BMC Bioinformatics

BioMed Central

Proceedings

Support Vector Machine Implementations for Classification & Clustering

Stephen Winters-Hilt*[1,2], Anil Yelundur[1], Charlie McChesney[1] and Matthew Landry[1]

Address: [1]Department of Computer Science, University of New Orleans, New Orleans, LA, 70148, USA and [2]The Research Institute for Children, 200 Henry Clay Ave., New Orleans, LA 70118, USA

Email: Stephen Winters-Hilt* - swinters@chnola-research.org; Anil Yelundur - ayelundu@cs.uno.edu; Charlie McChesney - cmcchesn@cs.uno.edu; Matthew Landry - mlandry@cs.uno.edu

* Corresponding author

from The Third Annual Conference of the MidSouth Computational Biology and Bioinformatics Society
Baton Rouge, Louisiana. 2–4 March, 2006

Published: 26 September 2006

BMC Bioinformatics 2006, **7**(Suppl 2):S4 doi:10.1186/1471-2105-7-S2-S4

Abstract

Background: We describe Support Vector Machine (SVM) applications to classification and clustering of channel current data. SVMs are variational-calculus based methods that are constrained to have structural risk minimization (SRM), i.e., they provide noise tolerant solutions for pattern recognition. The SVM approach encapsulates a significant amount of model-fitting information in the choice of its kernel. In work thus far, novel, information-theoretic, kernels have been successfully employed for notably better performance over standard kernels. Currently there are two approaches for implementing multiclass SVMs. One is called external multi-class that arranges several binary classifiers as a decision tree such that they perform a single-class decision making function, with each leaf corresponding to a unique class. The second approach, namely internal-multiclass, involves solving a single optimization problem corresponding to the entire data set (with multiple hyperplanes).

Results: Each SVM approach encapsulates a significant amount of model-fitting information in its choice of kernel. In work thus far, novel, information-theoretic, kernels were successfully employed for notably better performance over standard kernels. Two SVM approaches to multiclass discrimination are described: (1) internal multiclass (with a single optimization), and (2) external multiclass (using an optimized decision tree). We describe benefits of the internal-SVM approach, along with further refinements to the internal-multiclass SVM algorithms that offer significant improvement in training time without sacrificing accuracy. In situations where the data isn't clearly separable, making for poor discrimination, signal clustering is used to provide robust and useful information – to this end, novel, SVM-based clustering methods are also described. As with the classification, there are Internal and External SVM Clustering algorithms, both of which are briefly described.

Background

Support Vector Machine

SVMs are fast, easily trained, discriminators [1,2], for which strong discrimination is possible without the overfitting complications common to neural net discriminators [1]. SVMs strongly draw upon variational methods in their construction and are designed to yield the best estimate of the optimal separating hyperplane (for classifier, see Fig. 1) with confidence parameter information included (via hyperplane with margin optimization used in structural risk minimization). The SVM approach also encapsulates a significant amount of model fitting and discriminatory information in the choice of kernel in the SVM, and a number of novel kernels have been developed. In [3], novel, information-theoretic, kernels were introduced for notably better performance over standard kernels (with discrete probability distributions as part of feature vector data). The classification approach adopted in [3] is designed to scale well to multi-species classification (or a few species in a very noisy environment). The scaling is possible due to use of a decision tree architecture and an SVM approach that permits rejection on weak data. SVMs are usually implemented as binary classifiers, are in many ways superior to neural nets, and may be grouped in a decision tree to arrive at a multi-class discriminator. SVMs are much less susceptible to over-training than neural nets, allowing for a much more hands-off training process that is easily deployable and scalable. A multiclass implementation for an SVM is also possible – where multiple hyperplanes are optimized simultaneously. A (single-optimization, multi-hyperplane) multiclass SVM has a much more complicated implementation, but the reward is a classifier that is much easier to tune and train, especially when considering data rejection. The (single) multiclass SVM, doesn't have as non-scalable a throughput problem (with tree depth), and even appears to offer a natural drop zone via its margin definition, so is being considered in further refinements of the method.

SVMs use variational methods in their construction and encapsulate a significant amount of discriminatory information in their choice of kernel. In reference [3] information-theoretic kernels provided notably better performance than standard kernels. Feature extraction was designed to arrive at probability vectors (i.e., discrete probability distributions) on a predefined, and complete, space of possibilities. (The different blockade levels, and their frequencies, the emission probabilities, and the transition probabilities, for example.) This turns out to be a very general formulation, wherein feature extraction makes use of signal decomposition into a complete set of separable states that can be interpreted or represented as a probability vector. A probability vector formulation also provides a straightforward hand-off to the SVM classifiers since all feature vectors have the same length with such an approach. What this means for the SVM, however, is that geometric notions of distance are no longer the best measure for comparing feature vectors. For probability vectors (i.e., discrete distributions), the best measures of similarity are the various information-theoretic divergences: Kullback-Leibler, Renyi, etc. By symmetrizing over the arguments of those divergences a rich source of kernels is

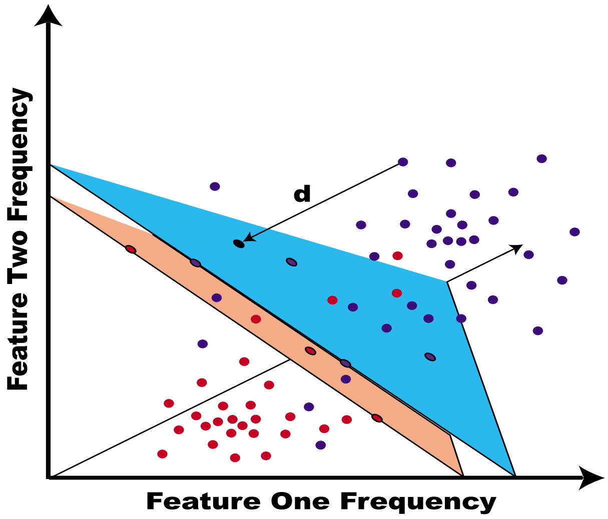


Figure 1

A sketch of the hyperplane separability heuristic for SVM binary classification. An SVM is trained to find an optimal hyperplane that separates positive and negative instances, while also constrained by structural risk minimization (SRM) criteria, which here manifests as the hyperplane having a thickness, or "margin," that is made as large as possible in seeking a separating hyperplane. A benefit of using SRM is much less complication due to overfitting (a common problem with Neural Network discrimination approaches). Given its geometric expression, it is not surprising that a key construct in the SVM formulation (via the choice of kernel) is the notion of "nearness" between instances (or nearness to the hyperplane, where it gives a measure of confidence in the classification, i.e., instances further from the decision hyperplane are called with greater confidence). Most notions of nearness explored in this context have stayed with the geometric paradigm and are known as "distance kernels," one example being the familiar Gaussian kernel which is based on the Euclidean distance: $K_{Gaussian}(x,y) = \exp(-D_{Eucl.}(x,y)^2/2\sigma^2)$, where $D_{Eucl.}(x,y) = [\sum_k(x_k\text{-}y_k)2]^{1/2}$ is the usual Euclidean distance. Those kernels are used in the signal pattern recognition analysis in Figure 8 along with a new class of kernels, "divergence kernels," based on a notion of nearness appropriate when comparing probability distributions (or probability feature vectors). The main example of this is the Entropic Divergence Kernel: $K_{Entropic} = \exp(-D_{Entropic.}(x,y)^2/2\sigma^2)$, where $D_{Entropic.}(x,y) = D(x||y) + D(y||x)$ and $D(..||..)$ is the Kullback-Leibler Divergence (or relative entropy) between x and y.

obtained that works well with the types of probabilistic data obtained.

The SVM discriminators are trained by solving their KKT relations using the Sequential Minimal Optimization (SMO) procedure [4]. A chunking [5,6] variant of SMO also is employed to manage the large training task at each SVM node. The multi-class SVM training generally involves thousands of blockade signatures for each signal class. The data cleaning needed on the training data is accomplished by an extra SVM training round.

Binary Support Vector Machines

Binary Support Vector Machines (SVMs) are based on a decision-hyperplane heuristic that incorporates structural risk management by attempting to impose a training-instance void, or "margin," around the decision hyperplane [1].

Feature vectors are denoted by x_{ik}, where index i labels the M feature vectors ($1 \le i \le M$) and index k labels the N feature vector components ($1 \le i \le N$). For the binary SVM, labeling of training data is done using label variable $y_i = \pm 1$ (with sign according to whether the training instance was from the positive or negative class). For hyperplane separability, elements of the training set must satisfy the following conditions: $w_\beta x_{i\beta}\text{-}b \ge +1$ for i such that $y_i = +1$, and $w_\beta x_{i\beta}\text{-}b \le -1$ for $y_i = -1$, for some values of the coefficients $w_1, ..., w_N$, and b (using the convention of implied sum on repeated Greek indices). This can be written more concisely as: $y_i(w_\beta x_{i\beta}\text{-}b) - 1 \ge 0$. Data points that satisfy the equality in the above are known as "support vectors" (or "active constraints").

Once training is complete, discrimination is based solely on position relative to the discriminating hyperplane: $w_\beta x_{i\beta} - b = 0$. The boundary hyperplanes on the two classes of data are separated by a distance 2/w, known as the "margin," where $w^2 = w_\beta w_\beta$. By increasing the margin between the separated data as much as possible the optimal separating hyperplane is obtained. In the usual SVM formulation, the goal to maximize w^{-1} is restated as the goal to minimize w^2. The Lagrangian variational formulation then selects an optimum defined at a saddle point of $L(w,b;\alpha) = (w_\beta w_\beta)/2 - \alpha_\gamma y_\gamma(w_\beta x_{\gamma\beta}\text{-}b) - \alpha_0$, where $\alpha_0 = \Sigma_\gamma \alpha_\gamma$, $\alpha_\gamma \ge 0$ ($1 \le \gamma \le M$). The saddle point is obtained by minimizing with respect to $\{w_1, ..., w_N, b\}$ and maximizing with respect to $\{\alpha_1, ..., \alpha_M\}$. If $y_i(w_\beta x_{i\beta}\text{-}b) - 1 \ge 0$, then maximization on α_i is achieved for $\alpha_i = 0$. If $y_i(w_\beta x_{i\beta}\text{-}b) - 1 = 0$, then there is no constraint on α_i. If $y_i(w_\beta x_{i\beta}\text{-}b) - 1 < 0$, there is a constraint violation, and $\alpha_i \to \infty$. If absolute separability is possible the last case will eventually be eliminated for all α_i, otherwise it's natural to limit the size of α_i by some constant upper bound, i.e., $\max(\alpha_i) = C$, for all i. This is equivalent to another set of inequality constraints with $\alpha_i \le C$. Introducing sets of Lagrange multipliers, ξ_γ and μ_γ($1 \le \gamma \le M$), to achieve this, the Lagrangian becomes:

$L(w,b;\alpha,\xi,\mu) = (w_\beta w_\beta)/2 - \alpha_\gamma[y_\gamma(w_\beta x_{\gamma\beta}\text{-}b)+\xi_\gamma] + \alpha_0 + \xi_0 C - \mu_\gamma \xi_\gamma$, where $\xi_0 = \Sigma_\gamma \xi_\gamma$, $\alpha_0 = \Sigma_\gamma \alpha_\gamma$, and $\alpha_\gamma \ge 0$ and $\xi_\xi \ge 0$ ($1 \le \gamma \le M$).

