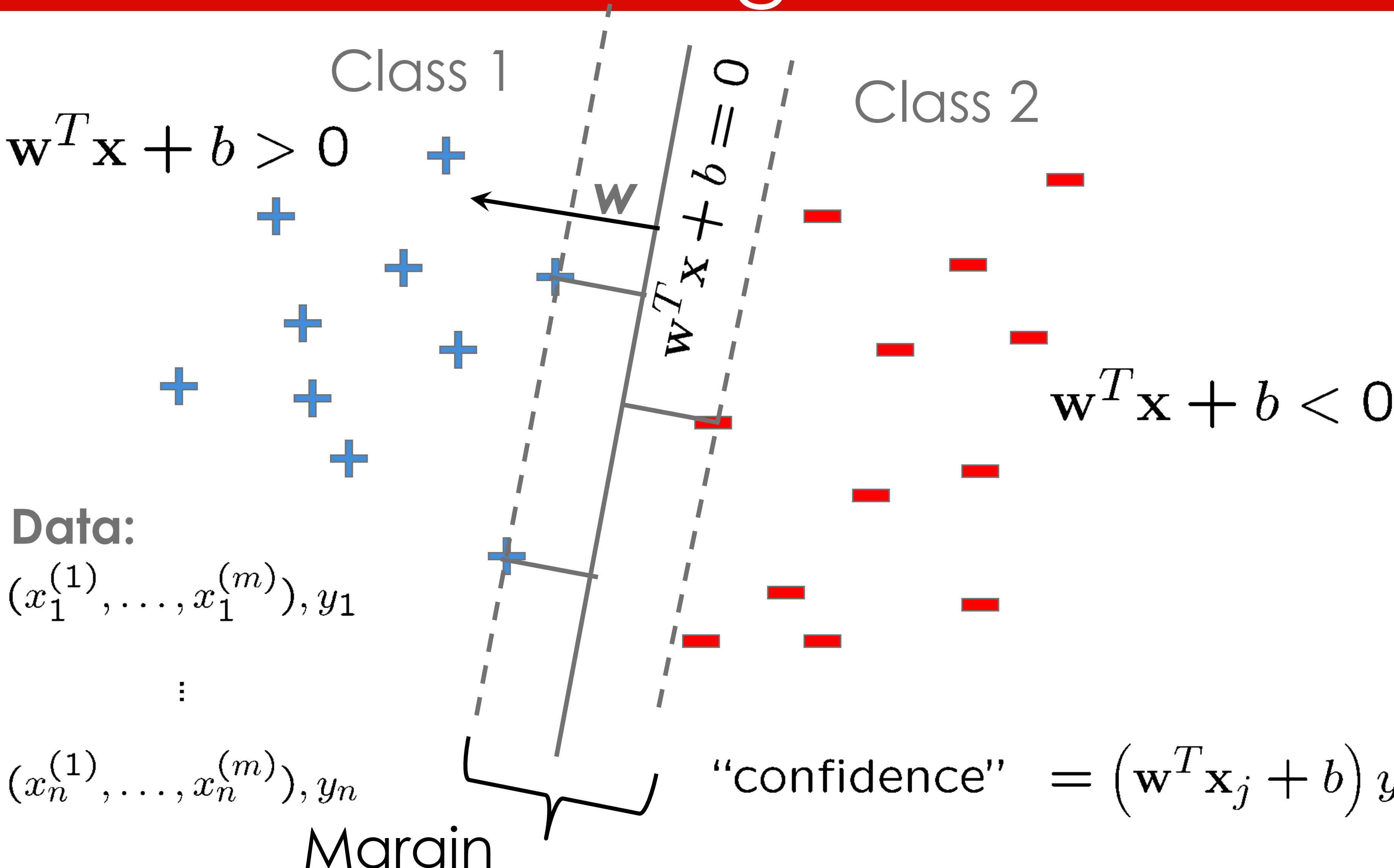
Scalable ML

10605-10805

Support Vector Machines

Barnabás Póczos

Pick the one with the largest margin!



