Support vector machines and kernel-based methods

Louis Wehenkel & Pierre Geurts

Institut Montefiore, University of Liège, Belgium

ELEN062-1
Introduction to Machine Learning
September 1, 2020

Outline

Outline

Linear classification model

- Given a $LS = \{(x_k, y_k)\}_{k=1}^{N}$, where $y_k \in \{-1, 1\}$, and $x_k \in \mathbb{R}^n$.
- Find a classifier in the form

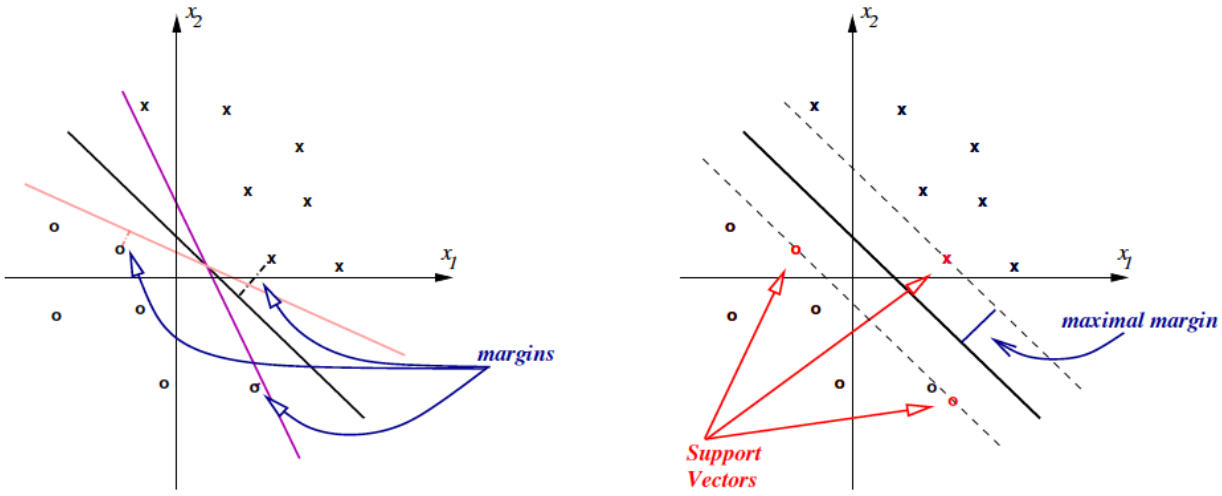
$$\hat{y}(x) = \text{sgn}(w^T x + b),$$

which classifies the LS correctly, i.e. that minimizes

$$\sum_{k=1}^{N} 1(y_k \neq \hat{y}(x_k))$$

- Several methods to find one such classifier: perceptron, linear discriminant analysis, naive bayes...

Maximal margin hyperplane



- When the data is linearly separable in the feature space, the separating hyperplane is not unique
- SVM maximizes the distance from the hyperplane to the nearest points in LS, i.e.

$$\max_{w,b} \min\{||x - x_k|| : w^T x + b = 0, k = 1, \ldots, N\}.$$

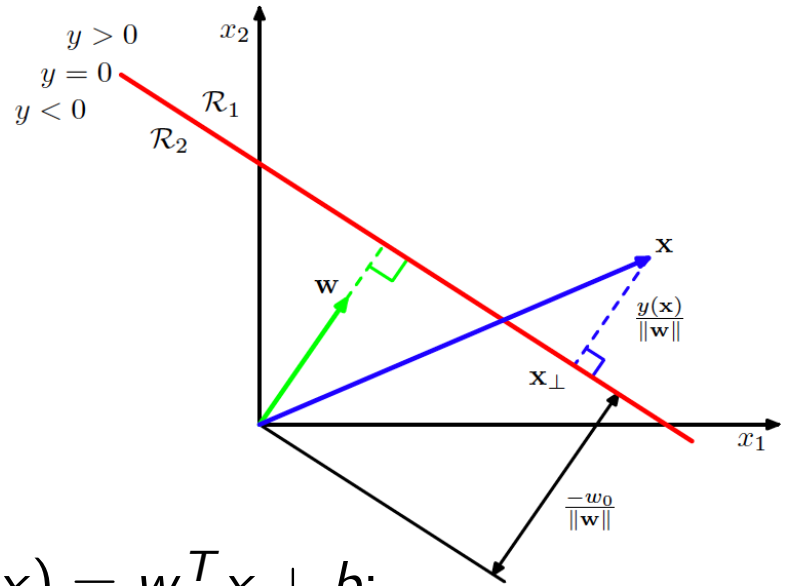
Why maximizing the margin?

- Intuitively, it feels safest
- There exist theoretical bounds on the generalization error that depend on the margin

$$\text{Err}(TS) < O(1/\gamma),$$

  where γ is the margin. But these bounds are often loose.
- It works very well in practice.
- It yields a convex optimization problem whose solution can be written in terms of dot-products only. But this is the case with other criterion as well.

Some geometry



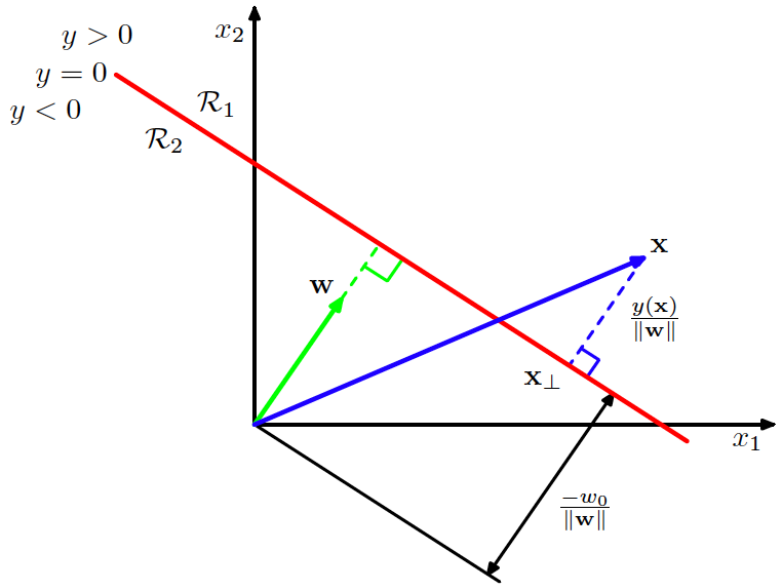
- w is perpendicular to the line $y(x) = w^T x + b$:

$$y(x_a) = 0 = y(x_b) \Rightarrow w^T(x_A - x_B) = 0$$

- Let x such that $y(x) = 0$. The distance from the origin to the line is:

$$||x||cos(w, x) = ||x|| \frac{w^T x}{||w||||x||} = \frac{w^T x}{||w||} = \frac{-b}{||w||}$$

