TERMINATED RAMP - SVM
APPLYING A NON PARAMETRIC DATA-DRIVEN KERNEL
ON MICROARRAY AND PROTEOMIC DATA

Stefano Merler     Giuseppe Jurman

Third Bioinformatics Meeting on Machine Learning
for Microarray Studies of Disease
Genova, 15-16 September 2006
1 INTRODUCTION

2 THE MODEL

3 EXAMPLES

4 FEATURE SELECTION

5 CONCLUSIONS
OUTLINE

1. Introduction
2. The model
3. Examples
4. Feature selection
5. Conclusions
ADVANCES IN KERNEL-BASED METHODS

Classical kernel-based theories required:
- to choose learning rate $\eta$
  \[ P_{i+1} = P_i - \eta \nabla E(P_i) \]
- to configure initial state, i.e. set
  - the number of kernels
  - their positions
  - their (and also other free) parameters

Kernel-based algorithms are now formulated in terms of convex optimization problems.

CONVERGENCE TO THE GLOBAL MINIMUM IS GUARANTEED.
Advances in kernel-based methods

Classical kernel-based theories required:

- to choose learning rate $\eta$
  \[ P_{i+1} = P_i - \eta \nabla E(P_i) \]
- to configure initial state, i.e. set
  - the number of kernels
  - their positions
  - their (and also other free) parameters

Kernel-based algorithms are now formulated in terms of convex optimization problems.
Classical kernel-based theories required:

- to choose learning rate $\eta$
  
  $$P_{i+1} = P_i - \eta \nabla E(P_i)$$

- to configure initial state, i.e. set
  - the number of kernels
  - their positions
  - their (and also other free) parameters

Kernel-based algorithms are now formulated in terms of convex optimization problems.
It would be desirable to explore model selection methods that allow kernels to be chosen in a more automatic way based on data.

Three main tasks:

1. to automatize the learning process,
2. to avoid the selection of the optimal kernel for the task at hand,
3. to introduce nonlinearity without introducing additional parameters.

Points 2 and 3 are particularly important when dealing with small size datasets, like microarray and proteomic datasets.
It would be desirable to explore model selection methods that allow kernels to be chosen in a more automatic way based on data.

Three main tasks:

1. to automatize the learning process,
2. to avoid the selection of the optimal kernel for the task at hand,
3. to introduce nonlinearity without introducing additional parameters.

Points 2 and 3 are particularly important when dealing with small size datasets, like microarray and proteomic datasets.
**Vapnik, 1995**

The optimal hyperplane is the one that maximizes the distance between the hyperplane and the closest example.

In simple classification tasks this is the hyperplane passing through \( \frac{x^+ + x^-}{2} \) which has normal vector parallel to the vector \( v \) connecting \( x^+ \) and \( x^- \). The couple \( (x^+, x^-) \) is, along all the couples of oppositely labelled points, the one with minimum distance.

**Cristianini and Shawe-Taylor, 2000**

The optimal hyperplane is the one that maximizes the geometric margin of the couple of oppositely labelled points \( (x^+, x^-) \) with minimum functional margin.
The optimal hyperplane is the one that maximizes the distance between the hyperplane and the closest example.

The optimal hyperplane is the one that maximizes the geometric margin of the couple of oppositely labelled points \((x^+, x^-)\) with minimum functional margin.

In simple classification tasks this is the hyperplane passing through \(\frac{x^+ + x^-}{2}\) which has normal vector parallel to the vector \(v\) connecting \(x^+\) and \(x^-\). The couple \((x^+, x^-)\) is, along all the couples of oppositely labelled points, the one with minimum distance.
Map the data to a high dimensional (possibly infinite) feature space and then linearly separate the mapped data.

Drawbacks and issues:

1. the solution depends on the specific choice of the kernel $K(x, x')$.

Which is the “best” $K$ for the task at hand?

2. In general, the kernel depends on some parameter

\( (e.g. \ K(x, x') = e^{-\frac{||x - x'||^2}{\sigma}}, \ K(x, x') = 1 + (x, x')^d). \)

Is this practical when the number of training data is very limited?
Classical approach

Map the data to a high dimensional (possibly infinite) feature space and then linearly separate the mapped data.