Data:

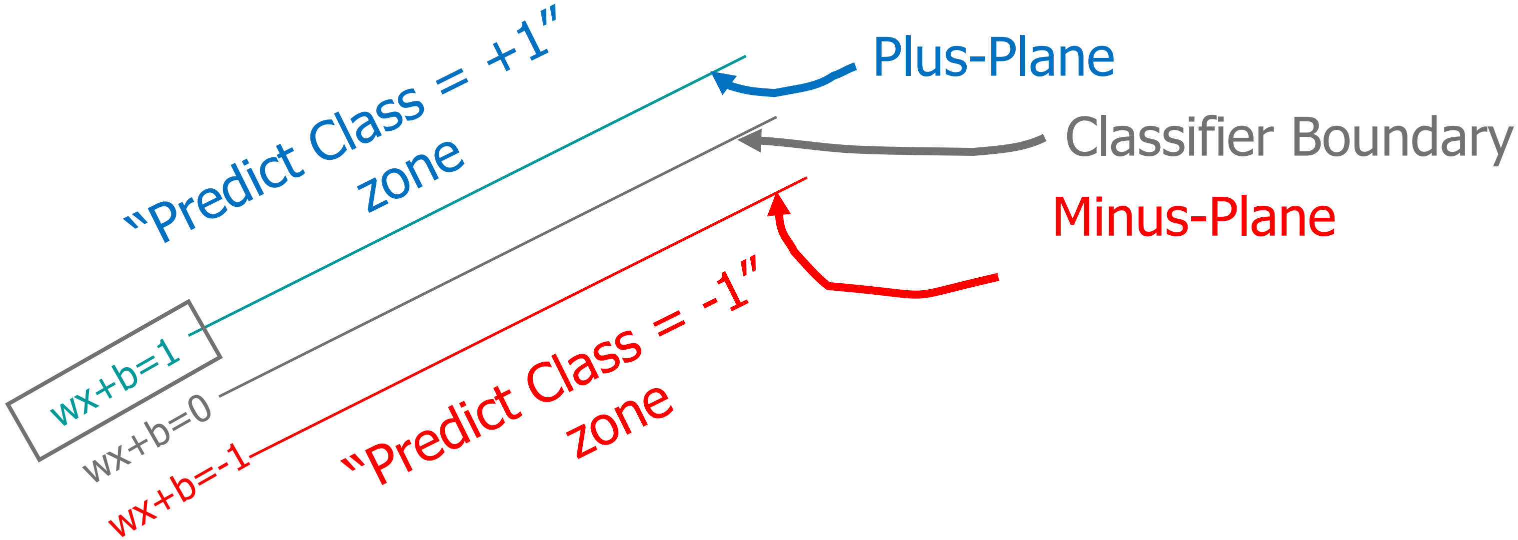
$$(x_1^{(1)}, \ldots, x_1^{(m)}), y_1$$

$$\vdots$$

$$(x_n^{(1)}, \ldots, x_n^{(m)}), y_n$$

"confidence" $= \left(\mathbf{w}^T\mathbf{x}_j + b\right) y_j$

Scaling



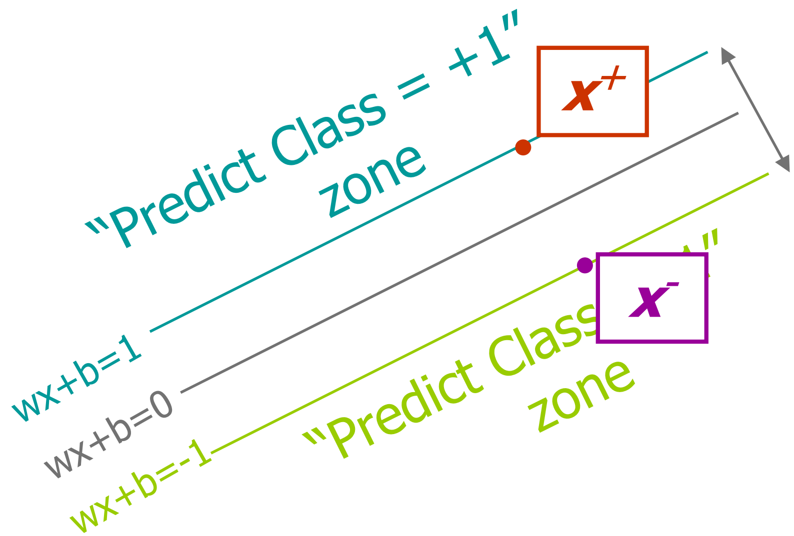
Classification rule:

Classify as..	+1	if	$\mathbf{w}^T\mathbf{x} + b \geq 1$
	–1	if	$\mathbf{w}^T\mathbf{x} + b \leq -1$
	Universe explodes	if	$-1 < \mathbf{w}^T\mathbf{x} + b < +1$

Goal: Find the maximum margin classifier

How large is the margin of this classifier?