Some geometry



- Any point x can be written as:

$$x = x_\perp + r\frac{w}{||w||},$$

  where $|r|$ is the distance from x to the line.
- Multiplying both sides by w^T and adding b, one gets:

$$w^T x + b = w^T x_\perp + b + r\frac{w^T w}{||w||} = 0 + r \cdot ||w|| \Rightarrow r = \frac{y(x)}{||w||}$$

Optimisation problem formulation

- The optimisation problem can be written:

$$\arg\max_{w,b} \left\{ \frac{1}{||w||} \min_n [y_n \cdot (w^T x_n + b)] \right\}.$$

- The solution is not unique as the hyperplane is unchanged if we multiply w and b by a constant $c > 0$.
- To impose unicity, one typically chooses $|w^T x + b| = 1$ for the point x that is closest to the surface (support vector).
- The problem is then equivalent to maximizing $\frac{1}{||w||}$ (or minimizing $||w||$) with the constraints:

$$y_k(w^T x_k + b) \geqslant 1, \forall k = 1, \ldots, N.$$

(Show that $|w^T x_k + b| = 1$ for at least two points x_k at the solution)

Optimisation problem formulation

The SVM problem is equivalent to

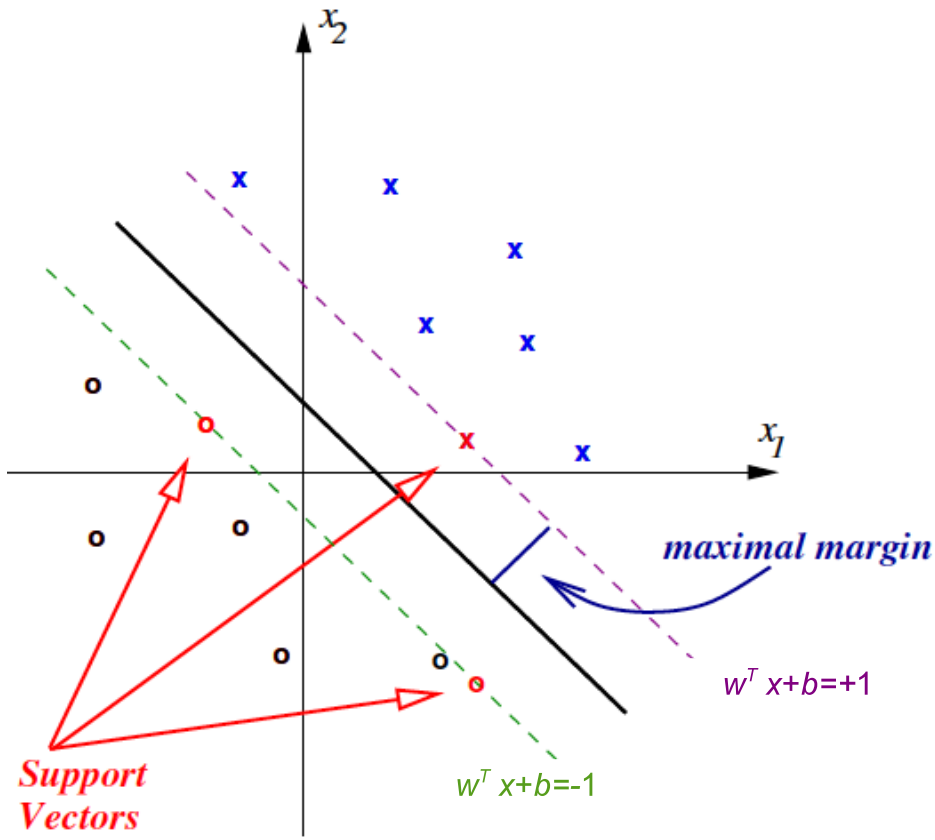
$$\min_{w,b} \mathcal{E}(w,b) = \frac{1}{2}||w||^2$$

subject to the N inequality constraints

$$y_k(w^T x_k + b) \geqslant 1, \forall k = 1, \ldots, N.$$

NB

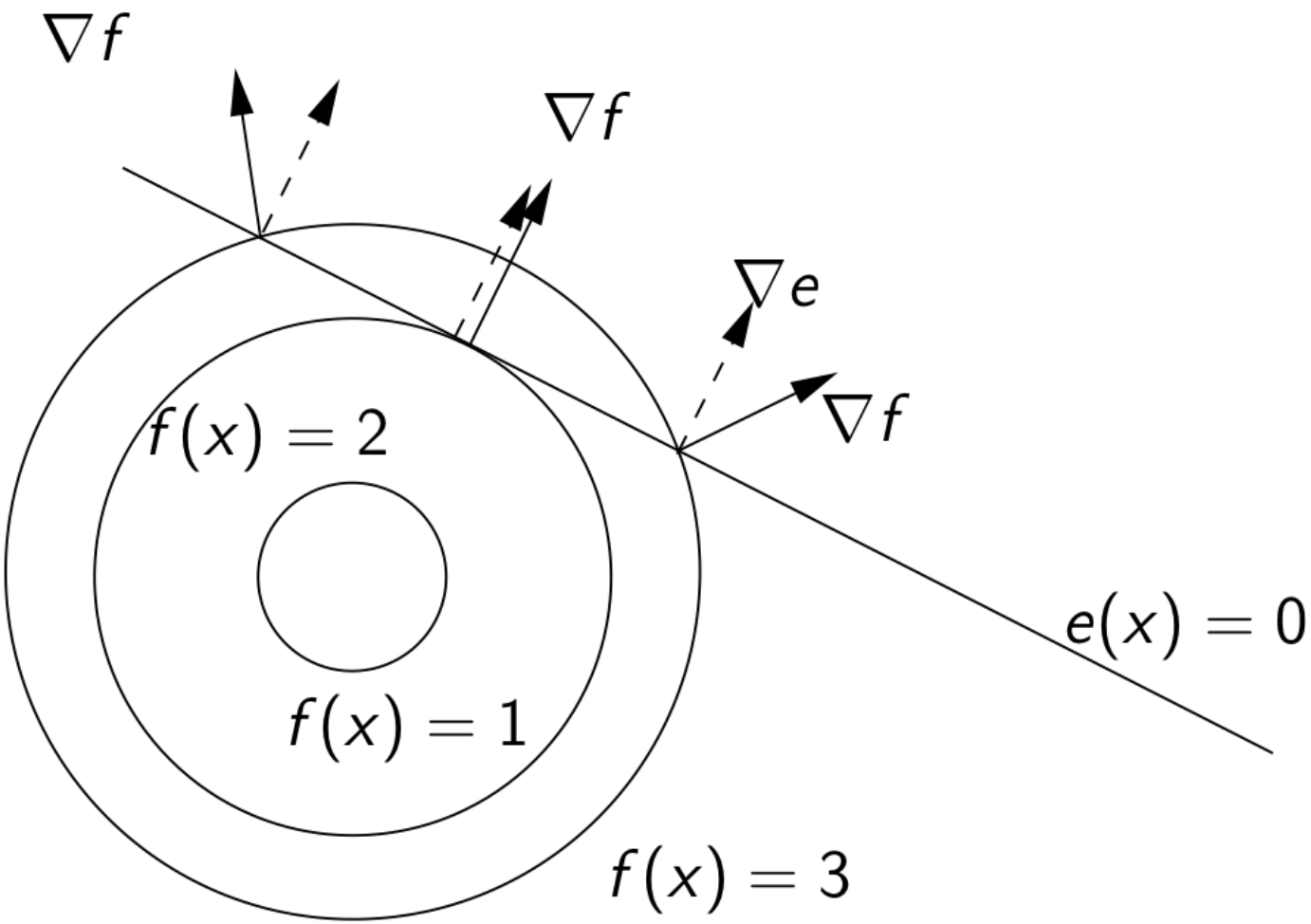
- $||w|| \rightarrow \frac{1}{2}||w||^2$ for mathematical convenience
- This is a quadratic programming problem
- A solution exists only when the data is linearly separable



