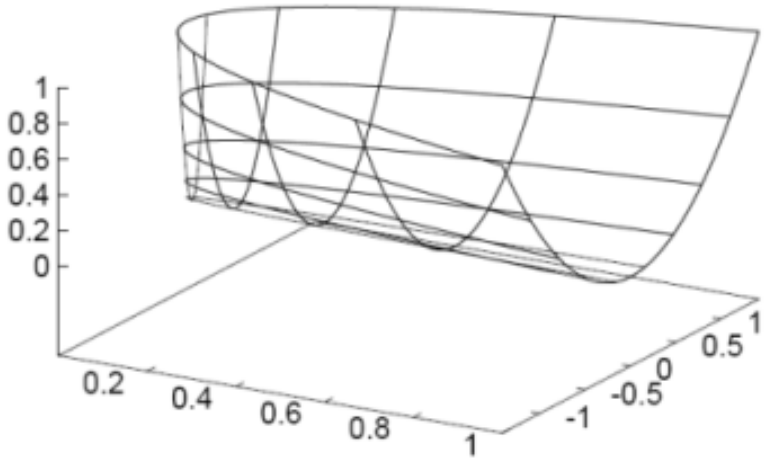
Introduction to Supervised Learning



Week 4:
Logistic Regression, continued
Support Vector Machines

Iasonas Kokkinos

i.kokkinos@cs.ucl.ac.uk

University College London

Q: background

"..my background is not appropriate this course, but I have to pass this course because it is compulsory.."

Linear algebra: http://math.mit.edu/~gs/linearalgebra/

Gilbert Strang, Introduction to Linear Algebra

+ newly-attached pdf on "resources" tab

Probability: https://moodle.ucl.ac.uk/pluginfile.php/3650305/mod_label/intro/STAT1005_CourseNotes%281516%29.pdf

Tsitsiklis and Bertsekas, Introduction to Probability

+ 1st chapter in D. Barber's book

Calculus: Tom M. Apostol, Calculus, Vol. I and II

Q: background

"..my background is not appropriate this course, but I have to pass this course because it is compulsory.."

We are working on arranging a weekly help session for the course (starting hopefully from next week)

Meeting hours (myself): Gower Street 66-72, 110B
Tue: 4-5:30 pm
Wed: 9-10:00 pm

Idea: replace computational assignments in practicals with analytical exercise sessions, Q&A.

Some clarifications

Last announcement:
“since we provide the "solution", we will be more strict on evaluating your result**.”**

Q: I do not understand why assessment will be strict suddenly

Assessment will be the same.
We will be however double-checking with your code.

Some apologies

Last announcement:
“We will run your code to make sure we can get the result you present, so DO NOT copy our "solution" figures. In addition, we would use watermarking techniques to detect whether the solution figures in your report are copied from our assignment description. This is an individual assignment, so please DO NOT give your code to any classmate.”

This relates to the few 1(?)% who would consider cheating. This was our internal discussion and was not meant to be addressing everyone.

Code of honor (first session): A copies from B - A and B get zero.

Lecture outline

Logistic Regression, continued

Introduction to Support Vector Machines

Large margins and generalization

Optimization

Kernels

Applications to vision

Loss function for logistic regression

Training: given $S = \{(\mathbf{x}^i, y^i)\}, i = 1, \ldots, N$, estimate optimal $\mathbf{w}$

Loss function: quantify appropriateness of $\mathbf{w}$

$$L(S, \mathbf{w}) = -\log P(\mathbf{y}|\mathbf{X}; \mathbf{w})$$

$$= -\sum_{i=1}^{N} \log P(Y = y^i | X = \mathbf{x}^i; \mathbf{w})$$

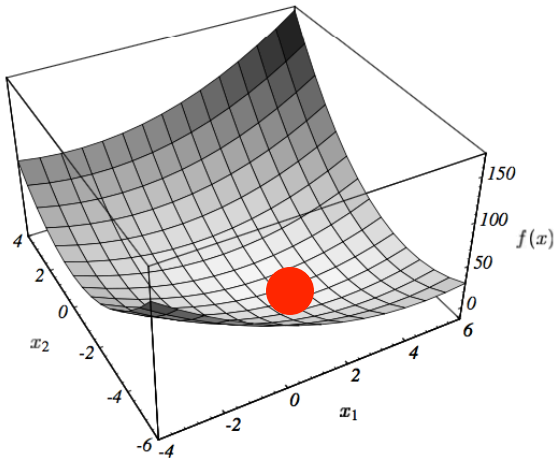
$$= -\sum_{i=1}^{N} y^i \log g(\mathbf{w}^T \mathbf{x}^i) + (1 - y^i) \log(1 - g(\mathbf{w}^T \mathbf{x}^i))$$

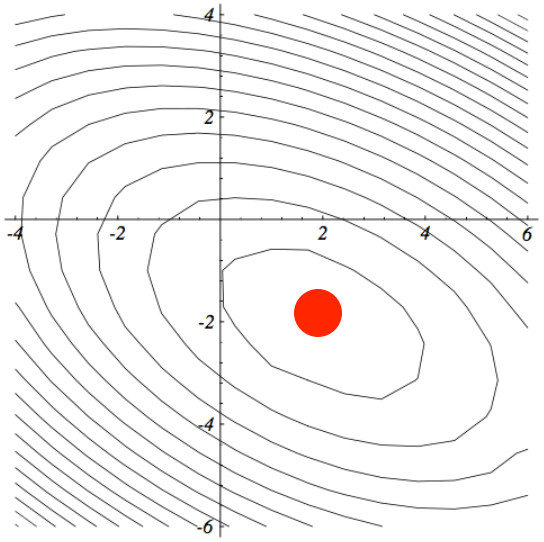
$$= \sum_{i=1}^{N} l(y^i, f_{\mathbf{w}}(\mathbf{x}^i))$$

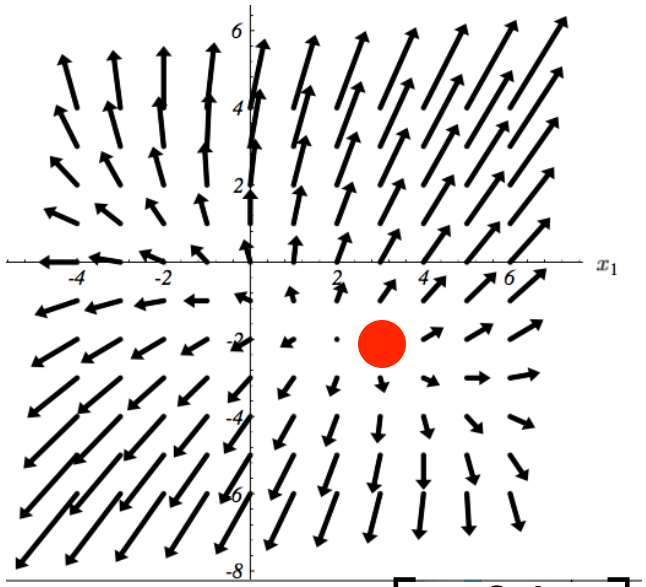
$$g(a) = \frac{1}{1 + \exp(-a)}$$

Linear discriminant: $f_{\mathbf{w}}(\mathbf{x}^i) = \mathbf{w}^T \mathbf{x}^i$

Gradient-based optimization







$f(\mathbf{x})$

$f(\mathbf{x}) = c$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

2D function graph

isocontours

gradient field

● **at minimum of function:** $\nabla f(\mathbf{x}) = \mathbf{0}$

Gradient of cross-entropy los

$$L(\mathbf{w}) = -\sum_{i=1} y^i \log g(\mathbf{w}^T\mathbf{x}^i) + (1-y^i)\log(1-g(\mathbf{w}^T\mathbf{x}^i))$$

$$\frac{\partial L(\mathbf{w})}{\partial w_k} = -\sum_{i=1}^{N}\left[y^i \frac{1}{g(\mathbf{w}^T\mathbf{x}^i)}\frac{\partial g(\mathbf{w}^T\mathbf{x}^i)}{\partial w_k} + (1-y^i)\frac{1}{1-g(\mathbf{w}^T\mathbf{x}^i)}(-\frac{\partial g(\mathbf{w}^T\mathbf{x}^i)}{\partial w_k})\right]$$

Fact: $g(x) = \frac{1}{1+\exp(-x)} \rightarrow \frac{dg}{dx} = g(x)(1-g(x))$

$$= -\sum_{i=1}^{N}\left[y^i \frac{1}{g(\mathbf{w}^T\mathbf{x}^i)} - (1-y^i)\frac{1}{1-g(\mathbf{w}^T\mathbf{x}^i)}\right] g(\mathbf{w}^T\mathbf{x}^i)(1-g(\mathbf{w}^T\mathbf{x}^i))\frac{\partial \mathbf{w}^T\mathbf{x}^i}{\partial w_k}$$

$$= -\sum_{i=1}^{N}\left[y^i(1-g(\mathbf{w}^T\mathbf{x}^i)) - (1-y^i)g(\mathbf{w}^T\mathbf{x}^i)\right] x_k^i$$

$$= -\sum_{i=1}^{N}\left[y^i - g(\mathbf{w}^T\mathbf{x}^i)\right] \mathbf{x}_k^i$$

Nonlinear system of equations!!

How can we find where this becomes zero?

Let's make it happen!

Second-order methods, multivariate case

First- order Taylor series approximation:

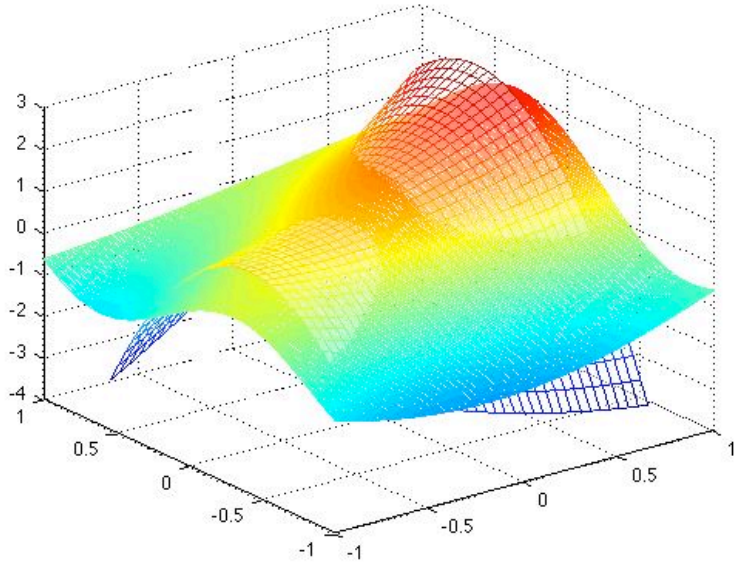
$$f(\mathbf{x}) \simeq f(\mathbf{x}_i) + (\mathbf{x} - \mathbf{x}_i)^T \nabla f(\mathbf{x}_i)$$