Drawbacks and issues:

1. the solution depends on the specific choice of the kernel $K(x, x')$.

Which is the “best” $K$ for the task at hand?

2. In general, the kernel depends on some parameter

(e.g. $K(x, x') = e^{-\frac{||x-x'||^2}{\sigma}}$, $K(x, x') = 1 + \langle x, x' \rangle^d$).

Is this practical when the number of training data is very limited?
INTRODUCTION

THE MODEL

- The geometrical approach
- The RKHS approach

EXAMPLES

FEATURE SELECTION

CONCLUSIONS
FOCUSSING ON LOCAL MODELS

1. Identify the couples of points oppositely labelled lying close to the "real" separation surface,
2. Reconstruct locally the separation surface,
3. Linearly combine the local models to get a global one.
FOCUSING ON LOCAL MODELS

1. Identify the couples of points oppositely labelled lying close to the “real” separation surface,
2. Reconstruct locally the separation surface,
3. Linearly combine the local models to get a global one.
AN ALTERNATIVE STRATEGY

FOCUSSING ON LOCAL MODELS

1. identify the couples of points oppositely labelled lying close to the “real” separation surface,
2. reconstruct locally the separation surface,
3. linearly combine the local models to get a global one.
Let $D = \{p_i = (x_i, y_i)\}_{i=1,...,N}$ be the training set, with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.

Given two oppositely labelled points $(p_i, p_j)$, a terminated ramp function is defined as follows:

$$k_{ij}(x) = \sigma(\langle w_{ij}, x \rangle + b_{ij}),$$

where $\sigma : \mathbb{R} \rightarrow [-1, 1]$ is a squashing function and $(w_{ij}, b_{ij})$ is the maximal margin hyperplane separating $(p_i, p_j)$.

\[
\sigma(x) = \begin{cases} 
-1 & \text{if } x < -1 \\
 x & \text{if } -1 \leq x \leq 1 \\
 1 & \text{if } x > 1.
\end{cases}
\]

Squashing function $\sigma(x)$
**Terminated ramp**

Let $D = \{ p_i = (x_i, y_i) \}_{i=1,...,N}$ be the training set, with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. Given two oppositely labelled points $(p_i, p_j)$, a terminated ramp function is defined as follows:

$$k_{ij}(x) = \sigma(\langle w_{ij}, x \rangle + b_{ij}),$$

where $\sigma : \mathbb{R} \to [-1, 1]$ is a squashing function and $(w_{ij}, b_{ij})$ is the maximal margin hyperplane separating $(p_i, p_j)$.

$$\sigma(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \leq x \leq 1 \\ 1 & \text{if } x > 1. \end{cases}$$

Squashing function $\sigma(x)$
The resulting global model can thus be defined as follows

\[ F(x) = \sum_{i \in I^+} \sum_{j \in I^-} a_{ij} k_{ij}(x) + b = \sum_{t \in I^+ \times I^-} a_t k_t(x) + b , \]

where \( I^+ \) and \( I^- \) are the set of indices of the points belonging to class 1 and \(-1\) respectively.

Pros and cons:

- we only need to estimate the weights \( a_t \) and the offset \( b \),
- we have \( \approx \frac{N^2}{4} \) terminated ramp functions.

Basically, this is a two-layer network: we balance the loss of flexibility (the hidden units parameters are fixed) by considering a very large number of (sometimes redundant) hidden units.
The resulting global model can thus be defined as follows

$$ F(x) = \sum_{i \in I^+} \sum_{j \in I^-} a_{ij} k_{ij}(x) + b = \sum_{t \in I^+ \times I^-} a_t k_t(x) + b, $$

where $I^+$ and $I^-$ are the set of indices of the points belonging to class 1 and $-1$ respectively.

Pros and cons:
- we only need to estimate the weights $a_t$ and the offset $b$,
- we have $\approx \frac{N^2}{4}$ terminated ramp functions.

Basically, this is a two-layer network: we balance the lost of flexibility (the hidden units parameters are fixed) by considering a very large number of (sometimes redundant) hidden units.
The resulting global model can thus be defined as follows

$$F(x) = \sum_{i \in I^+} \sum_{j \in I^-} a_{ij} k_{ij}(x) + b = \sum_{t \in I^+ \times I^-} a_t k_t(x) + b,$$

where $I^+$ and $I^-$ are the set of indices of the points belonging to class 1 and $-1$ respectively.

