point is

$$(3.9) \quad \Phi_a(i) = \frac{1}{\eta} \sum_{j \in S_i} \delta(i, j),$$

where S_i is the set of index j such that $0 < r_{ij} \leq R_c$ with the opposite sign of the i th ion, and η is a constant for scaling the approximate potential to the order of the exact potential energy $\Phi(i)$. Since $\delta(i, j)$ is always -1, the value of the summation in Eqn. 3.9 can be easily obtained by counting the number of opposite sign ions within the cutoff radius. This process can be efficiently performed by the range-counting algorithm [14].

The potential energy calculation step can be approximated by considering only nearest-neighbors within the cut-off radius or simplified potential energy. However, we calculated all potential energies considering only opposite sign ions to a selected desired percentage among given training data points in this paper. In this case, we do not have to optimize the cut-off radius (R_c) and threshold potential (Φ_s). The threshold potential can automatically be found when the percentage of selection is given. In Algorithm 1, IoI represents a selection method to choose data points which have lower potential based on the IoI model.

When selecting data according to potential energy, we did not choose a data point for which potential energy is too similar to that of the previous selected data point even though it has low enough potential energy to be chosen. We need only one representative point among many points that face a very similar environment for data reduction. In our experiments, we accept the data point only when the difference of potential energy between the data point and the previous selected data point is greater than the *data selection threshold value* (Φ_t). We found that the data selection threshold value with a range of $0.01 \leq \Phi_t \leq 2.0$ shows good performance for data reduction results through numerical experiments, though it depends on the scattering structure of data points. The IoI algorithm chooses data points that have a more different environment than a higher data selection threshold value. The percentage of chosen data points can be smaller than the percentage of desired selection due to a large Φ_t value. If the difference of the magnitude orders between different features is too large, e.g. Cleveland Heart data set [4], an appropriate Φ_t value can be within the different range. We avoid this problem by normalization of feature values for the Cleveland Heart data set.

IoIB is a balanced selection around the percentage of data points required to balance the number of training points of binary classes. There are two ways to balance the data points for positive and negative classes: removing

Algorithm 2 : KIB2 (Selective Data Point Support Vector Machines by kernelized IB2)

Given m data points $\mathbf{x}_i$, $1 \leq i \leq m$, and a kernel function K with kernel parameters, this algorithm finds an *approximate support vector set* (ASV) from the input data points using the IB2 algorithm [1]. The distances between two data points x and y in the feature space can be computed by $r(x, y)^2 = \|\phi(x) - \phi(y)\|^2 = K(x, x) + K(y, y) - 2K(x, y)$, where K is a kernel function and $\phi(\cdot)$ is its corresponding mapping function that maps input data to a feature space.

ASV $\leftarrow \emptyset$

for each $x \in$ Input Data Set do

1. for each $y \in$ ASV do
 Sim[y] $\leftarrow 2K(x, y) - K(x, x) - K(y, y)$
 2. $y_{max} \leftarrow$ some $y \in$ ASV with maximal Sim[y]
 3. if class(x) = class(y_{max})
 then classification $\leftarrow$ correct
 else
 - (a) classification $\leftarrow$ incorrect
 - (b) ASV $\leftarrow$ ASV $\cup \{x\}$
-

data points from the larger class and adding data points to the smaller class. We chose the latter in order to prevent information loss from the larger class, as well as to add information to the smaller class. We added data points that have lower potential energy which had not already been included in the smaller class. IoIBX is a balanced selection excluding expected outliers. Using the IoI model, we can expect that data points that have potential energy substantially lower than others, i.e. surrounded by many ions with opposite sign, are outliers. This is one of the advantages of the IoI model.

Our second approach is based on the boundary hunting algorithm, IB2 [1]. The approximate support vectors are found by IB2 instead of the ionic interaction (IoI) model in the feature space (See Algorithm 2). Although it has been observed that the selected data points in the input space by IB2 can hardly represent the support vectors that yield high prediction accuracy [28], we found that IB2 performs well in the feature space obtained implicitly by a kernel, where the data points are assumed to be linearly separable. This approach is summarized in Algorithm 2 as kernelized IB2: KIB2.

