COMP 790-125: Goals for today

Max margin methods and kernelization

- SVMs in separable case
- Kernelization
- SVMs in non-separable cases
- Optimization – Pegasos

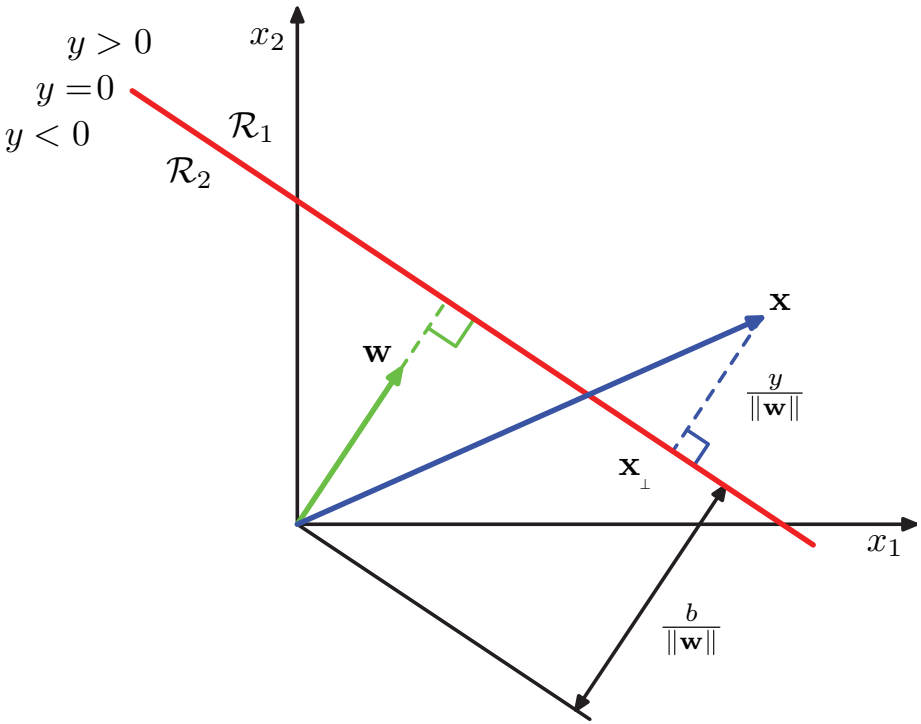
Supervised learning context

We are given training data instances $(y_t, \mathbf{x}_t)$ where $t = 1, \ldots T$ and $y_t \in \mathcal{Y}, \mathbf{x}_t \in \mathcal{X}$.

We want to learn a rule for predicting label $y \in \mathcal{Y}$ given its corresponding feature vector $\mathbf{x} \in \mathcal{X}$.

For now a single binary label, so $\mathcal{Y} = \{-1, +1\}$.