Computing the margin width



M = Margin Width

$$\mathbf{w}^T(\underbrace{\mathbf{x}^+ - \mathbf{x}^-}_{\lambda\mathbf{w}}) = 2$$

$$\Rightarrow \lambda = \frac{2}{\mathbf{w}^T\mathbf{w}}$$

$$\Rightarrow M = \|\mathbf{x}^+ - \mathbf{x}^-\|$$

$$= \|\lambda\mathbf{w}\|$$

$$= \frac{\|2\mathbf{w}\|}{\mathbf{w}^T\mathbf{w}} = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$$

Let $\mathbf{x}^+$ and $\mathbf{x}^-$ be such that

$$\mathbf{w}^T\mathbf{x}^+ + b = 1$$

$$\mathbf{w}^T\mathbf{x}^- + b = -1$$

$$\mathbf{x}^+ = \mathbf{x}^- + \lambda\mathbf{w}$$

$$\|\mathbf{x}^+ - \mathbf{x}^-\| = M = ?(\text{Margin})$$

Maximize $M \equiv$ minimize w^Tw **!**

The Primal Hard SVM

- Given $D = \{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ training data set.
- Assume that D is **linearly separable**.

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{subject to } y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1, \ \forall i = 1, \ldots, n$$

Prediction: $f_{\hat{\mathbf{w}}}(\mathbf{x}) = \text{sign}(\langle \hat{\mathbf{w}}, \mathbf{x} \rangle)$

This is a QP problem (m-dimensional)
(Quadratic cost function, linear constraints)

Quadratic Programming

Find

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^m} \mathbf{w}^T H \mathbf{w} + \mathbf{w}^T \mathbf{q} + e$$

subject to

$$A\mathbf{w} \leq \mathbf{b}, \qquad A \in \mathbb{R}^{n\times m}, \mathbf{b} \in \mathbb{R}^n$$
$$C\mathbf{w} = \mathbf{d}, \qquad C \in \mathbb{R}^{s\times m}, \mathbf{d} \in \mathbb{R}^s$$

Efficient Algorithms exist for QP.
They often solve the dual problem instead of the primal.

The Dual Hard SVM

$$\boldsymbol{Y} \doteq diag(y_1, \ldots, y_n),\ \ y_i \in \{-1, 1\}^n$$

$\boldsymbol{G} \in \mathbb{R}^{n \times n} \doteq \{G_{ij}\}_{i,j}^{n,n}$, where $G_{ij} \doteq \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ Gram matrix.

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \boldsymbol{\alpha}^T \mathbf{1}_n - \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Y} \boldsymbol{G} \boldsymbol{Y} \boldsymbol{\alpha}$$

subject to $\alpha_i \geq 0,\ \forall i = 1, \ldots, n$

Quadratic Programming (n-dimensional)

Lemma $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$

Prediction: $f_{\hat{\mathbf{w}}}(x) = \text{sign}(\langle \hat{\mathbf{w}}, \mathbf{x} \rangle) = \text{sign}(\sum_{i=1}^{n} \hat{\alpha}_i y_i \underbrace{\langle \mathbf{x}_i, \mathbf{x} \rangle}_{k(\mathbf{x}_i, \mathbf{x})})$

The Problem with Hard SVM

It assumes samples are linearly separable...