Figure 1: Classification of the original checker data set by SVM with RBF kernel. Ten-fold cross validated accuracy for the checker data set was 96.1%. The actual decision boundary was drawn in solid line. Support vectors are shown in star points.

Though the performance of KIB2 was good, it could not select additional data points that are able to improve the testing correctness since it only selected a predetermined number of data points. The ionic interaction model gave a rule to append data points to the selected data point by KIB2. The KIB2-IoI method is a combinational approach using the IoI model to append data points after obtaining candidate support vectors by KIB2 in order to generate a more accurate optimal separating hyperplane with a minimal number of selected points among the input data. The performance of all introduced algorithms depends on the input kernel parameter set. We have to optimize the kernel parameters in order to achieve better data reduction performance producing higher prediction accuracy in the classification problem by trying various parameter sets.

When training data is unbalanced, an accuracy maximizing classifier tends to classify data points as a class that has much greater number of points. For dealing with this probable situation, we used different penalty parameters (C_+ and C_-) in the SVM formulation [22]:

$$\max_{\alpha_i} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

Figure 2: Classification of the selected points of checker data set by kernelized IB2 (KIB2). The prediction accuracy of test fold with this decision boundary was 96.0%. The decision boundary was drawn in solid line. Support vectors are shown in star points.

$$\begin{aligned} s.t. \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C_+, \text{ if } y_i = 1 \\ & 0 \leq \alpha_i \leq C_-, \text{ if } y_i = -1. \end{aligned}$$

We used a common heuristic in choosing C_+ and C_- to satisfy the ratio,

$$\frac{C_+}{C_-} = \frac{\# \text{ of data points in the negative class}}{\# \text{ of data points in the positive class}},$$

for the initial guess of the appropriate weighting parameters. We optimized the weighting parameters to obtain a better classifier that has low generalization error by trying several ratios from the initial guess.

4 Visualization of the Input Data

In order to visualize data points in the input space, we need to project high-dimensional data to a 3 or 2-dimensional space. Several widely accepted linear dimensionality reduction methods can be used for this purpose. The principal component analysis (PCA) [16] finds linear projections of great variance, by computing the eigenvectors of the data covariance matrix coresponding to the largest eigenvalues. The multidimensional scaling (MDS)

Figure 3: Data distribution of Pima Indians data set (left) and selected data points (right) by applying IB2 boundary hunting algorithm in the mapped high dimensional feature space by kernel. The 8 features in the input space were reduced to 3 dimensional space by LLE to draw the structure of the data points.

[7] finds projections which minimize the stress of violating metric pairwise distances between examples. The linear discriminant analysis (LDA) [10] finds a projection which minimizes the within-cluster scatter and maximizes between-cluster scatter. These methods perform well for the linearly separable cases and have been generalized to handle nonlinear cases by using Mercer kernels [25, 21, 23]. However, these kernel based dimension reduction methods are suitable for visualization of the the kernel feature space instead of the input space. We used nonlinear dimensionality reduction from locally linear embedding (LLE) to visualize the input space [24, 9]. The LLE discovers a compact representations of high-dimensional data by finding lower dimensional vectors that best match reconstruction weights computed by k -neighbors of each example. In Figure 3, this was used to draw the distribution of the Pima Indian data set and the selected points by KIB2 in three-dimensional space.

5 Results and Discussion

The numerical results in Table 2 were achieved on UC Irvine test problems [4]: Ionosphere, BUPA Liver, Pima Liver, Cleveland Heart, Tic-Tac-Toe, and Mushroom to compare variations of the ionic interaction model-based data selection algorithm in terms of ten-fold cross-validation testing correctness. In cross-validation, we

Figure 4: (a) Ten-fold cross validated accuracy (b) Computing time to obtain ten-fold cross validation accuracy. (c) The number of support vectors. All results were drawn against the percentage of selected data points on the UC Irvine 8124 $\times$ 22 Mushroom data set. The solid, dashed, and dashdot lines were the results of IoI for the selection threshold values Φ_t of 0.1, 0.01, and 0.0, respectively. The circle points were the results of KIB2. The star, square, and x-mark points were the results of KIB2-IoI with 10% additional data point for the selection threshold values Φ_t of 0.1, 0.01, and 0.0, respectively.

broke the training data into 10 subsets. A classifier is trained on the union of 9 of the subsets, and evaluated on the 10th subset. The process is repeated 10 times, using each of the subsets as the validation subset once. The results were combined from the validation subsets to obtain an overall estimate of the effectiveness of the training procedure. This cross validation was also used to choose a parameter setting by making a cross-validated estimate of effectiveness at several values of the parameter.