Distances and projections

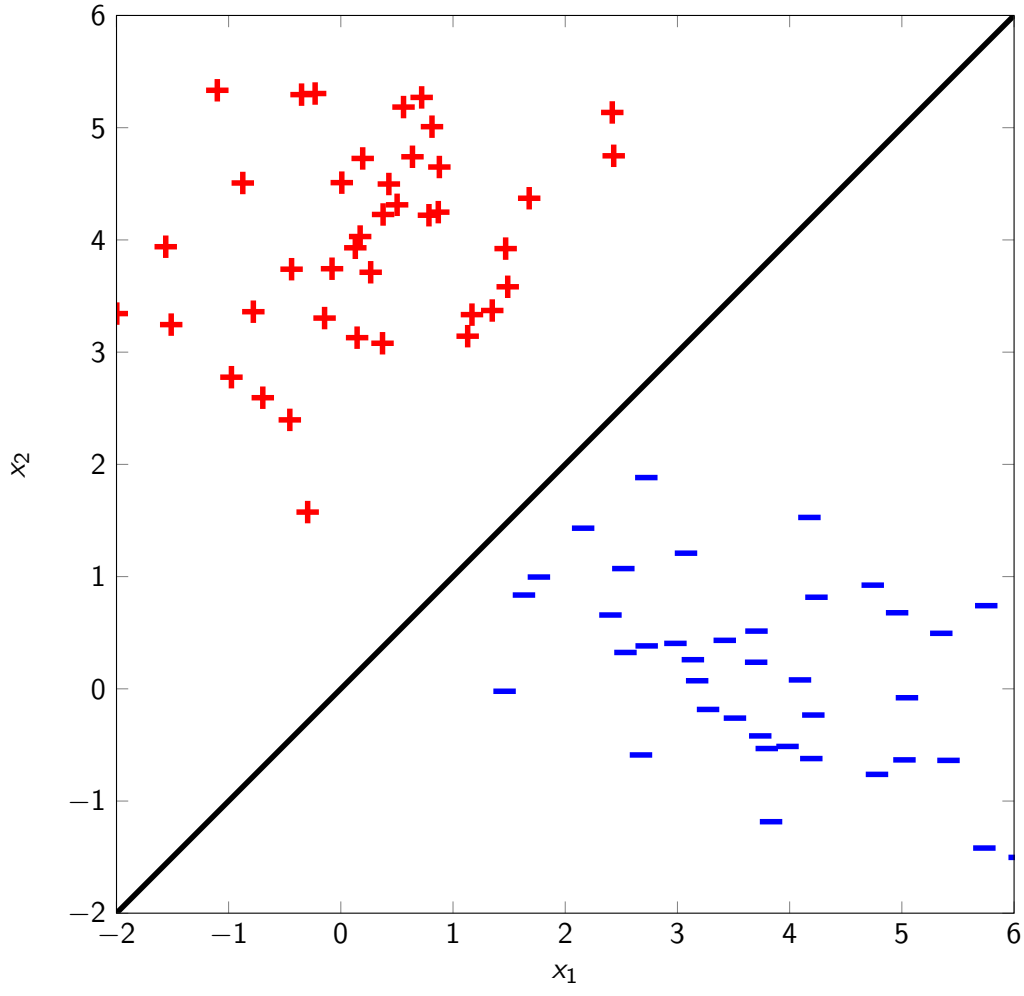


$$
\begin{aligned}
y(x) &= w^T x + b \\
w^T x_\perp + b &= 0 \\
x_\perp + r\frac{w}{\|w\|} &= x
\end{aligned}
$$

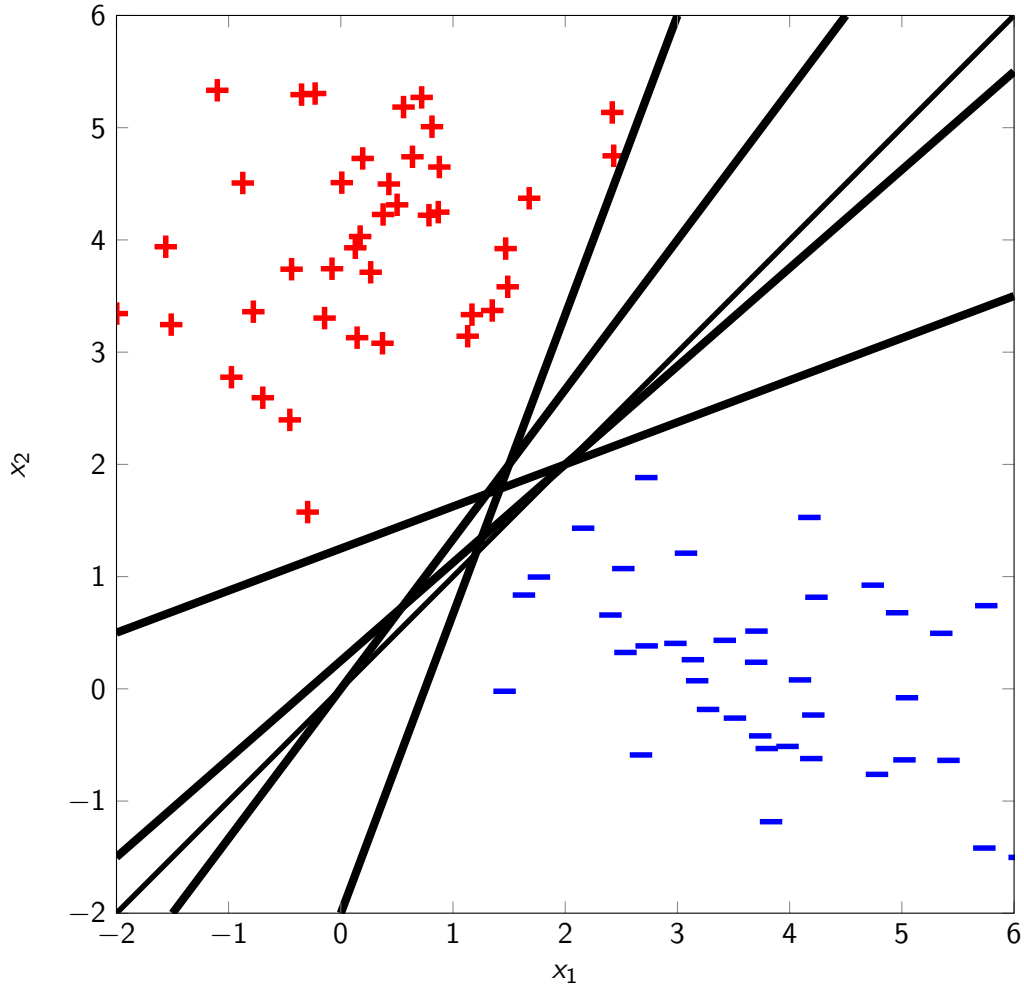
We wish to compute distance r from x to hyperplane defined by $y(x) = 0$

$$
\begin{aligned}
wx_\perp + b + r\frac{w^T w}{\|w\|} &= w^T x + b \\
r\,\|w\| &= w^T x + b \\
r &= \frac{w^T x + b}{\|w\|} \\
r &= \frac{y(x)}{\|w\|}
\end{aligned}
$$