At the variational minimum on the $\{w_1, ..., w_N, b\}$ variables, $w_\beta = \alpha_\gamma y_\gamma x_{\gamma\beta}$, and the Lagrangian simplifies to: $L(\alpha) = \alpha_0 - (\alpha_\delta y_\delta x_{\delta\beta}\, \alpha_\gamma y_\gamma x_{\gamma\beta}/2$, with $0 \le \alpha_\gamma \le C$ ($1 \le \gamma \le M$) and $\alpha_\gamma y_\gamma = 0$, where only the variations that maximize in terms of the α_γ remain (known as the Wolfe Transformation). In this form the computational task can be greatly simplified. By introducing an expression for the discriminating hyperplane: $f_i = w_\beta x_{i\beta} - b = \alpha_\gamma y_\gamma x_{\gamma\beta} x_{i\beta} - b$, the variational solution for $L(\alpha)$ reduces to the following set of relations (known as the Karush-Kuhn-Tucker, or KKT, relations): (i) $\alpha_i = 0 \Leftrightarrow y_i f_i \ge 1$, (ii) $0 < \alpha_i < C \Leftrightarrow y_i f_i = 1$, and (iii) $\alpha_i = C \Leftrightarrow y_i f_i \le 1$. When the KKT relations are satisfied for all of the α_γ (with $\alpha_\gamma y_\gamma = 0$ maintained) the solution is achieved. (The constraint $\alpha_\gamma y_\gamma = 0$ is satisfied for the initial choice of multipliers by setting the α's associated with the positive training instances to $1/N^{(+)}$ and the α's associated with the negatives to $1/N^{(-)}$, where $N^{(+)}$ is the number of positives and $N^{(-)}$ is the number of negatives.) Once the Wolfe transformation is performed it is apparent that the training data (support vectors in particular, KKT class (ii) above) enter into the Lagrangian solely via the inner product $x_{i\beta} x_{j\beta}$. Likewise, the discriminator f_i, and KKT relations, are also dependent on the data solely via the $x_{i\beta} x_{j\beta}$ inner product.

Generalization of the SVM formulation to data-dependent inner products other than $x_{i\beta} x_{j\beta}$ are possible and are usually formulated in terms of the family of symmetric positive definite functions (reproducing kernels) satisfying Mercer's conditions [1].

Binary SVM Discriminator Implementation

The SVM discriminators are trained by solving their KKT relations using the Sequential Minimal Optimization (SMO) procedure of [4]. The method described here follows the description of [4] and begins by selecting a pair of Lagrange multipliers, $\{\alpha_1, \alpha_2\}$, where at least one of the multipliers has a violation of its associated KKT relations (for simplicity it is assumed in what follows that the multipliers selected are those associated with the first and second feature vectors: $\{x_1, x_2\}$). The SMO procedure then "freezes" variations in all but the two selected Lagrange multipliers, permitting much of the computation to be circumvented by use of analytical reductions:

$L(\alpha_1,\alpha_2;\alpha_{\beta'\ge 3}) = \alpha_1 + \alpha_2 - (\alpha_1{}^2 K_{11} + \alpha_2{}^2 K_{22} + 2\alpha_1\alpha_2 y_1 y_2 K_{12})/2 - \alpha_1 y_1 v_1 - \alpha_2 y_2 v_2 + \alpha_{\beta'} U_{\beta'} - (\alpha_{\beta'}\alpha_{\gamma'} y_{\beta'} K_{\beta'\gamma'})/2$, with $\beta',\gamma' \geq 3$, and where $K_{ij} \equiv K(x_i, x_j)$, and $v_i \equiv \alpha_{\beta'} y_{\beta'} K_{i\beta'}$ with $\beta' \geq 3$. Due to the constraint $\alpha_\beta y_\beta = 0$, we have the relation: $\alpha_1 + s\alpha_2 = -\gamma$, where $\gamma \equiv y_1 \alpha_{\beta'} y_{\beta'}$ with $\beta' \geq 3$ and $s \equiv y_1 y_2$. Substituting the constraint to eliminate references to α_1, and performing the variation on α_2: $\partial L(\alpha_2;\alpha_{\beta' \geq 3})/\partial\alpha_2 = (1 - s) + \eta\alpha_2 + s\gamma(K_{11} - K_{22}) + sy_1v_1 - y_2v_2$, where $\eta \equiv (2K_{12} - K_{11} + K_{22})$. Since v_i can be rewritten as $v_i = w_\beta x_{i\beta} - \alpha_1 y_1 K_{i1} - \alpha_2 y_2 K_{i2}$, the variational maximum $\partial L(\alpha_2;\alpha_{\beta' \geq 3})/\partial\alpha_2 = 0$ leads to the following update rule:

$$\alpha_2^{\text{new}} = \alpha_2^{\text{old}} - y_2((w_\beta x_{1\beta} - y_1) - (w_\beta x_{2\beta} - y_2))/\eta.$$

Once α_2^{new} is obtained, the constraint $\alpha_2^{\text{new}} \leq C$ must be re-verified in conjunction with the $\alpha_\beta y_\beta = 0$ constraint. If the $L(\alpha_2;\alpha_{\beta' \geq 3})$ maximization leads to a α_2^{new} that grows too large, the new α_2 must be "clipped" to the maximum value satisfying the constraints. For example, if $y_1 \neq y_2$, then increases in α_2 are matched by increases in α_1. So, depending on whether α_2 or α_1 is nearer its maximum of C, we have $\max(\alpha_2) = \text{argmin}\{\alpha_2 + (C - \alpha_2); \alpha_2 + (C - \alpha_1)\}$. Similar arguments provide the following boundary conditions: (i) if $s = -1$, $\max(\alpha_2) = \text{argmin}\{\alpha_2; C + \alpha_2 - \alpha_1\}$, and $\min(\alpha_2) = \text{argmax}\{0; \alpha_2 - \alpha_1\}$, and (ii) if $s = +1$, $\max(\alpha_2) = \text{argmin}\{C; \alpha_2 + \alpha_1\}$, and $\min(\alpha_2) = \text{argmax}\{0; \alpha_2 + \alpha_1 - C\}$. In terms of the new $\alpha_2^{\text{new, clipped}}$, clipped as indicated above if necessary, the new α_1 becomes:

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + s(\alpha_2^{\text{old}} - \alpha_2^{\text{new, clipped}}),$$

where $s \equiv y_1 y_2$ as before. After the new α_1 and α_2 values are obtained there still remains the task of obtaining the new b value. If the new α_1 is not "clipped" then the update must satisfy the non-boundary KKT relation: $y_1 f(x_1) = 1$, i.e., $f^{\text{new}}(x_1) - y_1 = 0$. By relating f^{new} to f^{old} the following update on b is obtained:

$$b^{\text{new1}} = b - (f^{\text{new}}(x_1) - y_1) - y_1(\alpha_1^{\text{new} - \alpha}{}_1^{\text{old}})K_{11} - y_2(\alpha_2^{\text{new, clipped}} - \alpha_2^{\text{old}})K_{12}.$$

If α_1 is clipped but α_2 is not, the above argument holds for the α_2 multiplier and the new b is:

$$b^{\text{new2}} = b - (f^{\text{new}}(x_2) - y_2) - y_2(\alpha_2^{\text{new} - \alpha}{}_2^{\text{old}})K_{22} - y_1(\alpha_1^{\text{new, clipped}} - \alpha_1^{\text{old}})K_{12}.$$

If both α_1 and α_2 values are clipped then any of the b values between b^{new1} and b^{new2} is acceptable, and following the SMO convention, the new b is chosen to be:

$$b^{\text{new}} = (b^{\text{new1}} + b^{\text{new2}})/2.$$

Multiclass SVM Methods

The SVM binary discriminator offers high performance and is very robust in the presence of noise. This allows a variety of reductionist multiclass approaches, where each reduction is a binary classification (for classifying cards by suit, maybe classify as red or black first, then as heart or diamond for red and spade or club for black, for example). The SVM Decision Tree is one such approach, and a collection of them (a SVM Decision Forest) can be used to avoid problems with throughput biasing. Alternatively, the variational formalism can be modified to perform a multi-hyperplane optimization situation for a direct multiclass solution [7-9], and that is what is described next.

SVM-Internal Multiclass

In the formulation in [7], there are 'k' classes and hence 'k' linear decision functions – a description of their approach is given here. For a given input 'x', the output vector corresponds to the output from each of these decision functions. The class of the largest element of the output vector gives the class of 'x'.

Each decision function is given by: $f_m(x) = w_m.x + b_m$ for all $m = (1,2, ..., k)$. If y_i is the class of the input x_i, then for each input data point, the misclassification error is defined as follows: $\max_m\{f_m(x_i) + 1 - \delta_i^m\} - f_{yi}(x_i)$, where δ_i^m is 1 if $m = y_i$ and 0 if $m \neq y_i$. We add the slack variable ζ_i where $\zeta_i \geq 0$ for all i that is proportional to the misclassification error: $\max_m\{f_m(x_i) + 1 - \delta_i^m\} - f_{yi}(x_i) = \zeta_i$, hence $f_{yi}(x_i) - f_m(x_i) + \delta_i^m \geq 1 - \zeta_i$ for all i, m. To minimize this classification error and maximize the distance between the hyper-planes (Structural Risk Minimization) we have the following formulation:

Minimize: $\sum_i \zeta_i + \beta(1/2)\sum_m w_m^T w_m + (1/2)\sum_m b_m^2$,

where $\beta > 0$ is defined as a regularization constant.

Constraint: $w_{yi}.x_i + b_{yi} - w_m.x_i - b_m - 1 + \zeta_i + \delta_i^m \geq 0$ for all i,m

Note: the term $\mathbf{(1/2)\sum_m b_m^2}$ is added for de-coupling, $1/\beta = C$, and $m = y_i$ in the above constraint is consistent with $\zeta_i \geq 0$. The Lagrangian is:

$$L(w,b,\zeta) = \sum_i \zeta_i + \beta(1/2)\sum_m w_m^T w_m + (1/2)\sum_m b_m^2 - \sum_i \sum_m \alpha_i^m (w_{yi} x_i + b_{yi} - w_m.x_i - b_m - 1 + \zeta_i + \delta_i^m)$$

Where all α_i^ms are positive Lagrange multipliers. Now taking partial derivatives of the Lagrangian and equating them to zero (Saddle Point solution): $\partial L/\partial\zeta_i = 1 - \sum_m \alpha_i^m = 0$. This implies that $\sum_m \alpha_i^m = 1$ for all i. $\partial L/\partial b_m = b_m + \sum_i \alpha_i^m - \sum_i \delta_i^m = 0$ for all m. Hence $b_m = \sum_i(\delta_i^m - \alpha_i^m)$. Similarly: $\partial L/\partial w_m = \beta w_m + \sum_i \alpha_i^m x_i - \sum_i \delta_i^m x_i = 0$ for all m. Hence $w_m = (1/\beta)[\sum_i(\delta_i^m - \alpha_i^m)x_i]$ Substituting the above equations into the Lagrangian and after simplification reduces into the dual formalism:

Maximize: $-1/2\Sigma_{i,j}\Sigma_m(\delta_i^m - \alpha_i^m)(\delta_j^m - \alpha_j^m)(K_{ij} + \beta) - \beta\Sigma_{i,m}\delta_i^m\alpha_i^m$

Constraint: $0 \le \alpha_i^m$, $\Sigma_m\alpha_i^m = 1$, i = 1...1; m = 1...k

Where $K_{ij} = x_i.x_j$ is the Kernel generalization. In vector notation:

Maximize: $-1/2\Sigma_{i,j}(\Delta_{yi} - A_i)(\Delta_{yj} - A_j)(K_{ij} + \beta) - \beta\Sigma_i\Delta_{yi}A_i$

Constraint: $0 \le A_i$, A_i. 1 = 1, i = 1 ...1

Let $\tau_i = \Delta_{yi} - A_i$. Hence after ignoring the constant: $-1/2\Sigma_{i,j}\tau_i.\tau_j(K_{ij} + \beta) + \beta\Sigma_i\Delta;_{yi}\tau_i$, subject to: $\tau_i \le \Delta_{yi}$, $\tau_i.1 = 0$, i = 1 ...l. The dual is solved (determine the optimum values of all the τs) using the decomposition method.