All our results were run on a 600MHz Pentium II with 500 megabytes of memory. We used the Gaussian RBF kernel in all our experiments using SVMlight [15]. In Table 2, only 50% of data points were selected for each algorithm in order to compare them, though more accurate results can be obtained by selecting more data points. The IoIB algorithm generally showed good performance

Data set points $\times$ features	Full data time(sec), (SV [†])	IoI time(sec)	IoIB time(sec)	IoIBX time(sec)
Cleveland Heart	85.18%	83.18%	83.51%	83.16%
297 $\times$ 13	1.69, (115.6)	0.80	0.85	0.92
BUPA Liver	74.79%	73.62%	70.66%	71.31%
345 $\times$ 6	6.80, (201.6)	0.99	1.97	1.01
Ionosphere	94.58%	94.31%	94.88%	94.28%
351 $\times$ 34	3.16, (69.3)	1.52	1.82	1.35
Pima Indians	76.95%	72.79%	74.99%	74.60%
768 $\times$ 8	9.86, (395.7)	6.63	4.62	4.54
Tic-Tac-Toe	98.23%	93.84%	96.56%	96.35%
958 $\times$ 9	50.50, (425.1)	18.67	24.24	24.14
Mushroom	100.0%	99.51%	99.51%	99.43%
8124 $\times$ 22	106.15, (628.1)	6.85	6.79	6.81

Table 2: Ten-fold cross-validation testing correctness (%) on UC Irvine test problems: Ionosphere, BUPA Liver, Pima Liver, Cleveland Heart, Tic-Tac-Toe, and Mushroom. Execution times is computing time to obtain ten-fold cross validation accuracy. SV[†] represents the average number of support vectors for ten-folds. IoI: results with the desired data selection percentage 50% of the data points that have lower potential by IoI model. IoIB: results with balanced selection around 50% of data points. IoIBX: results with balanced selection excluding expected outliers.

compared to IoI. This means that balancing the data points in the selected data points is also helpful to obtain a more generalized optimal separating hyperplane. The number of points that will be excluded for IoIBX can be fixed or optimized for specific supervised machine learning problems. In this experiment, we excluded 2 points that have the lowest potential energy. However, we could not obtain good results by IoIBX. We think that this is due to the fact that the data sets that we used do not contain serious outliers.

Table 3 shows the ten-fold cross validation accuracy and computing time to train and test for ten folds to compare the optimal ionic interaction (IoI) model-based algorithm, the kernelized instance based (KIB2) algorithm, and the hybrid (KIB2-IoI) algorithm. The number of 50% data points are approximately similar to the number of support vectors that was found after training with the full number of data points, except the Ionosphere and Mushroom data sets. For the Mushroom data set, the percentage of the points predicted as support vectors is much less than 50% though the desired selection percentage was 50% in the IoI algorithm. This was possible through setting the data selection threshold parameter (Φ_t) to 0.1. This means that the data set contains many data points that have similar information. By selecting representatives from each group that have a similar ionic environment, we achieved data reduction using the IoI model. Figure 4 shows that an appropriate data selection threshold value enables the IoI

algorithm to find a smaller number of data points without any significant decrease in the prediction accuracy. When the parameter is 0.0, the jumping selection function based on the potential energy difference between current data point and previous chosen data point is off. A too small Φ_t value generates a *poor representation error* (i.e. the chosen data points may contain redundant data points and miss meaningful data points in spite of a higher potential energy), while a too large Φ_t value has the IoI data selector choosing too few of the data points.