Second-order Taylor series approximation:

$$f(\mathbf{x}) \simeq f(\mathbf{x}_i) + (\mathbf{x} - \mathbf{x}_i)^T \nabla f(\mathbf{x}_i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T \mathbf{H} (\mathbf{x} - \mathbf{x}_i)$$

$$\doteq q(\mathbf{x})$$

$$H_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$



Second-order minimization, N-D (Newton-Raphson)

Start from some inititial position, $\mathbf{x}_0$

At any point, form quadratic approximation:

$$f(\mathbf{x}) \simeq q(\mathbf{x}) = f(\mathbf{x}_i) + (\mathbf{x} - \mathbf{x}_i)^T \nabla f(\mathbf{x}_i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)$$

Condition for minimum of quadratic approximation:

$$\nabla q(\mathbf{x}) = 0 \rightarrow \nabla f(\mathbf{x}_i) + (\mathbf{x} - \mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i) = 0$$

Set point in next iteration to be at the minimum of present approximation

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\mathbf{H}(\mathbf{x}_i))^{-1} \nabla f(\mathbf{x}_i)$$

Until update is too small

Newton-Raphson for Logistic Regression

Gradient: $$\frac{\partial L(\mathbf{w})}{\partial w_k} = -\sum_{i=1}^{N} \left[y^i - g(\mathbf{w}^T\mathbf{x}^i)\right] \mathbf{x}_k^i$$

Hessian:

$$\frac{\partial^2 L(\mathbf{w})}{\partial w_k \partial w_j} = \frac{\partial \left(-\sum_{i=1}^{N} \left[y^i - g(\mathbf{w}^T\mathbf{x}^i)\right] \mathbf{x}_k^i\right)}{\partial w_j}$$

$$= \sum_{i=1}^{N} \mathbf{x}_k^i \frac{\partial g(\mathbf{w}^T\mathbf{x}^i)}{\partial w_j} = \sum_{i=1}^{N} \mathbf{x}_k^i g(\mathbf{w}^T\mathbf{x}^i)(1 - g(\mathbf{w}^T\mathbf{x}^i))\mathbf{x}_j^i$$

Matrix version of same result:

$$H(\mathbf{w}) = \mathbf{X}^T\mathbf{R}\mathbf{X}, \quad R_{i,i} = g(\mathbf{w}^T\mathbf{x}^i)(1 - g(\mathbf{w}^T\mathbf{x}^i))$$

Summation- and matrix-based expressions

$$H_{k,j} = \frac{\partial^2 L(\mathbf{w})}{\partial w_k \partial w_j} = \sum_{i=1}^{N} \mathbf{x}_k^i g(\mathbf{w}^T \mathbf{x}^i)(1 - g(\mathbf{w}^T \mathbf{x}^i)\mathbf{x}_j^i$$

$$H(\mathbf{w}) = \mathbf{X}^T \mathbf{R} \mathbf{X}, \quad R_{i,i} = g(\mathbf{w}^T \mathbf{x}^i)(1 - g(\mathbf{w}^T \mathbf{x}^i))$$

Matrix version of same result:

From two to many

- So far: binary classification
- How about multi-class classification?

Multiple classes & linear regression

C classes: one-of-c coding (or one-hot encoding)

4 classes, i-th sample is in 3rd class: $\mathbf{y}^i = (0, 0, 1, 0)$

Matrix notation:

$$\mathbf{Y} = \left[\begin{array}{c} \mathbf{y}^1 \\ \hline \vdots \\ \hline \mathbf{y}^N \end{array}\right] = \left[\begin{array}{c|c|c} \mathbf{y}_1 & \dots & \mathbf{y}_C \end{array}\right] \qquad \text{where} \quad \mathbf{y}_c = \begin{bmatrix} y_c^1 \\ \vdots \\ y_c^N \end{bmatrix}$$

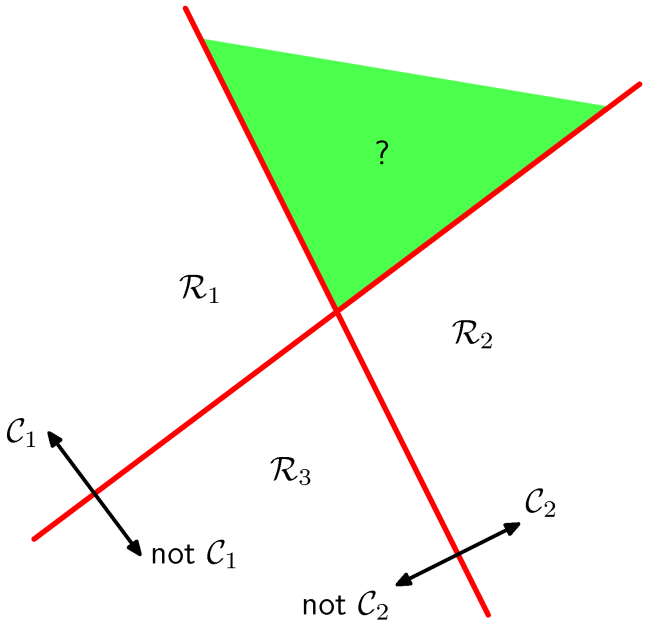
$$\mathbf{W} = \left[\begin{array}{c|c|c} \mathbf{w}_1 & \dots & \mathbf{w}_C \end{array}\right]$$

Loss function: $$L(\mathbf{W}) = \sum_{c=1}^{C} (\mathbf{y}_c - \mathbf{X}\mathbf{w}_c)^T (\mathbf{y}_c - \mathbf{X}\mathbf{w}_c)$$

Least squares fit (decouples per class):

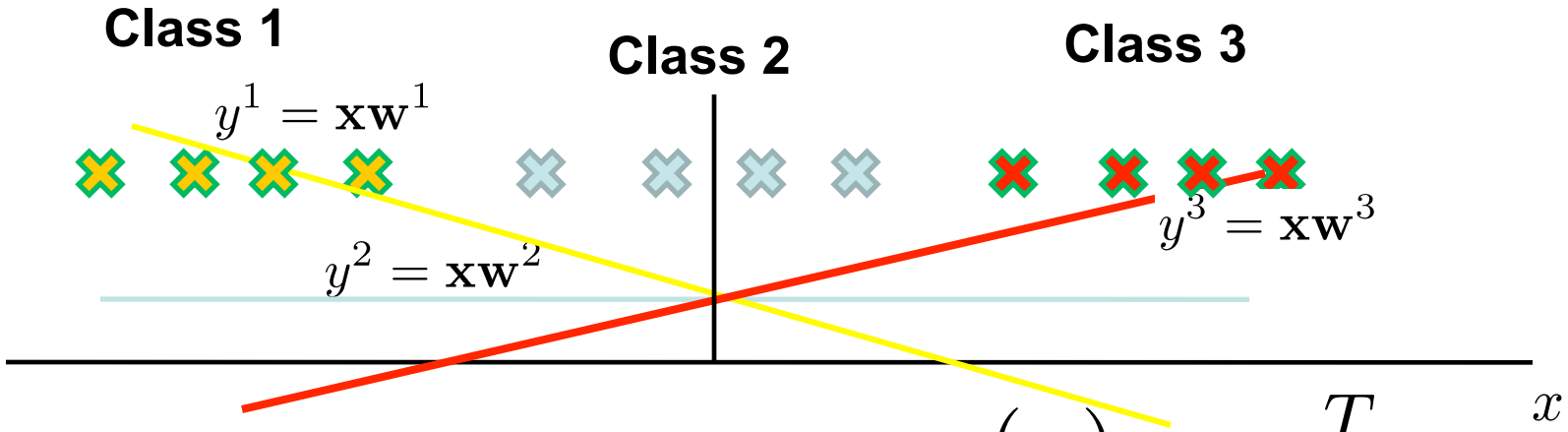
$$\mathbf{w}_c^* = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}_c$$

Multiple classes & linear regression



Solution: assign to discriminant with largest score

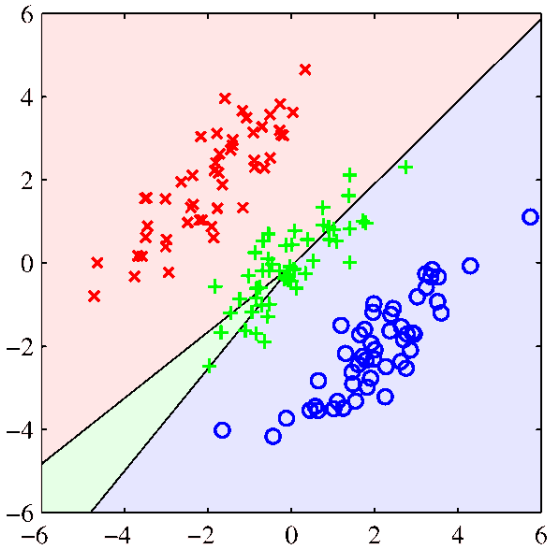
Masking Problem in linear regression



One linear discriminant per class: $s_c(\mathbf{x}) = \mathbf{w}_c^T \mathbf{x}$

Nothing ever gets assigned to class 2!

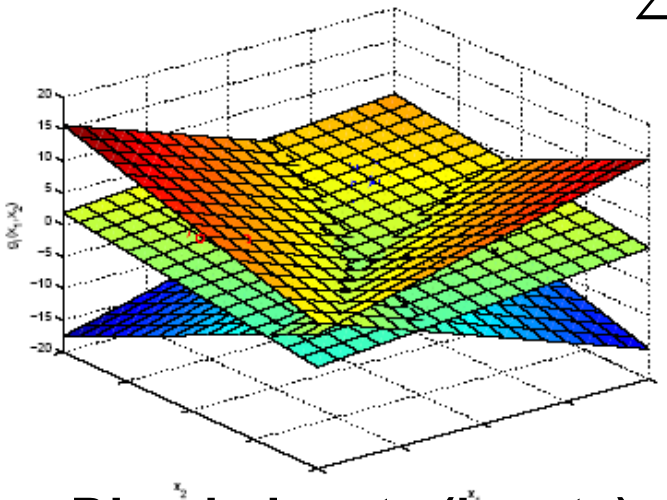
2D version:



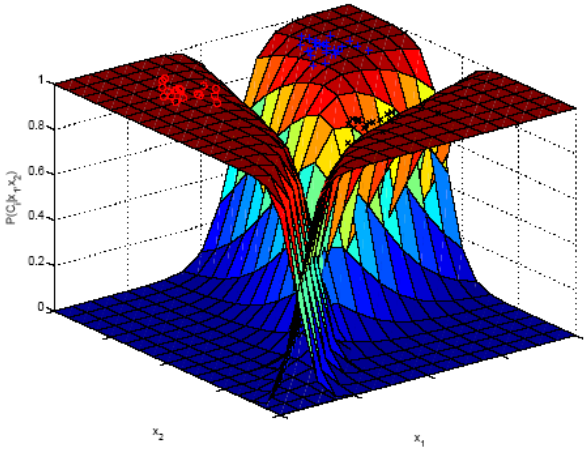
Multiple classes & logistic regression

Soft maximum (softmax) of competing classes:

$$P(y = c|\mathbf{x}; \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x})}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T \mathbf{x})} \doteq g_c(\mathbf{x}, \mathbf{W})$$