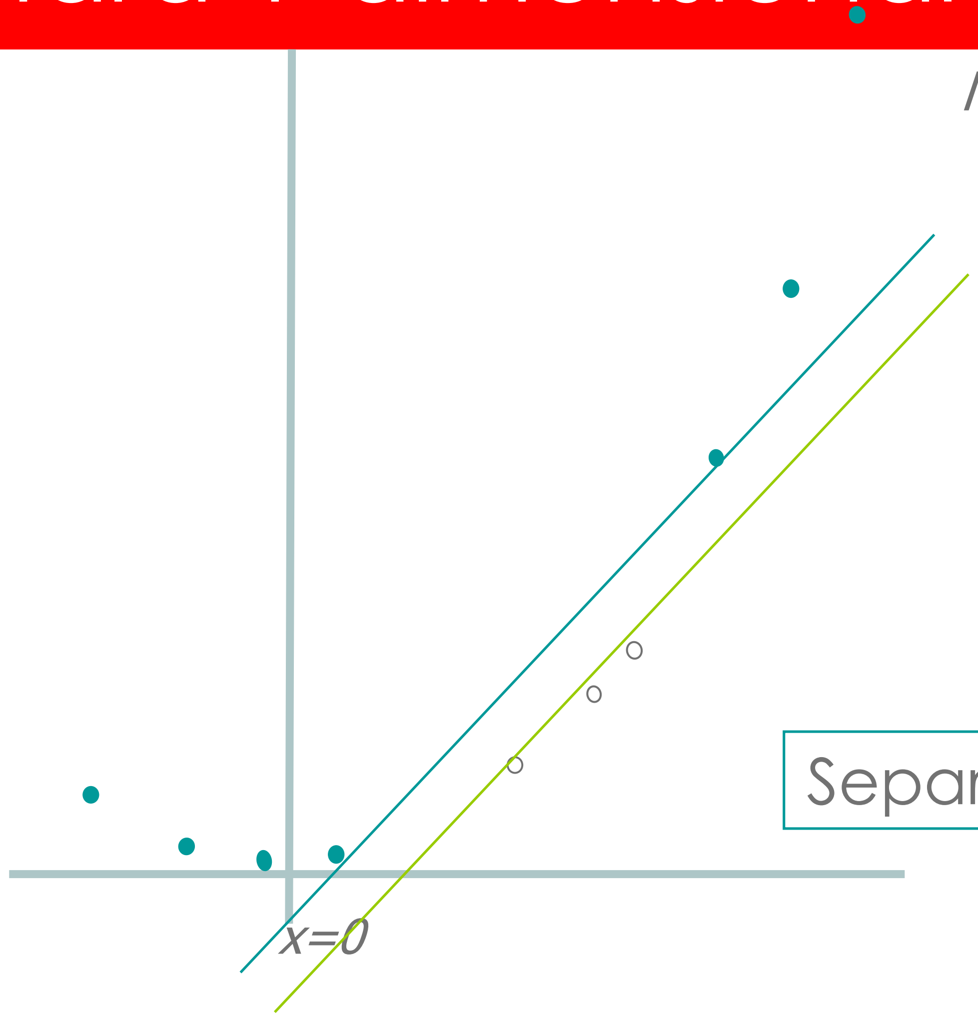
What can we do if data is not linearly separable???

Hard 1-dimensional Dataset

If the data set is **not** linearly separable, then adding new features (mapping the data to a larger feature space) the data might become linearly separable

Hard 1-dimensional Dataset

Make up new features!



$$\mathbf{z}_k = (x_k, x_k^2)$$

Separable! MAGIC!

Now drop this "augmented" data into our linear SVM.

How to do feature mapping?

Let the original features be denoted by $\mathbf{x} = [\ x_1, \mathbf{x}_2] \in \mathbb{R}^2$

Let $\phi(\mathbf{x}) \doteq \underbrace{[sin(\mathbf{x}_2), \exp(\mathbf{x}_2 + \mathbf{x}_1), \mathbf{x}_1, \mathbf{x}_2^{\tan(\mathbf{x}_1)}, \ldots]}_{\infty}$

Use features of features of features of features....

The Problem with Hard SVM

It assumes samples are linearly separable...

Solutions:

1. Use feature transformation to a larger space
   $\Rightarrow$ training samples are linearly separable in the high dim feature space
   $\Rightarrow$ Hard SVM can be applied ☺
   $\Rightarrow$ overfitting... ☹

2. **Soft margin** SVM instead of Hard SVM
   - Slack variables... We will discuss them now

Hard SVM

The Hard SVM problem can be rewritten:

$$\widehat{\mathbf{w}}_{hard} = \arg\min_{\mathbf{w}\in\mathbb{R}^m} \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } y_i\langle \mathbf{x}_i, \mathbf{w}\rangle \geq 1,\ \forall i = 1,\dots,n$$

$$\Downarrow$$

$$\boxed{\widehat{\mathbf{w}}_{hard} = \arg\min_{\mathbf{w}\in\mathbb{R}^m} \sum_{i=1}^{n} l_{0-\infty}(\langle \mathbf{x}_i, \mathbf{w}\rangle, y_i) + \frac{1}{2}\|\mathbf{w}\|^2}$$

where

$$l_{0-\infty}(a,b) \doteq \begin{cases} \infty : ab < 1 & \text{Misclassification, or inside the margin} \\ 0 : ab \geq 1 & \text{Correct classification and outside of the margin} \end{cases}$$

From Hard to Soft constraints

Instead of using hard constraints (points are linearly separable)

$$\hat{\mathbf{w}}_{hard} = \arg\min_{\mathbf{w}\in\mathbb{R}^m} \sum_{i=1}^{n} l_{0-\infty}(\langle \mathbf{x}_i, \mathbf{w}\rangle, y_i) + \frac{1}{2}\|\mathbf{w}\|^2$$