The ionic interaction model-based methods tend to need more data points than KIB2 to obtain a similar level of accuracy. This tendency can be seen in Figure 4 again. Therefore, the computing time of KIB2 is generally much smaller than that of IoI. KIB2 only needs about 0.6% of the data points to produce 96.61% testing correctness for the Mushroom data set. However, IoI is a more general tool if we want to obtain better prediction accuracy than KIB2 since it can be used for balancing selected data points and exclusion of outliers. For IoI, the accuracy slowly decreased until 50% selection, while the computing time dramatically decreased and the number of support vectors decreased only a small amount as shown in Figure 4. The number of support vectors of KIB2 is much smaller than that of IoI. Generally speaking, we can say that KIB2 can select more compact data points that can be meaningful support vectors. However, we observed that the IB2-based data selector performed poorly due to

Data set points $\times$ features	Full data set time(sec), (SV [†])	IoI time(sec), (ASV [‡])	KIB2 time(sec), (ASV [‡])	KIB2-IoI time(sec), (ASV [‡])
Cleveland Heart	85.18%	83.18%	82.83%	80.48%
297 $\times$ 13	1.69, (115.6)	0.80, (128.5)	0.97, (82.7)	1.23, (108.7)
BUPA Liver	74.79%	73.62%	71.28%	72.72%
345 $\times$ 6	6.80, (201.6)	0.99, (159.5)	1.45, (139.7)	1.83, (170.7)
Ionosphere	94.58%	94.31%	92.26%	94.59%
351 $\times$ 34	3.16, (69.3)	1.52, (161.8)	0.72, (55.7)	0.97, (88.4)
Pima Indians	76.95%	72.79%	73.95%	75.38%
768 $\times$ 8	9.85, (395.7)	6.63, (300.1)	2.63, (235.1)	4.67, (329.1)
Tic-Tac-Toe	98.23%	93.84%	77.86%	91.54%
958 $\times$ 9	50.50, (425.1)	18.67, (435.8)	1.71, (180.2)	14.28, (296.2)
Mushroom	100.0%	99.51%	96.61%	99.53%
8124 $\times$ 22	106.15, (628.1)	6.85, (797.4)	1.96, (52.9)	6.34, (730.0)

Table 3: Ten-fold cross-validation testing correctness (%) on UC Irvine test problems: Ionosphere, BUPA Liver, Pima Liver, Cleveland Heart, Tic-Tac-Toe, and Mushroom. The execution times are the computed time to obtain ten-fold cross validation accuracy. SV[†] represents the average number of support vectors for ten-folds. ASV[‡] represents the average number of data points predicted as support vectors. IoI: results with the desired selection percentage 50% of the data points which have lower potential energy by IoI model. KIB2: results with the selected data points by KIB2. KIB2-IoI: results with the data points of selected data points by KIB2 and 10% additional data points by IoI model.

the characteristics of the IB2 boundary hunting algorithm when the data points are too unbalanced. The Tic-Tac-Toe UC Irvine test problem has the size of one class only about 53% of the size of the other class. For this problem, the balanced data selectors such as IoIB and IoIBX showed better performance than that of KIB2.

We also tested the checker data set [20] to see how well KIB2 works on a data set containing 2 dimensional features, which easily allows drawing of the data points and decision boundary. In Figure 1, the actual decision boundary was drawn in the input space. The ten-fold cross validated accuracy for the checker data set was 96.1%. The support vectors are shown in star points. After applying KIB2 in the mapped kernel feature space, we could obtain a relatively small number of data points. The decision boundary was drawn in the input space by these selected data points. The testing accuracy was 96.0%. The results of KIB2-IoI exhibited good performance in terms of the testing correctness as well as a short learning and testing times (See Table 3 and Figure 4).

When we treat a huge number of data points in kernel-based supervised machine learning, it is very time consuming to search for the optimal kernel parameters. The newly introduced KIB2 data selector is a novel efficient method to find the optimal kernel parameters, or at least rough ranges of the parameters. After obtaining cross-validation rates for different parameter sets which

are input values of KIB2 algorithm for reduction of training data points, we can find a parameter set to maximize the cross-validation rate. This is one of the major achievements in this paper.