Discriminants (inputs)



Softmax (outputs)

Parameter estimation, multi-class case

One-hot label encoding: $\mathbf{y}^i = (0, 0, 1, 0)$

Likelihood of training sample: $(\mathbf{y}^i, \mathbf{x}^i)$

$$P(\mathbf{y}^i|\mathbf{x}^i; \mathbf{w}) = \prod_{i=1}^{N} \prod_{c=1}^{C} \left(g_c(\mathbf{x}, \mathbf{W})\right)^{\mathbf{y}_c^i}$$

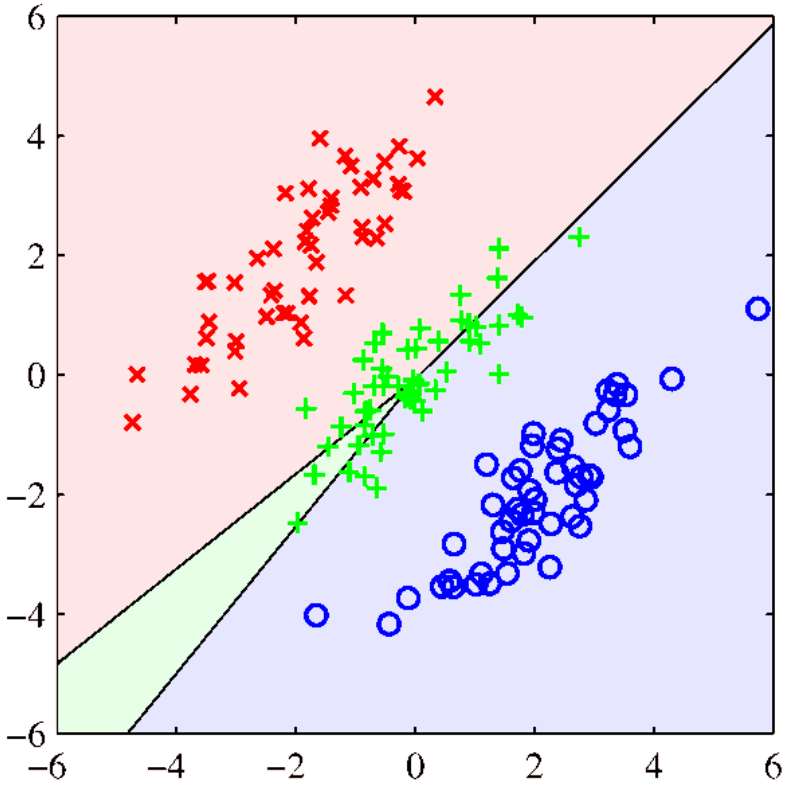
Optimization criterion:

$$L(\mathbf{W}) = -\sum_{i=1}^{N} \sum_{c=1}^{C} \mathbf{y}_c^i \log \left(g_c(\mathbf{x}, \mathbf{W})\right)$$

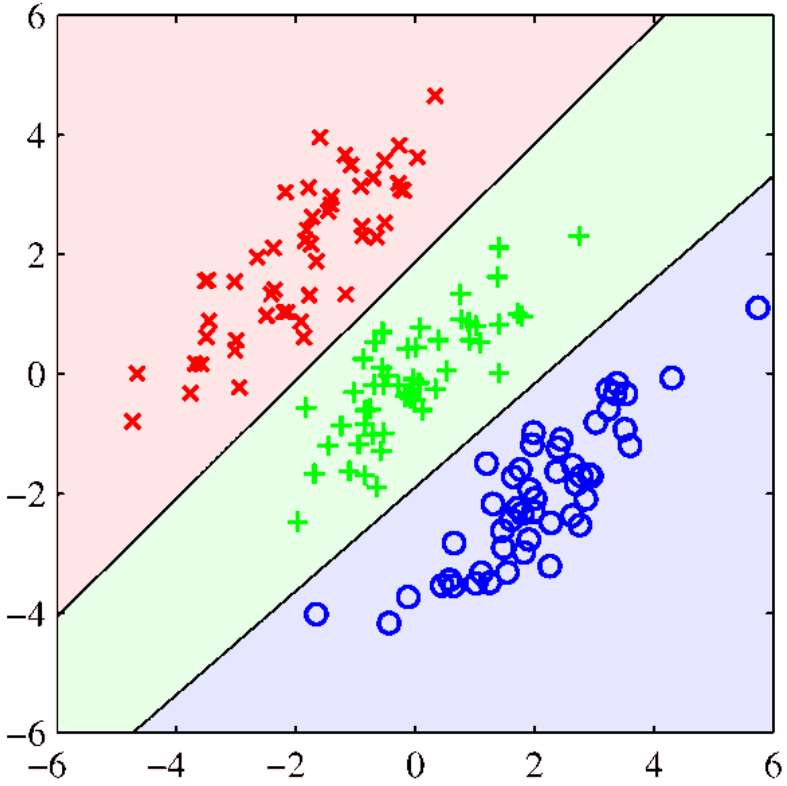
Parameter estimation: adaptation of Newton-Raphson

Logistic vs Linear Regression, n>2 classes

Linear regression



Logistic regression



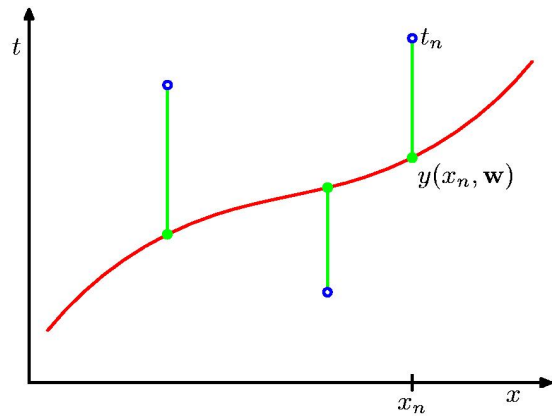
Logistic regression does not exhibit the masking problem

Lecture outline

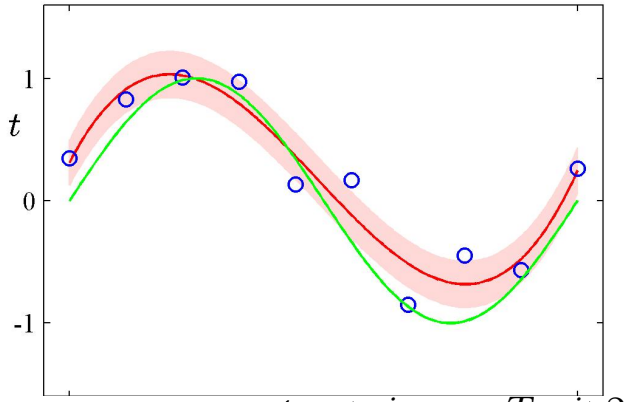
- Introduction to Support Vector Machines
  - Geometric margins
  - Training criterion & hinge loss
- Large margins and generalization
- Optimization
- Kernels
- Applications to vision

Our path so far (week 2-3)

Week 2 - regression: geometric

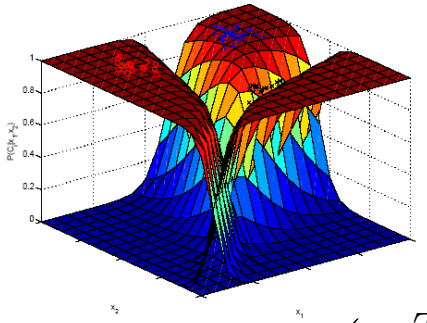


Week 3: probabilistic interpretation



$$P(y^i|\mathbf{x}^i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^i - \mathbf{w}^T\mathbf{x}^i)^2}{2\sigma^2}\right)$$

Week 3: switch to classification

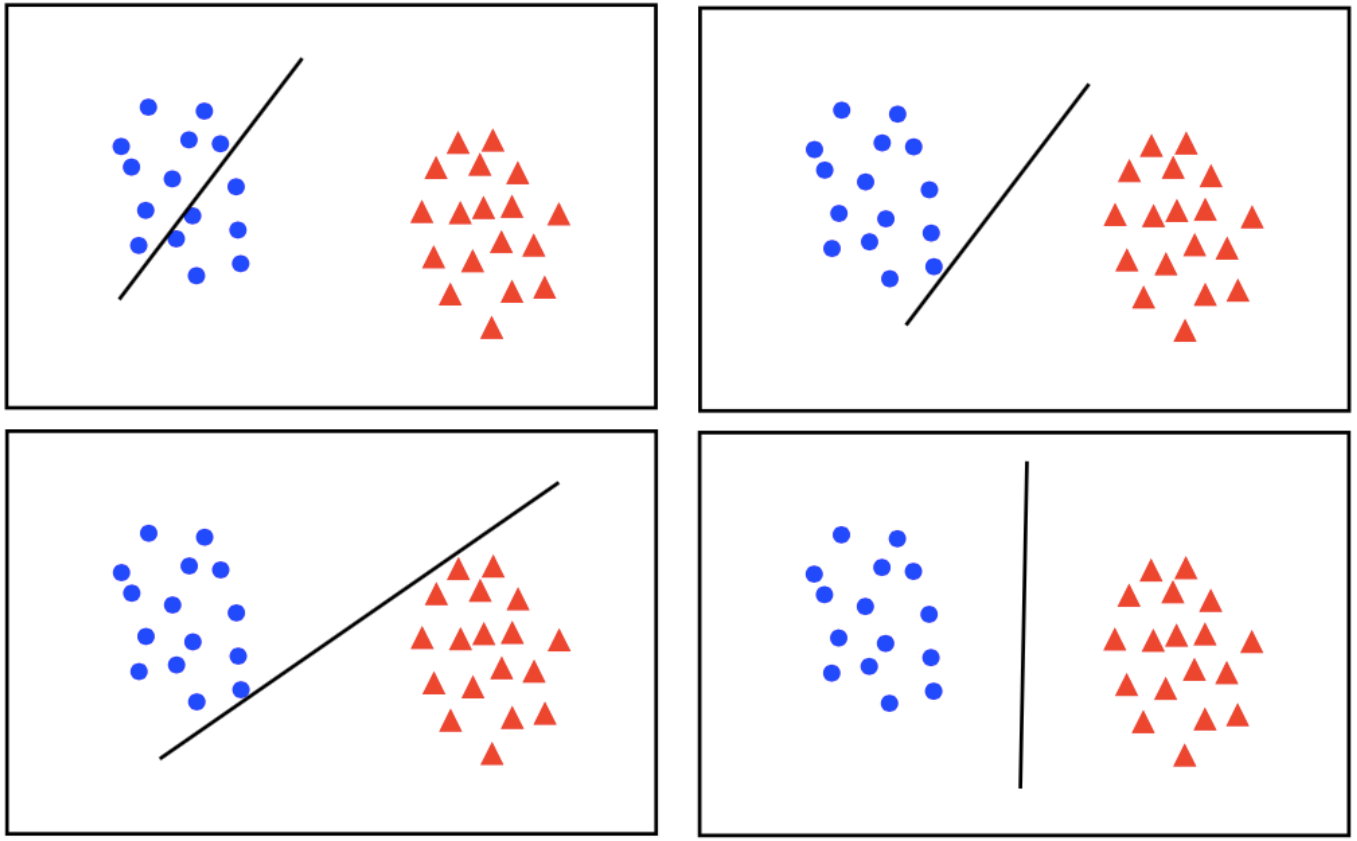


$$P(y^i|\mathbf{x}^i) = \frac{\exp(\mathbf{w}_c^T\mathbf{x})}{\sum_{c'=1}^{C}\exp(\mathbf{w}_{c'}^T\mathbf{x})}$$

?

geometry + classification?