Minimize: $1/2\Sigma_{i,j}\tau_i^m.\tau_j^m(K_{ij} + \beta) - \beta\Sigma_{i,m}\delta_i^m\tau_i^m$

Constraint: $\tau_i \le \Delta_{yi}$, $\tau_i.1 = 0$, i = 1 ...l

The Lagrangian of the dual is:

$L = 1/2\Sigma_{i,j,m}\tau_i^m.\tau_j^m(K_{ij} + \beta) - \phi\Sigma_{i,m}\delta_i^m\tau_i^m - \Sigma_{i,m}u_i^m(\delta_i^m - \tau_i^m) - \Sigma_i v_i\Sigma_m\tau_i^m$

Subject to $u_i^m \ge 0$

We take the gradient of the Lagrangian with respect to τ_i^m:

$\blacktriangledown_{\tau}{}^m[L] = \Sigma_i\tau_j^m(K_{ij} + \beta) - \beta\delta_i^m + u_i^m - v_i = 0$

Introducing $f(\tau) = \Sigma_i\tau_j^m(K_{ij} + \beta) - \beta\delta_i^m + u_i^m - v_i = 0$ and $f_i^m = \Sigma_i\tau_j^m(K_{ij} + \beta) - \beta\delta_i^m$, then $f(\tau) = f_i^m + u_i^m - v_i = 0$. By KKT conditions we get two more equations:

$u_i^m(\delta_i^m - \tau_i^m) = 0$ and $u_i^m \ge 0$

Case I: if $\delta_i^m = \tau_i^m$, then $u_i^m \ge 0$, hence $f_i^m \le v_i$. Case II: if $\tau_i^m < \delta_i^m$, then $u_i^m = 0$, hence $f_i^m = v_i$. Note: There is atleast one 'm' for all i such that $\tau_i^m < \delta_i^m$ is satisfied.

Therefore combining Case I & II, we get:

$\max_m\{f_i^m\} \le v_i \le \min_{m:\,\tau i^m < \delta_i^m}\{f_i^m\}$

Or $\max_m\{f_i^m\} \le \min_{m:\,\tau i^m < \delta_i^m}\{f_i^m\}$

Or $\max_m\{f_i^m\} - \min_{m:\,\tau i^m < \delta_i^m}\{f_i^m\} \le \varepsilon$

Note: $\tau_i^m < \delta_i^m$ implies that $\alpha_i^m > 0$. Since $\Sigma_m\alpha_i^m = 1$, for any i each α_i^m is treated as the probability that the data point belongs to class m. Hence we define KKT violators as:

$\max_m\{f_i^m\} - \min_{m:\,\tau i^m < \delta_i^m}\{f_i^m\} > \varepsilon$ for all i.

Decomposition Method to Solve the Dual

Using the method in [7] to solve the Dual, maximize

$Q(\tau) = -1/2\Sigma_{i,j}\tau_i.\tau_j(K_{ij} + \beta) + \beta\Sigma_i\Delta_{yi}\tau_i$

Subject to: $\tau_i \le \Delta_{yi}$, $\tau_i.1 = 0$, i = 1 ...l

Expanding in terms of a single 'τ' vector:

$Q_p(\tau_p) = -1/2A_p(\tau_p.\ \tau_p) - B_p.\tau_p + C_p$

Where:

$A_p = K_{pp} + \beta$

$B_p = -\beta\Delta_{yp} + \Sigma_{i\ne p}\tau_i(K_{ip} + \beta)$

$C_p = -1/2\Sigma_{i,j\ne p}\tau_i.\tau_j(K_{ij} + \beta) + \beta\Sigma_{i\ne p}\tau_i\Delta_{yi}$

Therefore ignoring the constant term 'C_p', we have to minimize:

$Q_p(\tau_p) = 1/2A_p(\tau_p.\ \tau_p) + B_p.\tau_p$

Subject to: $\tau_p \le \Delta_{yp}$ and $\tau_p.1 = 0$

The above equation can also be written as:

$Q_p(\tau_p) = 1/2A_p(\tau_p + B_p/A_p).(\tau_p + B_p/A_p) - B_p.B_p/2A_p$

Substitute $v = (\tau_p + B_p/A_p)$ & $D = (\Delta_{yp} + B_p/A_p)$ in the above equation. Hence, after ignoring the constant term $B_p.B_p/2A_p$ and the multiplicative factor 'A_p' we have to minimize:

$Q(v) = 1/2v.v = 1/2||v||^2$

Subject to: $v \le D$ and $v.1 = D.1 - 1$

The Lagrangian is given by:

$L(v) = 1/2||v||^2 - \Sigma_m\rho_m(D_m - v_m) - \sigma[\Sigma_m(v_m - D_m) + 1]$

Subject to: $\rho_m \le 0$

Hence $\partial L/\partial v_m = v_m + \rho_m - \sigma = 0$. By KKT conditions we have: $\rho_m(D_m - v_m) = 0$ & $\rho_m \ge 0$, also $v_m \le D_m$. Hence by combining the above in-equalities, we have: $v_m = Min\{D_m, \sigma\}$, or $\Sigma_m v_m = \Sigma_m Min\{D_m, \sigma\} = \Sigma_m D_m - 1$. The above equation uniquely defines the 'σ' that satisfies the above equation AND that 'σ' is the optimal solution of the quadratic optimization problem. (Refer to [7] for a formal proof).

Solve for 'σ': We have $\mathrm{Min}\{D_m, \sigma\} + \mathrm{Max}\{D_m, \sigma\} = D_m + \sigma$, hence $\Sigma_m[D_m + \sigma - \mathrm{Max}\{D_m, \sigma\}] = \Sigma_m D_m - 1$, or $\sigma = 1/K[\Sigma_m \mathrm{Max}\{D_m, \sigma\} - 1]$, hence we find σ (iteratively) that satisfies the equation: $|(\sigma_l - \sigma_{l+1})/\sigma_l| \leq$ tolerance. The initial value for 'σ' is set to $\sigma_1 = 1/K[\Sigma_m D_m - 1]$.

Update rule for 'τ': Once we have 'σ', $\tau_{new}{}^m = v_m - B_p{}^m/(K_{pp} + \beta)$, or:

$\tau_{new}{}^m = v_m - f_p{}^m/(K_{pp} + \beta) + \tau_{old}{}^m$

SVM-Internal Clustering

Let $\{x_i\}$ be a data set of 'N' points in R^d. Using a non-linear transformation ϕ, we transform 'x' to some high-dimensional space called Kernel space and look for the smallest enclosing sphere of radius 'R'. Hence we have: $||\phi(x_j) - a||^2 \leq R^2$ for all j = 1,...,N; where 'a' is the center of the sphere. Soft constraints are incorporated by adding slack variables 'ζ_j':

$||\phi(x_j) - a||^2 \leq R^2 + \zeta_j$ for all j = 1,...,N

Subject to: $\zeta_j \geq 0$

We introduce the Lagrangian as:

$L = R^2 - \Sigma_j \beta_j (R^2 + \zeta_j - ||\phi(x_j) - a||^2) - \Sigma_j \zeta_j \mu_j + C\Sigma_j \zeta_j$

Subject to: $\beta_j \geq 0$, $\mu_j \geq 0$,

where C is the cost for outliers and hence $C\Sigma_j \zeta_j$ is a penalty term. Setting to zero the derivative of 'L' w.r.t. R, a and ζ we have: $\Sigma_j \beta_j = 1$; $a = \Sigma_j \beta_j \phi(x_j)$; and $\beta_j = C - \mu_j$.

Substituting the above equations into the Lagrangian, we have the dual formalism as:

$W = 1 - \Sigma_{i,j} \beta_i \beta_j K_{ij}$ where $0 \leq \beta_i \leq C$; $K_{ij} = \exp(-||x_i - x_j||^2/2\sigma^2)$

Subject to: $\Sigma_i \beta_i = 1$

By KKT conditions we have: $\zeta_j \mu_j = 0$ and $\beta_j(R^2 + \zeta_j - ||\phi(x_j) - a||^2) = 0$.

In the kernel space of a data point 'x_j' if $\zeta_j > 0$, then $\beta_j = C$ and hence it lies outside of the sphere i.e. $R^2 < ||\phi(x_j) - a||^2$. This point becomes a bounded support vector or BSV. Similarly if $\zeta_j = 0$, and $0 < \beta_j < C$, then it lies on the surface of the sphere i.e. $R^2 = ||\phi(x_j) - a||^2$. This point becomes a support vector or SV. If $\zeta_j = 0$, and $\beta_j = 0$, then $R^2 > ||\phi(x_j) - a||^2$ and hence this point is enclosed with-in the sphere.

Nanopore Detector based Channel Current Cheminformatics

All data analyzed is obtained from a nanopore detector and relates to single molecule blockades of a single protein channel. The protein channel is the α-hemolysin pore-forming toxin from *Staphylococcus aureus*, which has a molecule-sized channel opening for partial capture, if not translocation, of biomolecules drawn in by electrophoretic forces (such as DNA) [3,10-20]. Further details on the detector and signal processing architecture are shown in Fig. 2. Further detail on the components of the extracted SVM feature vectors (on events due to individual blockade events), are given in the Methods. Although the figure can only show one SVM classifier implementation (that used in [3]), the data sets examined by all the SVMs described are kept the same (for comparative purposes), so the signal acquisition and feature extraction stages show how the SVM feature vectors are obtained.

Information measures

The fundamental information measures are Shannon entropy, mutual information, and relative entropy (also known as the Kullback-Leibler divergence or distance). Shannon entropy, $\sigma = -\Sigma_x \mathbf{p(x)log(p(x))}$, is a measure of the information in distribution $\mathbf{p(x)}$. Mutual Information, $\mu = \Sigma_x \Sigma_y \mathbf{p(xy)log(p(xy)/p(x)p(y))}$, is a measure of information one random variable has about another random variable. Relative Entropy (Kullback-Leibler distance): $\rho = \Sigma_x \mathbf{p(x)\ log(p(x)/q(x))}$, is a measure of distance between two probability distributions. Mutual information is a special case of relative entropy between a joint probability (two-component in simplest form) and the product of component probabilities.

Khinchin derivation of Shannon entropy

In his now famous 1948 paper, Claude Shannon [21] provided a qualitative measure for entropy in connection with communication theory. The Shannon entropy measure was later put on a more formal footing by A. I. Khinchin in an article where he proves that with certain reasonable assumptions the Shannon entropy is unique [22]. A statement of the theorem is as follows:

Khinchine Uniqueness Theorem

Let $\mathbf{H(p_1,p_2,...,p_n)}$ be a function defined for any integer $\mathbf{n}$ and for all values $\mathbf{p_1,p_2,...,p_n}$ such that $\mathbf{p_k \geq 0\ (k = 1,2,...,n)}$, and $\mathbf{\Sigma_k p_k = 1}$. If for any function $\mathbf{n}$ this function is continuous with respect to its arguments, and if the function obeys the three properties listed below, then $\mathbf{H(p_1,p_2,...,p_n) = -\lambda\Sigma_k p_k log(p_k)}$, where λ is a positive constant (with Shannon entropy recovered for convention $\lambda = \mathbf{1}$). The three properties are:

(1) For given $\mathbf{n}$ and for $\mathbf{\Sigma_k p_k = 1}$, the function takes its largest value for $\mathbf{p_k = 1/n\ (k = 1,2,...,n)}$. This is equivalent to