Pros and cons:

- we only need to estimate the weights $a_t$ and the offset $b$,
- we have $\approx \frac{N^2}{4}$ terminated ramp functions.

Basically, this is a two-layer network: we balance the lost of flexibility (the hidden units parameters are fixed) by considering a very large number of (sometimes redundant) hidden units.
TWO FACTS ABOUT TWO-LAYER NETWORKS

\[ F(x) = \sum_{t=1}^{T} a_t \sigma(\langle w_t, x \rangle + b_t) + b, \]

where \( \sigma : \mathbb{R} \rightarrow [-1, 1] \) is an increasing function.

\[ \text{VC}(F) \approx T \log T \implies \text{poor generalization if } T \gg N \]

**Bartlett, 1998**

For any \( 0 \leq \gamma \leq 1 \), with probability \( 1 - \delta \) we have

\[ P[yF(x) \leq 0] \leq \epsilon_D^\gamma + \sqrt{\frac{c}{N} \left( \frac{A^2 d}{\gamma^2} \log \left( \frac{A}{\gamma} \right) \log^2 N + \log(1/\delta) \right)}, \]

where

\[ \epsilon_D^\gamma = \frac{1}{N} \sum_{i=1}^{N} I[y_i F(x_i) < \gamma] \text{ and } A = \sum_{t=1}^{T} |a_t|. \]

\[ \implies \text{generalization does not depend on } T. \]
TWO FACTS ABOUT TWO-LAYER NETWORKS

$$F(x) = \sum_{t=1}^{T} a_t \sigma(\langle w_t, x \rangle + b_t) + b,$$

where $\sigma : \mathbb{R} \rightarrow [-1, 1]$ is an increasing function.

$$\text{VC}(F) \approx T \log T \implies \text{poor generalization if } T \gg N$$

**Bartlett, 1998**

For any $0 \leq \gamma \leq 1$, with probability $1 - \delta$ we have

$$P[yF(x) \leq 0] \leq \epsilon_D^\gamma + \sqrt{\frac{c}{N} \left( \frac{A^2 d}{\gamma^2} \log \left( \frac{A}{\gamma} \right) \log^2 N + \log(1/\delta) \right)},$$

where

$$\epsilon_D^\gamma = \frac{1}{N} \sum_{i=1}^{N} I[y_i F(x_i) < \gamma] \text{ and } A = \sum_{t=1}^{T} |a_t|.$$

$\implies$ generalization does not depend on $T$. 

S. Merler, G. Jurman

Data-driven kernels
**TWO FACTS ABOUT TWO-LAYER NETWORKS**

\[ F(x) = \sum_{t=1}^{T} a_t \sigma(\langle w_t, x \rangle + b_t) + b, \]

where \( \sigma : \mathbb{R} \to [-1, 1] \) is an increasing function.

\[ \text{VC}(F) \approx T \log T \implies \text{poor generalization if } T \gg N \]

**Bartlett, 1998**

For any \( 0 \leq \gamma \leq 1 \), with probability \( 1 - \delta \) we have

\[ P[yF(x) \leq 0] \leq \epsilon_D^\gamma + \sqrt{\frac{c}{N} \left( \frac{A^2 d}{\gamma^2} \log \left( \frac{A}{\gamma} \right) \log^2 N + \log(1/\delta) \right)}, \]

where

\[ \epsilon_D^\gamma = \frac{1}{N} \sum_{i=1}^{N} I[y_i F(x_i) < \gamma] \quad \text{and} \quad A = \sum_{t=1}^{T} |a_t|. \]

\( \implies \) generalization does not depend on \( T \).
The optimal strategy is to minimize the following cost function:

\[ E(a, b) = \epsilon_D^\gamma + \frac{\|a\|_{\ell_2}}{\gamma}. \]

**Hard strategy**

To separate all the data we require

\[ y_i F(x_i) \geq \gamma^\star, \quad i = 1, \ldots, N \]

for some \( \gamma^\star > 0 \) (we set \( \gamma^\star = 1 \)). This requirement is equivalent to \( \epsilon_D^\star = 0 \).
The optimal strategy is to minimize the following cost function:

$$\mathcal{E}(a, b) = \epsilon_D^* + \frac{\|a\|_2^2}{\gamma}.$$  

**Hard strategy**

To separate all the data we require

$$y_i F(x_i) \geq \gamma^*, \quad i = 1, \ldots, N$$

for some $\gamma^* > 0$ (we set $\gamma^* = 1$).