Which classifier is best?



All points should lie **clearly** on the correct side of the boundary

How can we quantify this?

How can we enforce this?

Functional Margins

Consider Logistic Regression:

$$P(y = 1|\mathbf{x}; \mathbf{w}) = g(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

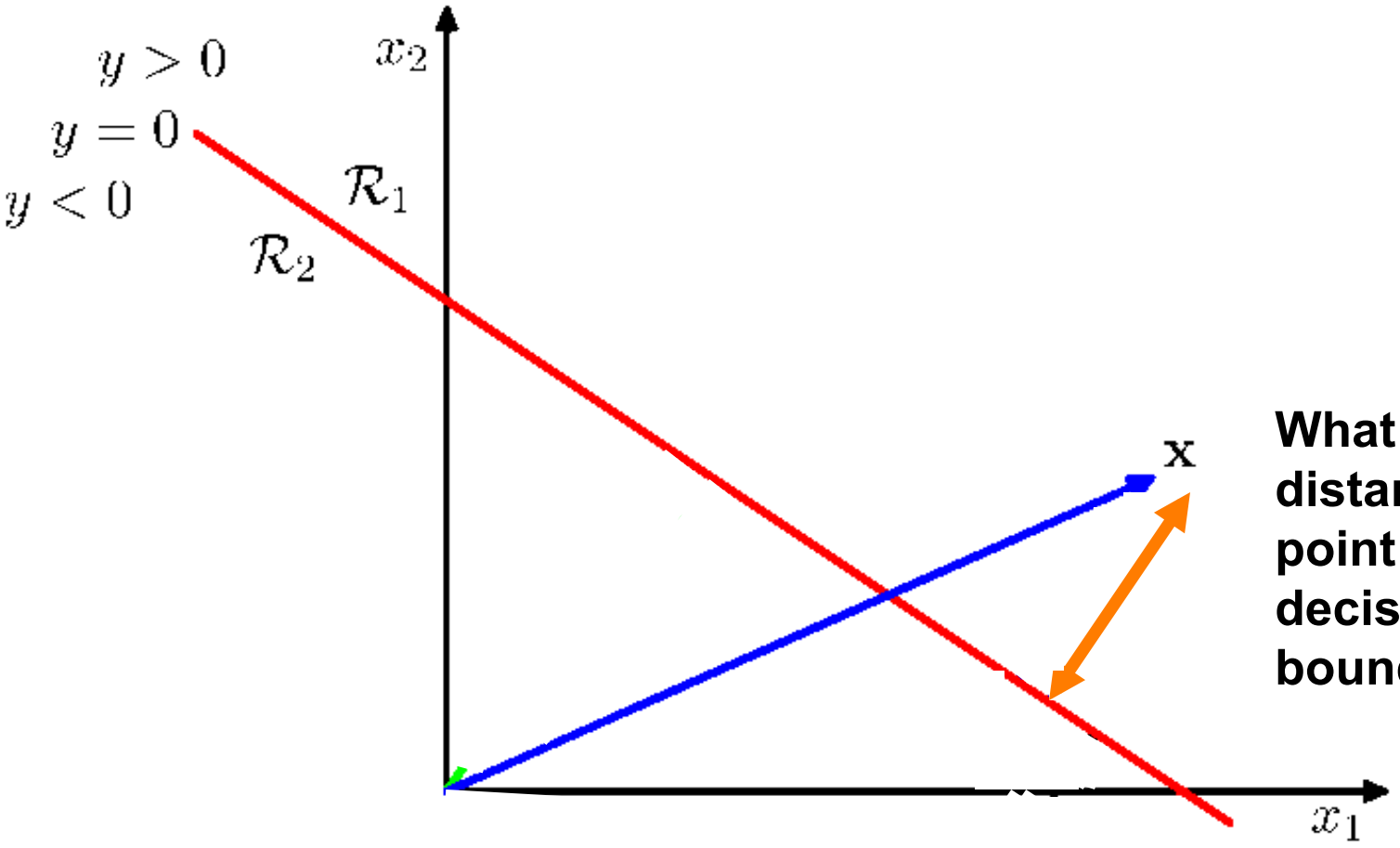
Ideally:
$$\mathbf{w}^T\mathbf{x}^i \gg 0, \quad \text{if} \quad y^i = 1$$
$$\mathbf{w}^T\mathbf{x}^i \ll 0, \quad \text{if} \quad y^i = -1$$

Put together: $y^i(\mathbf{w}^T\mathbf{x}^i) \gg 0$

`functional margin'

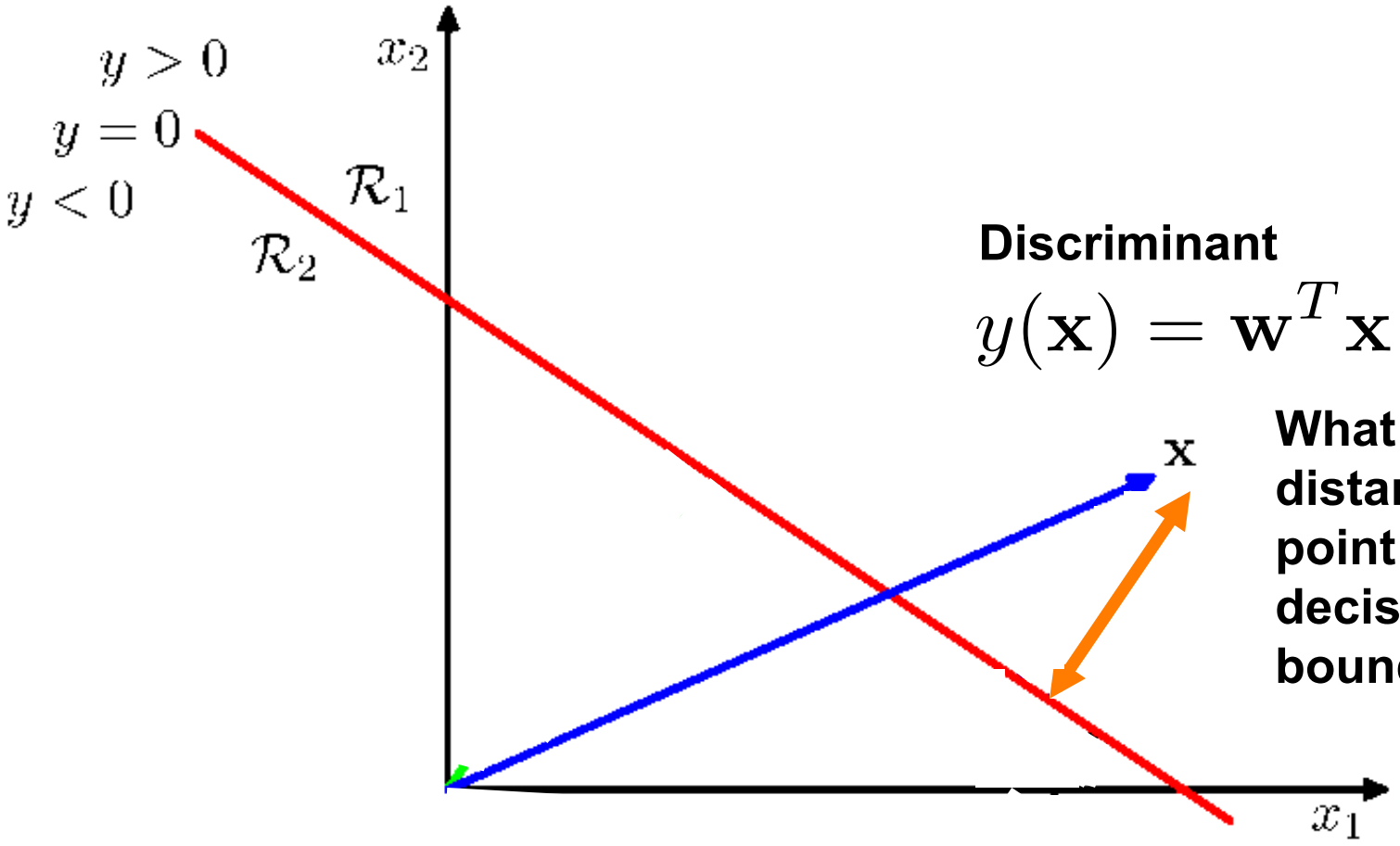
Problem: scaling **w** changes functional margin, but not decision boundary

Geometric Margins



What is the distance of this point from the decision boundary?

