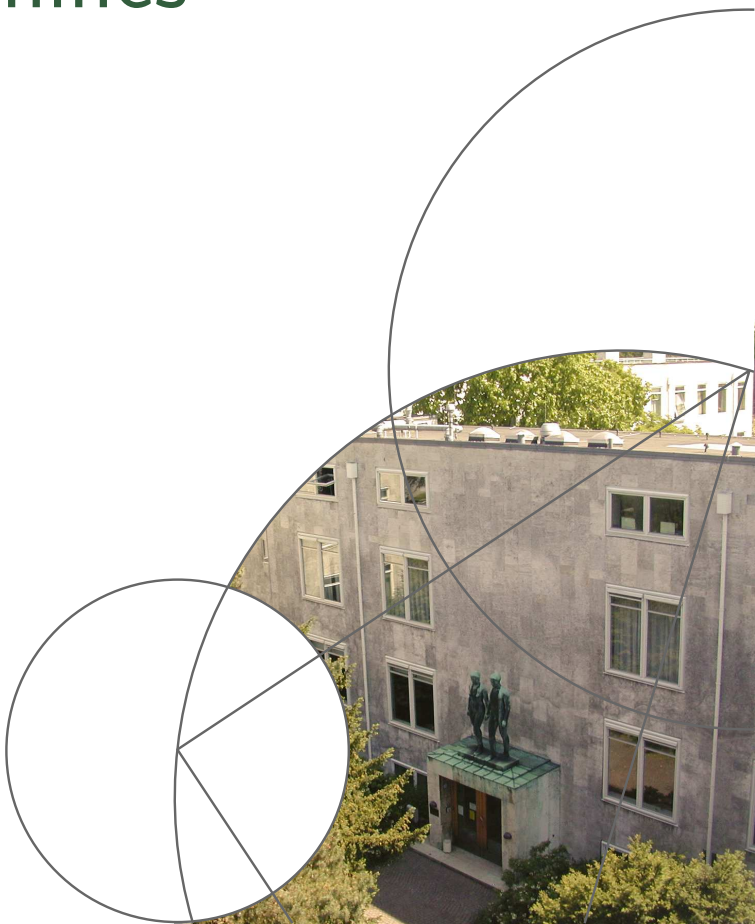
Faculty of Science

Training Support Vector Machines

Advanced Topics in Machine Learning

Christian Igel
Department of Computer Science



igel@diku.dk

What's up?

- We consider training of *non-linear* support vector machines.
- We focus on batch learning using sequential minimal optimization (SMO), which can be adapted for online and active learning.
- For training linear SVMs, other methods – working directly on the primal optimization problem – are preferable.

Why?

- ...to deepen our understanding of SVMs, one of the most important classification methods, which helps using them successfully in practice.
- ...to learn general techniques from optimization that are widely applicable.

Outline

Outline

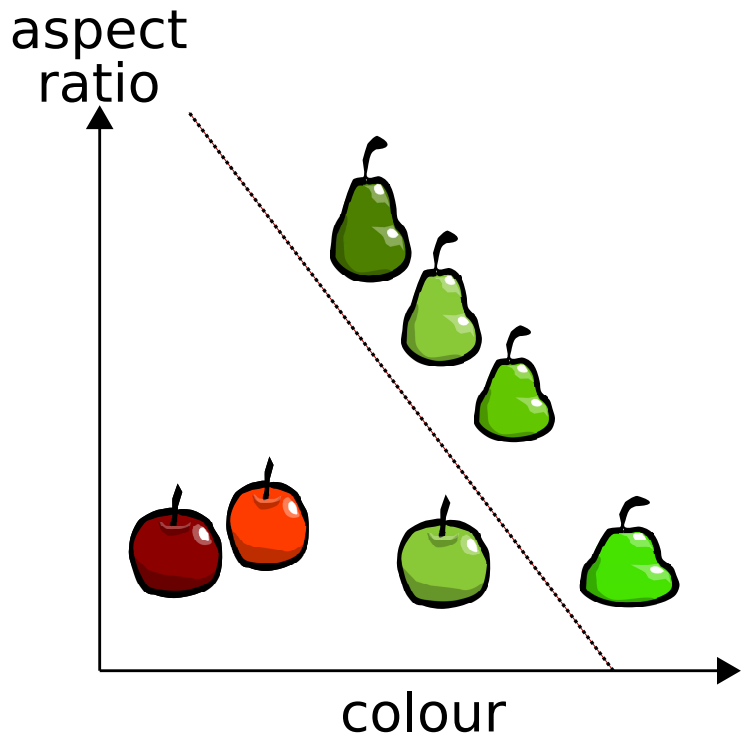
Affine linear decision functions I

Affine linear decision functions

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$$

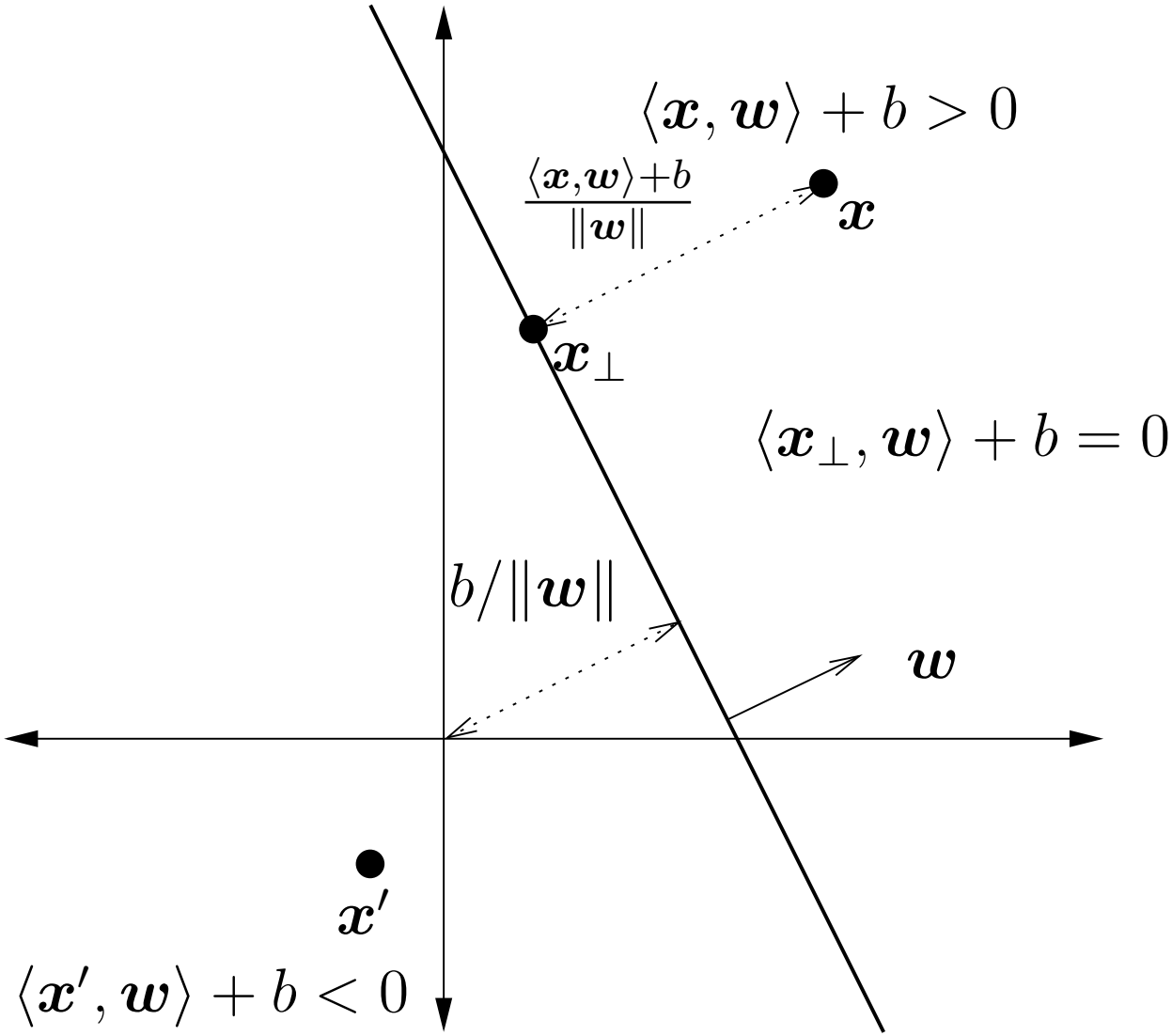
with $\mathcal{X} = \mathbb{R}^n, \boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^n, b \in \mathbb{R}$ lead to hypothesis with (affine) linear decision boundaries:

$$\boldsymbol{x} \mapsto \begin{cases} 1 & \text{if } \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 0 \\ -1 & \text{otherwise} \end{cases}$$