This requirement is equivalent to $\epsilon_D^* = 0$.

$$\begin{aligned}
\text{minimize} & \quad \frac{1}{2} \sum_{t=1}^{T} a_t^2 \\
\text{subject to} & \quad y_i F(x_i) \geq 1, \quad i = 1, \ldots, N.
\end{aligned}$$
The solution is a Support Vector Machine

\[ F(x) = \sum_{i=1}^{N} \alpha_i y_i \langle x_i, x \rangle_K + b , \]

with kernel \( \langle x_i, x_j \rangle_K := \sum_{t=1}^{T} k_t(x_i) k_t(x_j) \) and the \( \alpha_i \) are the solution of

\[
\begin{align*}
\max_{\alpha \in \mathbb{R}^N} & -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle_K + \sum_{i=1}^{N} \alpha_i \\
\text{subject to} & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& \alpha_i \geq 0, \quad i = 1, \ldots, N.
\end{align*}
\]
SOFT (REGULARIZED) VERSION

\[
\begin{aligned}
\text{minimize} & \quad \frac{1}{2} \sum_{t=1}^{T} a_t^2 + \lambda \sum_{i=1}^{N} \xi_i \\
\text{subject to} & \quad y_i F(x_i) \geq 1 - \xi_i, \quad i = 1, \ldots, N \\
& \quad \xi_i \geq 0, \quad i = 1, \ldots, N.
\end{aligned}
\]

\[
\begin{aligned}
\text{max} & \quad -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle_k + \sum_{i=1}^{N} \alpha_i \\
\text{subject to} & \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& \quad 0 \leq \alpha_i \leq \lambda, \quad i = 1, \ldots, N.
\end{aligned}
\]
Kernel Shape and Sparseness

- Automatically determined by the training data
- No kernel parameter to optimize
- Able to follow the local behavior of the separation surface

\[ a_t = \sum_{i \in I_S} \alpha_i y_i k_t(x_i) \]

How many \( a_t \) are different from zero?
Kernel shape and sparseness

- Automatically determined by the training data
- No kernel parameter to optimize
- Able to follow the local behavior of the separation surface

\[ a_t = \sum_{i \in I_S} \alpha_i y_i k_t(x_i) \]

How many \( a_t \) are different from zero?
OUTLINE

1 INTRODUCTION

2 THE MODEL
   • The geometrical approach
   • The RKHS approach

3 EXAMPLES

4 FEATURE SELECTION

5 CONCLUSIONS
RKHS OF A FINITE SETS

Any given finite set of linearly independent functions

\[ G_0 = \{ k_1(x), \ldots, k_T(x) \} \]

gives rise to a family of kernels inducing a RKHS.

In fact, let

\[ G = \left\{ f(x) = \sum_{t=1}^{T} a_t k_t(x) \mid a_t \in \mathbb{R} \right\} \]

be the linear span of \( G_0 \). Any positive–definite \( T \times T \) matrix \( A \) induces a scalar product on \( G \) by setting

\[ \left\langle \sum_{t=1}^{T} a_t k_t(x), \sum_{s=1}^{T} a'_s k_s(x) \right\rangle = \sum_{s,t=1}^{T} a_s a'_t A_{st} \, . \]

\[ K(x, x') = \sum_{s,t=1}^{T} k_s(x) k_t(x') A_{st}^{-1} \]

is the reproducing kernel of the class \( G_0 \) for the matrix \( A \), defining the RKHS \( H_K \).
RKHS of a finite sets

Any given finite set of linearly independent functions

\[ G_0 = \{ k_1(x), \ldots, k_T(x) \} \]

gives rise to a family of kernels inducing a RKHS.