Geometric Margins

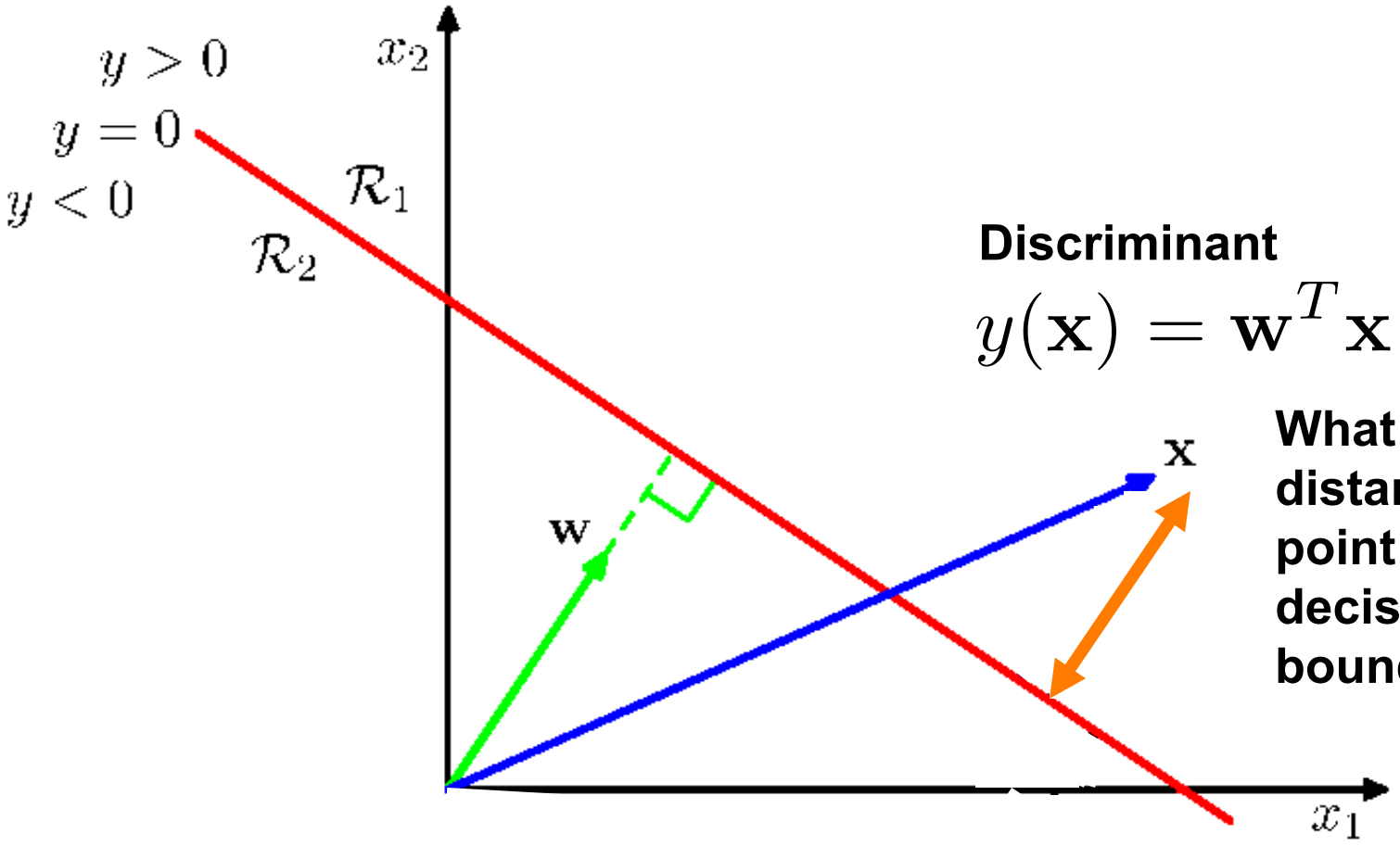


Discriminant

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

What is the distance of this point from the decision boundary?

Geometric Margins



Discriminant

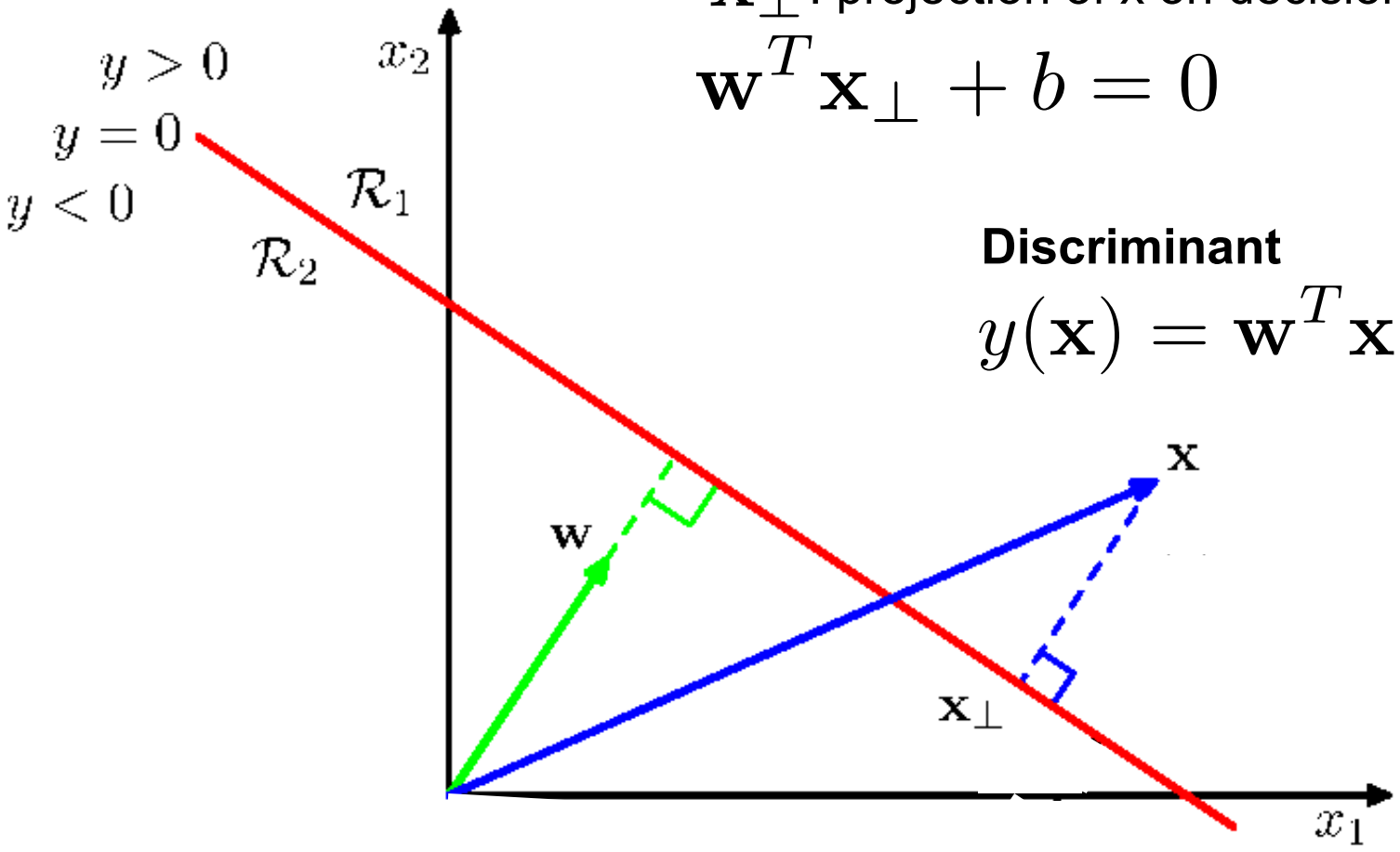
$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

What is the distance of this point from the decision boundary?