Linear decision functions II

$$f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle + b$$



Margins

The *functional margin* of an example $(\boldsymbol{x}_i, y_i)$ with respect to a hyperplane $(\boldsymbol{w}, b)$ is

$$\gamma_i := y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \ .$$

and its *geometric margin* is

$$\rho_i := y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)/\|\boldsymbol{w}\| = \gamma_i/\|\boldsymbol{w}\| \ .$$

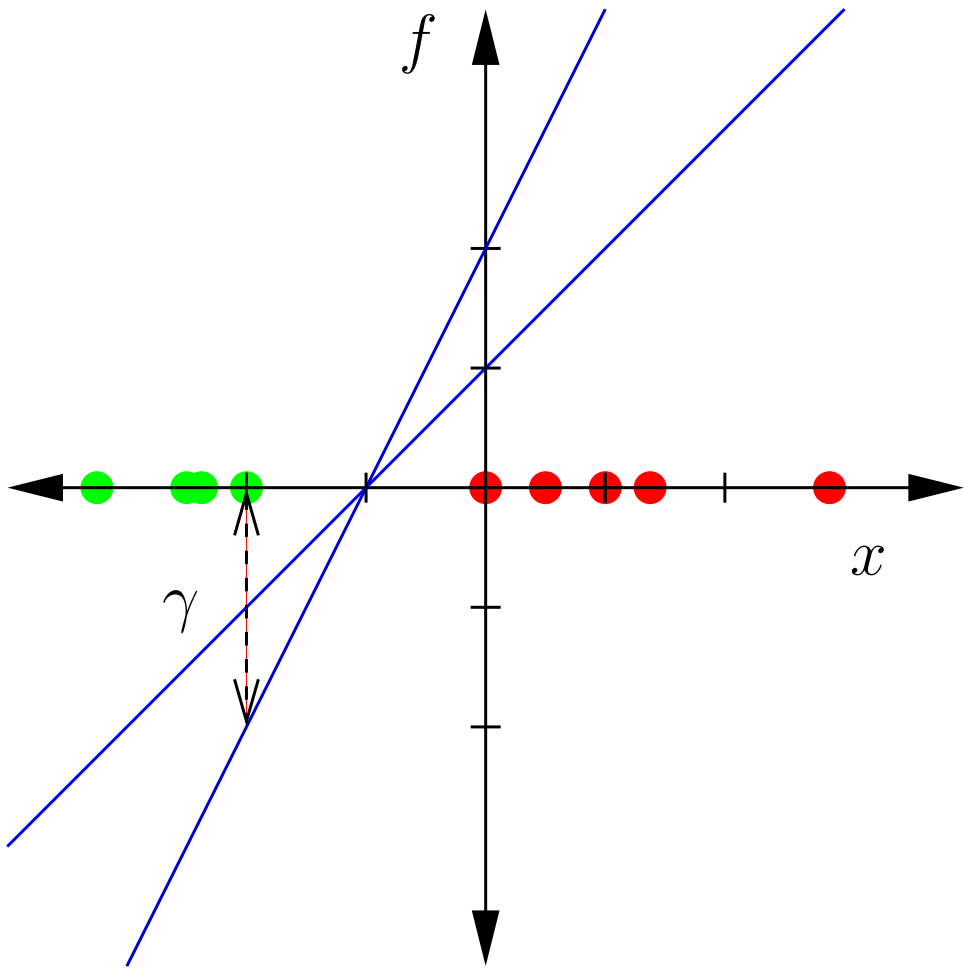
A positive margin implies correct classification.

The functional margin γ_S of a hyperplane $(\boldsymbol{w}, b)$ with respect to a training set S is $\min_i \gamma_i$.

Outline

"Inherent degree of freedom"

Inherent degree of freedom: $(c\boldsymbol{w}, cb)$ leads to same decision boundary for all $c \in \mathbb{R}^+$



Large margin classifier for separable data

Given linearly separable training data $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_\ell, y_\ell)\}$, we get rid of the inherent degree of freedom in

$$\begin{aligned} \text{maximize}_{\boldsymbol{w},b} \quad & \rho = \gamma / \|\boldsymbol{w}\| \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq \gamma \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

by fixing $\gamma = 1$

$$\begin{aligned} \text{maximize}_{\boldsymbol{w},b} \quad & \rho = 1 / \|\boldsymbol{w}\| \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

is equal to:

$$\begin{aligned} \text{minimize}_{\boldsymbol{w},b} \quad & \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

Linear hard margin SVM primal

Given linearly separable data $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_\ell, y_\ell)\}$ the hyperplane $(\boldsymbol{w}, b)$ solving

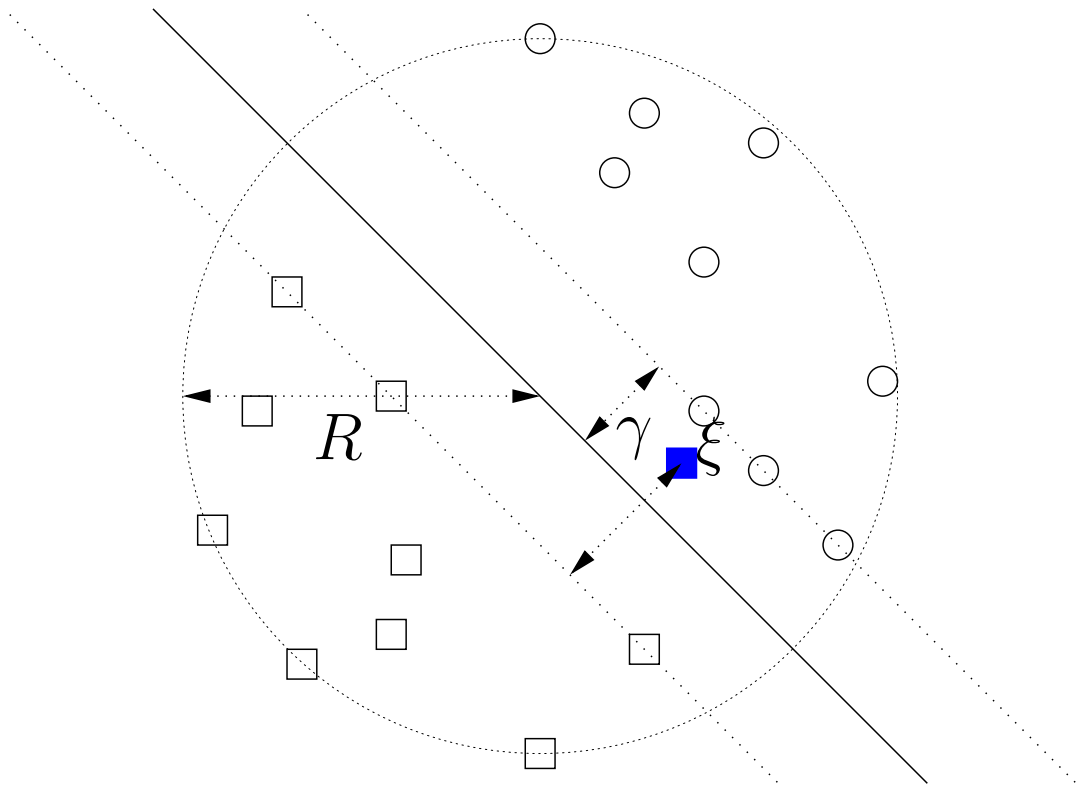
$$\begin{aligned} \text{minimize}_{\boldsymbol{w},b} \quad & \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

realizes the maximal margin hyperplane with margin $\rho = 1/\|\boldsymbol{w}\|$.

Tolerating margin violations: Slack variables

For fixed target margin $\gamma > 0$, the margin *slack variable* ξ_i of $(\boldsymbol{x}_i, y_i)$ with respect to the hyperplane $(\boldsymbol{w}, b)$ is

$$\xi((\boldsymbol{x}_i, y_i), (\boldsymbol{w}, b), \gamma) = \xi_i := \max(0, \gamma - y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)) \ .$$



1-norm linear soft margin SVM primal

Penalizing the absolute values of the slack variables gives:

Given $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_\ell, y_\ell)\} \in (\mathbb{R}^d \times \{-1, 1\})^\ell$ and a regularization parameter $C \geq 0$, a 1-norm linear soft margin SVM computes an affine linear decision function $f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle + b$ by solving:

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi}, \boldsymbol{w}, b} \quad & \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i=1}^{\ell} \xi_i \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i \;\; , \;\; i = 1, \ldots, \ell \\ & \xi_i \geq 0 \;\; , \;\; i = 1, \ldots, \ell \end{aligned}$$

Non-linear SVM

- In SVMs elements from $\mathcal{X}$ occur only in scalar products – we can apply the "kernel trick"!
- Consider an arbitrary input space $\mathcal{X}$ and a feature map $\Phi : \mathcal{X} \to \mathcal{H}_k$, where