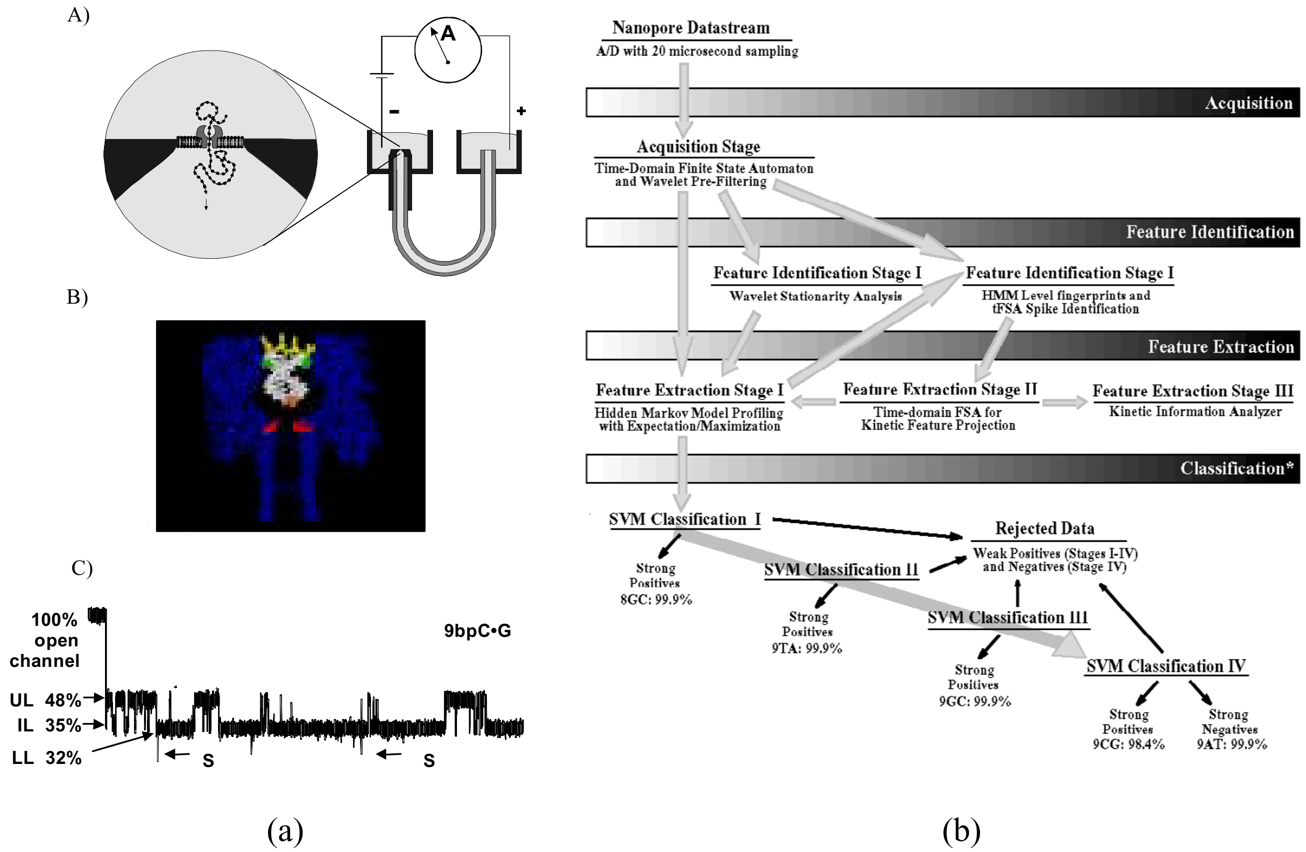

A)
A
-
+
B)
C)
100% open channel
9bpC•G
UL 48%
IL 35%
LL 32%
S
S
Nanopore Datastream
A/D with 20 microsecond sampling
Acquisition
Acquisition Stage
Time-Domain Finite State Automaton and Wavelet Pre-Filtering
Feature Identification
Feature Identification Stage I
Wavelet Stationarity Analysis
Feature Identification Stage I
HMM Level fingerprints and tFSA Spike Identification
Feature Extraction
Feature Extraction Stage I
Hidden Markov Model Profiling with Expectation/Maximization
Feature Extraction Stage II
Time-domain FSA for Kinetic Feature Projection
Feature Extraction Stage III
Kinetic Information Analyzer
Classification*
SVM Classification I
Rejected Data
Weak Positives (Stages I-IV) and Negatives (Stage IV)
Strong Positives 8GC: 99.9%
SVM Classification II
Strong Positives 9TA: 99.9%
SVM Classification III
Strong Positives 9GC: 99.9%
SVM Classification IV
Strong Positives 9CG: 98.4%
Strong Negatives 9AT: 99.9%
(a)
(b)

Figure 2

a. (A) shows a nanopore device based on the α-hemolysin channel. It has been used for analysis of single DNA molecules, such as ssDNA, shown, and dsDNA, a nine base-pair DNA hairpin is shown in (B) superimposed on the channel geometry. The channel current blockade trace for the nine base-pair DNA hairpin blockade from (B) is shown in (C). **b** shows the signal processing architecture that was used to classify DNA hairpins with this approach: Signal acquisition was performed using a time-domain, thresholding, Finite State Automaton, followed by adaptive pre-filtering using a wavelet-domain Finite State Automaton. Hidden Markov Model processing with Expectation-Maximization was used for feature extraction on acquired channel blockades. Classification was then done by Support Vector Machine on five DNA molecules: four DNA hairpin molecules with nine base-pair stem lengths that only differed in their blunt-ended DNA termini, and an eight base-pair DNA hairpin. The accuracy shown is obtained upon completing the 15th single molecule sampling/classification (in approx. 6 seconds), where SVM-based rejection on noisy signals was employed.

Laplace's principle of insufficient reason, which says if you don't know anything assume the uniform distribution (also agrees with Occam's Razor assumption of minimum structure).

(2) $\mathbf{H(ab) = H(a) + H_a(b)}$, where $\mathbf{H_a(b) = -\Sigma_a p(a)log(p(b|a))}$, is the conditional entropy. This is consistent with $\mathbf{H(ab)=H(a)+H(b)}$, for probabilities of **a** and **b** independent, with modifications involving conditional probability being used when not independent.

(3) $\mathbf{H(p_1,p_2,\dots,p_n,0) = H(p_1,p_2,\dots,p_n)}$. This reductive relationship, or something like it, is implicitly assumed when describing any system in "isolation."

Relative Entropy Uniqueness

This falls out of a geometric formalism on families of distributions: the Information Geometry formalism described by S. Amari [23-25]. Together with Laplace's principle of insufficient reason on the choice of "reference" distribution in the relative entropy expression, this will reduce to Shannon entropy, and thus uniqueness on Shannon entropy from a geometric context. The parallel with geometry is the Euclidean distance for "flat" geometry (simplest assumption of structure), vs. the "distance" between distributions as described by the Kullback-Leibler divergence.

The Success of Distributions of Nature suggests Generalization from Geometric Feature-Space Kernels to Distribution Feature-Space Kernels

Using the Shannon entropy measure it is possible to derive the classic probability distributions of statistical physics by maximizing the Shannon measure subject to appropriate linear momentum constraints. Constrained variational optimizations involving the Shannon entropy measure can, thus, provide a unified framework with which to describe all, or most, of statistical mechanics. The distributions derivable within the maximum entropy formalism include the Maxwell-Boltzmann, Bose-Einstein, Fermi-Dirac, and Intermediate distributions. The maximum entropy method for defining statistical mechanical systems has been extensively studied by [26].

Both statistical estimation and maximum entropy estimation are concerned with drawing inferences from partial information. The maximum entropy approach estimates a probability density function when only a few moments are known (where there are an infinite number of higher moments). The statistical approach estimates the density function when only one random sample is available out of an infinity of possible samples. The maximum entropy estimation may be significantly more robust (against over-fitting, for example) in that it has an Occam's Razor argument that "cuts both ways" – use *all* of the information given and avoid using any information not given. This means that out of all of the probability distributions consistent with the set of constraints, choose the one that has maximum uncertainty, i.e., maximum entropy [27].

At the same time that Jaynes was doing his work, essentially an optimization principle based on Shannon entropy, Soloman Kullback was exploring optimizations involving a notion of probabilistic distance known as the Kullback-Leibler distance, referred to above as the relative entropy [28]. The resulting minimum relative entropy (MRE) formalism reduces to the maximum entropy formalism of Jaynes when the reference distribution is uniform. The information distance that Kullback and Leibler defined was an oriented measure of "distance" between two probability distributions. The MRE formalism can be understood to be an extension of Laplace's *Principle of Insufficient Reason* (e.g., if nothing known assume the uniform distribution) in a manner like that employed by Khinchine in his uniqueness proof, but now incorporating constraints.

In their book *Entropy Optimization Principles with Applications* [27], Kapur and Kesavan argue for a generalized entropy optimization approach to the description of distributions. They believe every probability distribution, theoretical or observed, is an entropy optimization distribution, i.e., it can be obtained by maximizing an appropriate entropy measure, or by minimizing a relative entropy measure with respect to an appropriate *a priori* distribution. The primary objective in such a modeling procedure is to represent the problem as a simple combination of probabilistic entities that have a simple set of moment constraints. Generalized measures of distributional distance can also be explored along the lines of generalized measures of geometric distance. In physics, not every geometric distance is of interest, however, since the special theory of relativity tells us that spacetime is locally flat (Lorentzian, which is Euclidean on spatial slices), with metric generalization the Riemannian metrics. Likewise, perhaps not all distributional distance measures are created equal either. What the formalism of Information Geometry [23-25] reveals, among other things, is that relative entropy is uniquely structureless (like flat geometry) and is perturbatively stable, i.e., has a well-defined Taylor expansion at short divergence range, just like the locally Euclidean metrics at short distance range.

Results

SVM Kernel/Algorithm Variants

The SVM Kernels of interest are "regularized" distances or divergences, where they are regularized if in the form of an exponential with argument the negative of some distance-measure squared ($d^2(x,y)$) or symmetrized divergence measure ($D(x,y)$), the former if using a geometric heuristic for comparison of feature vectors, the latter if using a distributional heuristic. For the Gaussian Kernel: $d^2(x,y) = \Sigma k(xk-yk)^2$; for the Absdiff Kernel $d^2(x,y)=(\Sigma k|xk-yk|)^{1/2}$; and for the Symmetrized Relative Entropy Kernel $D(x,y)= D(x||y)+D(y||x)$, where $D(x||y)$ is the standard relative entropy. Results are shown in Fig. 3.

The SVM algorithm variants being explored are only briefly mentioned here. In the standard Platt SMO algorithm, η = 2*K12-K11-K22, and speedup variations are described to avoid calculation of this value entirely. A middle ground is sought with the following definition "η = 2*K12-2; If (η >= 0) { η = -1;}" (labeled WH SMO in Fig. 3, underflow handling and other implementations differ slightly in the implementation shown as well).

SVM-Internal Speedup via differentiating BSVs and SVs

Fig. 4 shows the percent increase in iterations-to-convergence against the 'C' value. Fig. 5 shows the number of bounded support vectors (BSV) as a function of 'C' value. Since the algorithm presented in [7] does not differentiate between SV and BSV, a lot of time is spent in trying to adjust the weights of the BSV i.e. weak data. The weight of a BSV may range from [0, 0.5) in their algorithm. In our modification to the algorithm, shown below, as soon as we identify the BSV (as specified by Case III conditions), its weight is no longer adjusted. Hence faster convergence is achieved without sacrificing accuracy:

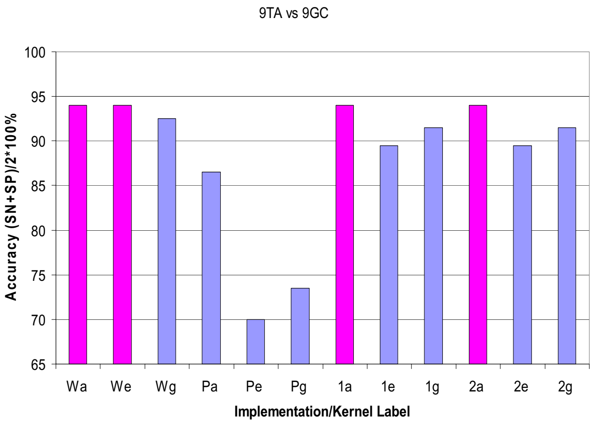


Figure 3
Comparative results are shown on performance of Kernels and algorithmic variants. The classification is between two DNA hairpins (in terms of features from the blockade signals they produce when occluding ion flow through a nanometer-scale channel). Implementations: WH SMO (W); Platt SMO (P); Keerthi1 (1); and Keerthi2 (2). Kernels: Absdiff (a); Entropic (e); and Gaussian (g). The best algorithm/kernel on this and other channel blockade data studied has consistently been the WH SMO variant and the Absdiff and Entropic Kernels. Another benefit of the WH SMO variant is its significant speedup over the other methods (about half the time of Platt SMO and one fourth the time of Keerthi 1 or 2).

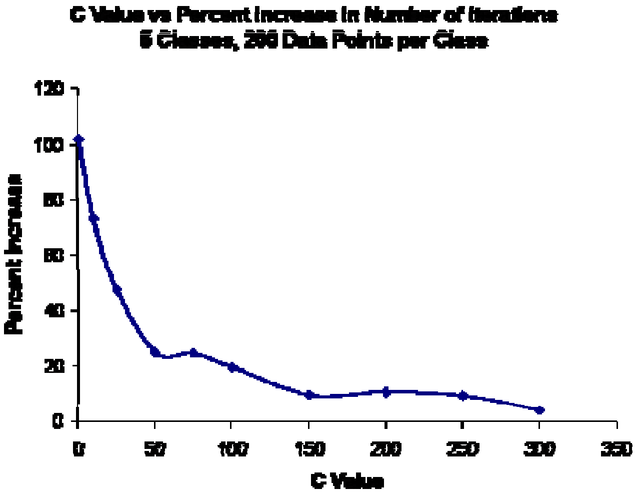


Figure 4
The percent increase in iterations-to-convergence against the 'C' value. For very low values of 'C' the gain is doubled while for very large values of 'C' the gain is low (almost constant for C > 150). Thus we note the dependence of the gain on 'C' value.