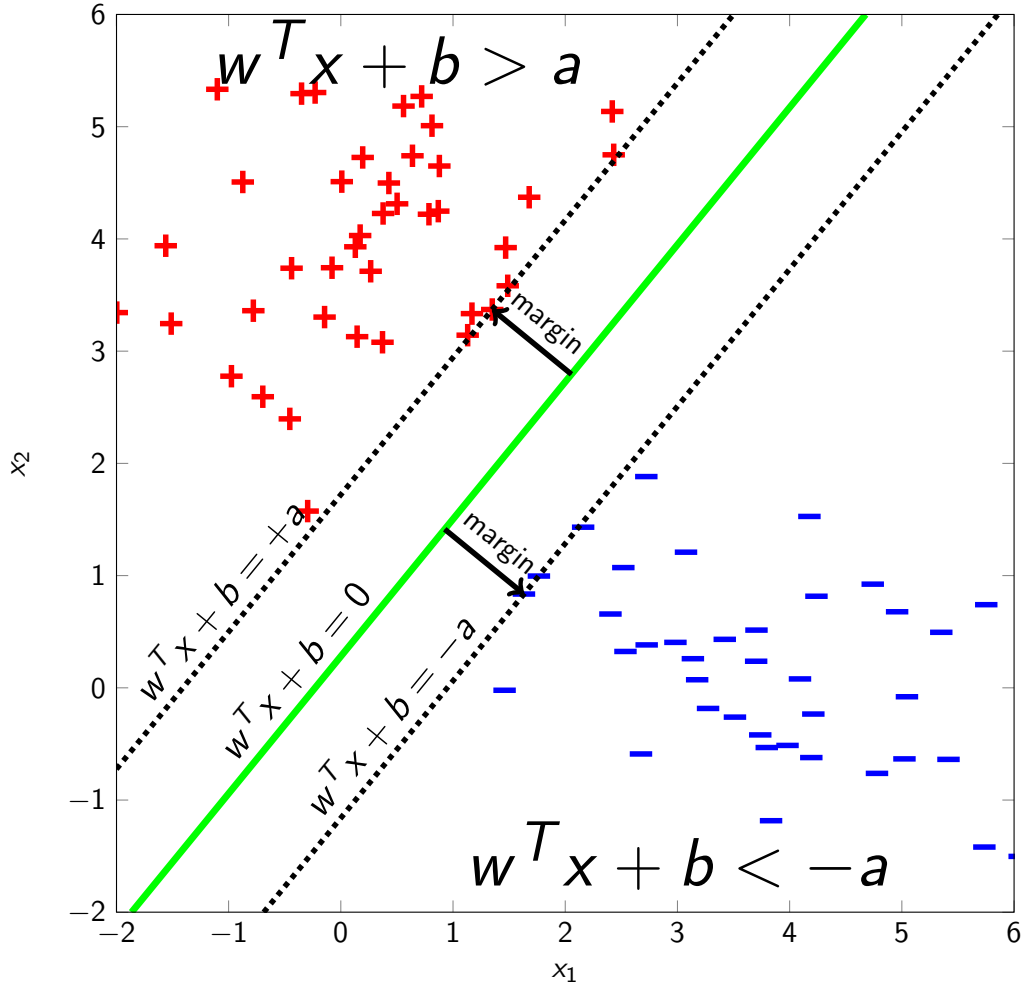
Separating hyperplane



Separating hyperplanes



Separating hyperplane and margin



Separating and supporting hyperplanes

Three relevant parallel hyperplanes

- $w^T \mathbf{x} + b = 0$ the actual separating hyperplane
- $w^T \mathbf{x} + b = -a$ passes through a negative example (supports negative set)
- $w^T \mathbf{x} + b = +a$ passes through a positive example (supports positive set)

Separating hyperplane

All positive examples $w^T \mathbf{x} + b \geq +a$ and all negative examples $w^T \mathbf{x} + b \leq -a$.

More formally, for any instance $(\mathbf{x}_t, y_t)$ in training set ($t \in \text{Train}$)

$$\begin{aligned} w^T \mathbf{x}_t + b &\geq +a, \text{if } y_t = +1 \\ w^T \mathbf{x}_t + b &\leq -a, \text{if } y_t = -1 \end{aligned}$$

and then we can say that we have a separating hyperplane with margin $\frac{|a|}{\|w\|}$.

However, this parameterization is overcomplete: multiplying equations by a constant $(1/a)$ gives the same planes *and* does not change margin.

Simpler parameterization

We can write the constraints on the separating hyperplane in terms of positive and negative examples as

$$\begin{aligned} w^T \mathbf{x}_t + b &\geq +1, \text{if } y_t = +1 \\ w^T \mathbf{x}_t + b &\leq -1, \text{if } y_t = -1 \end{aligned}$$

or equivalently

$$y_t(w^T \mathbf{x}_t + b) \geq 1, \forall t \in \text{Train}$$

This defines the constraints that we want the separating hyperplane to honor.
And we note that after simplification the margin is $\frac{1}{\|w\|}$

Optimization problem

$$\begin{aligned} \text{maximize} \quad & \frac{1}{\|w\|} = \frac{1}{\sqrt{\sum_i w_i^2}} \\ \text{subject to} \quad & y_t(w^T \mathbf{x}_t + b) - 1 \geq 0, \forall t \in \text{Train} \end{aligned}$$

or equivalently

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & 1/2 \, \|w\|_2^2 = 1/2 \sum_i w_i^2 \\ \text{subject to} \quad & y_t(w^T \mathbf{x}_t + b) - 1 \geq 0, \forall t \in \text{Train} \end{aligned}$$

Dual problem

Lagrangian

$$L(w, b, \alpha) = 1/2 \, \|w\|_2^2 - \sum_t \alpha_t (y_t (w^T \mathbf{x}_t + b) - 1)$$

where $\alpha_t \geq 0$ for all t

At saddle point

$$\begin{aligned} w - \sum_t \alpha_t y_t x_t &= 0 \qquad \left(\frac{\partial}{\partial w} L(w, b, \alpha) = 0 \right) \\ \sum_t \alpha_t y_t &= 0 \qquad \left(\frac{\partial}{\partial b} L(w, b, \alpha) = 0 \right) \end{aligned}$$

and after plugging in $w = \sum_t \alpha_t y_t x_t$ into the Lagrangian we obtain the dual problem. Note also that we obtain term $b \sum_t \alpha_t y_t$ but under the constraint this is 0.