Geometric Margins

$\mathbf{x}_\perp$: projection of x on decision boundary

$$\mathbf{w}^T\mathbf{x}_\perp + b = 0$$

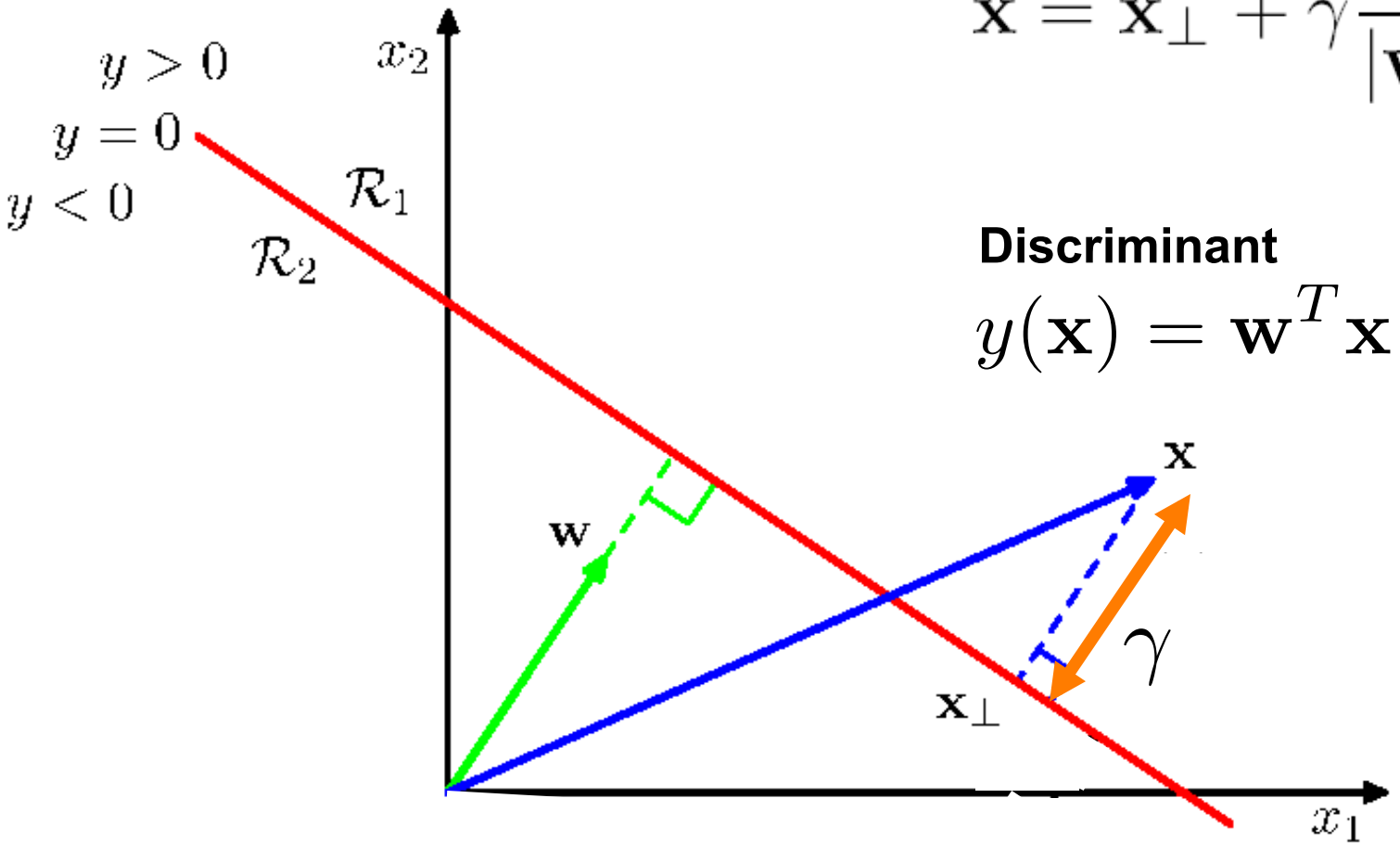


Discriminant

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

Geometric Margins

$$\mathbf{x} = \mathbf{x}_\perp + \gamma \frac{\mathbf{w}}{|\mathbf{w}|}$$



Discriminant

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

Geometric Margins

Point = projection + distance* direction

$$\mathbf{x} = \mathbf{x}_\perp + \gamma \frac{\mathbf{w}}{|\mathbf{w}|}$$

Note: γ is independent of |w|

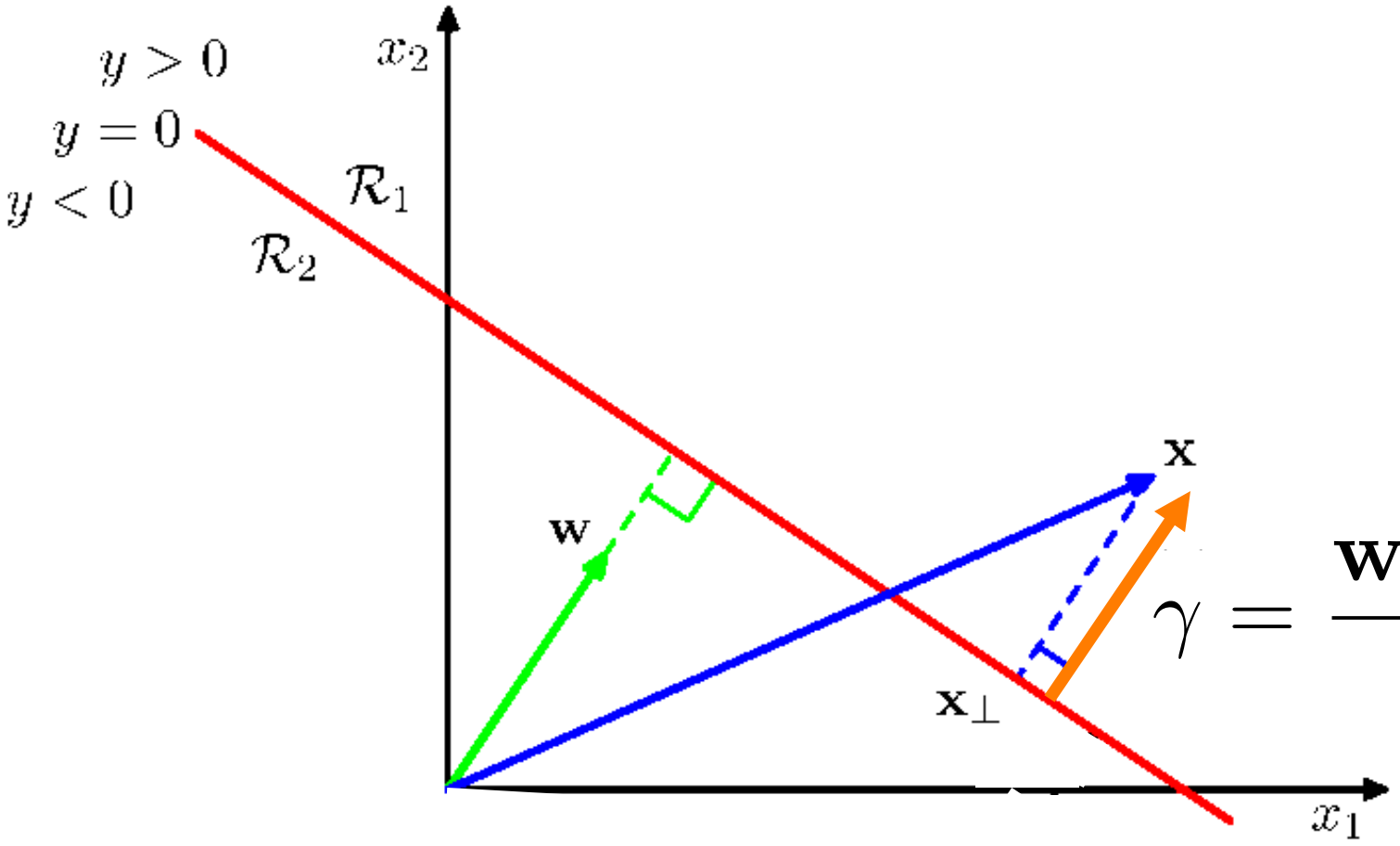
Multiply: $$\mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{x}_\perp + \mathbf{w}^T\gamma \frac{\mathbf{w}}{|\mathbf{w}|}$$

Rewrite ($\mathbf{w}^T\mathbf{x}_\perp + b = 0$) :

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

Solve for γ: $$\gamma = \frac{\mathbf{w}^T\mathbf{x} + b}{|\mathbf{w}|} = \frac{\mathbf{w}^T}{|\mathbf{w}|}\mathbf{x} + \frac{b}{|\mathbf{w}|}$$

Geometric Margins

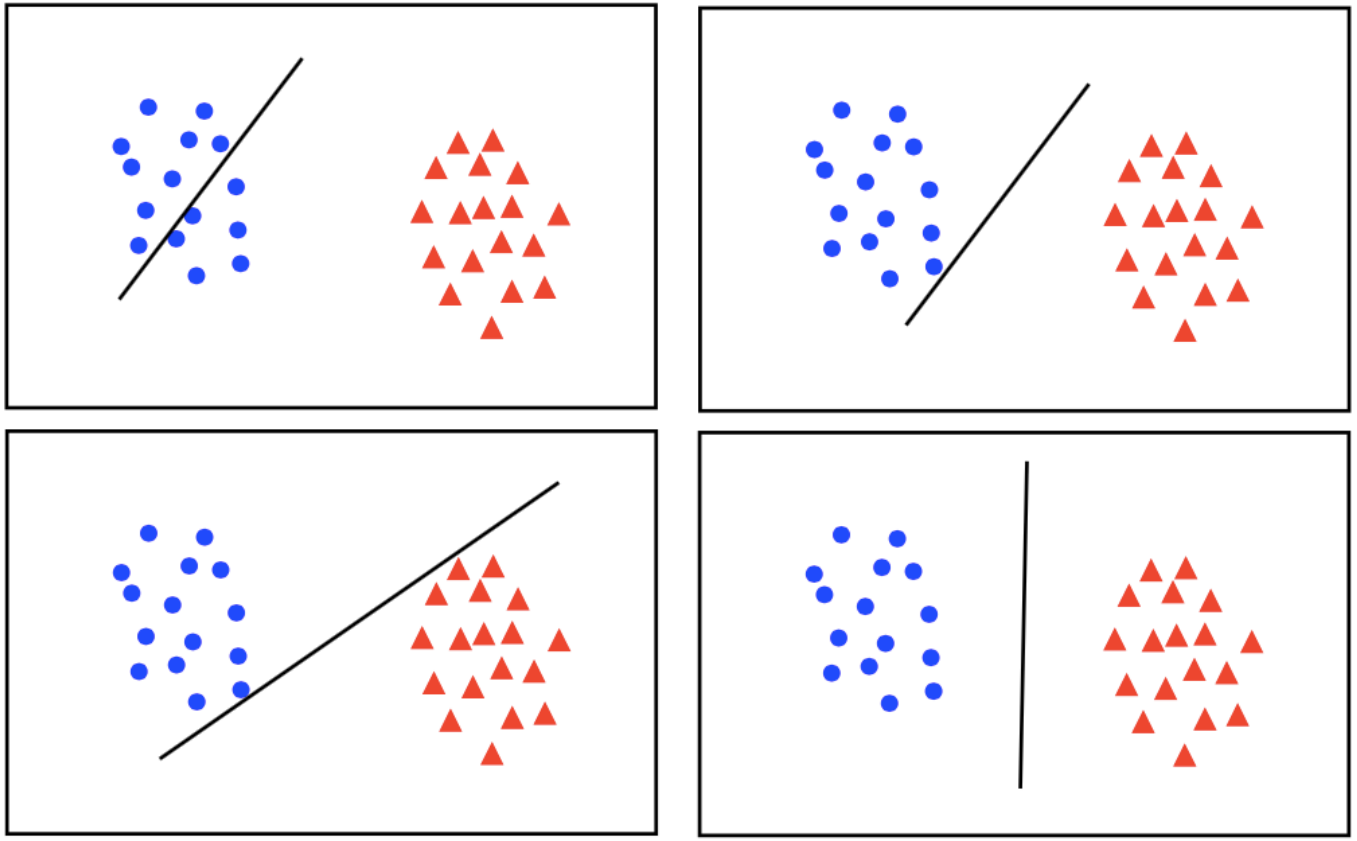


$$\gamma = \frac{\mathbf{w}^T\mathbf{x} + b}{|\mathbf{w}|}$$

Geometric Margin: $\gamma^i = y^i \left(\frac{\mathbf{w}^T}{|\mathbf{w}|}\mathbf{x}^i + \frac{b}{|\mathbf{w}|} \right)$

(positive if x is on the correct size of the decision boundary)

Which classifier is best?



All points should lie **clearly** on the correct side of the boundary

How can we quantify this? (large margins!)

How can we enforce this?

Lecture outline

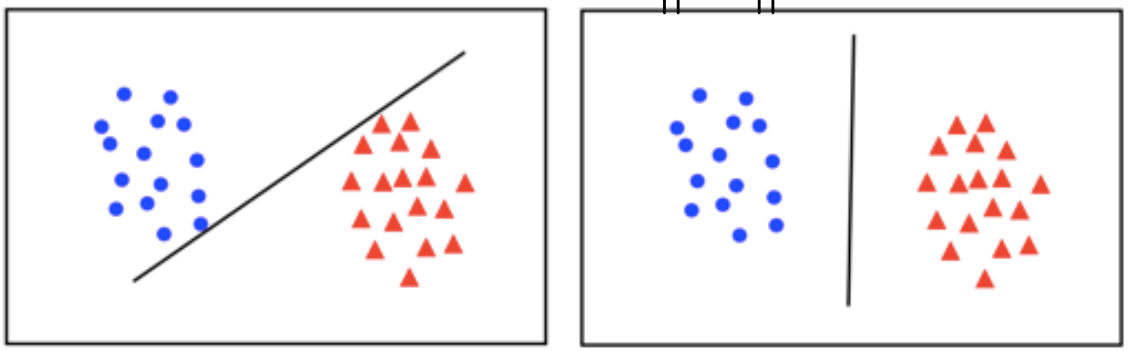
- Introduction to Support Vector Machines
  - Geometric margins
  - Training criterion & hinge loss
- Large margins and generalization
- Optimization
- Kernels
- Applications to vision

What should we be optimizing?