A brief introduction to constrained optimisation

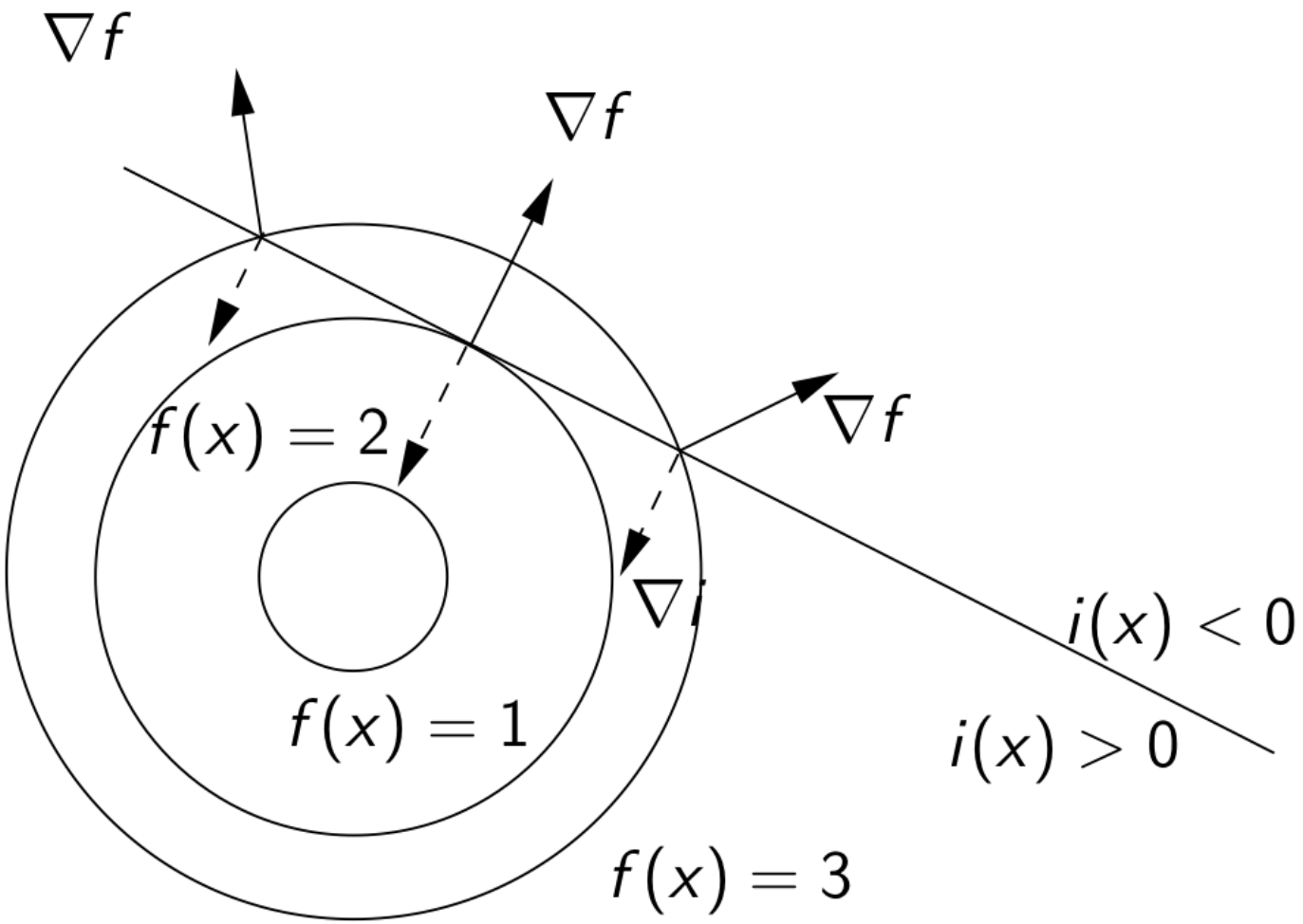
- Equality constrained optimisation problem
  - Minimise (or maximise) $f(x) \in \mathbb{R}$, with $x \in \mathbb{R}^n$
  - subject to p equality constraints $e_i(x) = 0, i = 1, \ldots, p$
- Inequality constrained optimisation problem
  - Minimise (or maximise) $f(x) \in \mathbb{R}$, with $x \in \mathbb{R}^n$
  - subject to p equality constraints $e_i(x) = 0, i = 1, \ldots, p$
  - subject to r inequality constraints $i_j(x) \leqslant 0, j = 1, \ldots, r$
- Feasibillity: existence of at least one x satisfying all equality and inequality constraints

Pictorial view: equality constraints



At the optimum, $\nabla f(x) + \alpha \nabla e(x) = 0$ and $e(x) = 0$.

Pictorial view: active inequality constraint



At the optimum, $\nabla f(x) + \alpha \nabla i(x) = 0$, $i(x) = 0$ and $\alpha > 0$.