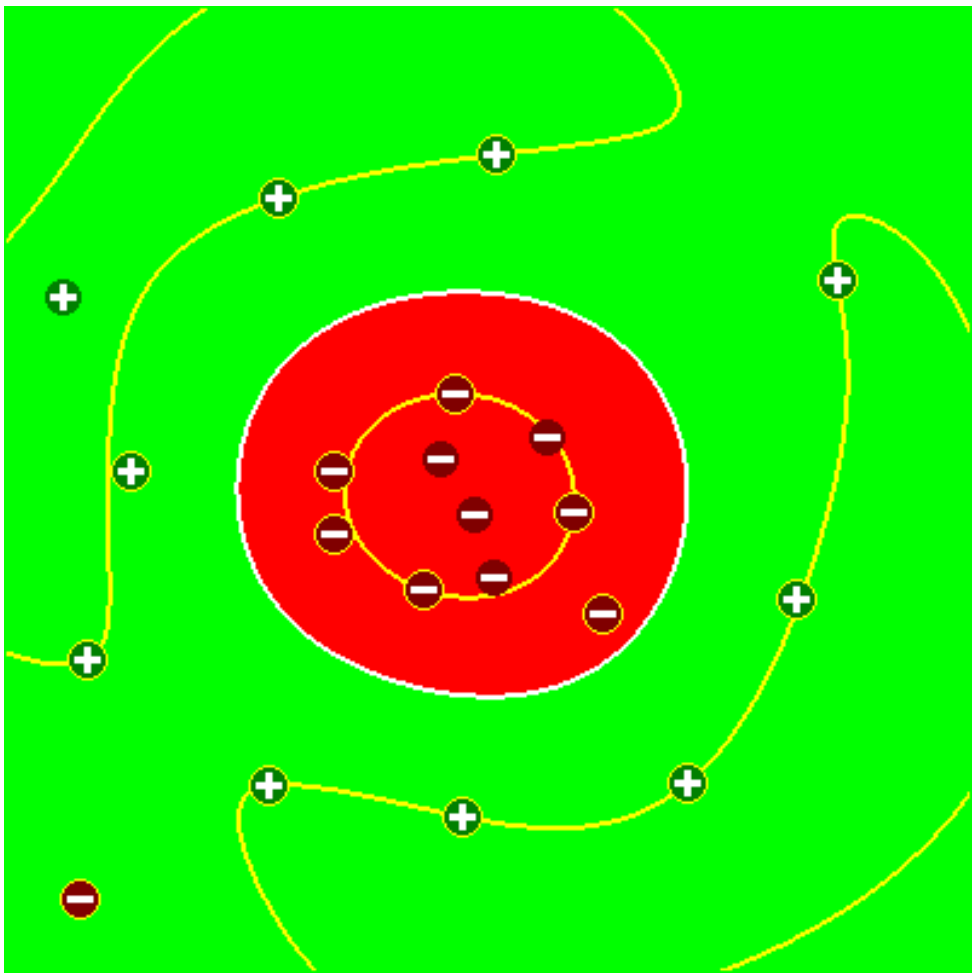
$$\mathcal{H}_k = \operatorname{span}\{k(x, \cdot) \,|\, x \in \mathcal{X}\}$$

  is the RKHS induced by the (symmetric, positive-definite) kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.
- The non-linear SVM learns decision functions of the form

$$f(x) = \langle \Phi(x), \boldsymbol{w} \rangle + b \ .$$

  Here $\boldsymbol{w} \in \mathcal{H}_k$ and $\Phi : x \mapsto k(x, \cdot)$.
- The scalar product $\langle \Phi(x), \boldsymbol{w} \rangle$ will be computed using the kernel trick.

Regularization and kernel representation



Kernel k: Represent data for linear classification (ideally, $h^{\text{Bayes}} \in \mathcal{H}_k^b$)
Slack variables: Deal with noise and outliers (i.e., $\mathcal{R}_p^{\text{Bayes}} > 0$)

1-norm non-linear soft margin SVM primal

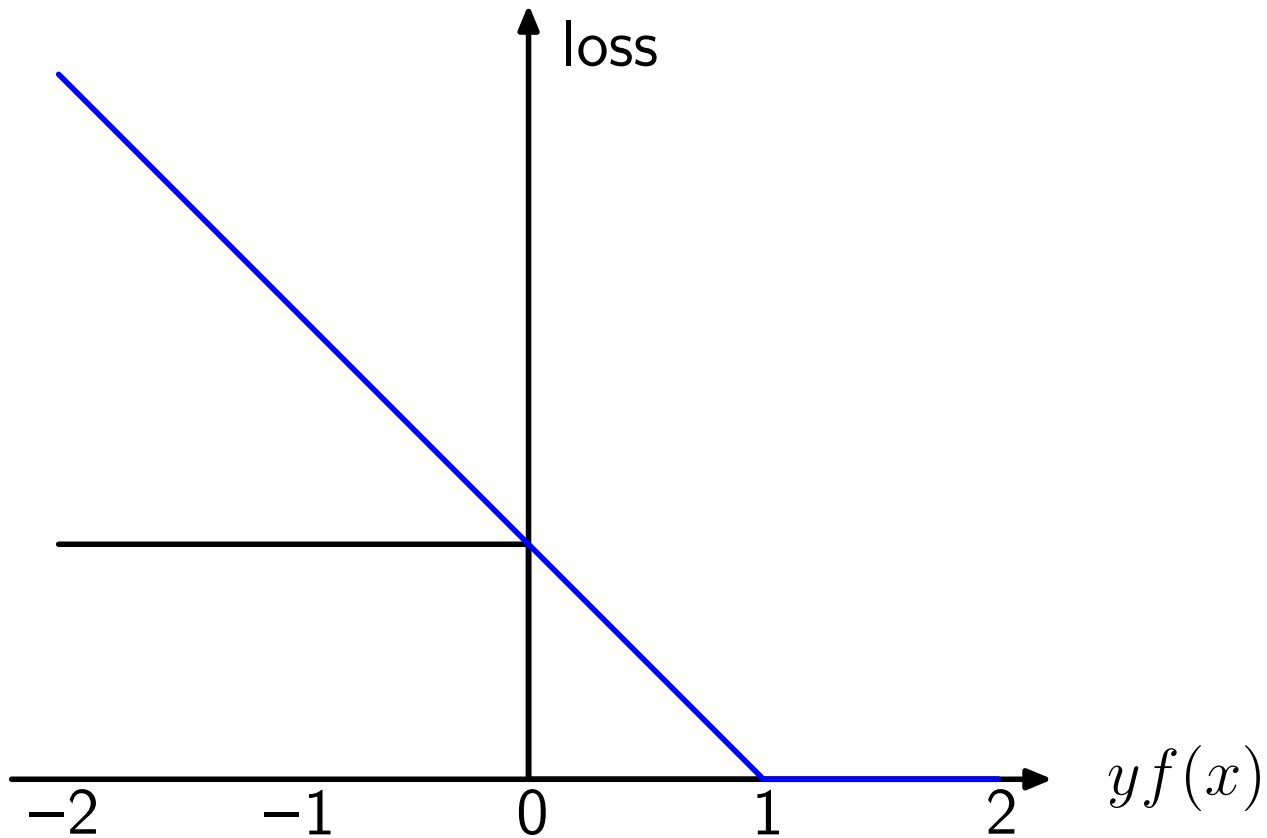
Given $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\} \in (\mathcal{X} \times \{-1, 1\})^\ell$ a regularization parameter $C \geq 0$, and a kernel k on $\mathcal{X}$, a 1-norm soft margin SVM computes a linear decision function $f(x) = \langle \Phi(x), \boldsymbol{w} \rangle + b$ by solving:

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi}, \boldsymbol{w}, b} \quad & \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i=1}^{\ell} \xi_i \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \Phi(x_i) \rangle + b) \geq 1 - \xi_i \quad , \quad i = 1, \ldots, \ell \\ & \xi_i \geq 0 \quad , \quad i = 1, \ldots, \ell \quad , \end{aligned}$$

where $\Phi(x) = k(x, \cdot)$.