Dual problem

$$\begin{aligned} \text{maximize}_{\alpha} \sum_{t=1} \alpha_t - 1/2 \sum_{t=1} \sum_{r=1} \alpha_t \alpha_r y_t y_r k(\mathbf{x}_t, \mathbf{x}_r) & \\ \text{subject to } \sum_t \alpha_t y_t = 0 & \quad \forall t \in \text{Train} \\ \alpha_t \geq 0 & \quad \forall t \in \text{Train} \end{aligned}$$

where

$$k(\mathbf{x}_t, \mathbf{x}_r) = \mathbf{x}_t^T \mathbf{x}_r$$

an inner product of feature vectors for two instances.

The dual problem has as many variables (α_t) as there are data instances (T).

The primal problem worked on w – same dimensionality as the feature vector.[1]

[1] Ok, there was also a bias variable b but it is a scalar

Support vectors

KKT conditions derived from the Lagrangian are

$$\begin{aligned} w - \sum_t \alpha_t y_t x^t &= 0 \\ \sum_t \alpha_t y_t &= 0 \\ y_t(w^T \mathbf{x}_t + b) - 1 &\geq 0 \\ \alpha_t &\geq 0 \\ \alpha_t(y_t(w^T \mathbf{x}_t + b) - 1) &= 0 \end{aligned}$$

From complementary slackness either $y_t(w^T \mathbf{x}_t + b) = 1$ or $\alpha_t = 0$.

If $\alpha_t \neq 0$ we can tell that instance t is on one of the support hyperplanes. These data instances are **support vectors.**

Primal variables from dual

Suppose we solved the dual problem and obtained optimal α^*. How do we find optimal w^*, b^*?

$$
\begin{aligned}
w^* &= \sum_t \alpha_t^* y_t \mathbf{x}_t \\
b^* &= \frac{\sum_{t:\alpha_t^* \neq 0} y_t - {w^*}^T \mathbf{x}_t}{\sum_{t:\alpha_t^* \neq 0} 1}
\end{aligned}
$$

The optimal w^* and b^* depend only on instances that have a non-zero α_t^* – the support vectors.

Prediction

Prediction rule is then

$$f(\mathbf{x}) = \text{sign}\left(w^{*T}\mathbf{x} + b^*\right)$$

but also

$$f(\mathbf{x}) = \text{sign}\left(\sum_t \alpha_t^* y_t \mathbf{x}_t^T \mathbf{x} + \frac{\sum_{t:\alpha_t^* \neq 0} y_t - \sum_r \alpha_r^* y_r \mathbf{x}_r^T \mathbf{x}_t}{\sum_{t:\alpha_t^* \neq 0} 1}\right)$$

or

$$f(\mathbf{x}) = \text{sign}\left(\sum_t \alpha_t^* y_t k(\mathbf{x}_t, \mathbf{x}) + \frac{\sum_{t:\alpha_t^* \neq 0} y_t - \sum_r \alpha_r^* y_r k(\mathbf{x}_r, \mathbf{x}_t)}{\sum_{t:\alpha_t^* \neq 0} 1}\right)$$

Another glance – kernelization

Optimization problem

$$\begin{aligned} \text{maximize } & \textstyle\sum_{t=1} \alpha_t - 1/2 \sum_{t=1} \sum_{r=1} \alpha_t \alpha_r y_t y_r k(\mathbf{x}_t, \mathbf{x}_r) \\ \text{subject to } & \textstyle\sum_t \alpha_t y_t = 0 \qquad \forall t \in \text{Train} \\ & \alpha_t \geq 0 \qquad \forall t \in \text{Train} \end{aligned}$$

and prediction rule

$$f(\mathbf{x}) = \text{sign}\left(\sum_t \alpha_t^* y_t k(\mathbf{x}_t, \mathbf{x}) + \frac{\sum_{t:\alpha_t^* \neq 0} y_t - \sum_r \alpha_r^* y_r k(\mathbf{x}_r, \mathbf{x}_t)}{\sum_{t:\alpha_t^* \neq 0} 1} \right)$$

only refer to features through k – a **kernel.**

Kernelization

The process of taking a learning method and transforming it so it refers to features only through inner products and consequently kernels is called kernelization.

You can kernelize many methods not just SVMs (e.g. kernelized ridge regression).

What is a kernel?

Kernel $k(\mathbf{x}_r, \mathbf{x}_t)$ corresponds to an inner product in some space

$$k(\mathbf{x}_r, \mathbf{x}_t) = \phi(\mathbf{x}_r)^T \phi(\mathbf{x}_t)$$

where $\phi(\cdot)$ is mapping the original features $\mathbf{x}$ into a new (potentially infinite) space.