In fact, let

\[ G = \left\{ f(x) = \sum_{t=1}^{T} a_t k_t(x) \mid a_t \in \mathbb{R} \right\} \]

be the linear span of \( G_0 \). Any positive–definite \( T \times T \) matrix \( A \) induces a scalar product on \( G \) by setting

\[
\left\langle \sum_{t=1}^{T} a_t k_t(x), \sum_{s=1}^{T} a'_s k_s(x) \right\rangle = \sum_{s,t=1}^{T} a_s a'_t A_{st}.
\]

\[ K(x, x') = \sum_{s,t=1}^{T} k_s(x) k_t(x') A_{st}^{-1} \]

is the reproducing kernel of the class \( G_0 \) for the matrix \( A \), defining the RKHS \( H_K \).
Thus the geometrical problem translates in solving

\[
f_\lambda = \arg \min_{f \in H_K} \frac{1}{N} \sum_{i=1}^{N} V(y_i, f(x_i)) + \lambda \|f\|_{H_K}^2,
\]

where
- \(V(y, z)\) is a loss function,
- \(\|f\|_{H_K}\) is the norm of \(f\) in the RKHS \(H_K\),
- \(\lambda\) is the regularizer.

**Classical loss function**

Binary classification: hinge loss \(V(y, f(x)) = \max\{0, 1 - yf(x)\} \implies\) SVM.
Regression: \(L_2\) loss function \(V(y, f(x)) = (y - f(x))^2\).
Consider
\[
f_\lambda(x) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda J(f),
\]
where \( J \) is a suitable penalization term.

**Penalized regression in RKHS**

Suppose that
- \( f(x) = \sum_{t=1}^{T} \beta_t h_t(x) \), i.e. a linear combination of basis functions
- \( J(f) = \beta^T \Omega \beta \), for a suitable positive-definite block-diagonal \( \Omega \in \mathcal{M}(T, \mathbb{R}) \),
then we can define \( G_0 = \{h_1, \ldots, h_T\} \) and obtain the solution \( f_\lambda \) by minimizing the Tikhonov functional with kernel
\[
K(x, x') = \sum_{s,t=1}^{T} h_s(x) h_t(x') \Omega_{st}^{-1}.
\]

A standard choice is to consider the union of spline basis functions for each coordinate function as the set of basis functions.
Consider

\[ f_\lambda(x) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda J(f), \]

where \( J \) is a suitable penalization term.

### Penalized Regression in RKHS

Suppose that

- \( f(x) = \sum_{t=1}^{T} \beta_t h_t(x) \), i.e. a linear combination of basis functions
- \( J(f) = \beta^T \Omega \beta \), for a suitable positive-definite block-diagonal \( \Omega \in \mathcal{M}(T, \mathbb{R}) \),

then we can define \( G_0 = \{ h_1, \ldots, h_T \} \) and obtain the solution \( f_\lambda \) by minimizing the Tikhonov functional with kernel

\[ K(x, x') = \sum_{s,t=1}^{T} h_s(x) h_t(x') \Omega_{st}^{-1}. \]

A standard choice is to consider the union of spline basis functions for each coordinate function as the set of basis functions.
Analogously, the choice

\[ G_0 = \{1\} \cup \{k_{ij}(x) \mid 1 \leq i < j \leq N, y_i \neq y_j\}, \]

\[ k_{ij}(x) = \sigma_{ij}(\langle w_{ij}, x \rangle + b_{ij}). \]

gives the terminated ramp solution.

The adopted shape for \( \sigma_{ij} \) is the following piecewise-linear function:

\[ \sigma_{ij}(t) = \begin{cases} 
    y_{\text{min}} & \text{if } t < y_{\text{min}} \\
    t & \text{if } y_{\text{min}} \leq t \leq y_{\text{max}} \\
    y_{\text{max}} & \text{if } t > y_{\text{max}}
\end{cases}, \]

where \( y_{\text{max}} = \max\{y_i, y_j\} \) and \( y_{\text{min}} = \min\{y_i, y_j\}. \)
Analogously, the choice

\[ G_0 = \{1\} \cup \{k_{ij}(x) \mid 1 \leq i < j \leq N, \ y_i \neq y_j\} , \]

\[ k_{ij}(x) = \sigma_{ij}(\langle w_{ij}, x \rangle + b_{ij}) . \]

gives the terminated ramp solution.