The Ionic Interaction model (IoI)-based data selector is a new method to exclude outliers. In other methods, kernel-based machine learning algorithms can be prevented from finding a good generalized separating boundary. The hybrid KIB2-IoI data selector is a more reliable data reduction method compared to the KIB2-based approach, especially for unbalanced data. However, we would like to point out that KIB2 is still a comparable data selector when the data are well balanced.

In this paper, we applied the ionic interaction concept of Physics to the data reduction problem by considering data points of positive and negative classes as *p-experons* and *n-experons*, respectively, i.e. experience particles in the metaphysical space of a human decision model based on experiences. We would suppose that the principle of the negative and the positive, i.e. the cosmic dual forces, in the material world may be present in the metaphysical world as well. While some classification problems, for example, text categorization [17], suffers from a large number of features, many Bioinformatics problems, for example, protein secondary structure prediction [19] and solvent accessibility prediction [18], suffer from a large number of data points. They also suffer from a large

number of outliers. The IoI or KIB2-IoI data selector can be applied to those problems to generate a more accurate optimal separating hyperplane that has lower generalization error and overfitting.

Acknowledgements

We would like to thank Dr. D. W. Aha for his kind help and Dr. Dennis DeCoste for valuable comments on kernelized locally-linear embeddings.

References

- [1] D. W. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
- [2] J. L. Bentley. Multidimensional divide-and-conquer. *Communications of the ACM*, 23(4):214–229, 1980.
- [3] J. L. Bentley and J. H. Friedman. Data structures for range searching. *Computing Surveys*, 11(4):398–409, 1979.
- [4] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.
- [5] P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 13:1–10, 2000.
- [6] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.
- [7] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.
- [8] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods*. University Press, Cambridge, 2000.
- [9] D. DeCoste. Visualizing Mercer kernel feature spaces via kernelized locally-linear embeddings. In *The 8th International Conference on Neural Information Processing (ICONIP2001)*, Nov. 2001.
- [10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-interscience, New York, 2001.
- [11] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representation, 2000. Research Report RC 21911, IBM T.J. Watson Research Center.
- [12] G. Fung and O. L. Mangasarian. Data selection for support vector machine classification. In *Proceedings KDD2000: Knowledge Discovery and Data Mining*, pages 64–70, Boston, MA, 2000.
- [13] G. H. Golub and C. F. Van Loan. *Matrix Computations, third edition*. Johns Hopkins University Press, Baltimore, 1996.
- [14] A. G. Gray and A. W. Moore. ‘N-body’ problems in statistical learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 521–527, 2000.
- [15] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 41–56, Cambridge, 1999. MIT-Press.
- [16] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
- [17] H. Kim, P. Howland, and H. Park. Text classification using support vector machines with dimension reduction. In M. W. Berry and W. M. Pottenger, editors, *Proceedings of the Workshop on Text Mining SIAM Third International Conference on Data Mining (SDM03)*, pages 48–59, 2003.
- [18] H. Kim and H. Park. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor, 2003. *Proteins: Struct. Funct. Genet.*, to appear.
- [19] H. Kim and H. Park. Protein secondary structure prediction based on an improved support vector machines approach, 2003. *Protein Eng.*, to appear.
- [20] Y. J. Lee and O. L. Mangasarian. RSVM: Reduced support vector machines. In *Proceedings of the First SIAM International Conference on Data Mining*, Chicago, 2001. SIAM.
- [21] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. In Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.
- [22] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report AI Memo 1602, MIT A.I. Lab, 1997.
- [23] C. Park and H. Park. Kernel discriminant analysis based on the generalized singular value decomposition. Technical Report 03-017, Department of Computer Science and Engineering, University of Minnesota, 2003.
- [24] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- [25] B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
- [26] A. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proc. 17th International Conf. on Machine Learning*, pages 911–918. Morgan Kaufmann, San Francisco, CA, 2000.
- [27] N. A. Syed, H. Liu, and K. K. Sung. Incremental learning with support vector machines. In *Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, 1999.
- [28] N. A. Syed, H. Liu, and K. K. Sung. A study of support vectors on model independent example selection. In *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining*, pages 272–276, 1999.
- [29] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.
- [30] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

- [31] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Neural Information Processing Systems 13*. MIT Press, 2001.