Karush-Kuhn-Tucker conditions

- To minimise $f(x) \in \mathbb{R}$, with $x \in \mathbb{R}^n$, subject to equality constraints $e_i(x) = 0, i = 1, \ldots, p$ and inequality constraints $i_j(x) \leqslant 0, j = 1, \ldots, r$, define the Lagrangian

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \alpha^T e(x) + \beta^T i(x) \qquad (\alpha \in \mathbb{R}^p, \beta \in \mathbb{R}^r).$$

- At the optimum, there must exist some $\alpha \in \mathbb{R}^p$ and $\beta \in \mathbb{R}^r$ such that the following conditions hold:

$$\begin{array}{lll}
\nabla_x \mathcal{L} = 0 & \rightarrow & \nabla_x f(x) + \alpha^T \nabla_x e(x) + \beta^T \nabla_x i(x) = 0 \\
\nabla_\alpha \mathcal{L} = 0 & \rightarrow & e_i(x) = 0, \qquad \forall i = 1, \ldots, p \\
\nabla_\beta \mathcal{L} \leqslant 0 & \rightarrow & i_j(x) \leqslant 0, \qquad \forall j = 1, \ldots, r \\
 & & \beta_j i_j(x) = 0, \text{(complementary slackness conditions)} \\
 & & \beta_j \geqslant 0.
\end{array}$$

Convex optimization problems

If the optimization problem is convex (ie. f and $i_j, j = 1, \ldots, r$ are convex and $e_i, i = 1, \ldots, p$ are affine (linear) functions) and some conditions on the constraints are satisfied, we have (roughly):

- KKT conditions are necessary and sufficient conditions for optimality $\Rightarrow$ in some cases, an optimum x^* may be obtained analytically from these conditions
- Let $\mathcal{W}(\alpha, \beta) = \min_x \mathcal{L}(x, \alpha, \beta)$ and let α^* and β^* be the solution of the (Lagrange) dual problem:

$$\begin{aligned} &\max_{\alpha,\beta} \mathcal{W}(\alpha, \beta) \\ &\text{subject to } \beta_j \geqslant 0, \forall j = 1, \ldots, r, \end{aligned}$$

  then the minimizer x^* of $\mathcal{L}(x, \alpha^*, \beta^*)$ is the solution of the original optimization problem.

Back to the optimisation problem formulation

- The SVM optimization problem is formulated as:

$$\min_{w,b} \mathcal{E}(w,b) = \frac{1}{2}||w||^2$$
subject to the N inequality constraints
$$y_k(w^T x_k + b) \geqslant 1, \forall k = 1, \ldots, N.$$

- Let $\alpha_k \geqslant 0, k = 1, \ldots, N$ and let us construct the Lagrangian

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{k=1}^{N} \alpha_k (y_k(w^T x_k + b) - 1)$$

This function must be minimized w.r.t. w and b, and maximized w.r.t. α.

Lagrangian equations

By deriving the Lagrangian

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{k=1}^{N} \alpha_k (y_k (w^T x_k + b) - 1)$$

according to the primal variables w and b, we obtain the following optimality conditions:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial_w} = 0 & \rightarrow & w = \sum_{j=1}^{N} \alpha_j y_j x_j \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum_{k=1}^{N} \alpha_k y_k = 0 \end{cases}$$

(with $\alpha_k \geqslant 0$).

Dual problem

Substituting in the Lagrangian

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{k=1}^{N} \alpha_k (y_k (w^T x_k + b) - 1)$$

the expressions $w = \sum_{j=1}^{N} \alpha_j y_j x_j$ and $\sum_{k=1}^{N} \alpha_k y_k = 0$ yields the dual (quadratic) maximisation problem

$$\max_\alpha \mathcal{W}(\alpha) = \sum_{k=1}^{N} \alpha_k - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to the N inequality constraints

$$\alpha_k \geqslant 0, \forall k = 1, \ldots, N.$$

and one equality constraint

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Support vectors

Back to the primal problem:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{k=1}^{N} \alpha_k (y_k (w^T x_k + b) - 1)$$

According to the KKT complementary conditions, the solution vector w is such that:

$$\alpha_k (y_k (w^T x_k + b) - 1) = 0, \forall k = 1, \ldots, N$$

- $\alpha_k = 0$ if the constraint is satisfied as a strict inequality $y_k (w^T x_k + b) > 1$ because this is the way to maximize $\mathcal{L}$
- $\alpha_k > 0$ if the constraint is satisfied as an equality $y_k (w^T x_k + b) = 1$, in which case x_k is a support vector

Final model

Once the optimal values of α have been determined, the final model may be written as

$$\hat{y}(x) = \text{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i x_i^T x + b\right),$$

where the α_k values that are different from zero (i.e. strictly positive) are corresponding to the support vectors.

NB: b is computed by exploiting the fact that for any $\alpha_k > 0$ we have necessarily $y_k(w^T x_k + b) - 1 = 0$.

Leave-one-out bound

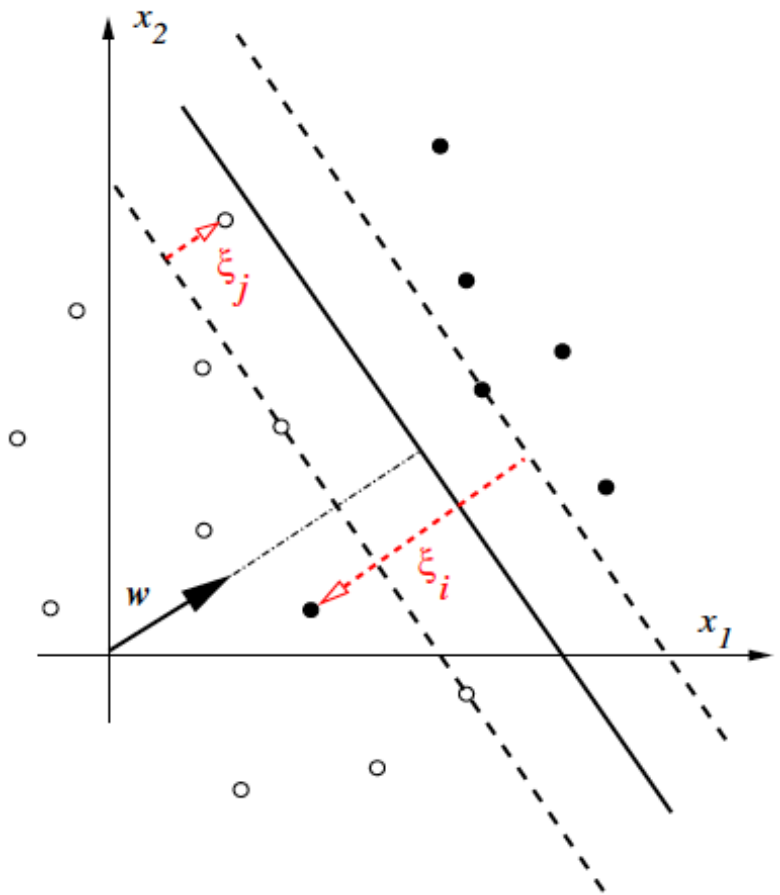
- Having a small set of support vectors is computationally efficient as only those vectors x_k and their weights α_k and class labels y_k need to be stored for classifying new examples