The adopted shape for \( \sigma_{ij} \) is the following piecewise-linear function:

\[ \sigma_{ij}(t) = \begin{cases} 
  y_{\text{min}} & \text{if } t < y_{\text{min}} \\
  t & \text{if } y_{\text{min}} \leq t \leq y_{\text{max}} \\
  y_{\max} & \text{if } t > y_{\max} 
\end{cases} , \]

where \( y_{\max} = \max\{y_i, y_j\} \) and \( y_{\min} = \min\{y_i, y_j\} \).
Suppose $0 < \delta < 1$, then with probability at least $1 - \delta$ every data-driven kernel SVM has:

$$P[yf_\lambda(x) \leq 0] \leq \epsilon_D^0 + \sqrt{|G_0|^2 n_s^2 4N^3 \lambda^2} + \sqrt{9 \log 2/\delta} 2N,$$

for any finite set $G_0$ of functions bounded between -1 and 1.

We recall that for any function class $\mathcal{F}$, with probability at least $1 - \delta$ every $f \in \mathcal{F}$ satisfies

$$P[yf(x) \leq 0] \leq \epsilon_D^0 + \hat{R}_N(\mathcal{F}) + \sqrt{9 \log 2/\delta} 2N,$$

where $\hat{R}_N(\mathcal{F})$ is the empirical Rademacher complexity of the class $\mathcal{F}$. 

---

**Rademacher generalization error**

S. MERLER, G. JURMAN

DATA-DRIVEN KERNELS
Suppose $0 < \delta < 1$, then with probability at least $1 - \delta$ every data-driven kernel SVM has:

$$P[yf_\lambda(x) \leq 0] \leq \epsilon_0^D + \sqrt{\frac{|G_0|^2 n_s^2}{4N^3 \lambda^2}} + \sqrt{\frac{9 \log 2/\delta}{2N}},$$

for any finite set $G_0$ of functions bounded between -1 and 1.

We recall that for any function class $\mathcal{F}$, with probability at least $1 - \delta$ every $f \in \mathcal{F}$ satisfies

$$P[yf(x) \leq 0] \leq \epsilon_0^D + \hat{R}_N(\mathcal{F}) + \sqrt{\frac{9 \log 2/\delta}{2N}},$$

where $\hat{R}_N(\mathcal{F})$ is the empirical Rademacher complexity of the class $\mathcal{F}$.
For kernel methods, i.e. \( f(x) = \sum_{i=1}^{N} c_i k_{x_i}(x) \), the following bound holds:

\[
\hat{R}_N(F) \leq \frac{B}{N} \sqrt{\text{tr}(K)} = \frac{B}{N} \sqrt{\sum_{i=1}^{N} K(x_i, x_i)},
\]

for a suitable \( B \) satisfying \( c^T K c \leq B^2 \), where \( c = (c_1, \ldots, c_N) \) and \( K \) is the Gram matrix.

Suppose that \( \|k_t\|_\infty \leq 1 \) for each \( k_t \in G_0 \). Then the following upper bound \( B^2 \) holds:

\[
\sum_{i,j=1}^{N} c_i c_j K(x_i, x_j) = \frac{1}{4\lambda^2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j \sum_{l=1}^{T} k_l(x_i) k_l(x_j) \leq \frac{|G_0| n_S^2}{4N^2 \lambda^2} = B^2,
\]

where \( n_S \) is the number of Support Vectors.
For kernel methods, i.e. \( f(x) = \sum_{i=1}^{N} c_i k(x_i) \), the following bound holds:

\[
\hat{R}_N(\mathcal{F}) \leq \frac{B}{N} \sqrt{\text{tr}(K)} = \frac{B}{N} \sqrt{\sum_{i=1}^{N} K(x_i, x_i)},
\]

for a suitable \( B \) satisfying \( c^T K c \leq B^2 \), where \( c = (c_1, \ldots, c_N) \) and \( K \) is the Gram matrix.