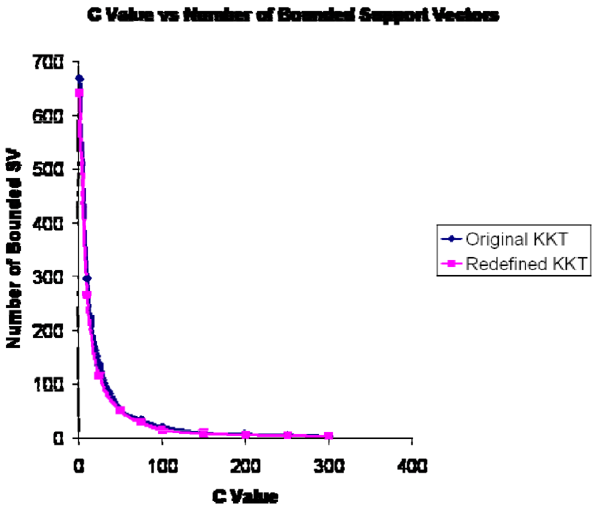


Figure 5
The number of bounded support vectors (BSV) as a function of 'C' value. There are many BSVs for very low values of 'C' and very few BSVs for large values of 'C'. Thus we can say that the number of BSVs plays a vital role in the speed of convergence of the algorithm.

For the BSV/SV-tracking speedup, the KKT violators are redefined as:

For all $m \neq y_i$ we have:

$\alpha_i^m\{f_{yi} - f_m - 1 + \zeta_i\} \geq 0$

Subject to: $1 \geq \alpha_i^m \geq 0$; $\sum_m \alpha_i^m = 1; \zeta_i \geq 0$ for all i,m

Where $f_m = (1/\beta)[w_m \cdot x_i + b_m]$ for all m

Case I:

If $\alpha_i^m = 0$ for m S.T $f_m = f_m^{max}$

Implies $\alpha_i^{yi} > 0$ and hence $\zeta_i = 0$

Hence $f_{yi} - f_m^{max} - 1 \geq 0$

Case II:

If $1 > \alpha_i^m > 0$ for m S.T $f_m = f_m^{max}$ and $\alpha_i^{yi} > \alpha_i^m$

Implies $\zeta_i = 0$

Hence $f_{yi} - f_m^{max} - 1 = 0$

Case III:

If $1 \geq \alpha_i^m > 0$ for m S.T $f_m = f_m^{max}$ and $\alpha_i^{yi} \leq \alpha_i^m$

Implies $\zeta_i > 0$

Hence $f_{yi} - f_m^{max} - 1 + \zeta_i = 0$

Or $f_{yi} - f_m^{max} - 1 < 0$

Data Rejection Tuning with SVM-Internal vs SVM-External Classifiers

The SVM Decision Tree shown in Fig. 2b obtained nearly perfect sensitivity and specificity, with a high data rejection rate, and a highly non-uniform class signal-calling throughput. In Fig. 6, the Percentage Data Rejection vs SN+SP curves are shown for test data classification runs with a binary classifier with one molecule (the positive, given by label) versus the rest (the negative). Since the signal calling wasn't passed through a Decision Tree, the way these curves were generated, they don't accurately reflect total throughput, and they don't benefit from the "shielding" shown in the Decision Tree in Fig. 2b prototype. In the SVM Decision Tree implementation described in Fig. 2b[3], this is managed more comprehensively, to arrive at a five-way signal-calling throughput at the furthest node of 16% (in Fig. 1a, 9CG and 9AT have to pass to the furthest node to be classified), while the best throughput, for signal calling on the 8GC molecules, is 75%.

The SVM Decision Tree classifier's high, non-uniform, rejection can be managed by generalizing to a collection of Decision Trees (with different species at the furthest node). The problem is that tuning and optimizing a single decision tree is already a large task, even for five species (as in Fig. 2). With a collection of trees this problem is seemingly compounded, but can actually be lessened in some ways in that now each individual tree need not be so well-tuned/optimized. Although more complicated to implement than an SVM-External method, the SVM-Internal multiclass methods are not similarly fraught with tuning/optimization complications. Fig. 7 shows the Percentage Data Rejection vs SN+SP curves on the same train/test data splits as used for Fig. 6, except now the drop curves are to be understood as *simultaneous* curves (not sequential application of such curves as in Fig. 6). Thus, comparable, or better, performance is obtained with the multiclass-internal approach and with far less effort since there is no managing and tuning of Decision Trees. Another surprise, and even stronger argument for the SVM-Internal approach to the problem, is that a natural drop zone is indicated by the margin.

Marginal Drop with SVM-Internal

Suppose we define the criteria for dropping weak data as the margin: For any data point x_i; let $\max_m\{f_m(x_i)\} = f_{yi}$, and Let $f_m = \max_m\{f_m(x_i)\}$ for all $m \neq yi$, then we define the margin as: $(f_{yi} - f_m)$, hence data point x_i is dropped if $(f_{yi} - f_m)$ = Confidence Parameter. (For this data set using Gaussian, AbsDiff & Sentropic kernel, a confidence parameter of at least (0.00001)*C was required to achieve 100% accuracy.) The results are shown in Table 1. Using the margin drop approach, there is even less tuning, and

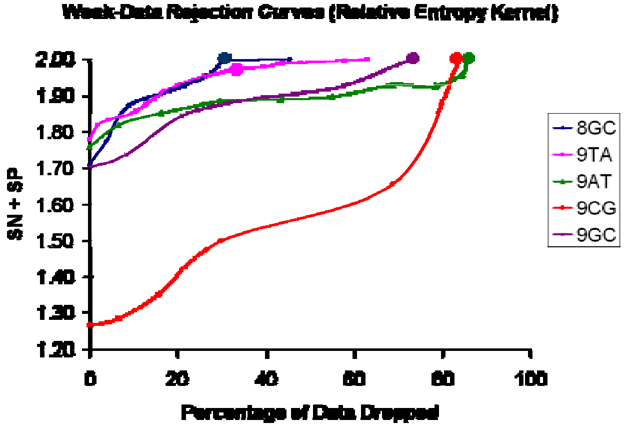


Figure 6

The Percentage Data Rejection vs SN+SP curves are shown for test data classification runs with a binary classifier with one molecule (the positive, given by label) versus the rest (the negative). Since the signal calling wasn't passed through a Decision Tree, it doesn't accurately reflect total throughput, and they don't benefit from the "shielding" shown in the Decision Tree in Fig. 1 prototype. The Relative Entropy Kernel is shown because it provided the best results (over Gaussian and Absdiff).

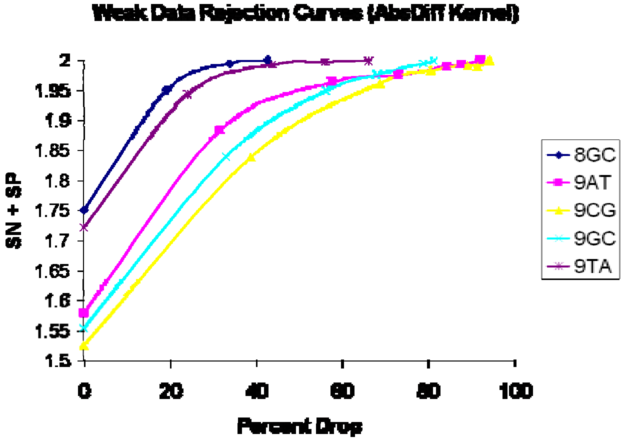


Figure 7

The Percentage Data Rejection vs SN+SP curves are shown for test data classification runs with a multiclass discriminator. The following criterion is used for dropping weak data: for any data point x_i; if $\max_m\{f_m(x_i)\} \leq$ Confidence Parameter, then the data point x_i is dropped. For this data set using AbsDiff kernel ($\sigma^2 = 0.2$) performed best, and a confidence parameter of 0.8 achieve 100% accuracy.

Table 1: The table shows the results of dropping data that falls in the margin. For any data point x_i; let $max_m\{f_m(x_i)\} = f_{yi}$, and Let $f_m = max_m\{f_m(x_i)\}$ for all $m \neq yi$, then we define the margin as: $(f_{yi} - f_m)$, hence data point x_i is dropped if $(f_{yi} - f_m) \leq$ Confidence Parameter. Using the margin drop approach, there is even less tuning, and there is improved throughput (approximately 75% for *all* species).

Kernel	8GC	9AT	9CG	9GC	9TA
Gaussian	P: 1268 TP: 1087 SN+SP: 1.76 P: 1087 TP: 1087 SN+SP: 2 **Drop = 9.42**	P: 1178 TP: 934 SN+SP: 1.57 P: 934 TP: 934 SN+SP: 2 **Drop = 22.17**	P: 1166 TP: 904 SN+SP: 1.53 P: 904 TP: 904 SN+SP: 2 **Drop = 24.67**	P: 1172 TP: 897 SN+SP: 1.51 P: 897 TP: 897 SN+SP: 2 **Drop = 25.25**	P: 1216 TP: 1027 SN+SP: 1.70 P: 1027 TP: 1027 SN+SP: 2 **Drop = 14.42**
AbsDiff	P: 1407 TP: 1134 SN+SP: 1.75 P: 1134 TP: 1134 SN+SP: 2 **Drop = 5.5**	P: 1151 TP: 928 SN+SP: 1. 58 P: 928 TP: 928 SN+SP: 2 **Drop = 22.67**	P: 1177 TP: 906 SN+SP: 1.53 P: 906 TP: 906 SN+SP: 2 **Drop = 24.5**	P: 1050 TP: 870 SN+SP: 1.55 P: 870 TP: 870 SN+SP: 2 **Drop = 27.5**	P: 1215 TP: 1040 SN+SP: 1.72 P: 1040 TP: 1040 SN+SP: 2 **Drop = 13.33**
Entropic	P: 1165 TP: 1038 SN+SP: 1.75 P: 1038 TP: 1038 SN+SP: 2 **Drop = 13.5**	P: 1480 TP: 995 SN+SP: 1.50 P: 991 TP: 991 SN+SP: 2 **Drop = 17.42**	P: 1348 TP: 922 SN+SP: 1.45 P: 920 TP: 920 SN+SP: 2 **Drop = 23.33**	P: 960 TP: 804 SN+SP: 1.50 P: 803 TP: 803 SN+SP: 2 **Drop = 33.08**	P: 1047 TP: 970 SN+SP: 1.73 P: 970 TP: 970 SN+SP: 2 **Drop = 19.17**

there is improved throughput (approximately 75% for *all* species).

SVM-Internal Clustering

The SVM-Internal approach to clustering was originally defined by [29]. Data points are mapped by means of a kernel to a high dimensional feature space where we search for the minimal enclosing sphere. In what follows, Keerthi's method is used to solve the dual (see Methods for further details).

The minimal enclosing sphere, when mapped back into the data space, can separate into several components; each enclosing a separate cluster of points. The width of the kernel (say Gaussian) controls the scale at which the data is probed while the soft margin constant helps to handle outliers and over-lapping clusters. The structure of a dataset is explored by varying these two parameters, maintaining a minimal number of support vectors to assure smooth cluster boundaries.

We have used the algorithm defined in [29] to identify the clusters, with methods adapted from [30,31 for their handling. If the number of data points is 'n', then we require n(n-1)/2 number of comparisons. We have made modifications to the algorithm such that we eliminate comparisons that do not have an impact on the cluster connectivity. Hence the number of comparisons required will be less than n(n-1)/2.