Hinge loss as convex surrogate for 0-1 loss

$$L_{\text{hinge}}(y, f(x)) = [1 - yf(x)]_+ = \max(0, 1 - yf(x))$$



0-1 loss applied to $\text{sgn}(f(x))$ and hinge loss

1-norm soft margin SVM and regularization I

- 1-norm soft margin SVM, primal

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi},\boldsymbol{w},b} \quad & \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w}\rangle + C\sum_{i=1}^{\ell}\xi_i \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle + b) \geq 1-\xi_i \quad , \quad i=1,\ldots,\ell \\ & \xi_i \geq 0 \quad , \quad i=1,\ldots,\ell \end{aligned}$$

- For fixed $\boldsymbol{w}$ optimal slack variables are:

$$\xi_i = \max(0, 1 - y_i(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle + b)) = L_{\text{hinge}}(y_i, f(x_i))$$

- Hypothesis classes
  - $\mathcal{H}_k$: RKHS induced by k
  - $\mathcal{H}_k^b = \{f(x) = g(x) + b \,|\, g \in \mathcal{H}_k, b \in \mathbb{R}\}$

1-norm soft margin SVM and regularization II

- $L_{\text{hinge}}(y, \hat{y}) = \max(0, 1 - y\hat{y})$
- 1-norm soft margin SVM

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi},\boldsymbol{w},b} \quad & \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i=1}^{\ell} \xi_i \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i) \rangle + b) \geq 1 - \xi_i \ , \ \ i = 1, \ldots, \ell \\ & \xi_i \geq 0 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

corresponds to

$$\text{minimize}_{f \in \mathcal{H}_k^b} \quad \frac{1}{\ell} \sum_{i=1}^{\ell} L_{\text{hinge}}(y_i, f(x_i)) + \gamma_\ell \| f \|_k^2$$

where $\gamma_\ell = (2\ell C)^{-1}$ and $\|.\|_k$ inherited from $\mathcal{H}_k$ to $\mathcal{H}_k^b$ is only a semi-norm

Representer theorem

Let $\Omega : [0, \infty[\to \mathbb{R}$ be a strictly monotonic increasing function, $\mathcal{H}$ a RKHS with kernel k on $\mathcal{X}$ and L a loss function. Given $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\} \subset (\mathcal{X} \times \mathbb{R})^\ell$, each minimizer $f \in \mathcal{H}^b$ of the regularized empirical risk

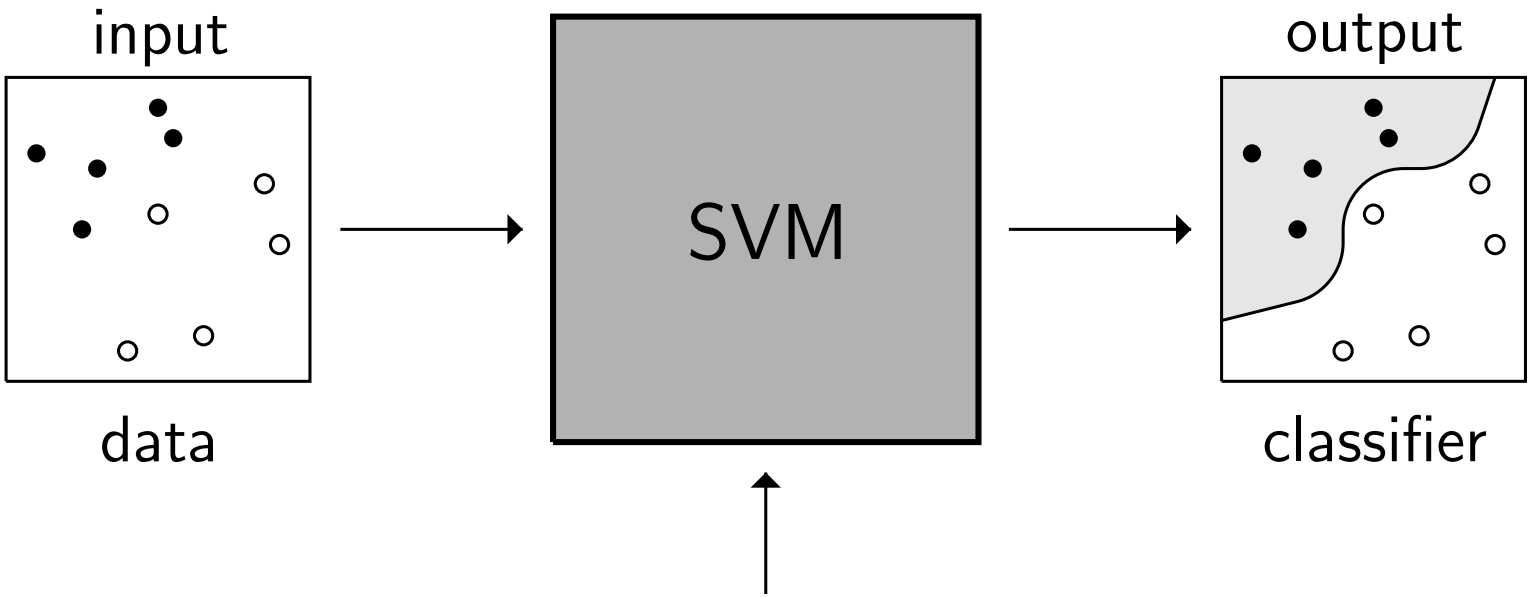
$$\sum_{i=1}^{\ell} L(y_i, f(x_i)) + \Omega(\|f\|_k^2)$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{\ell} \beta_i k(x_i, x) + b$$

with $\alpha_1, \ldots, \alpha_\ell, b \in \mathbb{R}$.

Binary SVMs



prior knowledge:
kernel function
regularisation parameter

Cortes, Vapnik: Support-Vector Networks, *Machine Learning* 20(3):273–297, 1995

Outline

Lagrangian Function

Given real-valued functions f, g_i, h_j ($i = 1, \ldots, k$ and $j = 1, \ldots, m$) defined on $M \subset \mathbb{R}^n$, we consider *(primal) optimization problems* of the form

$$\begin{array}{lll} \text{minimize} & f(\boldsymbol{w}) & \boldsymbol{w} \in M \\ \text{subject to} & g_i(\boldsymbol{w}) \leq 0 & i = 1, \ldots, k \\ & h_j(\boldsymbol{w}) = 0 & j = 1, \ldots, m \ . \end{array}$$

For a given primal optimization problem, the *Lagrangian function* is defined as

$$L(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{w}) + \sum_{i=1}^{m} \lambda_i h_i(\boldsymbol{w}) + \sum_{i=1}^{k} \mu_i g_i(\boldsymbol{w}) \ .$$

Karush-Kuhn-Tucker (KKT) theorem

Given a minimization problem with differentiable convex functions f, g_i, and affine functions h_j ($i = 1, \ldots, k$ and $j = 1, \ldots, m$) on a convex set $M \subset \mathbb{R}^n$. Suppose the differentials $\mathrm{d}h_i(\boldsymbol{w}^*)$ and $\mathrm{d}g_i(\boldsymbol{w}^*)$ are linearly independent at $\boldsymbol{w}^* \in M$. Then $\boldsymbol{w}^*$ is a global minimum iff there exist $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^k$ with

$$\frac{\partial L(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)}{\partial \boldsymbol{w}} = \mathbf{0} \ ,$$

$$\frac{\partial L(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)}{\partial \boldsymbol{\lambda}} = \mathbf{0} \ ,$$

$$\mu_i^* g_i(\boldsymbol{w}^*) = 0 \ \text{ for } i = 1, \ldots, k \ ,$$

$$g_i(\boldsymbol{w}^*) \leq 0 \ \text{ for } i = 1, \ldots, k \ ,$$

$$\mu_i^* \geq 0 \ \text{ for } i = 1, \ldots, k \ .$$

KKT Complementarity Condition

The conditions

$$\begin{aligned} \mu_i^* g_i(\boldsymbol{w}^*) &= 0 \text{ for } i = 1, \ldots, k \ , \\ g_i(\boldsymbol{w}^*) &\leq 0 \text{ for } i = 1, \ldots, k \ , \\ \mu_i^* &\geq 0 \text{ for } i = 1, \ldots, k \end{aligned}$$

are called Karush-Kuhn-Tucker complementarity conditions.

Dual optimization problem

Given a primal optimization problem with Lagrangian function L, the *dual optimization problem* is defined by

$$\begin{array}{lll} \text{maximize} & \vartheta(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{w} \in M} L(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) & \boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\mu} \in \mathbb{R}^k \\ \text{subject to} & \mu_i \geq 0 & i = 1, \ldots, k \ . \end{array}$$

Under the assumptions of the KKT theorem, the *duality gap*

$$f(\boldsymbol{w}^*) - \sup_{\boldsymbol{\lambda} \in \mathbb{R}^m, \mu_1, \ldots, \mu_k \in \mathbb{R}_0^+} \left[\inf_{\boldsymbol{w} \in M} L(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \right]$$

is zero.

Solution strategy

Given a constraint minimization problem, we maximize the dual, where we get rid of the inner minimization task using the following procedure:

1. Set derivatives of Lagrangian w.r.t. primal variables to zero;
2. Solve analytically w.r.t. primal variables;
3. Substitue primal variables into Lagrangian;
4. Maximize Lagrangian with w.r.t. dual variables.

Outline

Linear hard margin SVM: Primal to dual

Primal form:

$$\begin{aligned} \text{minimize}_{\boldsymbol{w},b} \quad & \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

Dual form:

$$\begin{aligned} \text{maximize}_{\boldsymbol{\alpha}} \quad & \inf_{\boldsymbol{w},b} \quad L(\boldsymbol{w}, b, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \alpha_i \geq 0 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

with Lagrangian:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i [y_i(\langle \boldsymbol{w}, x_i \rangle + b) - 1]$$

Eliminating primal variables I

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i [y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1]$$

Karush-Kuhn-Tucker (KKT) theorem requires

$$\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = 0 \qquad \frac{\partial}{\partial b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = 0$$

yielding

$$\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \boldsymbol{w} - \sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{x}_i \quad \text{and} \quad \frac{\partial}{\partial b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i y_i$$

implying $\boldsymbol{w} = \sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{x}_i$ and $0 = \sum_{i=1}^{\ell} \alpha_i y_i$.