Suppose that \( \|k_t\|_\infty \leq 1 \) for each \( k_t \in G_0 \). Then the following upper bound \( B^2 \) holds:

\[
\sum_{i,j=1}^{N} c_i c_j K(x_i, x_j) = \frac{1}{4\lambda^2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j \sum_{t=1}^{T} k_t(x_i) k_t(x_j) \leq \frac{|G_0| n_s^2}{4N^2 \lambda^2} = B^2,
\]

where \( n_s \) is the number of Support Vectors.
INTRODUCTION

THE MODEL

EXAMPLES
  - Classification
  - Regression
  - High-throughput data

FEATURE SELECTION

CONCLUSIONS
The box in \( \mathbb{R}^2 \), \( R \equiv [-10, 10] \times [-5, 5] \), is partitioned into two class regions \( R_1 \) (upper) and \( R_{-1} \) (lower) by means of the curve \( \Gamma \) of parametric equations:

\[
\Gamma \equiv \begin{cases} 
  x(t) = t & \text{if } t \leq 0 \\
  y(t) = 2 \sin(3t) & \text{if } t \geq 0 
\end{cases}
\]

A training set of 200 two-dimensional examples was generated by randomly sampling region \( R \), and labeled with either \(-1\) or \(1\) according to whether they belonged to \( R_{-1} \) or \( R_1 \). Additional 10000 points to be used as independent test set were generated on an evenly spaced grid and labeled according to the same rule.
4 sets of points were generated by sampling 4 2-dimensional Gaussian distributions, respectively centered in \((-1.0, 0.5), (0.0, -0.5), (0.0, 0.5)\) and \((1.0, -0.5)\). Covariance matrices were diagonal for all the 4 distributions; variance was constant and equal to 0.4. Points coming from the sampling of the first two Gaussians were labeled with class \(-1\); the others with class 1. This sampling schema was used to generate 200 points making up the training set. 10000 test points were generated on an evenly spaced grid and labeled according to the Bayes rule.
Training and test error

Training (dashed line) and test (solid line) error of Terminated Ramp kernel as a function of the regularizer on the sin (left) and gaussian (right) data sets (10-fold CV).

Training error: decreasing function
Test error: decreasing on the noiseless sin, showing the classical pattern of the regularizing methods on gaussian.
1. **Introduction**

2. **The Model**

3. **Examples**
   - Classification
   - Regression
   - High-throughput data

4. **Feature Selection**

5. **Conclusions**
A sinusoidal function

Approximation of $F(x) = \sin\left(\frac{\pi}{x}\right)$ via its relative minima and maxima.

$$D = \left\{ \left(\frac{2}{2k + 1}, (-1)^k\right) : 1 \leq k \leq 50, k \in \mathbb{N} \right\}$$

Dotted line: Gaussian kernel / Dashed line: Terminated ramp kernel

- Nearby zero performances are comparable
- Moving away from zero TR-K outperform G-K
Approximation of \( F(x) = 4.26 (e^{-|x|} - 4e^{-2|x|} + 3e^{-3|x|}) \), by using the training set

\[
D = \{(x_i, F(x_i) + \varepsilon): i = 1, \ldots, 150, x_i \in [-4, 4] \varepsilon \sim \mathcal{N}(0, \sigma^2)\}
\]

Dotted line: Gaussian kernel / Dashed line: Terminated ramp kernel

- If the training set is evenly spaced then G-K outperform TR-K
- If the training set is uniformly sampled then TR-K outperform G-K
INTRODUCTION
THE MODEL
EXAMPLES
FEATURE SELECTION
CONCLUSIONS

OUTLINE

1. INTRODUCTION
2. THE MODEL
3. EXAMPLES
   - Classification
   - Regression
   - High-throughput data
4. FEATURE SELECTION
5. CONCLUSIONS

S. Merler, G. Jurman
DATA-DRIVEN KERNELS
A comparison of performances of SVM with different kernels on some public microarray datasets, expressed in terms of error estimated by leave-one-out crossvalidation.