$$f(x) = \text{sgn}\left(\sum_{i|\alpha_i>0} y_i \alpha_i x_i^T x + b\right),$$

- Moreover, if a non-support vector x' is removed from the learning sample or moved around freely (outside the margin region), the solution would be unchanged.
- The proportion of support vectors in the learning sample gives a bound on the leave-one-out error:

$$\text{Err}_{\text{loo}} \leqslant \frac{|\{k|\alpha_k > 0\}|}{N}$$

Soft margin



- Due to noise or outliers, the samples may not be linearly separable in the feature space
- Discrepancies with respect to the margin is measured by slack variables $\xi_i \geqslant 0$ with the associated relaxed constraints $y_i(w^T x_i + b) \geqslant 1 - \xi_i$
- By making ξ_i large enough, the constraints can always be met
  - if $0 < \xi_i < 1$ the margin is not satisfied but x_i is still correctly classified
  - if $\xi_i > 1$ then x_i is misclassified

1-Norm soft margin optimization problem

Primal problem:

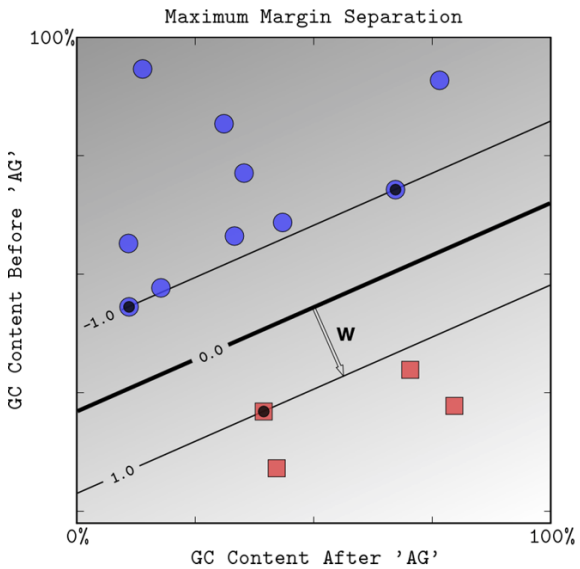
$$\min_{w,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i$$
subject to
$$y_i(w^T x_i + b) \geqslant 1 - \xi_i, \xi_i \geqslant 0, \forall i = 1, \ldots, N$$

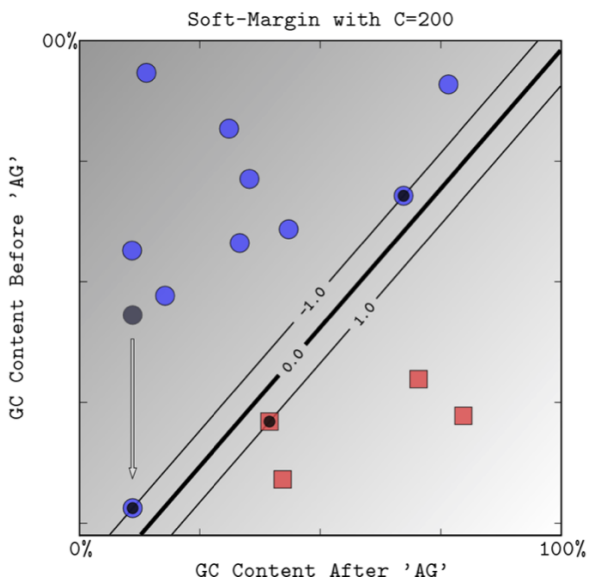
where C is a positive constant balancing the objective of maximizing the margin and minimizing the margin error

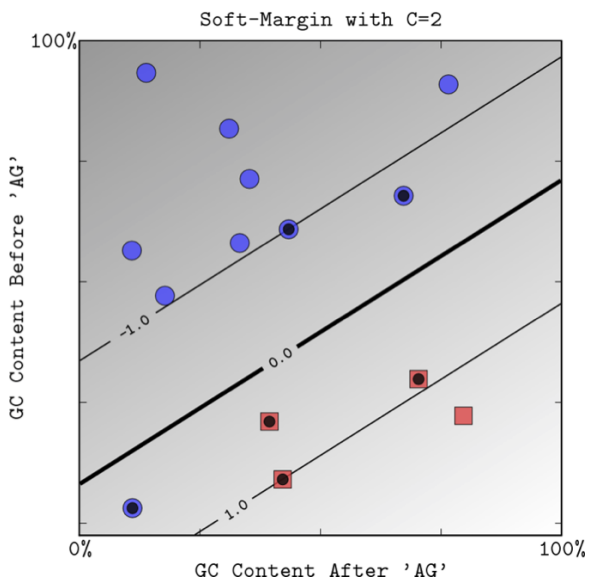
Dual problem:

$$\max_{\alpha} \mathcal{W}(\alpha) = \sum_{k=1}^{N}\alpha_k - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j$$
subject to
$$0 \leqslant \alpha_k \leqslant C, \forall k = 1, \ldots, N. \qquad \textit{(box constraints)}$$
$$\sum_{i=1}^{N}\alpha_i y_i = 0$$

Effect of C


Maximum Margin Separation
100%
0%
100%
GC Content Before 'AG'
GC Content After 'AG'
-1.0
0.0
1.0
w


Soft-Margin with C=200
00%
0%
100%
GC Content Before 'AG'
GC Content After 'AG'
-1.0
0.0
1.0


Soft-Margin with C=2
100%
0%
100%
GC Content Before 'AG'
GC Content After 'AG'
-1.0
0.0
1.0

Outline

Dot products

The quadratic optimization problem (hard and soft margin):

$$\max_{\alpha} \mathcal{W}(\alpha) = \sum_{k=1}^{N} \alpha_k - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

and the decision function:

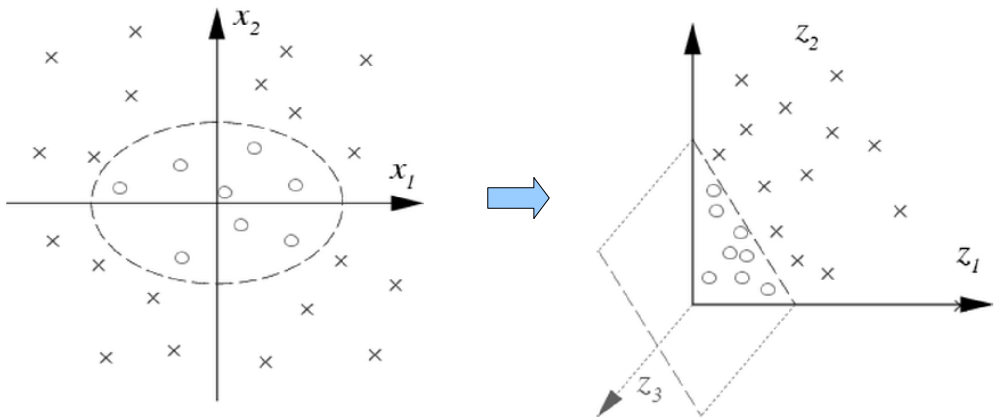
$$\hat{y}(x) = \text{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i x_i^T x + b \right)$$

make use of the learning samples x_i only through dot products.

Moreover the number of parameters to be estimated only depends on the number of training samples and is thus independent of the dimension of the input space (number of attributes).

Non-linear support vector machines

- The training samples may not be linearly separable in the input space (the above optimization problem does not have a solution)
- Consider a non-linear mapping ϕ to a new feature space
  $\phi(x) = [z_1, z_2, z_3]^T = [x_1^2, x_2^2, \sqrt{2}x_1x_2]^T$



- The dual problem becomes
  $\max_\alpha \mathcal{W}(\alpha) = \sum_{k=1}^{N} \alpha_k - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$

The kernel trick

- Instead of explicitly defining a mapping ϕ, we can directly specify the dot product $\phi^T(x)\phi(x')$ and leave the mapping implicit
- It is possible to characterise mathematically the functions $K(x, x')$ defined on pairs of objects that correspond to the dot product $\phi^T(x)\phi(x')$ for some implicit mapping ϕ (see next slides)
  $\Rightarrow$ Such function k is called a (positive or Mercer) kernel
- It can be thought of as a similarity measure between x and x'
- The kernel trick: Any learning algorithm that uses the data only via dot products can rely on this implicit mapping by replacing $x^T x'$ with $K(x, x')$

Kernel-based SVM classification

Assuming that we use a kernel $K(x, x')$ corresponding to the vectorisation map $\phi(x)$, we obtain straightforwardly

$$\hat{y}(x) = \text{sgn}\left(\sum_{k=1}^{N} y_k \alpha_k \phi(x_k)^T \phi(x) + b\right) = \text{sgn}\left(\sum_{k=1}^{N} y_k \alpha_k K(x, x_k) + b\right),$$

where the α_k can be determined by solving the following quadatric maximisation problem:

$$\max_\alpha \mathcal{W}(\alpha) = \sum_{k=1}^{N} \alpha_k - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to the N inequality constraints

$$\alpha_k \geqslant 0, \forall k = 1, \ldots, N.$$

and one equality constraint

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Mathematical notion of Kernel (with values in $\mathbb{R}$)

- Let U be a nonempty set of objects, then a function $K(\cdot, \cdot)$,

$$K(\cdot, \cdot) : U \times U \mapsto \mathbb{R}$$

such that for all $N \in \mathbb{N}$ and all $o_1, \ldots, o_N \in U$ the $N \times N$ matrix

$$K : K_{i,j} = K(o_i, o_j)$$

is symmetric and positive (semi)definite, is called a positive kernel.

- Example: let $\phi(\cdot)$ be a function defined on U with values in $\mathbb{R}^m$ (for some fixed m). Then

$$K_\phi(o, o') = \phi^T(o)\phi(o')$$

is a positive kernel. NB: $\phi(o)$ is a vector representation of o.

Mathematical notion of Kernel (with values in $\mathbb{R}$)

- The general result is as follows:

> *For any positive kernel K defined on U, there exists a scalar product space $\mathcal{V}$ and a function $\phi(\cdot) : U \mapsto \mathcal{V}$, such that*
>
> $$K(o, o') = \phi(o) \times \phi(o'),$$
>
> *where the operator $\times$ denotes the scalar product in $\mathcal{V}$.*

- In general, the space $\mathcal{V}$ is not necessarily of finite dimension.
- The kernel defines a "scalar product", hence a "norm" and a "distance" measure over U, which are inherited from $\mathcal{V}$:

$$\begin{aligned} d^2_U(o, o') &= d^2_{\mathcal{V}}(\phi(o), \phi(o')) = (\phi(o) - \phi(o'))^T(\phi(o) - \phi(o')) \\ &= \phi(o) \times \phi(o) + \phi(o') \times \phi(o') - 2\phi(o) \times \phi(o') \\ &= K(o, o) + K(o', o') - 2K(o, o'). \end{aligned}$$