In each comparison we sub-divide the line segment connecting the two data points into 20 parts; hence we obtain 19 different points on this line segment. The two data points belong to the same cluster only if all the 19 points lie inside the cluster. Given the cost of evaluating utmost 19 points for every comparison, the need to eliminate comparisons that do not have an impact on the cluster connectivity becomes even more important. Finally we have used Depth First Search (DFS) algorithm for the cluster harvest. Results are shown in Tables 2 and 3. The approach to the solving the Dual problem is shown in the Methods.

SVM-External Clustering

As with the multiclass SVM discriminator implementations, the strong performance of the binary SVM enables SVM-External as well as SVM-Internal approaches to clustering. Our external-SVM clustering algorithm clusters data vectors with no *a priori* knowledge of each vector's class. The algorithm works by first running a Binary SVM against a data set, with each vector in the set randomly labeled, until the SVM converges (Fig. 8). In order to obtain convergence, an acceptable number of KKT violators must be found. This is done through running the SVM on the randomly labeled data with different numbers of allowed violators until the number of violators allowed is near the lower bound of violators needed for the SVM to converge on the particular data set. Choice of an appropriate kernel and an acceptable sigma value also will affect

Table 2: The table shows clustering predictions when working with 400 Samples (200 each of 9GC & 9CG) with a Gaussian Kernel with Width = 50 ($\sigma^2 = 0.01$).

C Value	Number of SV	Percent of Outliers	Number of Clusters	Number of Comparisons
0.25	91	0	10	39005
0.025	87	1.25	5	37020
0.0125	44	13.75	4	29202
0.01	29	21.75	2	24145

Table 3: The table shows clustering predictions when working with 1200 Samples (600 each of 9GC & 9CG) with a Gaussian Kernel with Width = 50 ($\sigma^2 = 0.01$).

C Value	Number of SV	Percent of Outliers	Number of Clusters	Number of Comparisons
0.00833	106	5.8	4	10873
0.00417	37	18.25	2	232021
0.00333	31	23.8	2	203278
0.00278	23	29.08	2	177533

convergence. After the initial convergence is achieved, the sensitivity + specificity will be low, likely near 1. The algorithm now improves this result by iteratively relabeling the worst misclassified vectors, which have confidence factor values beyond some threshold, followed by rerunning the SVM on the newly relabeled data set. This continues until no more progress can be made. Progress is determined by an increasing value of sensitivity + specificity, hopefully nearly reaching 2. After this process, a high percentage of the previously unknown class labels of the data set will be known. With sub-cluster identification upon iterating the overall algorithm on the positive and negative clusters identified (until the clusters are no longer separable into sub-clusters), this method provides a way to cluster data sets without prior knowledge of the data's clustering characteristics, or the number of clusters. Figures 9 and 10 show clustering runs on a data set with a mixture of 8GC and 9GC DNA hairpin data. The set consists of 400 elements. Half of the elements belong to each class. The SVM uses a Gaussian Kernel and allows 3% KKT Violators.

Machine Learning and Cheminformatics Tools are Accessible via Website

The web-site provides an interface to several binary SVM variants (with other novel kernel selections), to a multiclass (internal) SVM, an FSA-based nanopore spike detector, and an HMM-based channel current feature extraction. New, web-accessible, channel current analysis tools, have also been developed for kinetic feature extraction (via channel current sub-level lifetimes), and clustering. The website is designed using HTML and CGI scripts that are executed to process the data sent when a form filled in by the user is received at the web server – results are then e-mailed to the address indicated by the user. The interface to this and all other software described is available via the group Home Page: http://logos.cs.uno.edu/~nano/ (see Fig. 11). The SVM interface offers options on chunk processing for large training sets (SV-carry by appending to next training chunk and SV-carry by maintaining state and injecting ("unfreezing") the next training chunk (a specialized α-heuristic). The interface offers use of arbitrary or structured feature vectors – where structured, in this case, corresponds to feature vector components that satisfy the properties of a non-trivial, non-reducible, discrete probability distribution. There is an SVM interface for a new single-optimization multiclass SVM discriminator (it simultaneously optimizes multiple hyperplanes). There is also an interface for our SVM-based clustering methods.

Discussion

Adaptive Feature Extraction/Discrimination

Adaptive feature extraction and discrimination, in the context of SVMs, can be accomplished by small batch reprocessing using the learned support vectors together with the new information to be learned. The benefit is that the easily deployed properties of SVMs can be retained while at the same time co-opting some of the on-line adaptive characteristics familiar from on-line learning with neural nets. This is also compatible with the chunking processing that is already implemented. A situation where such adaptation might prove necessary in nanopore signal analysis is if the instrumentation was found to have measurable, but steady, drift (at a new level of sensitivity for example). At the forefront of online adaptation, where the discrimination and feature extraction optimizations are inextricably mixed, further progress may derive

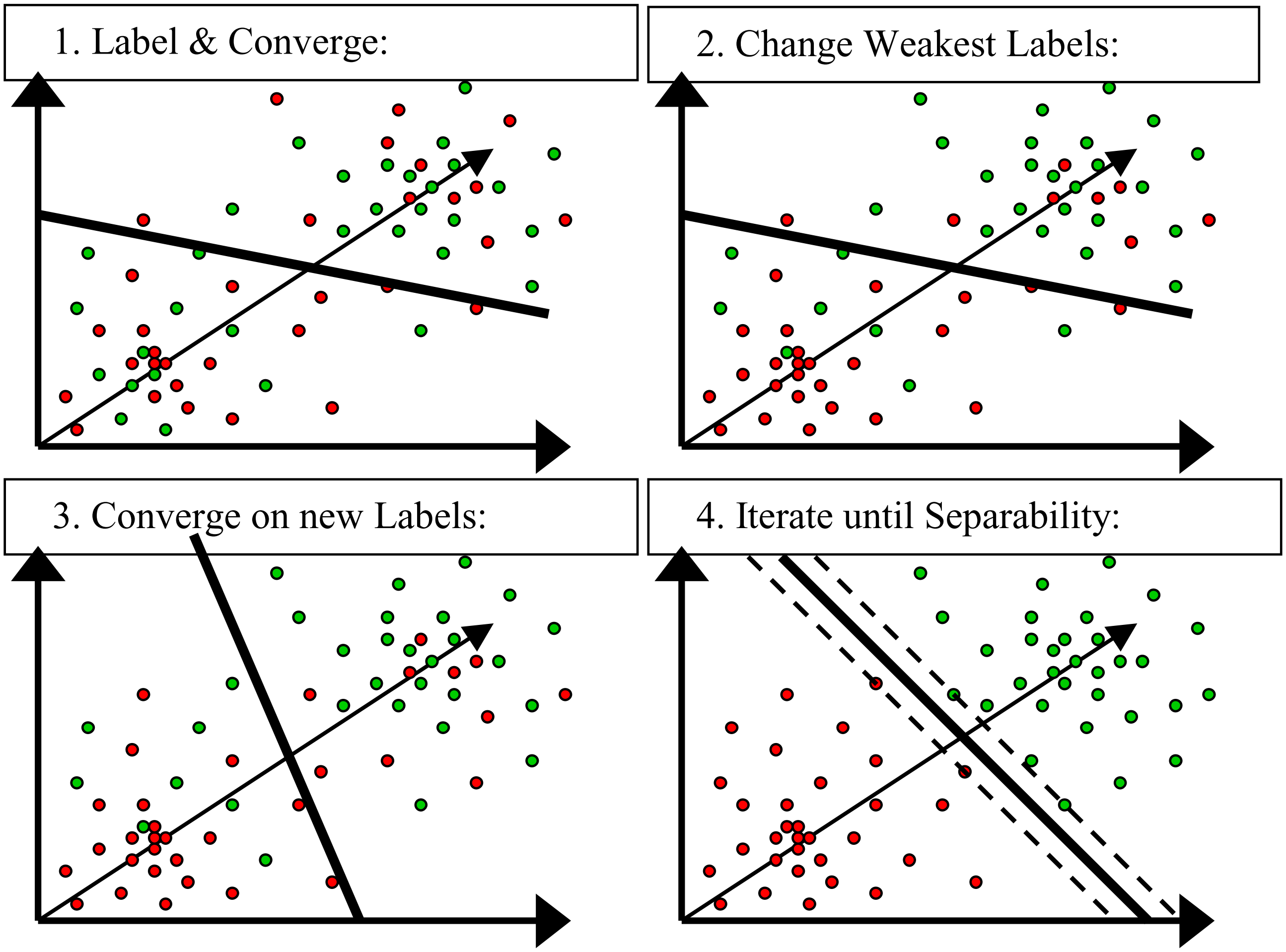

1. Label & Converge:
2. Change Weakest Labels:
3. Converge on new Labels:
4. Iterate until Separability:

Figure 8
Shown is the schematic for an "external" SVM clustering algorithm.

benefit from the Information-Geometrical methods of S. Amari [23-25].

Robust SVM performance in the presence of noise

In a parallel datarun to that indicated in Fig. 2a, with 150 component feature vectors, feature vectors with the full set of 2600 components were extracted (i.e., no compression was employed on the transition probabilities). SVM performance on the same train/test data splits, but with 2600 component feature vectors instead of 150 component feature vectors, offered similar performance after drop optimization. This demonstrates a significant robustness to what the SVM can "learn" in the presence of noise (some of the 2600 component have richer information, but even more are noise contributors).

AdaBoost Feature Selection

If SVM performance on the full HMM parameter set (the features extracted for each blockade signal) offers equivalent performance after rejecting weak data, then the possibility for significant improvement with selection on good parameters. An AdaBoost method is being used to select HMM parameters by representing each feature vector component as an independent Naïve Bayes classifier (trained on the data given), that then comprise the pool of experts in the AdaBoost algorithm [32-34]. The experts AdaBoost assigns heaviest weighting will then the components selected in the new, AdaBoost assigned, feature vector compression.

Conclusion

- External Multi-class SVM gave best results with Sentropic Kernel while Internal Multi-class SVM gave best results with AbsDiff kernel.

- Internal Multi-class approach overcomes the need to search for the best performing tree out of many possibili-

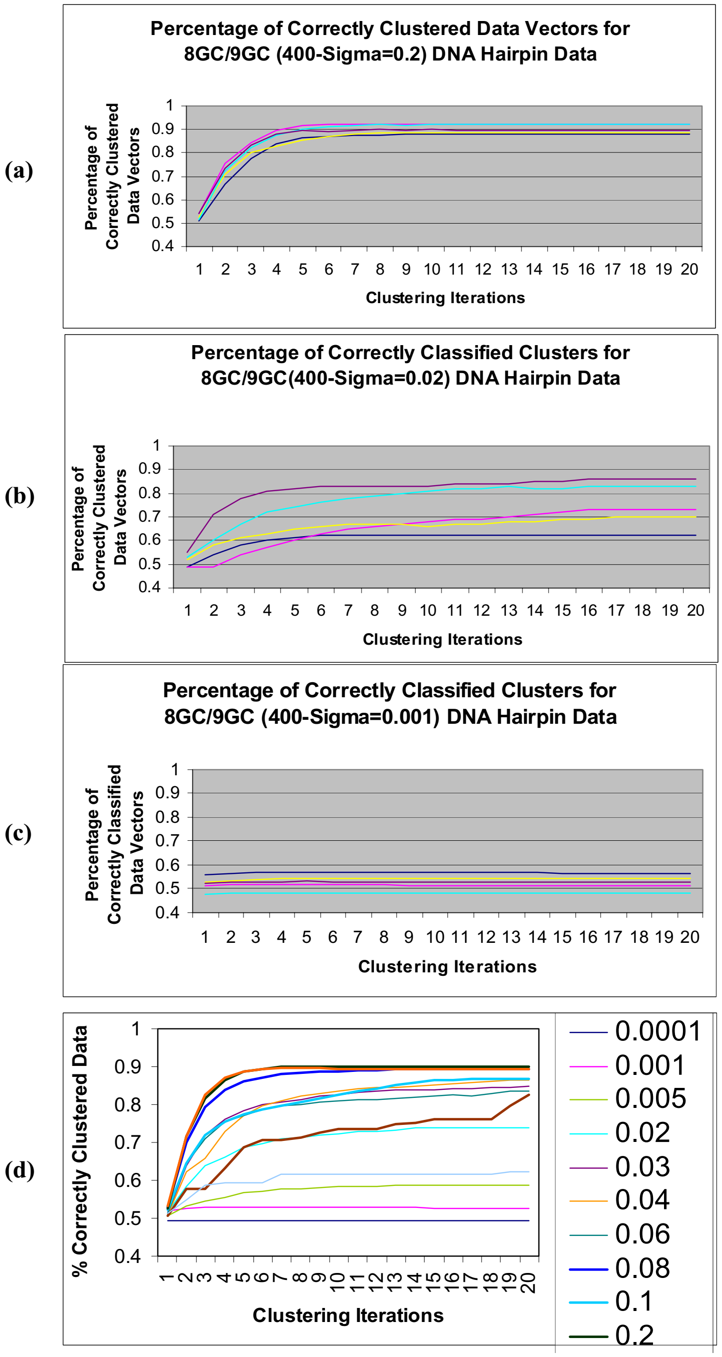


Figure 9

(a) The percentage correct classification (an indication of the clustering success) is shown with successive iteration of the clustering algorithm. Five separate test runs are shown, on different data from the same classes. Note that the plateau at around 0.9, this is approximately the performance of a supervised binary SVM on the same data (i.e., perfect separation isn't possible with this data without employing weak-data rejection). **(b)** The degradation in clustering performance for less optimal selection of kernel and tuning parameter (variance in case of Gaussian). **(c)** The degradation in clustering performance for non-optimal selection of kernel and tuning parameter (variance in case of Gaussian). **(d)** Summary of the degradation in clustering performance for less optimal selection of kernel and tuning parameter – with averages of the five test-runs are used as representative curves for that kernel/tuning selection in the above.