Eliminating primal variables II

Using $\boldsymbol{w} = \sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{x}_i$ gives

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle - \sum_{i=1}^{\ell} \alpha_i [y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1]$$

$$= \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle - \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^{\ell} \alpha_i$$

$$= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \quad .$$

Linear hard margin SVM dual

For linearly separable $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_\ell, y_\ell)\}$ solving

$$\text{maximize}_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$$

$$\text{subject to} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \ , \quad i = 1, \ldots, \ell$$

leads to the maximal margin hyperplane with margin $\rho = 1/\|\boldsymbol{w}^*\|$ with

$$\boldsymbol{w}^* = \sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{x}_i \ .$$

KKT complementarity condition I

- KKT complementarity condition requires

$$\alpha_i^* [y_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b^*) - 1] = 0$$

  for $i = 1, \ldots, \ell$.

- KKT condition can be used to compute b^*, e.g.:

$$b^* = -\frac{\max_{y_i=-1}(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle) + \min_{y_i=1}(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle)}{2}$$

- Solution is sparse:

$$\mathsf{SV} = \{\boldsymbol{x}_i \,|\, \alpha_i^* \neq 0\}$$

$$f(\boldsymbol{x}, \boldsymbol{\alpha}^*, b^*) = \sum_{i=1}^{\ell} y_i \alpha_i^* \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b^* = \sum_{\boldsymbol{x}_i \in \mathsf{SV}} y_i \alpha_i^* \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b^*$$

KKT complementarity condition II

For $\boldsymbol{x}_j \in \mathsf{SV}$

$$y_j f(\boldsymbol{x}_j, \boldsymbol{\alpha}^*, b^*) = y_j \left(\sum_{\boldsymbol{x}_i \in \mathsf{SV}} y_i \alpha_i^* \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + b^* \right) = 1$$

and therefore

$$\begin{aligned} \langle \boldsymbol{w}^*, \boldsymbol{w}^* \rangle &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ &= \sum_{\boldsymbol{x}_j \in \mathsf{SV}} \alpha_j^* y_j \sum_{\boldsymbol{x}_i \in \mathsf{SV}} \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ &= \sum_{\boldsymbol{x}_j \in \mathsf{SV}} \alpha_j^* (1 - y_j b^*) = \sum_{\boldsymbol{x}_j \in \mathsf{SV}} \alpha_j^* \ . \end{aligned}$$

Hard margin SVM dual

For data $\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ linearly separable in the feature space $\mathcal{H}_k^b$ defined by the kernel k the solution $\boldsymbol{\alpha}^*$, b^* of

$$\text{maximize}_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \;\;,\;\; \alpha_i \geq 0 \;\;,\;\; i = 1, \ldots, \ell$$

leads to the decision rule $\operatorname{sgn}\left(\sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, x) + b^*\right)$ corresponding to the maximal margin hyperplane in $\mathcal{H}_k^b$ with margin $\rho = 1/\|\boldsymbol{w}^*\| = 1/\sqrt{\sum_{x_j \in \mathsf{SV}} \alpha_j^*}$.

1-norm soft margin SVM: Primal to dual

Primal (linear case for ease of notation)

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi},\boldsymbol{w},b} \quad & \frac{1}{2}\langle \boldsymbol{w},\boldsymbol{w}\rangle + C\sum_{i=1}^{\ell}\xi_i \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w},\boldsymbol{x}_i\rangle + b) \geq 1-\xi_i \;\;,\;\; i=1,\ldots,\ell \\ & \xi_i \geq 0 \;\;,\;\; i=1,\ldots,\ell \end{aligned}$$

has Lagrangian under constraints $\alpha_i, \beta_i \geq 0, i = 1, \ldots, \ell$:

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) =$$

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\alpha_i[y_i(\langle \boldsymbol{w}, x_i\rangle + b) - 1 + \xi_i] - \sum_{i=1}^{\ell}\beta_i\xi_i$$

Eliminating primal variables III

Lagrangian under constraints $\alpha_i, \beta_i \geq 0, i = 1, \ldots, \ell$:

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) =$$

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\alpha_i[y_i(\langle \boldsymbol{w}, x_i\rangle + b) - 1 + \xi_i] - \sum_{i=1}^{\ell}\beta_i\xi_i$$

KKT:

$$\frac{\partial L}{\partial \boldsymbol{w}}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{w} - \sum_{i=1}^{\ell}\alpha_i y_i x_i = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{\ell}\alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial \xi_i}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = C - \alpha_i - \beta_i = 0 \quad , \quad i = 1, \ldots, \ell$$

$$\frac{\partial L}{\partial b}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{\ell}\alpha_i y_i = 0$$

Eliminating primal variables IV

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

$$= \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\alpha_i[y_i(\langle \boldsymbol{w}, x_i\rangle + b) - 1 + \xi_i] - \sum_{i=1}^{\ell}\beta_i\xi_i$$

$$= \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{\ell}[C\xi_i - \alpha_i\xi_i - \beta_i\xi_i] - \sum_{i=1}^{\ell}\alpha_i y_i\langle \boldsymbol{w}, x_i\rangle - \sum_{i=1}^{\ell}\alpha_i y_i b + \sum_{i=1}^{\ell}\alpha_i$$

$$= \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{\ell}[C - \alpha_i - \beta_i]\xi_i - \|\boldsymbol{w}\|^2 - b\sum_{i=1}^{\ell}\alpha_i y_i + \sum_{i=1}^{\ell}\alpha_i$$

$$= \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$$

Box constraint

KKT complementarity conditions:

$$\alpha_i[y_i(\langle \boldsymbol{w}, x_i\rangle + b) - 1 + \xi_i] = 0, \quad i = 1, \ldots, \ell$$
$$\beta_i \xi_i = 0, \quad i = 1, \ldots, \ell$$
$$\xi_i(\alpha_i - C) = 0, \quad i = 1, \ldots, \ell \quad \text{(using } \beta_i = C - \alpha_i \text{ and } \beta_i \xi_i = 0\text{)}$$

- Bounded SVs: $\xi \neq 0 \Rightarrow \alpha_i = C$
- Box constraint: $\alpha_i, \beta_i \geq 0$ and $\beta_i = C - \alpha_i$ imply $0 \leq \alpha_i \leq C$
- Free SVs: $0 < \alpha_i < C \Rightarrow \xi_i = 0$
- How to get b: $0 < \alpha_i < C \wedge \xi_i = 0 \Rightarrow y_i(\langle \boldsymbol{w}, x_i\rangle + b) = 1$

1-norm soft margin SVM

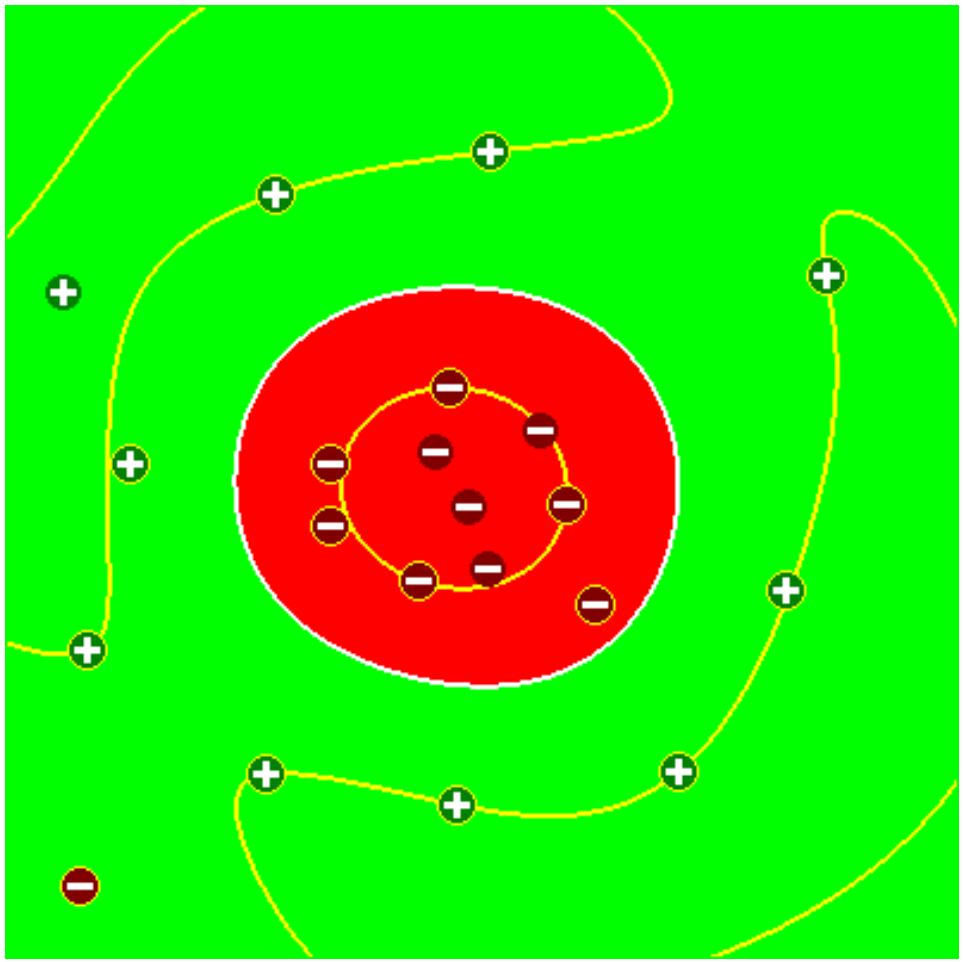
For $\{x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ and kernel k solving

$$\text{maximize}_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \ , \ C \geq \alpha_i \geq 0 \ , \ i = 1, \ldots, \ell$$

leads to the decision rule $h(x) = \operatorname{sgn}(f(x))$ with $f(x) = \sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, x) + b^*$, where b^* is chosen so that $y_i f(x_i) = 1$ for any i with $C > \alpha_i > 0$ and the slack variables of the "corresponding hyperplane" in $\mathcal{H}_k^b$ are defined relative to the margin $\rho = 1/\|\boldsymbol{w}^*\| = 1/\sqrt{\sum_{x_i, x_j \in \mathsf{SV}} y_i y_j \alpha_i^* \alpha_j^* k(x_i, x_j)}$.