Deconstructing a kernel

Let's take an example of a nonlinear function on 2D space

$$f(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2$$

in order to call it a kernel we will try to divine what the $\phi(\cdot)$ should look like

$$\begin{aligned} f(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2 &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2 x_1 z_2 x_2 z_2 + x_2^2 z_2^2 \\ &= [x_1^2 \quad \sqrt{2} x_1 x_2 \quad x_2^2][z_1^2 \quad \sqrt{2} z_1 z_2 \quad z_2^2]^T \\ &= \phi(x)^T \phi(z) \end{aligned}$$

where

$$\phi(x) = [x_1^2 \quad \sqrt{2} x_1 x_2 \quad x_2^2]^T$$

and we conclude that f is a kernel.

Of course, if using kernels required this kind of gymnastics every time it would not be so popular.

A different way

You do not have to find feature map $\phi(\cdot)$ for your kernel in order to use it.

What you have to be able to say is that such a feature map exists.

If for any choice of $\{\mathbf{x}_n\}$ matrix $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ is symmetric and positive definite then $k(\cdot, \cdot)$ is a kernel.

Popular kernels

- polynomial $k(x, z) = (1 + x^T z)^d$
- radial basis $k(x, z) = \exp\left\{-\|x - z\|^2 / c\right\}$
- tanh $k(x, z) = \tanh(\alpha x^T z + \beta)$

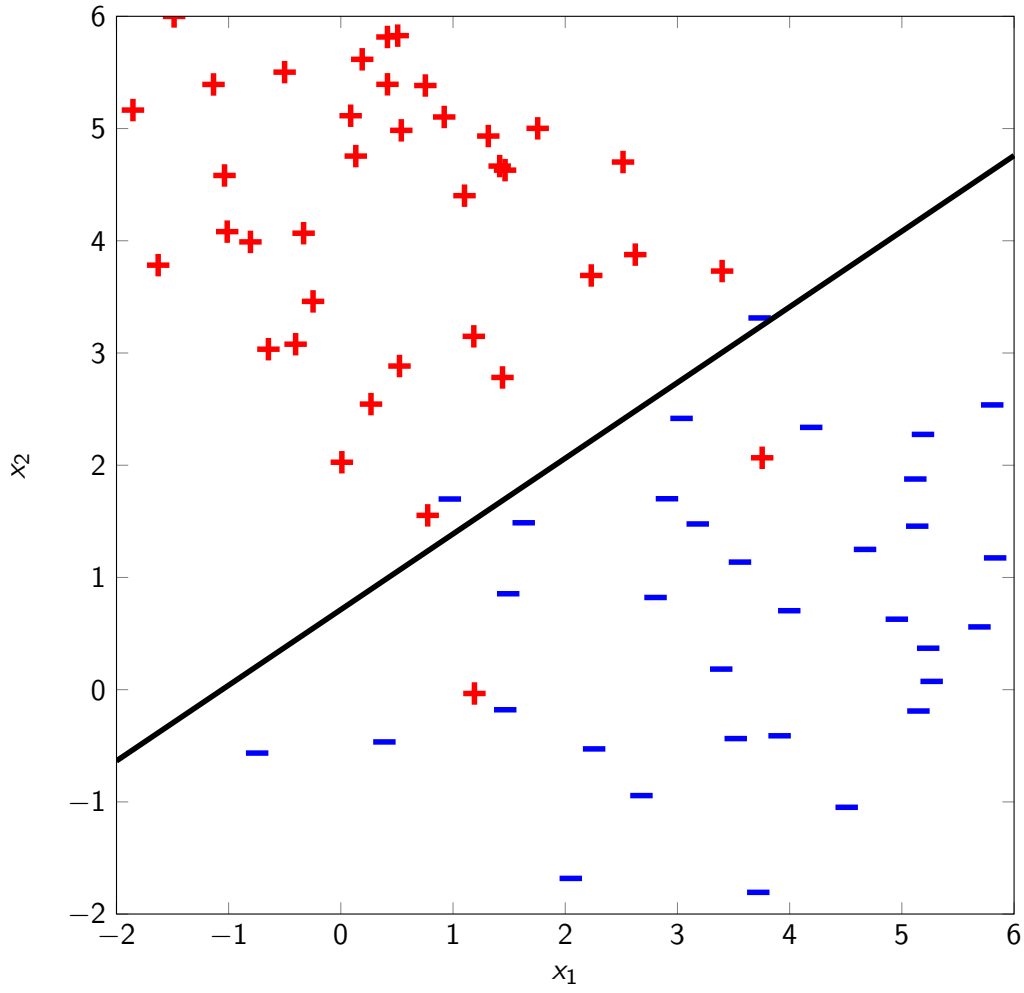
Building kernels

Assuming k_1 and k_2 are kernels, A is a symmetric psd matrix, a and b are subsets of coordinates of a feature vector, and q is a polynomial

- $k(x, z) = k_1(x, z) + k_2(x, z)$
- $k(x, z) = k_1(x, z) * k_2(x, z)$
- $k(x, z) = k_1(x_a, z_a) + k_2(x_b, z_b)$
- $k(x, z) = k_1(x_a, z_a) * k_2(x_b, z_b)$
- $k(x, z) = \exp\{k_1(x, z)\}$
- $k(x, z) = x^T A z$
- $k(x, z) = q(k_1(x, z))$

then k is a kernel as well.