We can try solve the soft version of it: Introduce a λ parameter, and let your loss be only 1 instead of ∞ if you misclassify an instance

$$\hat{\mathbf{w}}_{soft} = \arg\min_{\mathbf{w}\in\mathbb{R}^m} \sum_{i=1}^{n} l_{0-1}(\langle \mathbf{x}_i, \mathbf{w}\rangle, y_i) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where

$$l_{0-1}(y, f(\mathbf{x})) = \begin{cases} 1 : yf(\mathbf{x}) < 1 \\ 0 : yf(\mathbf{x}) \geq 1 \end{cases}$$

Misclassification, or inside the margin

Correct classification and outside of the margin

Problems with $l_{0\text{-}1}$ loss

$$\widehat{\mathbf{w}}_{soft} = \arg\min_{\mathbf{w}\in\mathbb{R}^m} \sum_{i=1}^{n} l_{0-1}(\langle \mathbf{x}_i, \mathbf{w}\rangle, y_i) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

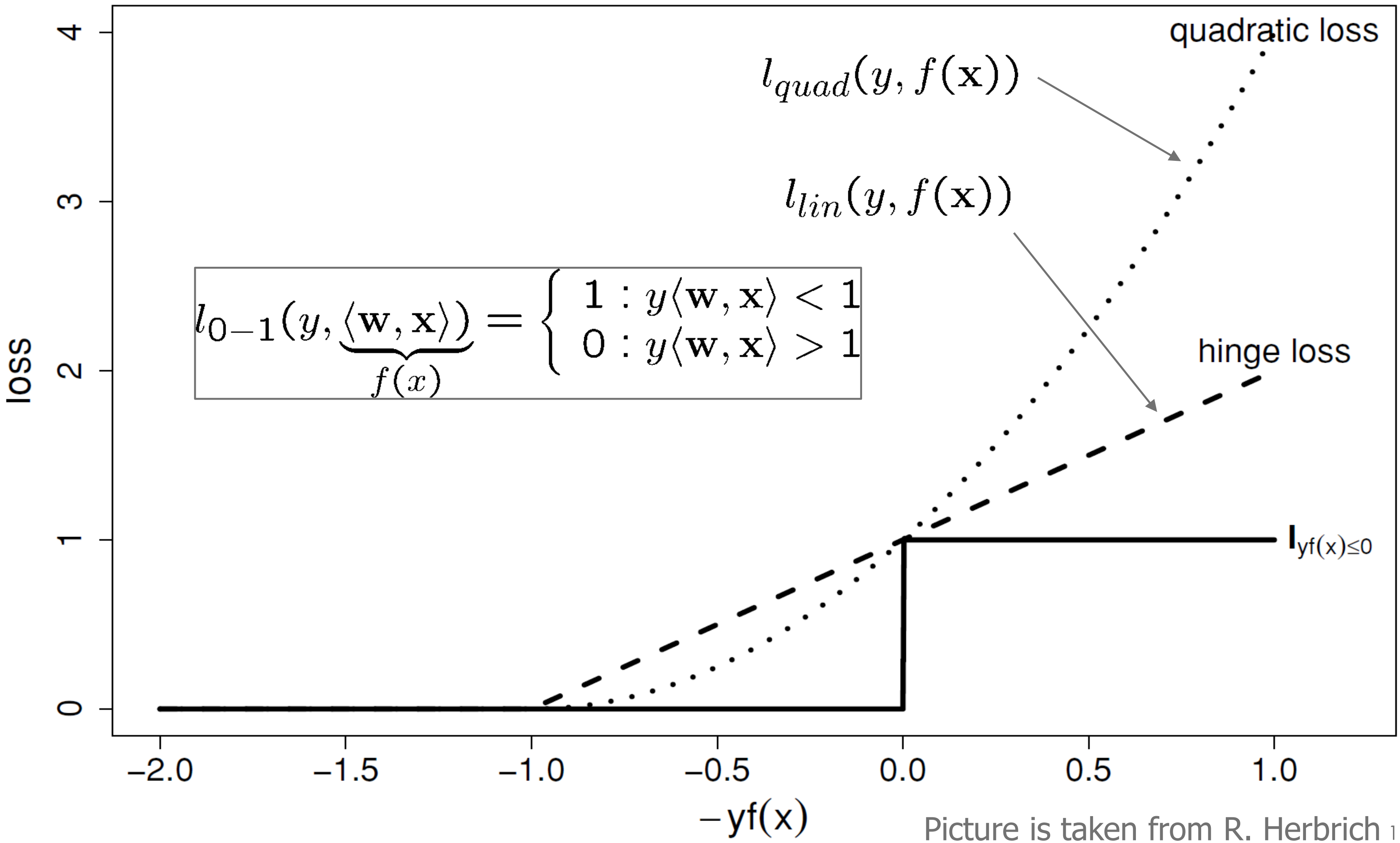
$$l_{0-1}(y, f(\mathbf{x})) = \begin{cases} 1 : yf(\mathbf{x}) < 1 \\ 0 : yf(\mathbf{x}) \geq 1 \end{cases}$$

It is not convex in **w**...