Inspecting the solution



Bounded SV: $\alpha_i = C,\ \xi_i \geq 0,\ y_i f(x_i) \leq 1$

Free SV: $0 < \alpha_i < C,\ \xi_i = 0,\ y_i f(x_i) = 1$

Non-SV: $\alpha_i = 0,\ \xi_i = 0,\ y_i f(x_i) > 1$

Outline

1-norm soft margin SVM: Dual

1-norm Soft Margin SVM Dual optimization problem:

$$\text{maximize}_{\boldsymbol{\alpha}} \quad \mathcal{D}(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \text{ and } \forall i \in \{1, \ldots, \ell\} :$$

$$C \geq \alpha_i \geq 0 y_i \alpha_i \in [a_i, b_i] = \begin{cases} [0, C] & \text{if } y_i = +1 \\ [-C, 0] & \text{if } y_i = -1 \end{cases}$$

Decision rule $\operatorname{sgn}(d(x))$ with $d(x) = \sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, x) + b^*$, where b^* is chosen so that $y_i d(x_i) = 1$ for any i with $C > \alpha_i > 0$.

Restricting the quadratic program

1-norm Soft Margin SVM dual optimization problem restricted to *working set* B:

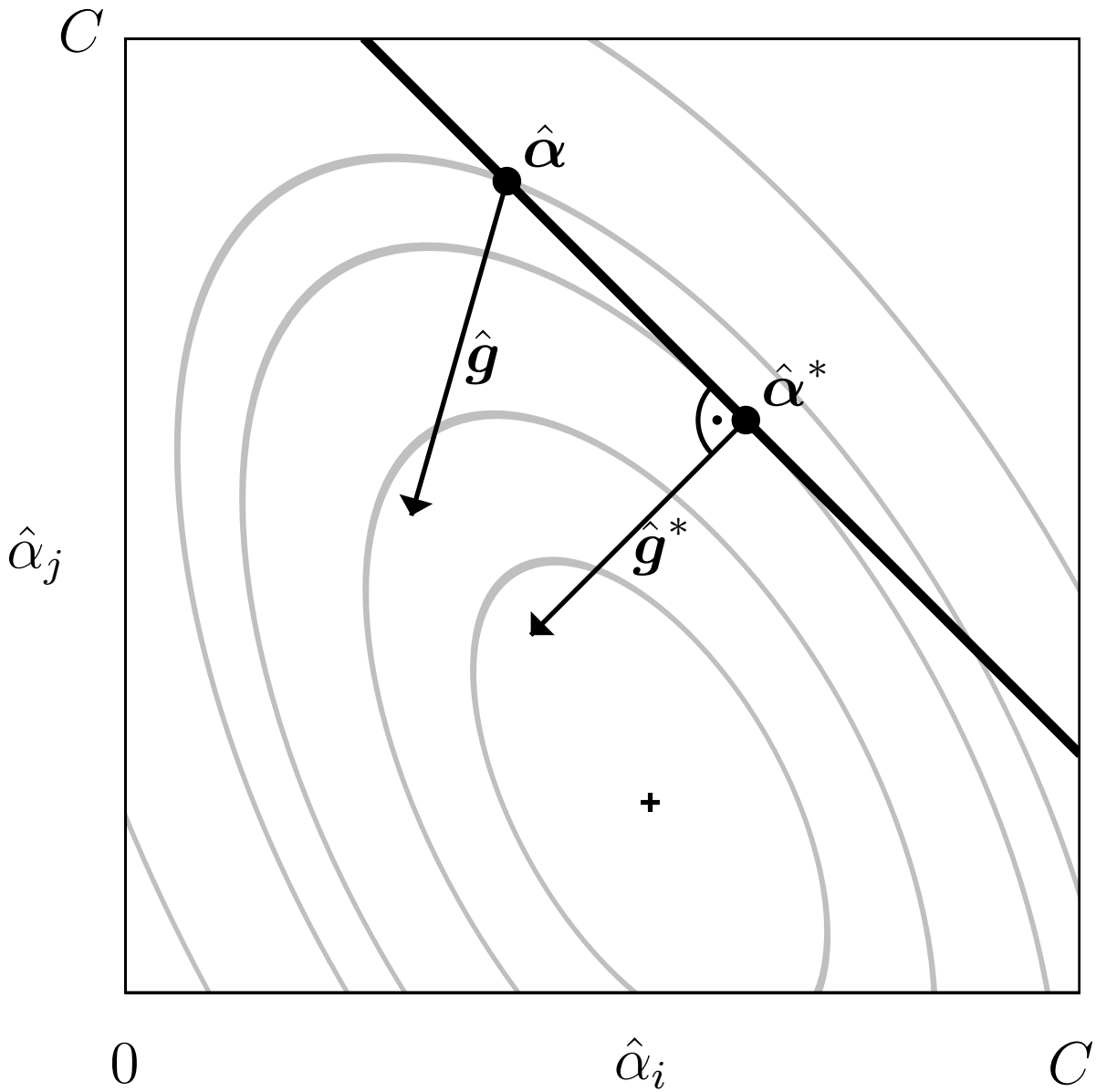
$$\text{maximize}_{\hat{\boldsymbol{\alpha}}} \quad \mathcal{D}(\hat{\boldsymbol{\alpha}}) = \sum_{i=1}^{\ell} \hat{\alpha}_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \hat{\alpha}_i \hat{\alpha}_j y_i y_j k(x_i, x_j)$$

$$\text{subject to} \quad \sum_{i=1}^{\ell} \hat{\alpha}_i y_i = 0$$

$$\forall i \in \{1, \ldots, \ell\} : C \geq \hat{\alpha}_i \geq 0$$

$$\forall i \notin B : \hat{\alpha}_i = \alpha_i$$

Two-dimensional subproblem



Some definitions

Let's define Gram matrix entry

$$K_{ij} = k(x_i, x_j)$$

gradient of dual

$$g_i = \frac{\partial \mathcal{D}(\boldsymbol{\alpha})}{\partial \alpha_i} = 1 - y_i \sum_{j=1}^{\ell} y_j \alpha_j K_{ij}$$

and the index sets:

$$I_{\text{up}} = \{i \,|\, y_i \alpha_i < b_i\} \qquad (y_i \alpha_i \text{ may increase})$$

$$I_{\text{down}} = \{i \,|\, y_i \alpha_i > a_i\} \qquad (y_i \alpha_i \text{ may decrease})$$

Decomposition algorithms

Strategy: Iteratively solve dual optimization problem

Decomposition Algorithm

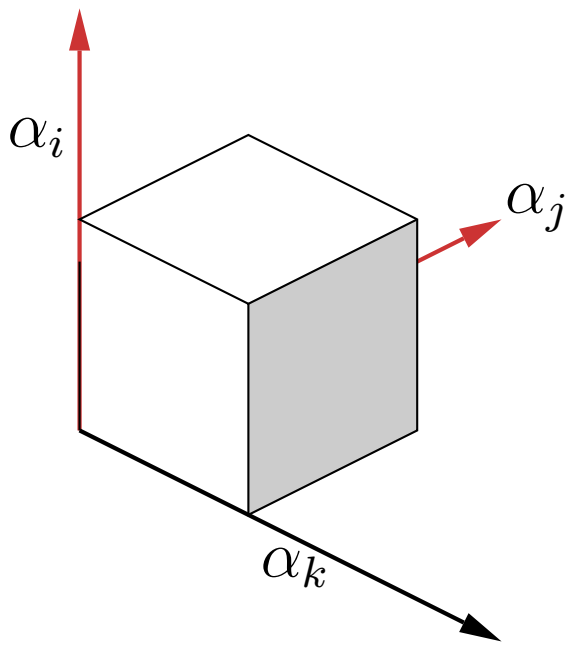
1 $\boldsymbol{\alpha} \leftarrow$ feasible starting point
2 **repeat**
3 | select working set B
4 | solve QP restricted to B resulting in $\hat{\boldsymbol{\alpha}}$
5 | $\boldsymbol{\alpha} \leftarrow \hat{\boldsymbol{\alpha}}$
6 **until** *stopping criterion is met*