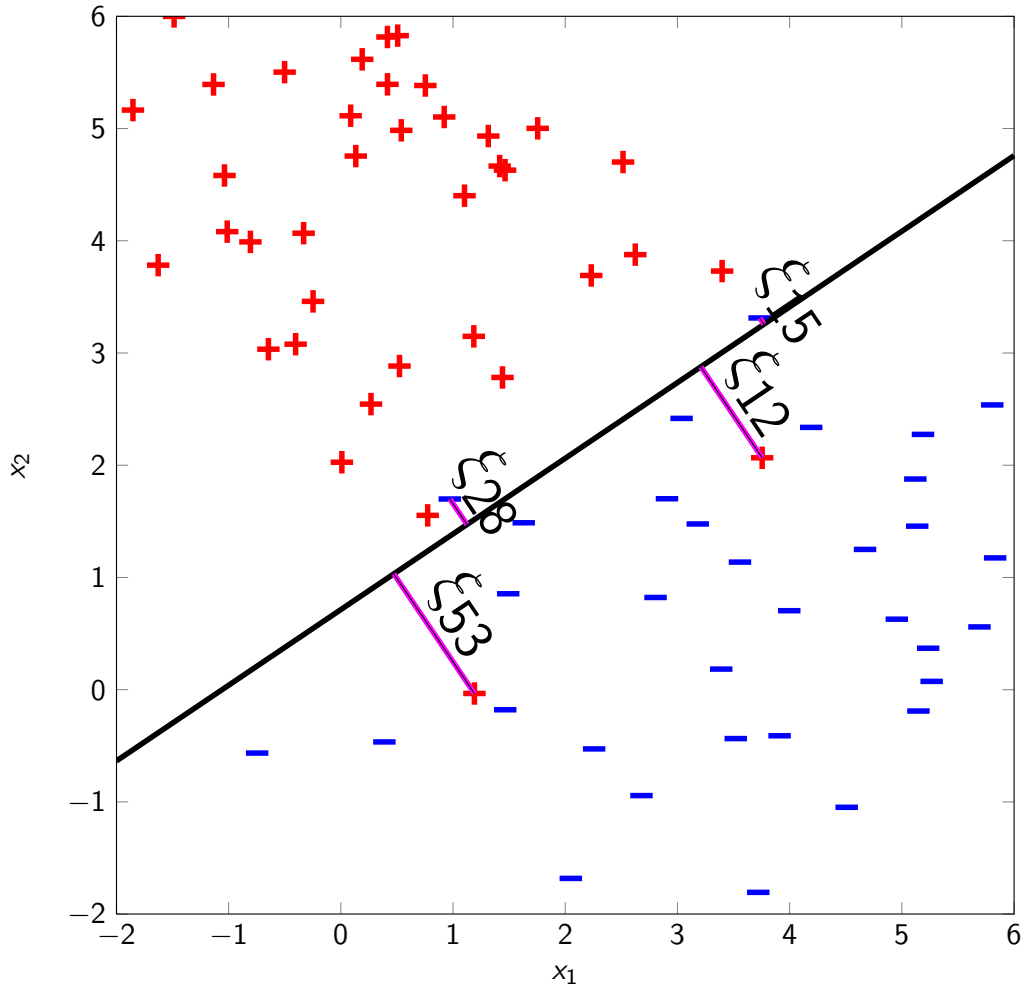
Trouble – hyperplane constraint violation



No separating hyperplane that satisfies

$$y_t(w^T \mathbf{x}_t + b) - 1 \geq 0, \forall t \in \text{Train}$$

Constraint violation



The values ξ_t tell us the amount of constraint violation

$$y_t(w^T \mathbf{x}_t + b) - 1 \geq -\xi_t, \forall t \in \text{Train}$$

Slack variables

In a nonseparable case the original optimization problem may be infeasible (set of hyperplanes that satisfy constraints is empty).

The **slack** variable ξ_t quantifies an instance specific constraint violation.

We will allow constraint violation/slack but penalize it

$$\begin{aligned} \underset{w,\xi}{\text{minimize}} \;\; & \|w\|_2^2 + C \sum_t \xi_t \\ \text{subject to } & y_t(w^T \mathbf{x}_t + b) - 1 + \xi_t \geq 0 \qquad \forall t \in \text{Train} \\ & \xi_t \geq 0 \qquad \forall t \in \text{Train} \end{aligned}$$

Note that the term $C \sum_t \xi_t = C \sum_t |\xi_t| = C \|\xi\|_1$ is an L1 norm since $\xi_t \geq 0$