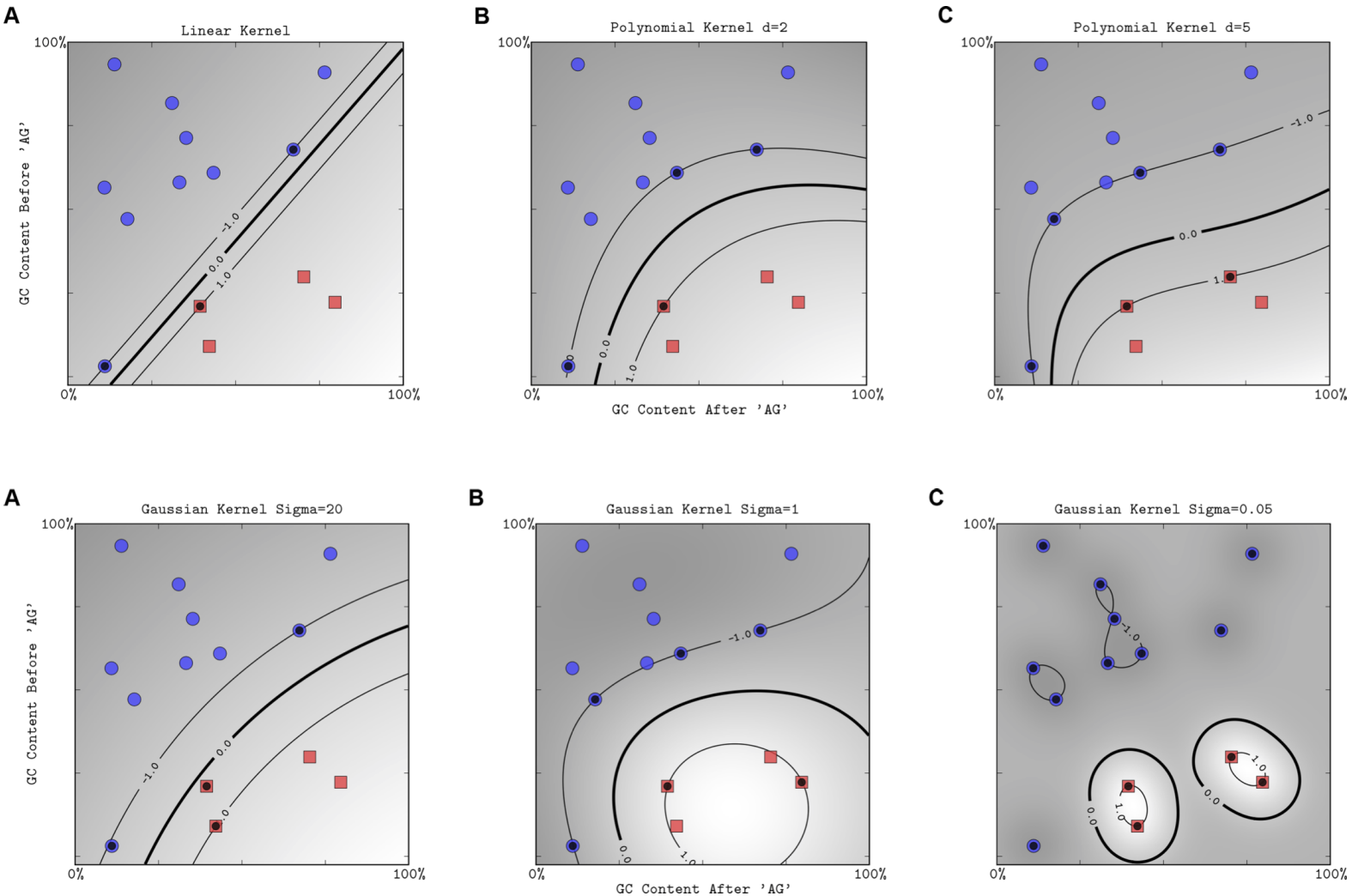
Examples of Kernels

- A trivial kernel: the constant kernel
  - $K(o, o') \equiv 1$
- Linear kernel defined on numerical attributes
  - $K(o, o') = \boldsymbol{a}^T(o)\boldsymbol{a}(o')$
- Hamming kernel for discrete attributes:
  - $K(o, o') = \sum_{i=1}^{m} \delta_{a_i(o), a_i(o')}$
- Text kernel
  - number of common substrings in o and o'
    (infinite dimension if text size is not a priori bounded)
- Combinations of kernels:
  - The sum of several (positive) kernels is also a (positive) kernel
  - The product of several kernels is also a kernel
  - Polynomial kernels: $\sum_{i=0}^{n} a_i (K(x, x'))^i$ if $\forall i : a_i \geqslant 0$

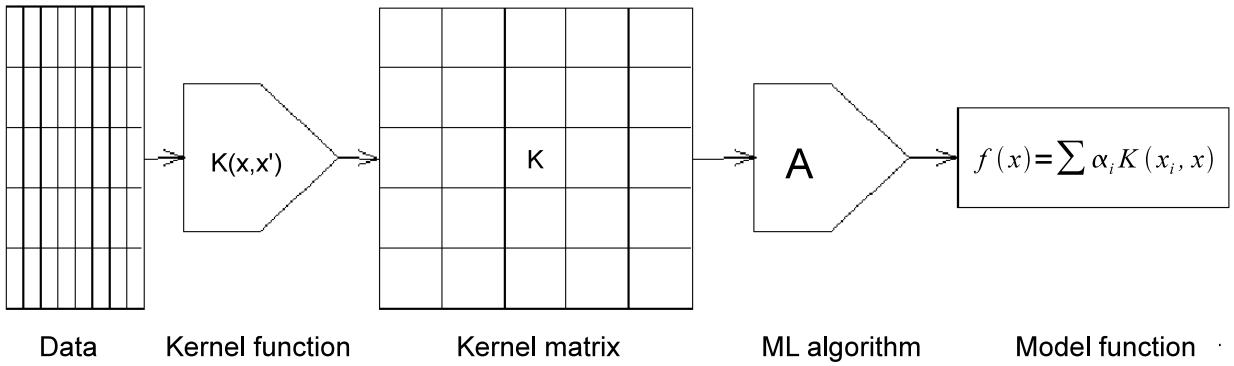
Examples of Kernels (for numerical attributes)

- Consider the kernel $K(x, x') = (x^T x')^2$.
  - We have in dimension 2, posing $x = (x_1, x_2)$ and $x' = (x'_1, x'_2)$:
  - $x^T x' = x_1 x'_1 + x_2 x'_2$, and thus
  - $(x^T x')^2 = (x_1 x'_1 + x_2 x'_2)^2 = x_1^2 {x'_1}^2 + 2 x_1 x_2 x'_1 x'_2 + {x_2}^2 {x'_2}^2$.
  - Hence, posing $\phi(x) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T$
  - we have $(x^T x')^2 = \phi(x)^T \phi(x')$,
  - $K(x, x')$ corresponds to the scalar product in the space of degree 2 products of the original features.
- In general $K(x, x') = (x^T x')^d$ corresponds to the scalar product in the space of all degree d product features.
- Another useful kernel
  - Gaussian kernel: $K(x, x') = \exp\left(\frac{-(x-x')^T (x-x')}{2\sigma^2}\right)$

Effect of the kernel



Kernel methods



- Modular approach: decouple the algorithm from the representation
  - Many algorithms can be "kernelized": ridge regression, fischer discriminant, PCA, k-means, etc.
  - Kernels have been defined for many data types: time series, images, graphs, sequences
- Main interests of kernel methods:
  - We can work in very high dimensional spaces (potentially infinite) efficiently as soon as the inner product is cheap to compute
  - Can apply classical algorithms on general, non-vectorial, data types (e.g. to build a linear model on graphs)

Outline

Ridge regression problem

- Let us assume that we are given a learning set

$$LS = \{(x_k, y_k)\}_{k=1}^{N},$$

and have chosen a kernel $K(x, x')$ defined on the input space, corresponding to a vectorisation mapping

$$\phi(x) \in \mathbb{R}^n \quad \text{i.e.} \quad \phi(x)^T \phi(x') = K(x, x').$$

- Least squares ridge regression consists in building a regression model in the form $\hat{y}(x) = w^T \phi(x) + b$, by minimising w.r.t. $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ the error function ($\gamma > 0$):

$$ERR(w, b) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{k=1}^{N} \Big(y_k - (w^T \phi(x_k) + b) \Big)^2.$$