$B = \{i, j\}$, $i < j$: Sequential minimal optimization (SMO)
search directions are just
$\pm(0, \ldots, y_i, 0, \ldots, 0, -y_j, 0, \ldots, 0) = \pm \boldsymbol{u}_{ij}$

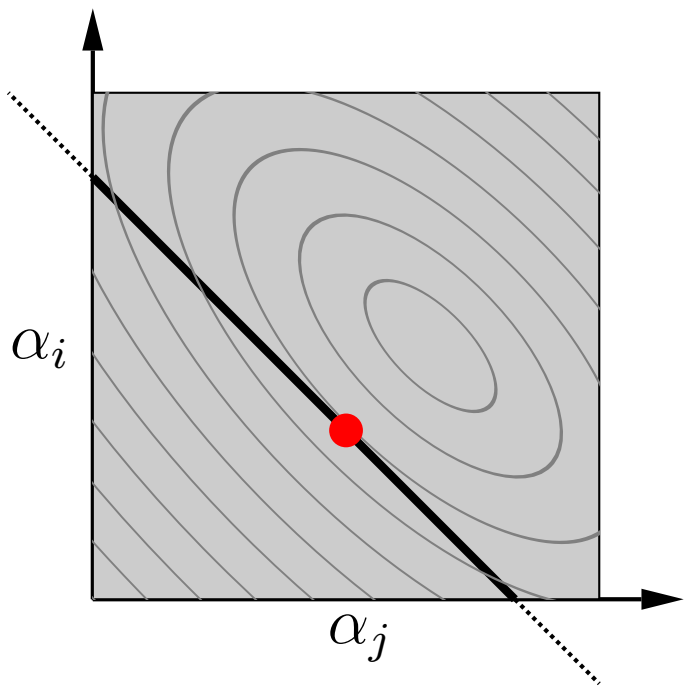
Sequential minimal optimization

repeat until target accuracy reached
{

select working set



solve optimally



}

Solving the two-dimensional subproblem

Hessian of dual problem has elements:

$$\frac{\partial^2 \mathcal{D}(\boldsymbol{\alpha})}{\partial \alpha_i \partial \alpha_j} = y_i y_j K_{ij}$$

Maximizing w.r.t. λ (ignoring box constraints)

$$\mathcal{D}(\boldsymbol{\alpha} + \lambda \boldsymbol{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha}) = \lambda(y_i g_i - y_j g_j) - \frac{\lambda^2}{2}(K_{ii} + K_{jj} - 2K_{ij})$$

by Newton step gives optimal λ^*:

$$\lambda^* = \frac{y_i g_i - y_j g_j}{K_{ii} + K_{jj} - 2K_{ij}}$$

Recomputing gradient & stopping criterion

Gradient of full problem can be adjusted after optimizing on B by:

$$\forall k \in \{1, \ldots, \ell\} : g_k \leftarrow g_k - y_k \sum_{i \in B} y_i (\hat{\alpha}_i - \alpha_i) K_{ik}$$

Stopping criterion (which needs gradient)

$$\max_{i \in I_{\text{up}}} y_i g_i - \min_{j \in I_{\text{down}}} y_j g_j \leq \epsilon$$

for $\epsilon > 0$.

Sequential minimal optimization

Sequential minimal optimization

1 $\boldsymbol{\alpha} \leftarrow \mathbf{0}$, $\boldsymbol{g} \leftarrow \mathbf{1}$

2 **repeat**

3 select indices $i \in I_{\text{up}}$ and $j \in I_{\text{down}}$

4 $\lambda = \min \left\{ b_i - y_i \alpha_i, y_j \alpha_j - a_j, \frac{y_i g_i - y_j g_j}{K_{ii} + K_{jj} - 2K_{ij}} \right\}$

5 $\forall k \in \{1, \ldots, \ell\} : g_k \leftarrow g_k - \lambda y_k K_{ik} + \lambda y_k K_{jk}$

6 $\alpha_i \leftarrow \alpha_i + y_i \lambda$

7 $\alpha_j \leftarrow \alpha_j - y_j \lambda$

8 **until** $\max_{i \in I_{up}} y_i g_i - \min_{j \in I_{down}} y_j g_j \leq \epsilon$

How long does training an SVM take?

Intuitive bounds:

- Assume an oracle tells us the unbounded SVs $F = \{x_i \,|\, 0 < \alpha_i < C\}$ and bounded SVs, then computing the αs takes $\mathcal{O}(|F|^3)$.
- Checking the optimality condition by computing the gradient from scratch takes $\mathcal{O}(\ell \cdot \#\text{SV})$.

Batch SVM training scales between quadraticly and cubicly in the number of training points.

Outline

Working set selection: Most violating pair

Problem: How to select the working set B such that

1. much progress is made / only few iterations are needed, and
2. few kernel evaluations are required?

We ignore Gram matrix caching/ "chunking" / "shrinking" in this course and just consider the selection of B.

Standard algorithm: *Most violating pair* working set selection.

1. first index $i = \operatorname{argmax}_{k \in I_{\mathsf{up}}} y_k g_k$
2. second index $j = \operatorname{argmin}_{k \in I_{\mathsf{down}}} y_k g_k$

- *first order working set selection*, recall that for $\lambda \to 0$

$$\mathcal{D}(\boldsymbol{\alpha} + \lambda \boldsymbol{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha}) = \lambda(y_i g_i - y_j g_j) + \mathcal{O}(\lambda^2)$$

- requires just $\mathcal{O}(\ell)$ computations

Maximum gain

Maximizing gain of subproblem in search direction $\boldsymbol{u}_{ij}$ ignoring box constraints corresponds to maximizing w.r.t. λ:

$$\mathcal{D}(\boldsymbol{\alpha} + \lambda \boldsymbol{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha}) = \lambda(y_i g_i - y_j g_j) - \frac{\lambda^2}{2}(K_{ii} + K_{jj} - 2K_{ij})$$

Newton step gives optimal λ^*

$$\lambda^* = \frac{y_i g_i - y_j g_j}{K_{ii} + K_{jj} - 2K_{ij}}$$

yielding a gain $\mathcal{D}(\boldsymbol{\alpha} + \lambda^* \boldsymbol{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha})$ of

$$\frac{(y_i g_i - y_j g_j)^2}{2(K_{ii} + K_{jj} - 2K_{ij})} \quad .$$

Maximum gain working set selection

Idea: Select i and j such that gain

$$\mathcal{D}(\boldsymbol{\alpha} + \lambda^* \boldsymbol{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha}) = \frac{(y_i g_i - y_j g_j)^2}{2(K_{ii} + K_{jj} - 2K_{ij})}$$

is maximized (ignoring box constraints).

Problem: Checking all $\ell(\ell-1)/2$ index pairs is not feasible.

Solution:

1. First index i is picked according to most violating pair heuristic.
2. Second index j is selected to maximize gain.

- *Second order working set selection*
- Requires just $\mathcal{O}(\ell)$ computations (given reasonable caching strategy).

Outline

Online learning

- We consider an ordered series of patterns $(x_1, y_1), (x_2, y_2), \ldots$ presented in a serial fashion.
- Online learning algorithms adapt hypotheses by successively processing new training examples.

At step $t > 0$, the goal of online learning is to infer a new hypothesis h_t given

1. the current sample (x_t, y_t),
2. the previous hypothesis h_{t-1} (where induction starts from some a priori hypothesis h_0), and
3. some finite memory $M \subseteq \{(x_1, y_1), (x_2, y_2), \ldots, (x_{t-1}, y_{t-1})\}$ of previous data points.

At each step t, h_t should minimize some loss function.

Online learning algorithms...

- provide usable solution without processing complete training set,
- allow to deploy an adaptive system that further improves over time (life-long learning),
- are ideal for streaming data,
- can be adopted for *novelty detection*,
- allow to track changes in the environment (i.e., can be applied if i.i.d. assumption is violated),
- need not be worse than batch learning algorithms even in batch scenario.

Ac

Active learning has successfully been applied to address sample selection bias.

If

- there is a wealth of input data from $\mathcal{X}$, but the corresponding targets are difficult or expensive to determine, or
- the data set is so large that processing every element is not feasible

then we need systems which autonomously choose the input data points from which they expect to learn most – that learn *actively*!