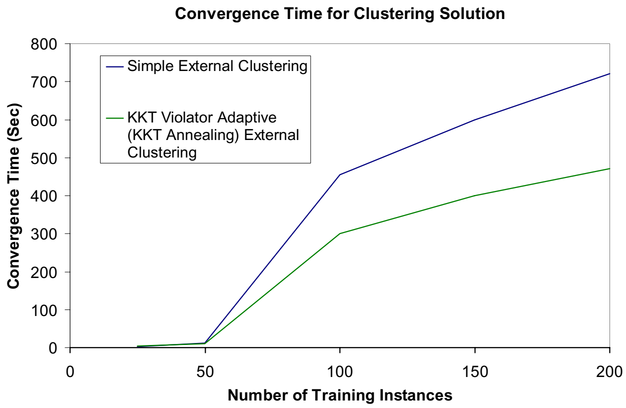


Figure 10
Efforts are underway to use simulated annealing in the number of KKT Violators tolerated on each iteration of the external clustering algorithm, to accelerate the convergence (clustering) process. Our current approach, results shown, approximately halves the cluster time needed.

ties. This is a huge advantage especially when the number of classes is large.

• Using a margin to define the drop zone for the internal multi-class approach produced far better results i.e. fewer data were dropped to achieve 100% accuracy.

• Additional benefit of using the margin is that the drop zone tuning to achieve 100% accuracy becomes trivial.

• External and Internal SVM *Clustering* Methods were also examined. The results show that our SVM-based clustering implementations can separate data into proper clusters without any prior knowledge of the elements' classification. this can be a powerful resource for insight into data linkages (topology).

Methods

The Feature Extraction used to obtain the Feature Vectors for SVM analysis

Signal Preprocessing Details

The Nanopore Detector is operated such that a stream of 100 ms samplings are obtained (throughput was approximately one sampling per 300 ms in [3]). Each 100 ms signal acquired by the time-domain FSA consists of a sequence of 5000 sub-blockade levels (with the 20 μs analog-to-digital sampling). Signal preprocessing is then used for adaptive low-pass filtering. For the data sets examined, the preprocessing is expected to permit compression on the sample sequence from 5000 to 625 samples (later HMM processing then only required construction of a dynamic programming table with 625 columns). The signal preprocessing makes use of an off-line wavelet stationarity analysis (Off-line Wavelet Stationarity Analysis, Figure 2b, also see [35]).

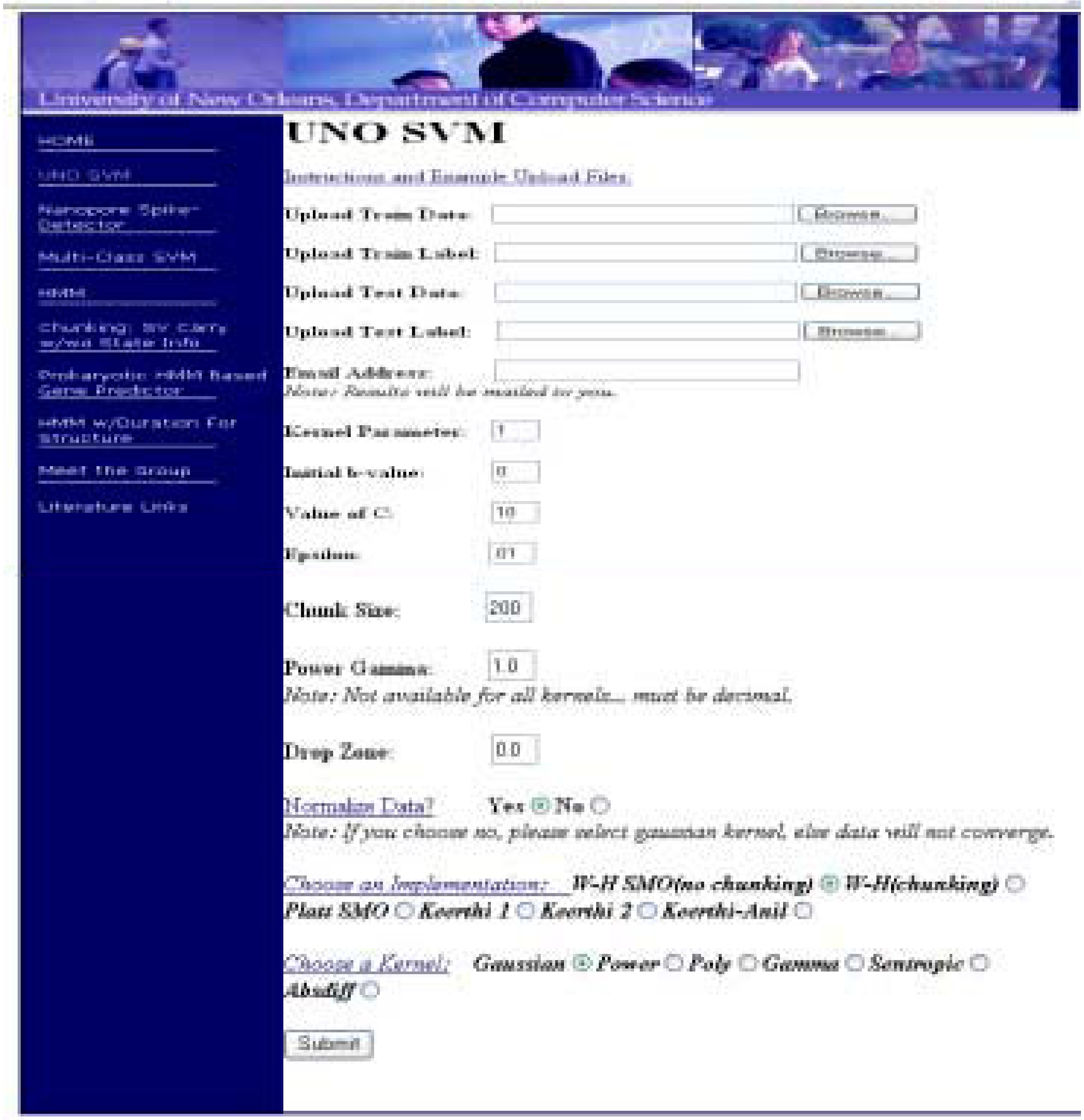
HMMs and Supervised Feature Extraction Details

With completion of preprocessing, an HMM [36] is used to remove noise from the acquired signals, and to extract features from them (Feature Extraction Stage, Fig. 2b). The HMM is, initially, implemented with fifty states, corresponding to current blockades in 1% increments ranging from 20% residual current to 69% residual current. The HMM states, numbered 0 to 49, corresponded to the 50 different current blockade levels in the sequences that are processed. The state emission parameters of the HMM are initially set so that the state j, 0 <= j <= 49 corresponding to level L = j+20, can emit all possible levels, with the probability distribution over emitted levels set to a discretized Gaussian with mean L and unit variance. All transitions between states are possible, and initially are equally likely. Each blockade signature is de-noised by 5 rounds of Expectation-Maximization (EM) training on the parameters of the HMM. After the EM iterations, 150 parameters are extracted from the HMM. The 150 feature vector components are extracted from the 50 parameterized emission probabilities, a 50-element compressed representation of the 50^2 transition probabilities, and an *a posteriori* information from the Viterbi path solution which is, essentially, a de-noised histogram of the bloackade sub-level occupation probabilities (further details in [3]). This information elucidates the blockade levels (states) characteristic of a given molecule, and the occupation probabilities for those levels, but doesn't directly provide kinetic information. An HMM-with-Duration has recently been introduced to better capture the latter information, but such feature vectors are not used in the studies shown in this paper, so this approach isn't discussed further in this paper.

Solving the Dual (Based on Keerthi's SMO [37])

The dual formalism is: $1 - \sum_{i,j}\beta_i\beta_j K_{ij}$ where $0 \le \beta_i \le C$; $K_{ij} = \exp(-||x_i - x_j||^2/2\sigma^2)$, also $\sum_i\beta_i = 1$. For any data point 'x_k', the distance of its image in kernel space from the center of the sphere is given by: $R^2(x_k) = 1 - 2\sum_i\beta_i K_{ik} + \sum_{i,j}\beta_i\beta_j K_{ij}$. The radius of the sphere is $R = \{R(x_k) \mid x_k \text{ is a Support Vectors}\}$, hence data points which are Support Vectors lie on cluster boundaries. Outliers are points that lie outside of the sphere and therefore they do not belong to any cluster i.e. they are Bounded Support Vectors. All other points are enclosed by the sphere and therefore they lie inside their respective cluster. KKT Violators are given as: (i) If $0 < \beta_i < C$ and $R(x_i) \neq R$; (ii) If $\beta_i = 0$ and $R(x_i) > R$; and (iii) If $\beta_i = C$ and $R(x_i) < R$.

The Wolfe dual is: $f(\beta) = \text{Min}_\beta \{\sum_{i,j}\beta_i\beta_j K_{ij} - 1\}$. In the SMO decomposition, in each iteration we select β_i & β_j and change them such that $f(\beta)$ reduces. All other β's are kept


University of New Orleans, Department of Computer Science
HOME
UNO SVM
Nanopore Spike Detector
Multi-Class SVM
HMM
chunking: SV carry w/wo State Info
Prokaryotic HMM based Gene Predictor
HMM w/Duration For Structure
Meet the Group
Literature Links
Browse...
Submit

UNO SVM

Instructions and Example Upload Files

Upload Train Data:

Upload Train Label:

Upload Test Data:

Upload Test Label:

Email Address:
Note: Results will be mailed to you.

Kernel Parameter: 1

Initial b-value: 0

Value of C: 10

Epsilon: .01

Chunk Size: 200

Power Gamma: 1.0
Note: Not available for all kernels... must be decimal.

Drop Zone: 0.0

Normalize Data? Yes ● No ○
Note: If you choose no, please select gaussian kernel, else data will not converge.