Alternate form and a hinge loss

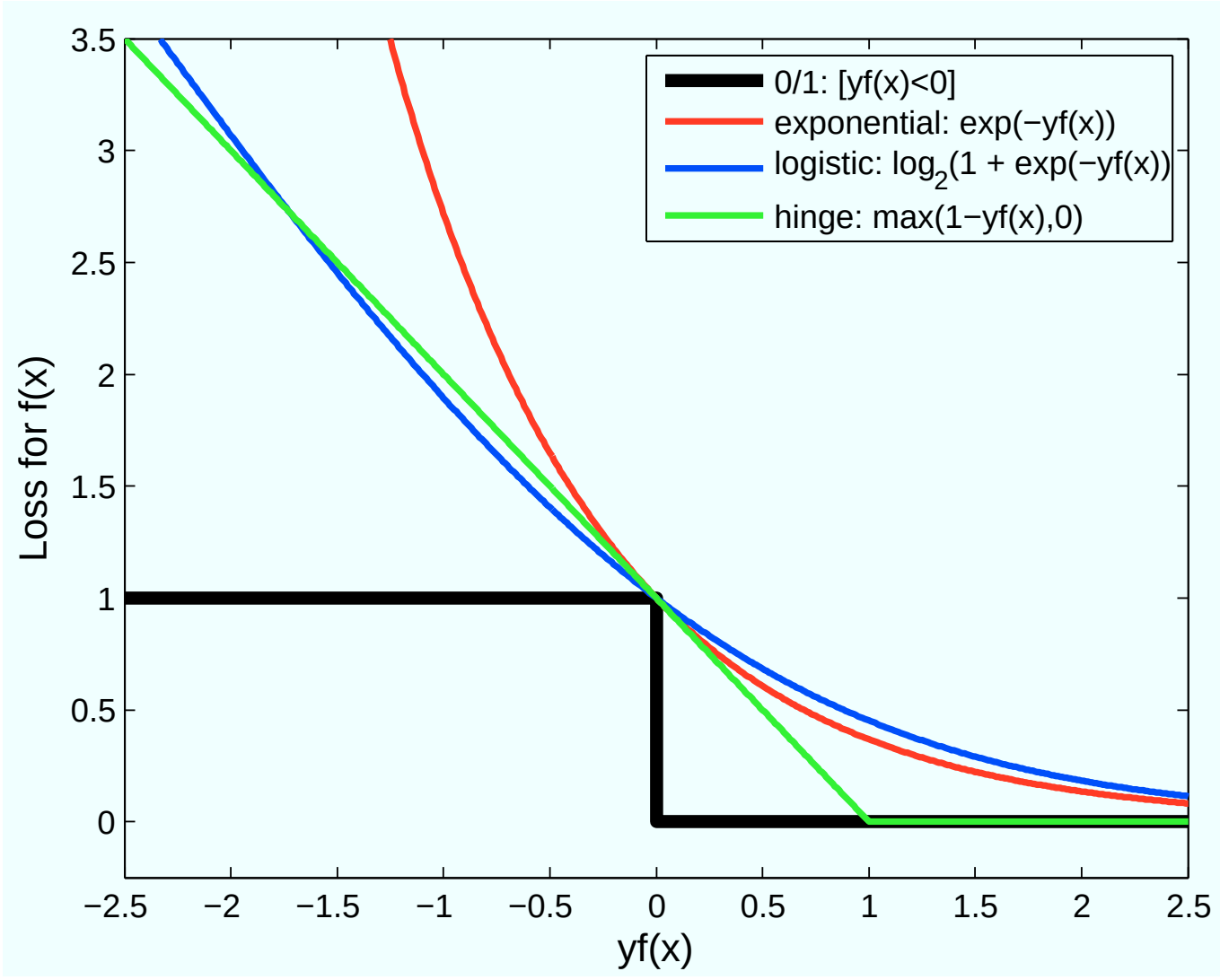
$$\begin{aligned}
&\underset{w,\xi}{\text{minimize}} \;\; \|w\|_2^2 + C\sum_t \xi_t \\
&\text{subject to } y_t(w^T\mathbf{x}_t + b) - 1 + \xi_t \geq 0 \qquad \forall t \in \text{Train} \\
&\qquad\qquad\qquad\qquad\qquad\quad \xi_t \geq 0 \qquad \forall t \in \text{Train}
\end{aligned}$$

is equivalent to

$$\underset{w}{\text{minimize}} \;\; \|w\|_2^2 + C\sum_t (1 - y_t(w^T\mathbf{x}_t + b))_+$$

where $(x)_+ = \max(0, x)$

Losses


0/1: [yf(x)<0]
exponential: exp(−yf(x))
logistic: log2(1 + exp(−yf(x))
hinge: max(1−yf(x),0)
Loss for f(x)
yf(x)
3.5
3
2.5
2
1.5
1
0.5
0
−2.5
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5