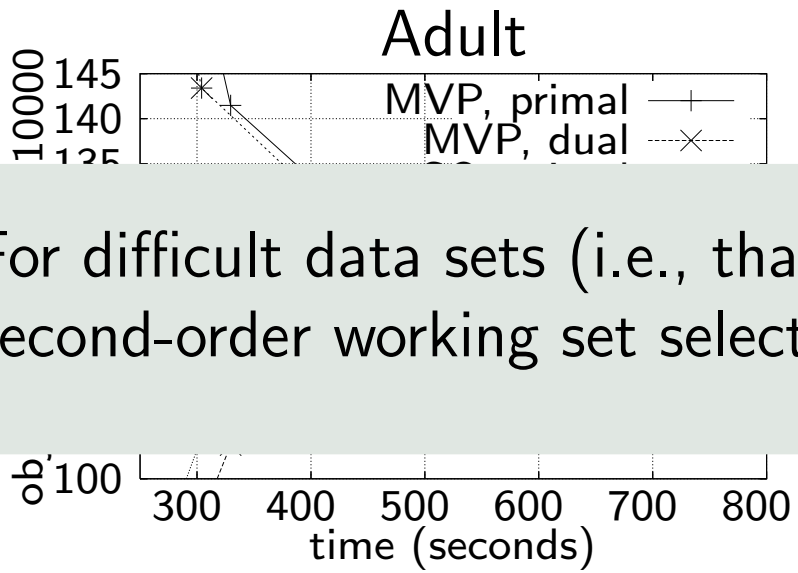
Active learning usually generates a new hypothesis based on an existing hypothesis and new sample data, that is, it requires online learning.

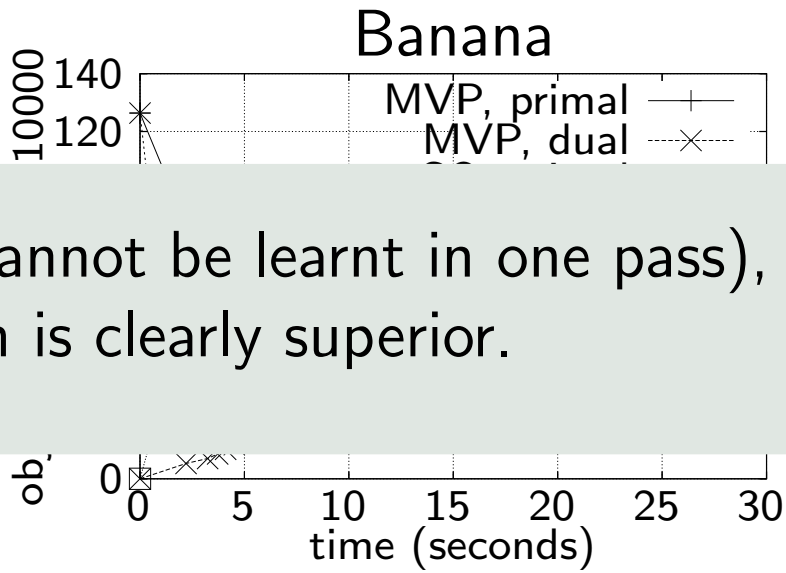
Online learning using LASVM

LASVM

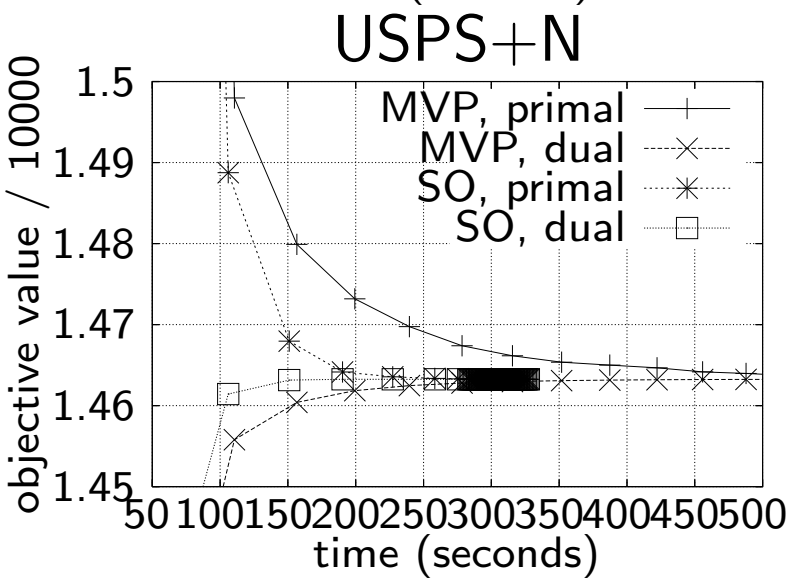
SVM with SVs M
repeat
  observe pattern (x_t, y_t)
  `// Step 1`
  select working set $B = \{t, i\}$, $i \in M$
  $M \leftarrow M \cup \{t\}$
  solve QP restricted to B
  remove non-SVs from M
  `// Step 2`
  select working set $B = \{i, j\}$, $i, j \in M$
  solve QP restricted to B
  remove non-SVs from M
until *stopping criterion is met*

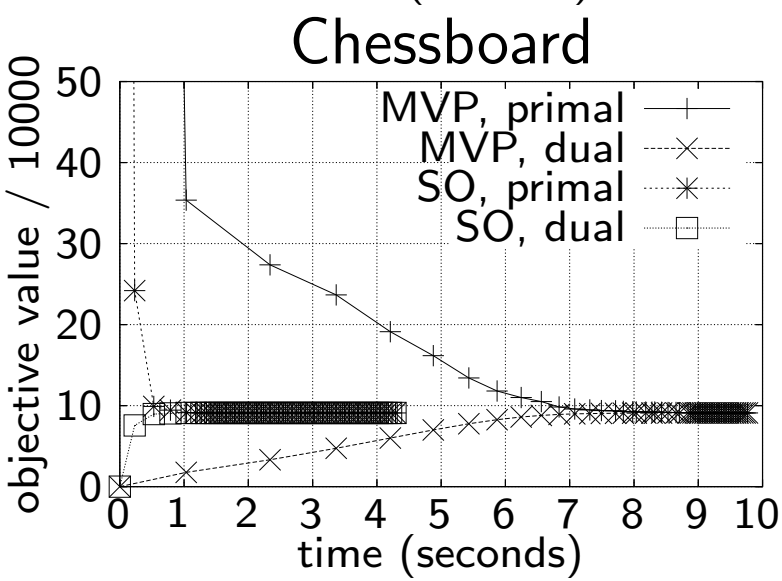
Online SVMs results I





- For difficult data sets (i.e., that cannot be learnt in one pass), second-order working set selection is clearly superior.





Online SVMs results II

dataset	algorithm	#SV	cl. rate	primal	dual
USPS+N	MVP	3870	99.09%	17,214	14,207
$\ell = 7,329$	SO	2501	99.36%	15,867	14,454
10 partitions	LIBSVM	2747	99.51%	14,635	14,633
Chessboard	MVP	3395	97.86%	782,969	60,771
$\ell = 5000$	SO	944	98.91%	143,614	83,686
1 partition	LIBSVM	873	99.40%	90,959	90,946

Second-order working set selection makes better decisions, which lead to more efficient solutions in terms of

- classification error, and
- number of support vectors.

Summary

- Machine learning requires solving optimization problems, and knowledge in mathematical optimization helps to scale-up learning algorithms.
- Decomposition algorithms are state-of-the-art for training *non-linear* SVMs.
- Decomposition algorithms can be used for online and active learning.
- Second-order working set selection is in general better than first order methods.

References I

Overview

J. Shawe-Taylor and S. Sun. A review of optimization methodologies in support vector machines. Neurocomputing 74(17): 3609–3618, 2011.

L. Bottou and C.-J. Lin. Support Vector Machine Solvers. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, eds.: Large Scale Kernel Machines, MIT Press, 2007.

SMO classics

T. Joachims. Making Large-Scale SVM Learning Practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, eds.: Advances in Kernel Methods – Support Vector Learning, pp. 185–208, MIT Press, 1999.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, eds.: Advances in Kernel Methods – Support Vector Learning, pp. 185–208, MIT Press, 1999.

H.-U. Simon, N. List. SVM-Optimization and Steepest-Descent Line Search. Proceedings of the 22nd Conference on Learning Theory (COLT), 2009.

References II

Second-order woking set selection

R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using the second order information for training SVM. Journal of Machine Learning Research 6:1889–1918, 2005.

T. Glasmachers and C. Igel. Maximum-Gain Working Set Selection for SVMs. Journal of Machine Learning Research 7: 1437–1466, 2006.

SVM online and active learning

A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast Kernel Classifiers with Online and Active Learning. *Journal of Machine Learning Research* 6:1579-1619, 2005

T. Glasmachers and C. Igel. Second-Order SMO Improves SVM Online and Active Learning, *Neural Computation* 20(2):374–382, 2008

Active learning and selection bias

J. Kremer, F. Gieseke, K. Steenstrup Pedersen, and C. Igel. Nearest Neighbor Density Ratio Estimation for Large-Scale Applications in Astronomy. *Astronomy and Computing* 12:67–72, 2015

Shark

`image.diku.dk/shark`

Igel, Glasmachers, Heidrich-Meisner: Shark, *Journal of Machine Learning Research* 9:993–996, 2008

Gold Prize at Open Source Software Wold Challenge 2011