... and we like convex functions...

Let us approximate it with convex functions!

Approximation of the Heaviside step function



Picture is taken from R. Herbrich

The hinge loss approximation of $l_{0\text{-}1}$

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}\in\mathbb{R}^m} \sum_{i=1}^{n} \underbrace{l_{lin}(\langle \mathbf{x}_i, \mathbf{w}\rangle, y_i)}_{\xi_i \geq 0} + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Where,

$$\xi_i \doteq l_{lin}(f(\mathbf{x}_i), y_i) = \max\{1 - y_i f(\mathbf{x}_i), 0)\}$$

The Primal Soft SVM problem

$$\hat{\mathbf{w}}_{soft} = \arg \min_{\mathbf{w}\in\mathbb{R}^m} \sum_{i=1}^{n} \underbrace{l_{lin}(\langle \mathbf{x}_i, \mathbf{w} \rangle, y_i)}_{\xi_i \geq 0} + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where

$$\xi_i \doteq l_{lin}(f(\mathbf{x}_i), y_i) = \max\{1 - y_i(\mathbf{w}^T \mathbf{x}_i), 0\}$$

Equivalently,

$$\hat{\mathbf{w}}_{soft} = \arg \min_{\mathbf{w}\in\mathbb{R}^m, \boldsymbol{\xi}\in\mathbb{R}^n} \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

subject to $y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1 - \xi_i, \ \forall i = 1, \ldots, n$

$\xi_i \geq 0, \ \forall i = 1, \ldots, n$

ξ_i: Slack variables

The Primal Soft SVM problem

$$\widehat{\mathbf{w}}_{soft} = \arg \min_{\mathbf{w}\in\mathbb{R}^m, \boldsymbol{\xi}\in\mathbb{R}^n} \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } y_i\langle \mathbf{x}_i, \mathbf{w}\rangle \geq 1 - \xi_i, \; \forall i = 1, \ldots, n$$

$$\xi_i \geq 0, \; \forall i = 1, \ldots, n$$

We can use this form, too... where $C = \frac{1}{\lambda}$

$$\widehat{\mathbf{w}}_{soft} = \arg \min_{\mathbf{w}\in\mathbb{R}^m, \boldsymbol{\xi}\in\mathbb{R}^n} C \underbrace{\sum_{i=1}^{n} \xi_i}_{\boldsymbol{\xi}^T \mathbf{1}_n} + \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } y_i\langle \mathbf{x}_i, \mathbf{w}\rangle \geq 1 - \xi_i, \; \forall i = 1, \ldots, n$$

$$\xi_i \geq 0, \; \forall i = 1, \ldots, n$$

What is the dual form of primal soft SVM?

The Dual Soft SVM (using hinge loss)

$\boldsymbol{Y} \doteq diag(y_1, \ldots, y_n) \in \{-1, 1\}^n$

$\boldsymbol{G} \in \mathbb{R}^{n \times n} \doteq \{G_{ij}\}_{i,j}^{n,n}$, where $G_{ij} \doteq \overbrace{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}^{k(\mathbf{x}_i, \mathbf{x}_j)}$, Gram matrix.

$$\hat{\alpha} = \arg\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \boldsymbol{\alpha}^T \mathbf{1}_n - \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Y} \boldsymbol{G} \boldsymbol{Y} \boldsymbol{\alpha}$$

subject to $0 \leq \alpha_i \leq C$

where $C = \frac{1}{\lambda}$

If $\lambda \to \mathbf{0} \Rightarrow$ soft-SVM $\to$ hard-SVM

This is the same as the dual hard-SVM problem,
but now we have the additional $0 \leq \alpha_i \leq C$ constraints.

SVM classification in the dual space

Solve the dual problem

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \boldsymbol{\alpha}^T \mathbf{1}_n - \tfrac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{YGY}\boldsymbol{\alpha}$$

$$\text{subject to } 0 \le \alpha_i \le C$$

where $C = \frac{1}{\lambda}$.

Let $\widehat{\mathbf{w}} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i \mathbf{x}_i$.