Lagrangian and dual problem

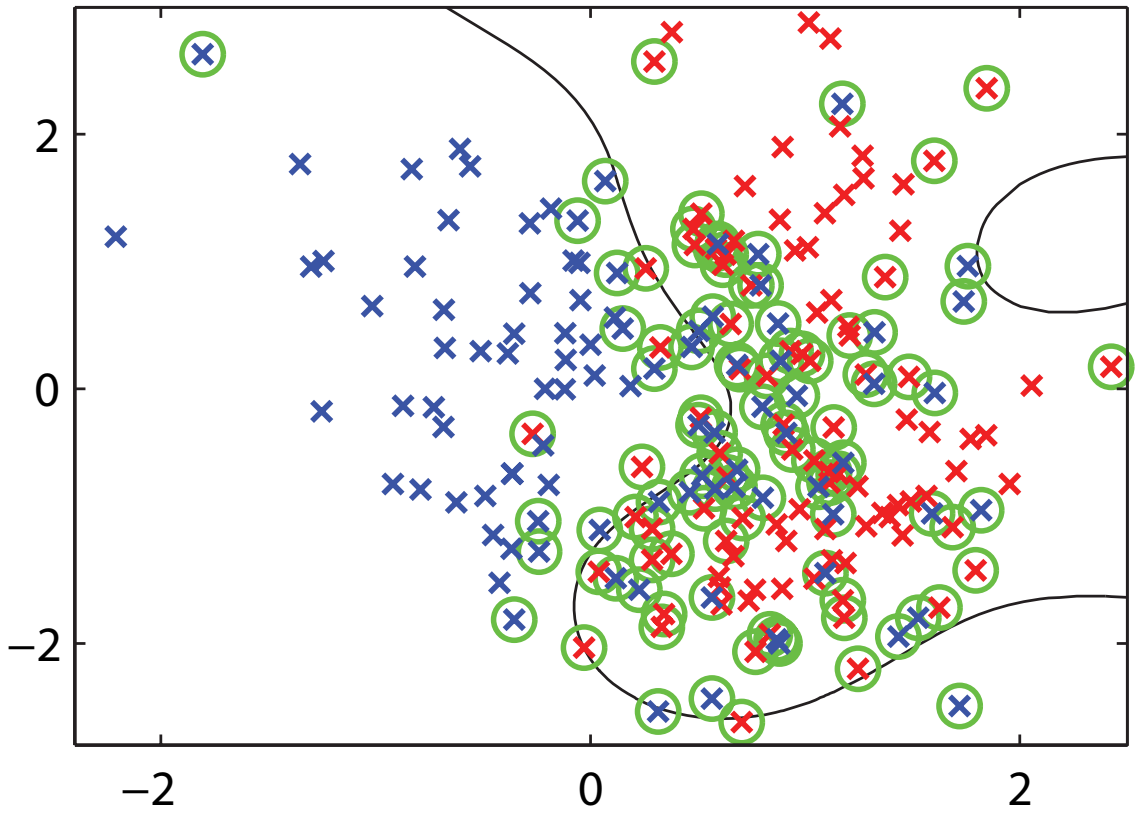
$$
\begin{aligned}
L(w, b, \xi, \alpha, \mu) &= 1/2 \, \|w\|_2^2 + C \sum_t \xi_t \\
&- \sum_{t=1} \alpha_t \left(y_t (w^T \mathbf{x}_t + b) - 1 + \xi_t \right) - \sum_{i=1} \mu_i \xi_i
\end{aligned}
$$

and the dual problem is

$$
\begin{aligned}
\text{maximize } & \textstyle\sum_{t=1} \alpha_t - 1/2 \sum_{t=1} \sum_{r=1} \alpha_t \alpha_r y_t y_r k(\mathbf{x}_t, \mathbf{x}_r) \\
\text{subject to } & \textstyle\sum_t \alpha_t y_t = 0 \qquad \forall t \in \text{Train} \\
& 0 \leq \alpha_t \leq C \qquad \forall t \in \text{Train}
\end{aligned}
$$

The support vector definition is still the same: instances for which $\alpha_t \neq 0$.

Example of decision boundary for a radial kernel



Pegasos – Primal Estimated Subgradient SOlver for SVM[2]

$$\underset{w}{\text{minimize}} \frac{\lambda}{2}\|w\|^2 + \frac{1}{T}\sum_{t=1}^{T}(1 - y_t w^T \mathbf{x}_t)_+$$

input : Train,λ,T,k

for $n = 1, 2, \ldots T$ **do**

Choose $A_n \subseteq \text{Train}$ where $\lvert A_n\rvert = k$	subset of k instances
$A_n^+ := \{t \in A_n : y_t w_n^T \mathbf{x}_t \leq 1\}$	violated/active contraints
$\nabla_n := \lambda w_n - \frac{1}{k}\sum_{t\in A_n^+} y_t \mathbf{x}_t$	(stochastic) gradient computation
$w_{n+\frac{1}{2}} = w_n - \frac{1}{\lambda n}\nabla_n$	diminishing step size
$w_{n+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\left\lVert w_{n+\frac{1}{2}}\right\rVert}\right\} w_{n+\frac{1}{2}}$	projection onto $\lVert w\rVert \leq \frac{1}{\sqrt{\lambda}}$

end

output: w_{N+1}

---

[2] ICML 2007

SVM implementations

- $\mathrm{SVM}^{\mathrm{light}}$ `http://svmlight.joachims.org/`
- LibLinear, LibSVM
  `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

Efficient learning of SVMs on large datasets an active field.

Max Margin Markov Networks

We used logistic regression to solve a single label task.

Then we saw generalizations of logistic regression to CRFs that tackle structured labels.

Similarly SVMs handle a single (binary) label task.

The generalization of SVMs to structured labels are Max Margin Markov Nets.

Again as in CRFs you run into a need to do approximate inference in order to compute marginals.

Kernels useful for Comp Bio

- String kernels
- Graph kernels
- Fisher kernels (constructed from generative models)
  - feature map $\phi(x) = \nabla_\theta \log p(x|\theta)$

We did ...

- SVMs in separable case
- Kernelization
- SVMs in non-separable cases
- Optimization – Pegasos