Equality-constrained formulation of ridge regression

- We can restate the unconstrained minimisation problem

$$\min_{w,b} ERR(w,b) = \frac{1}{2} w^T w + \frac{1}{2}\gamma \sum_{k=1}^{N} \Big(y_k - (w^T \phi(x_k) + b) \Big)^2,$$

as an (equality) constrained minimisation problem, namely

$$\min_{w,b,e} \mathcal{E}(w,b,e) = \tfrac{1}{2} w^T w + \tfrac{1}{2}\gamma \textstyle\sum_{k=1}^{N} {e_k}^2$$

subject to N equality constraints

$$y_k - (w^T \phi(x_k) + b) = e_k, k = 1, \ldots, N$$

Lagrangian formulation of kernel-based ridge regression

The optimisation problem to be solved:

$$\min_{w,b,e} \mathcal{E}(w,b,e) = \tfrac{1}{2} w^T w + \tfrac{1}{2}\gamma \sum_{k=1}^{N} e_k^2$$

subject to the N equality constraints

$$y_k = w^T \phi(x_k) + b + e_k, k = 1, \ldots, N.$$

This problem can be solved by defining the Lagrangian

$$\mathcal{L}(w,b,e;\alpha) = \frac{1}{2} w^T w + \frac{1}{2}\gamma \sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N} \alpha_k \Big(w^T \phi(x_k) + b + e_k - y_k \Big)$$

and stating the optimality conditions

$$\frac{\partial \mathcal{L}}{\partial_w} = 0; \frac{\partial \mathcal{L}}{\partial b} = 0; \frac{\partial \mathcal{L}}{\partial e_k} = 0; \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0.$$

Lagrangian equations

In other words, we have

$$\mathcal{L}(w, b, e; \alpha) = \frac{1}{2}||w||^2 + \frac{1}{2}\gamma \sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N} \alpha_k \Big(w^T \phi(x_k) + b + e_k - y_k \Big)$$

yielding the optimality conditions

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial_w} = 0 & \rightarrow \quad w = \sum_{j=1}^{N} \alpha_j \phi(x_j) & \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow \quad \sum_{k=1}^{N} \alpha_k = 0 & \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 & \rightarrow \quad \alpha_k = \gamma e_k, & \forall k = 1, \ldots, N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 & \rightarrow \quad w^T \phi(x_k) + b + e_k - y_k = 0, & \forall k = 1, \ldots, N \end{cases}$$

Lagrangian equations

Solution:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial_w} = 0 & \rightarrow & w = \sum_{j=1}^{N} \alpha_j \phi(x_j) & \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum_{k=1}^{N} \alpha_k = 0 & \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 & \rightarrow & \alpha_k = \gamma e_k, & \forall k = 1, \ldots, N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 & \rightarrow & w^T \phi(x_k) + b + e_k - y_k = 0, & \forall k = 1, \ldots, N \end{cases}$$

Substituting w from the first equation in the last one we can express e_k as

$$e_k = y_k - b - \sum_{j=1}^{N} \alpha_j \phi^T(x_j)\phi(x_k) = y_k - b - \sum_{j=1}^{N} \alpha_j K(x_j, x_k),$$

which allows us to eliminate e_k from the third equation.

Matrix equation for the unknowns α and b

We have to solve the following equations for $\alpha_1, \ldots, \alpha_N$ and b

$$\sum_{k=1}^{N} \alpha_k = 0$$
$$\alpha_k = \gamma e_k \quad \text{i.e.} \quad \alpha_k = \gamma \left(y_k - b - \sum_{j=1}^{N} \alpha_j K(x_j, x_k) \right)$$

in other words

$$\left(\begin{array}{c|c} 0 & 1 \cdots 1 \\ \hline 1 & \\ \vdots & K + \gamma^{-1} I \\ 1 & \end{array}\right) \left(\begin{array}{c} b \\ \hline \alpha_1 \\ \vdots \\ \alpha_N \end{array}\right) = \left(\begin{array}{c} 0 \\ \hline y_1 \\ \vdots \\ y_N \end{array}\right),$$

where the matrix K is defined by $K_{i,j} = K(x_i, x_j)$.

Kernel based ridge regression: discussion

- The model

$$\hat{y}(x) = b + \sum_{i=1}^{N} \alpha_i K(x, x_i)$$

  is expressed purely in terms of the Kernel function.
- Learning algorithm uses only the values of $K(x_i, x_j)$ and y_k to determine α and b.
- We don't need to know $\phi(x)$.
- Order N^3 operations for learning, and order N operations for prediction.
- Refinements:
  - Sparse approximations
  - Prune away the training points which correspond to small values of α.

Outline

Conclusion

- Support vector machines are based on two very smart ideas:
  - Margin maximization (to improve generalization)
  - The kernel trick (to handle very high dimensional or complex input spaces)
- SVMs are quite well motivated theoretically
- SVMs are among the most successful techniques for classification
- There exist very efficient implementations that can handle very large problems
- Drawbacks:
  - Essentially black-box models
  - The choice of a good kernel is difficult and critical to reach good accuracy

Conclusion

- Kernel methods are very useful tools in machine learning
  - Several generic frameworks have been proposed to define kernels for various data types
  - Many algorithms have been kernelized
  - Especially interesting in complex application domains like bioinformatics
- Convex optimization is a very useful tool in machine learning
  - Many machine learning problems can be formulated as convex optimization problems
  - Often, a unique solution implies a more stable model
  - Can benefit from the huge work made on optimization algorithms to focus on designing good objective functions

Further reading

- Bishop: Chapter 7 (7.1 and 7.1.1), Appendix E
- Hastie et al.: 12.2, 12.3 (12.3.1)
- Ben-Hur, A., Soon Ong, C., Sonnenburg, S., Schölkopf, B., Rätsch, G. (2008). *Support vector machines and kernels for computational biology.* PLOS Computational biology 4(10). http://www.ploscompbiol.org/

References

- Schölkopf, B. and Smola, A. (2002). *Learning with kernels: support vector machines, regularization, optimization and beyond.* MIT Press, Cambridge, MA.
- Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis.* Cambridge University Press.
- Vapnik, V. (2000). *The nature of statistical learning theory.* Springer.
- Boyd, S. and Vandenberghe, L (2004). *Convex optimization.* Cambridge University Press.
- Some slides borrowed from Pierre Dupont (UCL)

Softwares

- LIBSVM & LIBLINEAR
  http://www.csie.ntu.edu.tw/~cjlin/libsvm
- SVM$^{\text{light}}$
  http://svmlight.joachims.org/
- SHOGUN
  http://www.shogun-toolbox.org/