Choose an Implementation: ***W-H SMO(no chunking)*** ● ***W-H(chunking)*** ○ ***Platt SMO*** ○ ***Keerthi 1*** ○ ***Keerthi 2*** ○ ***Keerthi-Anil*** ○

Choose a Kernel: ***Gaussian*** ● ***Power*** ○ ***Poly*** ○ ***Gamma*** ○ ***Sentropic*** ○ ***Absdiff*** ○

Figure 11
Several channel current cheminformatics tools are available for use via web interfaces at http://logos.cs.uno.edu/~nano/. These tools include a variety of SVM interfaces for classification and clustering (binary and multiclass), and HMM tools for feature extraction and structure identification (with applications to both channel current cheminformatics and computational genomics).

constant for that iteration. Let us denote β_1 & β_2 as being modified in the current iteration. Also $\beta_1 + \beta_2 = (1 - \sum_{i=3}\beta_i) = s$, a constant. Let $\sum_{i=3}\beta_i K_{ik} = C_k$, then we obtain the SMO form: $f(\beta_1,\beta_2) = \beta^2_1 + \beta^2_2 + \sum_{i,j=3}\beta_i\beta_j K_{ij} + 2\beta_1\beta_2 K_{12} + 2\beta_1 C_1 + 2\beta_2 C_2$. Eliminating β_1: $f(\beta_2) = (s - \beta_2)^2 + \beta^2_2 + \sum_{i,j=3}\beta_i\beta_j K_{ij} + 2(s - \beta_2)\beta_2 K_{12} + 2(s - \beta_2)C_1 + 2\beta_2 C_2$. To minimize $f(\beta_2)$, we take the first derivative w.r.t. β_2 and equate it to zero, thus $f'(\beta_2) = 0 = 2\beta_2(1 - K_{12}) - s(1 - K_{12}) - (C_1 - C_2)$, and we get the update rule: $\beta_2^{new} = [(C_1 - C_2)/2(1 - K_{12})] + s/2$. We also have an expression for "$C_1 - C_2$" from:

$R(x_1^2) - R(x_2^2) = 2(\beta_2 - \beta_1)(1 - K_{12}) - 2(C_1 - C_2)$, thus $C_1 - C_2 = [R(x_2^2) - R(x_1^2)]/2 + (\beta_2 - \beta_1)(1 - K_{12})$, substituting, we have:

$\beta_1^{new} = \beta_1^{old} - [R(x_2^2) - R(x_1^2)]/[4(1 - K_{12})]$

Keerthi Algorithm

Compute 'C': if percent outliers = n and number data points = N, then: C = 100/(N*n)

Initialize β: Initialize m = int(1/C) - 1 number of randomly chosen indices to 'C'

Initialize two different randomly chosen indices to values less than 'C' such that $\sum_i \beta_i = 1$

Compute $R^2(x_i)$ for all 'i' based on the current value of β.

Divide data into three sets: Set I if $0 < \beta_i < C$; Set II if $\beta_i = 0$; and Set III if $\beta_i = C$.

Compute R^2_low = Max{ $R^2(x_i) \mid 0 \le \beta_i < C$} and R^2_up = Min{ $R^2(x_i) \mid 0 < \beta_i \le C$}.

In every iteration execute the following two paths alternatively until there are no KKT violators:

1. Loop through all examples (call Examine Example subroutine)

Keep count of number of KKT Violators.

2. Loop through examples belonging only to Set I (call Examine Example subroutine) until R^2_low - R^2_up < 2*tol.

Examine Example Subroutine

a. Check for KKT Violation. An example is a KKT violator if:

Set II and $R^2(x_i) > R^2$_up; choose R^2_up for joint optimization

Set III and $R^2(x_i) < R^2$_low; choose R^2_low for joint optimization

Set I and $R^2(x_i) > R^2$_up + 2*tol OR $R^2(x_i) < R^2$_low - 2*tol; choose R^2_low or R^2_up for joint optimization depending on which gives a worse KKT violator

b. Call the Joint Optimization subroutine

Joint Optimization Subroutine

a. Compute $\eta = 4(1 - K_{12})$ where K_{12} is the kernel evaluation of the pair chosen in Examine Example

b. Compute $D = [R^2(x_2) - R^2(x_1)]/\eta$

c. Compute Min{$(C - \beta_2), \beta_1$} = L1

d. Compute Min{$(C - \beta_1), \beta_2$} = L2

e. If D > 0; then D = Min{D, L1}

Else D = Max{D, -L2}

f. Update β_2 as: $\beta_2 = \beta_2 + D$

g. Update β_1 as: $\beta_1 = \beta_1 - D$

h. Re-compute $R^2(x_i)$ for all 'i' based on the changes in β_1 & β_2

i. Re-compute R^2_low & R^2_up based on elements in Set I, $R^2(x_1)$ & $R^2(x_2)$

The SVM-External Clustering Method

The SVM-clustering software is written in Perl. It runs data on a separate Binary SVM also written in Perl. This SVM uses a C file for kernel calculations. The data run on the SVM is created by running raw data through a tFSA/HMM(written in C), which creates a data set that contains 151 feature vectors for each element. The following is a simple step-by-step description of the basic algorithm used for SVM-clustering on this data:

1. Start with a set of data vectors (obtained through running raw data through tFSA/HMM feature extraction in Fig. 2b).

2. Randomly label each vector in the set as positive or negative.

3. Run the SVM on the randomly labeled data set until convergence is obtained (random relabeling is needed if prior random label scheme does not allow for convergence).

4. After initial convergence is obtained for the randomly labeled data set, relabel the misclassified data vectors, which have confidence factor values greater than some threshold.

5. Rerun the SVM on the newly relabeled data set.

6. Continue relabeling and rerunning SVM until no vectors in the data set are misclassified (or there is no improvement).

Authors' contributions

The paper was written by SWH and AY. The external clustering work was contributed by CM. The channel current feature vector extraction used to create the data sets was performed by ML.

Acknowledgements

SWH would like to thank MA and Prof. David Deamer at UCSC for strong collaborative support post-Katrina. Funding was provided by grants from the National Institutes for Health, The National Science Foundation, The Louisiana Board of Regents, and NASA.

References

1. Vapnik VN: **The Nature of Statistical Learning Theory.** 2nd edition. *Springer-Verlag, New York*; 1998.
2. Burges CJC: **A tutorial on support vector machines for pattern recognition.** *Data Min Knowl Discov* 1998, **2:**121-67.
3. Winters-Hilt S, Vercoutere W, DeGuzman VS, Deamer DW, Akeson M, Haussler D: **Highly Accurate Classification of Watson-Crick Basepairs on Termini of Single DNA Molecules.** *Biophys J* 2003, **84:**967-976.
4. Platt JC: **Fast Training of Support Vector Machines using Sequential Minimal Optimization.** In *Advances in Kernel Methods – Support Vector Learning Volume Ch. 12*. Edited by: *Scholkopf B, Burges CJC, Smola AJ. MIT Press, Cambridge, USA*; 1998.
5. Osuna E, Freund R, Girosi. F: **An improved training algorithm for support vector machines.** In *Neural Networks for Signal Processing VII* Edited by: *Principe J, Gile L, Morgan N, and Wilson E. IEEE, New York*; 1997:276-85.
6. Joachims T: **Making large-scale SVM learning practical.** In *Advances in Kernel Methods – Support Vector Learning Volume Ch. 11*. Edited by: *Scholkopf B, Burges CJC, Smola AJ. MIT Press, Cambridge, USA*; 1998.
7. Crammer K, Singer Y: **On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines.** *Journal of Machine Learning Research* 2001, **2:**265-292.
8. Hsu CW, Lin CJ: **A Comparison of Methods for Multi-class Support Vector Machines.** *IEEE Transactions on Neural Networks* 2002, **13;:**415-425.
9. Lee Y, Lin Y, Wahba G: **Multicategory Support Vector Machines.** *Technical Report 1043, Department of Statistics* 2001 [http://citeseer.ist.psu.edu/lee01multicategory.html]. *University of Wisconsin, Madison, WI*
10. Bezrukov SM, Vodyanoy I, Parsegian VA: **Counting polymers moving through a single ion channel.** *Nature* 1994, **370(6457):**279-281.
11. Kasianowicz JJ, Brandin E, Branton D, Deamer DW: **Characterization of Individual Polynucleotide Molecules Using a Membrane Channel.** *Proc Natl Acad Sci USA* 1996, **93(24):**13770-73.
12. Akeson M, Branton D, Kasianowicz JJ, Brandin E, Deamer DW: **Microsecond Time-Scale Discrimination Among Polycytidylic Acid, Polyadenylic Acid, and Polyuridylic Acid as Homopolymers or as Segments Within Single RNA Molecules.** *Biophys J* 1999, **77(6):**3227-3233.
13. Bezrukov SM: **Ion Channels as Molecular Coulter Counters to Probe Metabolite Transport.** *J Membr Biol* 2000, **174:**1-13.
14. Meller A, Nivon L, Brandin E, Golovchenko J, Branton D: **Rapid nanopore discrimination between single polynucleotide molecules.** *Proc Natl Acad Sci USA* 2000, **97(3):**1079-1084.
15. Meller A, Nivon L, Branton D: **Voltage-driven DNA translocations through a nanopore.** *Phys Rev Lett* 2001, **86(15):**3435-8.
16. Vercoutere W, Winters-Hilt S, Olsen H, Deamer DW, Haussler D, Akeson M: **Rapid discrimination among individual DNA hairpin molecules at single-nucleotide resolution using an ion channel.** *Nat Biotechnol* 2001, **19(3):**248-252.
17. Winters-Hilt S: **Highly Accurate Real-Time Classification of Channel-Captured DNA Termini.** *Third International Conference on Unsolved Problems of Noise and Fluctuations in Physics, Biology, and High Technology* 2003:355-368.
18. Vercoutere W, Winters-Hilt S, DeGuzman VS, Deamer D, Ridino S, Rogers JT, Olsen HE, Marziali A, Akeson M: **Discrimination Among Individual Watson-Crick Base-Pairs at the Termini of Single DNA Hairpin Molecules.** *Nucl Acids Res* 2003, **31:**1311-1318.
19. Winters-Hilt S: **Nanopore detection using channel current cheminformatics.** *SPIE Second International Symposium on Fluctuations and Noise* . 25–28 May, 2004
20. Winters-Hilt S, Akeson M: **Nanopore cheminformatics.** *DNA Cell Biol* 2004, **23(10):**675-83.
21. Shannon CE: **A mathematical theory of communication.** *Bell Sys Tech Journal* 1948, **27:**379-423. 623–656
22. Khinchine AI: *Mathematical foundations of information theory. Dover* 1957.
23. Amari S: **Dualistic Geometry of the Manifold of Higher-Order Neurons.** *Neural Networks* 1991, **4(4):**443-451.
24. Amari S: **Information Geometry of the EM and em Algorithms for Neural Networks.** *Neural Networks* 1995, **8(9):**1379-1408.
25. Amari S, Nagaoka H: **Methods of Information Geometry.** *Translations of Mathematical Monographs* 2000, **191:**.
26. Jaynes E: *Paradoxes of Probability Theory* 1997. Internet accessible book preprint: http://omega.albany.edu:8008/JaynesBook.html
27. Kapur JN, Kesavan HK: **Entropy optimization principles with applications.** *Academic Press*; 1992.
28. Kullback S: *Information Theory and Statistics. Dover* 1968.
29. Ben-Hur A, Horn D, Siegelmann HT, Vapnik V: **Support Vector Clustering.** *Journal of Machine Learning Research* 2001, **2:**125-137.
30. Scholkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC: **Estimating the Support of a High-Dimensional Distribution.** *Neural Comp* 2001, **13:**1443-1471.
31. Yang J, Estivill-Castro V, Chalup SK: **Support Vector Clustering Through Proximity Graph Modeling.** *Proceedings, 9th International Conference on Neural Information Processing (ICONIP'02)* 2002:898-903.
32. Freund Y, Schapire R: **A decision-theoretic generalization of on-line learning and an application to boosting.** *Journal of Computer and System Sciences* 1997, **55;:**119-139.
33. Freund Y, Schapire RE, Bartlett P, Lee WS: **Boosting the margin: a new explanation for the effectiveness of voting methods.** *Proc 14th International Conference on Machine Learning* 1998.
34. Schapire RE, Singer Y: **Improved Boosting Using Confiodence-Weighted Predictions.** *Machine Learning* 1999, **37(3):**297-336.
35. Diserbo M, Masson P, Gourmelon P, Caterini R: **Utility of the wavelet transform to analyze the stationarity of single ionic channel recordings.** *J Neurosci Methods* 2000, **99(1–2):**137-141.
36. Durbin R: **Biological sequence analysis: probabilistic models of proteins and nucleic acids.** *Cambridge, UK & New York: Cambridge University Press*; 1998.
37. Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK: **Improvements to Platt's SMO algorithm for SVM classifier design.** *Neural Computation* 2001, **13:**637-649.