<table>
<thead>
<tr>
<th>data set</th>
<th>$N$</th>
<th>$d$</th>
<th>TR-SVM</th>
<th>G-SVM</th>
<th>LIN-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast Cancer</td>
<td>37</td>
<td>12625</td>
<td>29.7</td>
<td>35.1</td>
<td>35.1</td>
</tr>
<tr>
<td>Colon Cancer</td>
<td>62</td>
<td>2000</td>
<td>12.9</td>
<td>14.5</td>
<td>12.9</td>
</tr>
<tr>
<td>Glioma</td>
<td>50</td>
<td>12625</td>
<td>18.0</td>
<td>18.0</td>
<td>20.0</td>
</tr>
<tr>
<td>Hypertrophy</td>
<td>36</td>
<td>10043</td>
<td>16.7</td>
<td>22.2</td>
<td>22.2</td>
</tr>
<tr>
<td>Lymphoma</td>
<td>96</td>
<td>4026</td>
<td>2.1</td>
<td>3.1</td>
<td>3.1</td>
</tr>
<tr>
<td>Metastases</td>
<td>76</td>
<td>16063</td>
<td>13.2</td>
<td>14.5</td>
<td>11.8</td>
</tr>
</tbody>
</table>
**Feature Importance**

\[ h_j = \sum_{t=1}^{T} |a_t| \frac{w_t(j)}{\|w_t\|} \]

**Ranking Score**

\[ s_j = \frac{d - p_j}{d - 1}, \]

where \( p_j \) is the position of the \( j \)-th feature ranked according to \( h_j \).

Example on the ma-synth dataset.

The data sets consist of 100 samples and 5000 features (genes). The first 50 samples are assigned to class 1, the others to class -1. Only the first 50 genes represent discriminant features for the induced binary problem. All expression values are generated as standard normally distributed numbers. Genes 1-50 in samples 1-50 have mean 1, genes 1-50 in samples 51-100 have mean 2. Initially, genes 51-5000 in all the samples have mean 0. Then three substitutions are performed, where a \( p\% \) of all genes from the \( a \)-th to the \( b \)-th are replaced by normally distributed numbers with mean \( m \), namely:

1. \( p = 40, a = 51, b = 100 \) and \( m = 2 \);
2. \( p = 50, a = 101, b = 200 \) and \( m = 1 \);
3. \( p = 70, a = 201, b = 300 \) and \( m = 0.5 \).
A suite of experiments have been conducted on high-throughput data in order to assess performances of different combinations of \(\star\)-SVM/\(\star\)-Recursive Feature Elimination (RFE) algorithms, in a complete validation setup.

In particular, we tested linear SVM versus terminated ramp SVM and 1-RFE (the non-recursive version) versus Entropy-RFE (E-RFE). Results are evaluated in terms of average test error and ranked lists distribution stability and saturation, two measures of disarray among lists.

**Main results**

- Although applying E-RFE results in a better accuracy, performances of 1-RFE are comparable or slightly worse, but this ranking methods produces more stable lists with a considerably lower computational cost;

- Terminated Ramp SVM are comparable (in terms of confidence intervals) or slightly worse than linear SVM for accuracy, but they perform consistently better in terms of list stability and saturating number.
This dataset was produced at Keck Laboratory at Yale by using a Micromass MALDI-L/R instrument. The data set includes 77 controls and 93 ovarian cancers. All spectra were preprocessed according to the technique described in [Barla et al., 2006], thus obtaining, for each spectrum, 123 peaks as describing features. Results are reported for both the original and the standardized (mean zero and standard deviation one) dataset.
IMPLEMENTATION-RELATED ISSUES

The algorithm as described is inefficient: computational cost ranges from $O(N^3)$ to $O(N^5)$.

However, in the case of microarray it is more efficient than SVMs: for a problem with $N$ samples and $d$ features, TR-SVM are projected into linear SVMs in $\mathbb{R}^{N^2/4}$, SVMs work in $\mathbb{R}^d$ and, in general, $\frac{N^2}{4} < d$.

Nevertheless, a pruning strategy based on $k$-NN can be employed to reduce the computational cost down to $O(N^4k)$.

WHY A DATA-DRIVEN KERNEL?

1. we do not have to choose any kernel
2. we do not have to optimize any kernel parameter
3. the solution is, in general, very sparse
4. accuracy comparable with linear/gaussian SVM
5. good results in terms of list stability and saturation (TR)