On test data $\mathbf{x}$: $f_{\widehat{\mathbf{w}}}(\mathbf{x}) = \langle \widehat{\mathbf{w}}, \mathbf{x} \rangle = \sum_{i=1}^{n} \widehat{\alpha}_i y_i \underbrace{\langle \mathbf{x}_i, \mathbf{x} \rangle}_{k(\mathbf{x}_i, \mathbf{x})}$

Constructing Kernels

Common Kernels

- Polynomials of degree d

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v})^d$$

- Polynomials of degree up to d

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v} + 1)^d$$

- Gaussian/Radial kernels

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{||\mathbf{u} - \mathbf{v}||^2}{2\sigma^2}\right)$$

Designing new kernels from kernels

$k_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are kernels $\Rightarrow$

1. $k(x, \tilde{x}) = k_1(x, \tilde{x}) + k_2(x, \tilde{x})$,
2. $k(x, \tilde{x}) = c \cdot k_1(x, \tilde{x})$, *for all* $c \in \mathbb{R}^+$,
3. $k(x, \tilde{x}) = k_1(x, \tilde{x}) + c$, *for all* $c \in \mathbb{R}^+$,
4. $k(x, \tilde{x}) = k_1(x, \tilde{x}) \cdot k_2(x, \tilde{x})$,
5. $k(x, \tilde{x}) = f(x) \cdot f(\tilde{x})$, *for any function* $f : \mathcal{X} \to \mathbb{R}$

are also kernels.

Picture is taken from R. Herbrich

Designing new kernels from kernels

1. $k(x,\tilde{x}) = (k_1(x,\tilde{x}) + \theta_1)^{\theta_2}$, *for all* $\theta_1 \in \mathbb{R}^+$ *and* $\theta_2 \in \mathbb{N}$

2. $k(x,\tilde{x}) = \exp\left(\frac{k_1(x,\tilde{x})}{\sigma^2}\right)$, *for all* $\sigma \in \mathbb{R}^+$,

3. $k(x,\tilde{x}) = \exp\left(-\frac{k_1(x,x) - 2k_1(x,\tilde{x}) + k_1(\tilde{x},\tilde{x})}{2\sigma^2}\right)$, *for all* $\sigma \in \mathbb{R}^+$

4. $k(x,\tilde{x}) = \frac{k_1(x,\tilde{x})}{\sqrt{k_1(x,x) \cdot k_1(\tilde{x},\tilde{x})}}$

Picture is taken from R. Herbrich

$dim(\mathcal{X}) = N$

Name	Kernel function	dim ($\mathcal{K}$)
pth degree polynomial	$k(\vec{u}, \vec{v}) = (\langle \vec{u}, \vec{v} \rangle_{\mathcal{X}})^p$ $p \in \mathbb{N}^+$	$\binom{N+p-1}{p}$
complete polynomial	$k(\vec{u}, \vec{v}) = (\langle \vec{u}, \vec{v} \rangle_{\mathcal{X}} + c)^p$ $c \in \mathbb{R}^+, \; p \in \mathbb{N}^+$	$\binom{N+p}{p}$
RBF kernel	$k(\vec{u}, \vec{v}) = \exp\left(-\frac{\|\vec{u}-\vec{v}\|_{\mathcal{X}}^2}{2\sigma^2}\right)$ $\sigma \in \mathbb{R}^+$	∞
Mahalanobis kernel	$k(\vec{u}, \vec{v}) = \exp\left(-(\vec{u}-\vec{v})' \boldsymbol{\Sigma} (\vec{u}-\vec{v})\right)$ $\boldsymbol{\Sigma} = \mathrm{diag}\left(\sigma_1^{-2}, \ldots, \sigma_N^{-2}\right)$, $\sigma_1, \ldots, \sigma_N \in \mathbb{R}^+$	∞

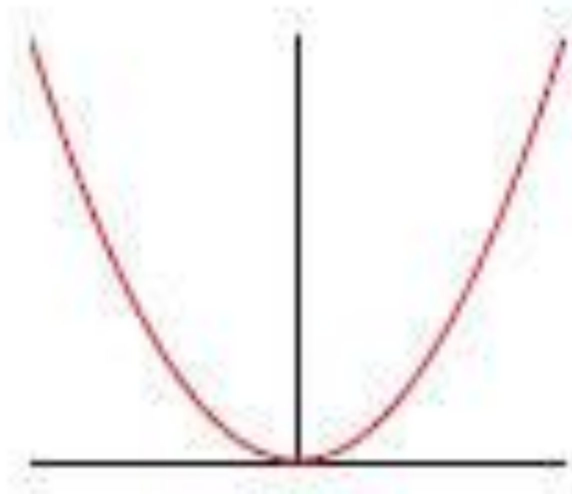
Picture is taken from R. Herbrich

SVM for Regression

SVM for Regression

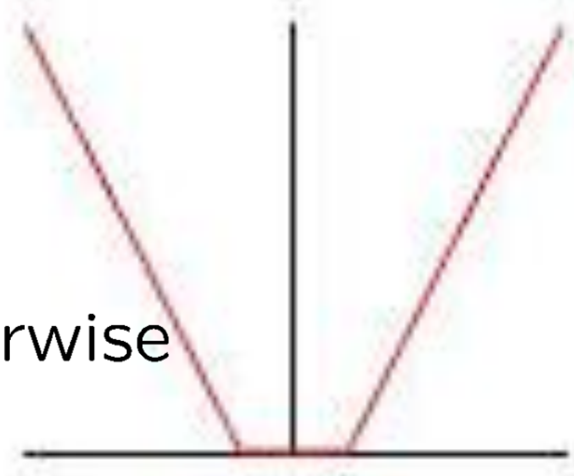
Quadratic loss

$$Loss(y, \mathbf{w}^T\mathbf{x}) = (y - \mathbf{w}^T\mathbf{x})^2$$



ϵ - insensitive loss

$$Loss(y, \mathbf{w}^T\mathbf{x}) = \begin{cases} 0, & \text{if } |y - \mathbf{w}^T\mathbf{x}| \leq \epsilon \\ |y - \mathbf{w}^T\mathbf{x}| - \epsilon & \text{otherwise} \end{cases}$$



Ridge Regression

Linear regression: $f(x) = \langle \mathbf{w}, \phi(x) \rangle$

Primal:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathcal{K}} \sum_{i=1}^{n} \xi_i^2$$

$$\text{subject to } y_i - \langle \underbrace{\phi(x_i)}_{\mathbf{x}_i}, \mathbf{w} \rangle = \xi_i, \ \forall i = 1, \ldots, n$$

$$\text{and } \|\mathbf{w}\| \leq B$$

Kernel Ridge Regression Algorithm

Dual:

Given $D = \{(x_i, y_i), i = 1, \dots, n\}$ training data set.

$k(\cdot,\cdot)$ kernel, $\lambda > 0$ parameter. $\mathbf{y} \doteq (y_1, \dots, y_n)^T \in \mathbb{R}^n$

- $\boldsymbol{G} \in \mathbb{R}^{n\times n} \doteq \{G_{ij}\}_{i,j}^{n,n}$,

  where $G_{ij} \doteq \overbrace{\langle \underbrace{\mathbf{x}_i}_{\phi(x_i)}, \underbrace{\mathbf{x}_j}_{\phi(x_j)} \rangle_{\mathcal{K}}}^{k(x_i,x_j)}$, Gram matrix.
- $\hat{\boldsymbol{\alpha}} = (\boldsymbol{G} + \lambda \boldsymbol{I}_n)^{-1}\mathbf{y}$
- $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i \phi(x_i)$.
- $f(x) = \langle \hat{\mathbf{w}}, \phi(x) \rangle = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, x)$

Distribution kernels

Euclidean:

$$K(p,q) = \int p(x)q(x)\,dx$$

Bhattacharyya's affinity:

$$K(p,q) = \int \sqrt{p(x)}\sqrt{q(x)}\,dx$$

Mean map:

$$K(p,q) = \mathbb{E}_{x\sim p}\mathbb{E}_{y\sim q}k(x,y)$$

$$\phi(p) = \mathbb{E}_{x\sim p}\left[k(\cdot,x)\right]$$

Set kernels

Mean map:

$$\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i, y_j) = \langle \{x_1, \ldots, x_n\}, \{y_1, \ldots, y_m\} \rangle$$

Intersection kernel:

$$k(A_1, A_2) = \int I_{A_1 \cap A_2}(x)\, dx = \mu(A_1 \cap A_2)$$

Union complement kernel: $1 - \mu(A_1 \cup A_2), \quad \mu(\Omega) = 1$

String kernels

P-spectrum kernel:

P=3:   s="statistics"   t="computation"

They contain the following substrings of length 3

"sta", "tat", "ati", "tis", "ist", "sti", "tic", "ics"
"com", "omp", "mpu", "put", "uta", "tat", "ati", "ti

Common substrings: "tat", "ati"

k(s,t)=2

Thanks for your attention ☺