Training set: $\{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)\}$

Candidate parameter vector: $(\mathbf{w}, b)$

Related margins: $\gamma^i = y^i \dfrac{\mathbf{w}^T \mathbf{x}^i + b}{\|\mathbf{w}\|}$



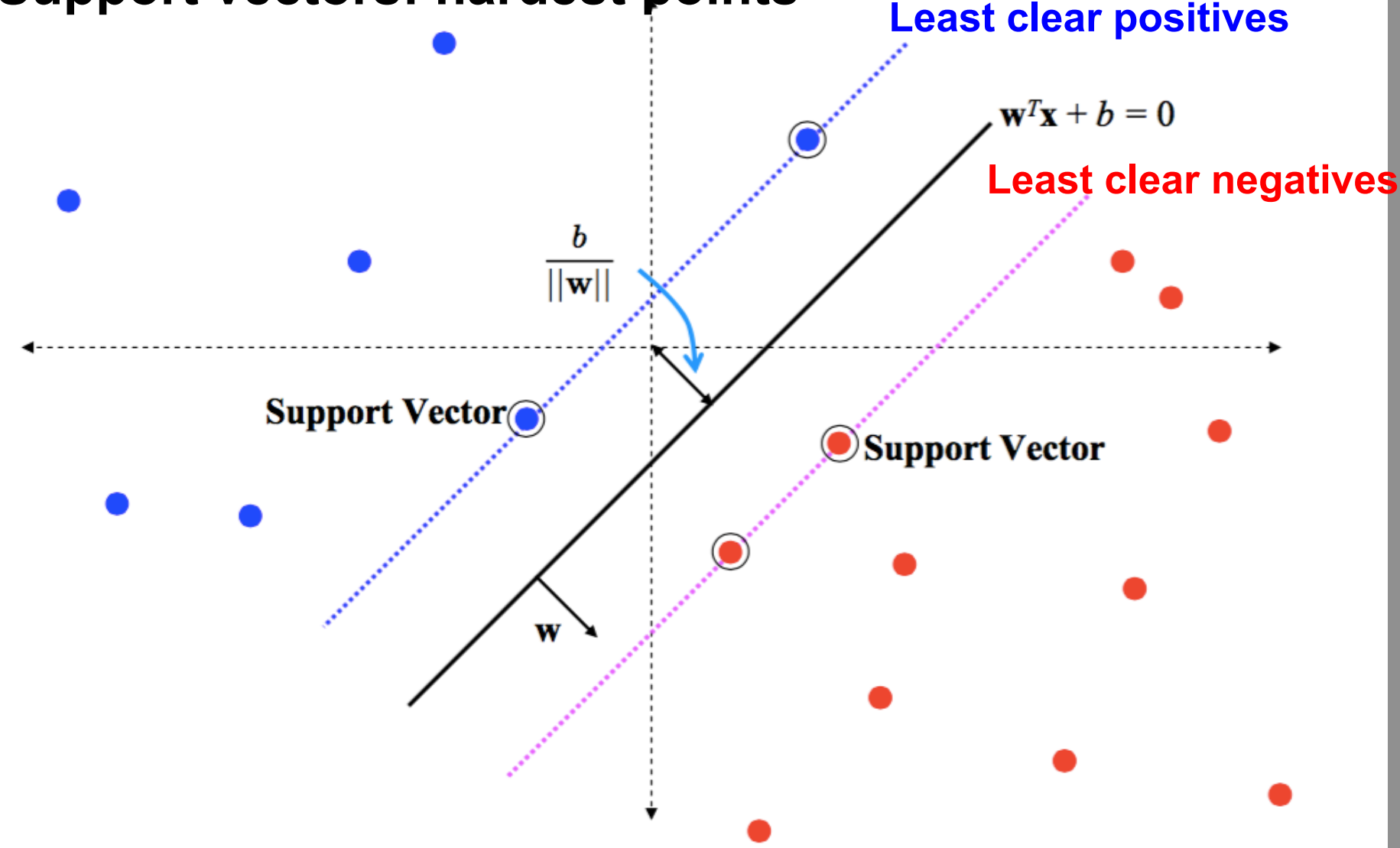
Should we be optimizing the mean, max, min margin?

All points should lie **clearly** on the correct side of the boundary

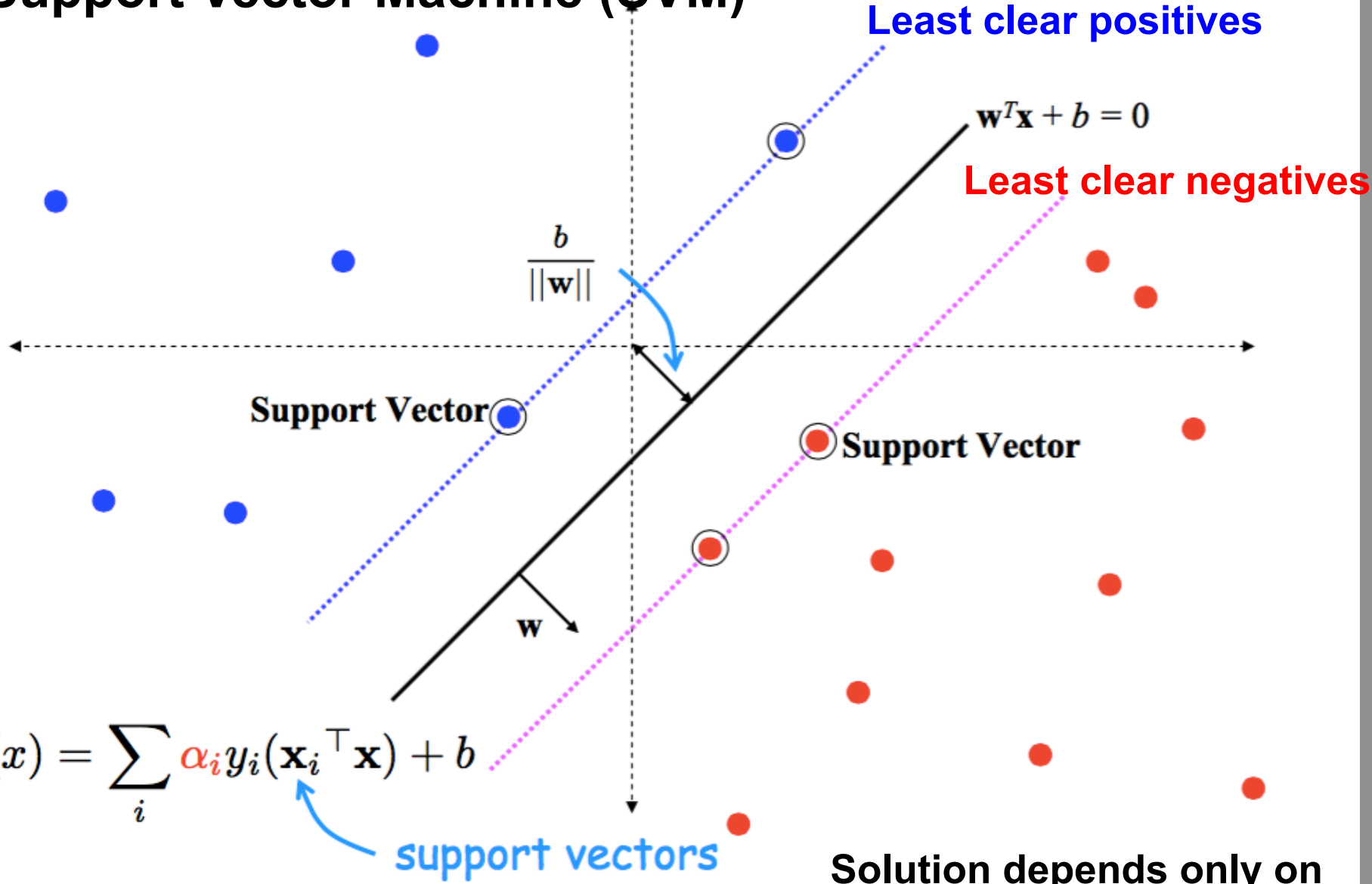
1) Take points that do not lie clearly on the correct side

2) Make sure they do

Support vectors: hardest points



Support Vector Machine (SVM)



$$f(x) = \sum_i \alpha_i y_i (\mathbf{x}_i{}^\top \mathbf{x}) + b$$

support vectors

Solution depends only on hardest points

SVM, sketch of derivation

- Since $\mathbf{w}^\top \mathbf{x} + b = 0$ and $c(\mathbf{w}^\top \mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization

SVM, sketch of derivation

- Since $\mathbf{w}^\top \mathbf{x} + b = 0$ and $c(\mathbf{w}^\top \mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization

- Choose normalization such that $\mathbf{w}^\top \mathbf{x}_+ + b = +1$ and $\mathbf{w}^\top \mathbf{x}_- + b = -1$ for the positive and negative support vectors respectively

SVM, sketch of derivation

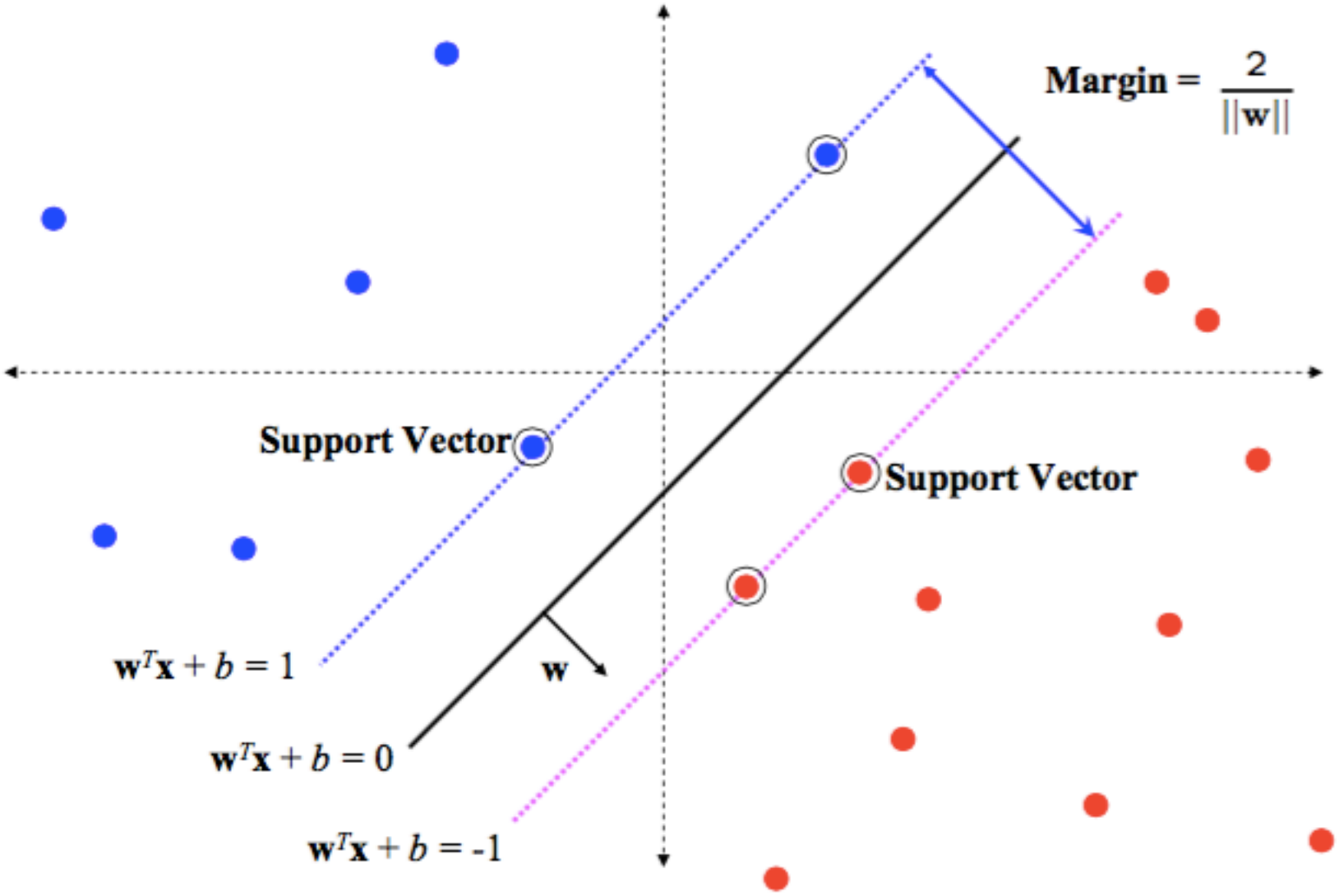
- Since $\mathbf{w}^\top \mathbf{x} + b = 0$ and $c(\mathbf{w}^\top \mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization

- Choose normalization such that $\mathbf{w}^\top \mathbf{x}_+ + b = +1$ and $\mathbf{w}^\top \mathbf{x}_- + b = -1$ for the positive and negative support vectors respectively

- Then the margin is given by

$$\frac{\mathbf{w}^\top \left(\mathbf{x}_+ - \mathbf{x}_-\right)}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$$

Support Vector Machine (SVM)



Representer theorem

Objective: find **w** that maximizes the margin subject to margin constraints

$$\min_{\mathbf{w}} \frac{2}{\|\mathbf{w}\|}$$

$$\text{s.t.} \quad y^i \left(\mathbf{w}^T \mathbf{x}^i + b\right) \geq 1 \quad \forall i$$

Equivalently:

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2$$

$$\text{s.t.} \quad y^i \left(\mathbf{w}^T \mathbf{x}^i + b\right) \geq 1 \quad \forall i$$

Representer Theorem: $\mathbf{w}^* = \sum_{i=1}^{N} \alpha^i \left(y^i \mathbf{x}^i\right)$

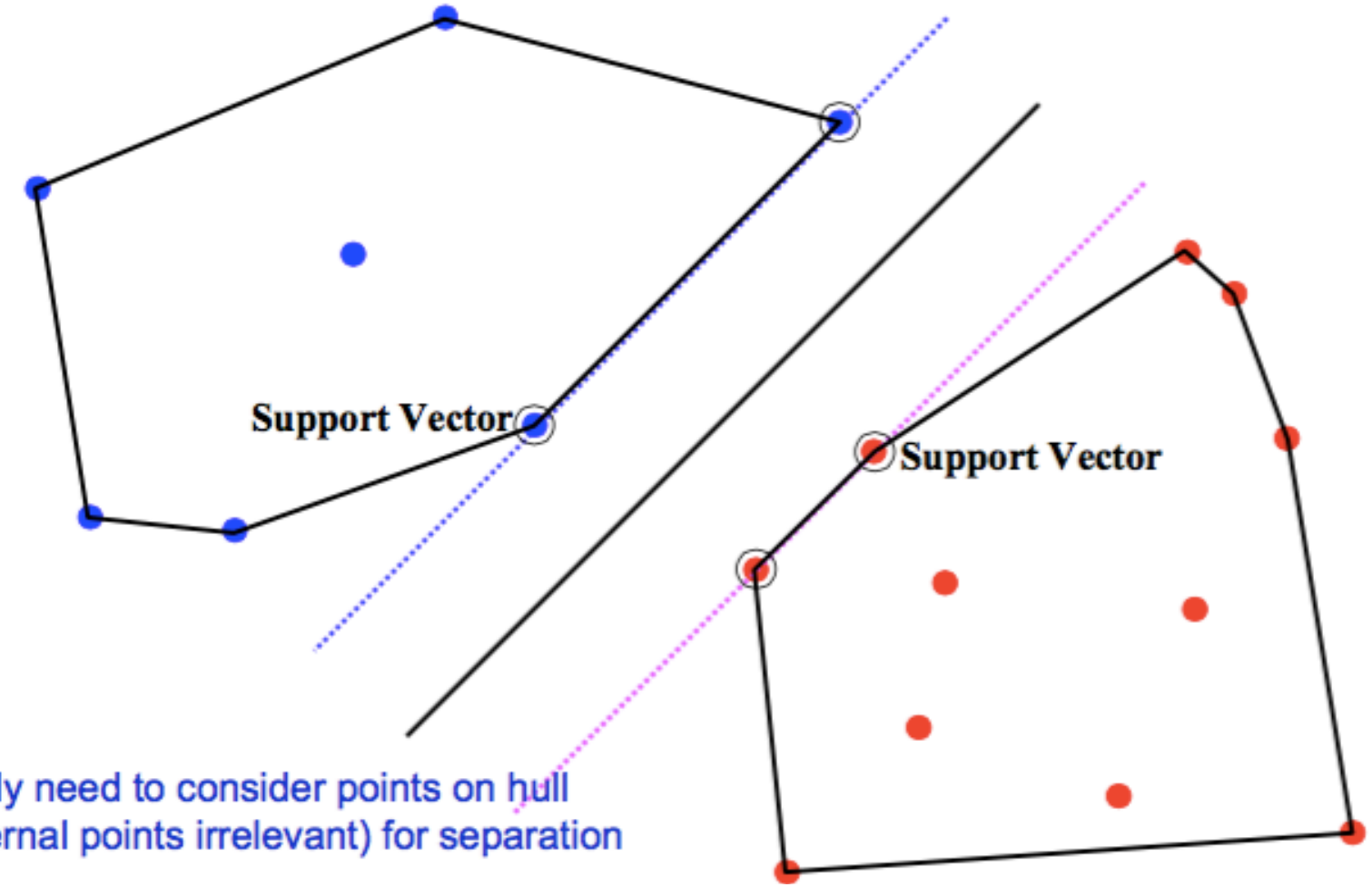
Proof idea: consider solution with component **v** in null-space of **X**

$$\mathbf{w}' = \sum_{i=1}^{N} b^i (y^i \mathbf{x}^i) + \mathbf{v}, \quad \mathbf{v}^T \mathbf{x}^i = 0, \forall i$$

We cannot influence constraints through **v -> b=a -> |w'|>|w|**

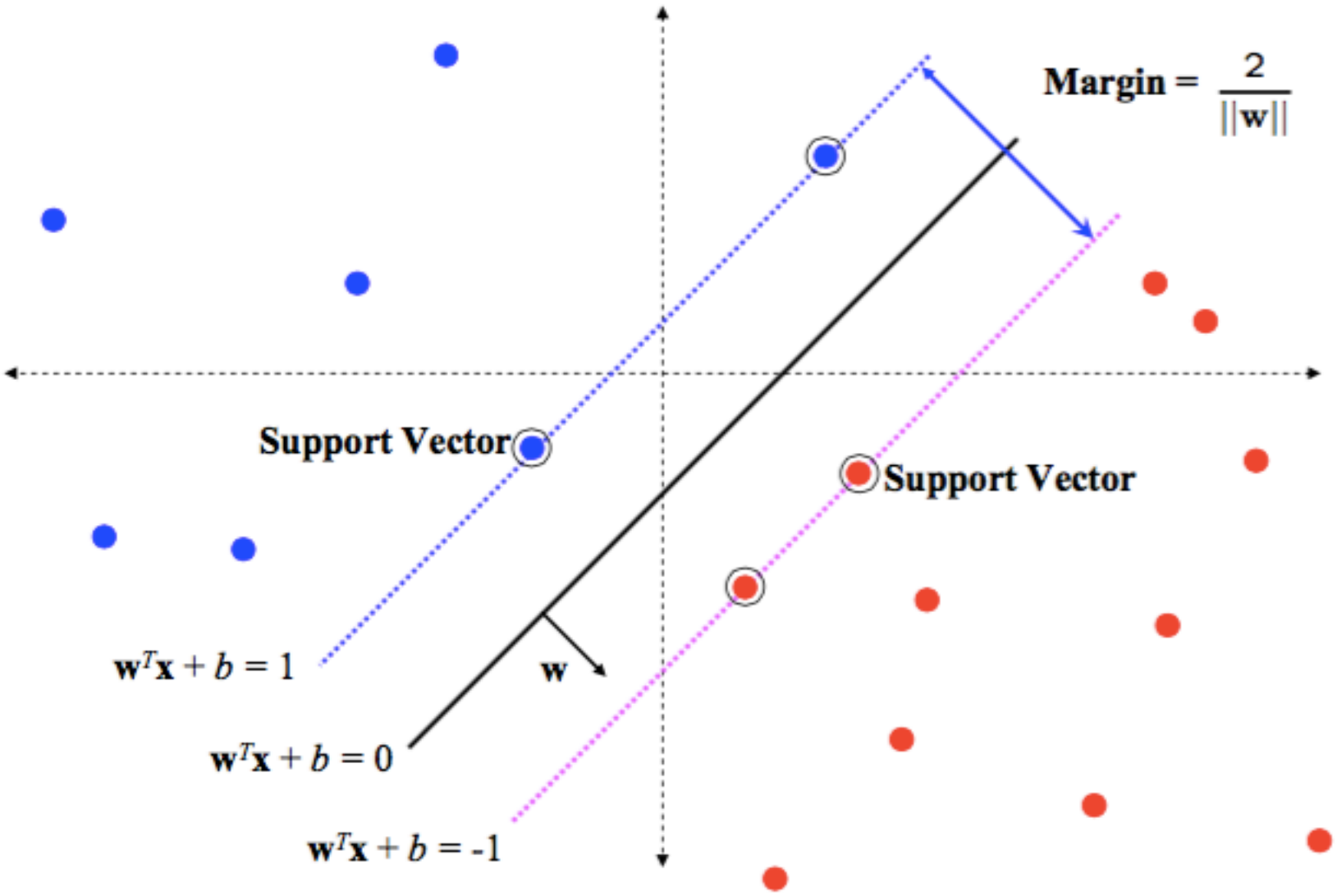
https://en.wikipedia.org/wiki/Representer_theorem + Appendix to slides

Intuitive justification of theorem



• only need to consider points on hull
(internal points irrelevant) for separation

Support Vector Machine (SVM)



Primal and dual problems

Primal, in terms of **w:** $\min_{\mathbf{w}} \|\mathbf{w}\|^2$

$$\text{s.t.}: \quad y^i(\mathbf{w}^T\mathbf{x}^i + b) \geq 1, \quad \forall i$$

But: $\|\mathbf{w}^*\|^2 = \langle \mathbf{w}^*, \mathbf{w}^* \rangle$

$$\mathbf{w}^* = \sum_{i=1}^{N} \alpha^i \left(y^i \mathbf{x}^i\right)$$

$$= \left\langle \sum_{i=1}^{N} \alpha^i y^i \mathbf{x}^i, \sum_{j=1}^{N} \alpha^j y^j \mathbf{x}^j \right\rangle = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha^i \alpha^j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle$$

Dual, in terms of $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$: $\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha^i \alpha^j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle$

$$\text{s.t.}: \quad y^i \left(\sum_{j=1}^{N} \alpha^j y^j \langle \mathbf{x}^j, \mathbf{x}^i \rangle + b \right) \geq 1, \quad i = 1, \ldots, N$$

Primal vs dual

Primal:
$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 \qquad \mathbf{w} \in \mathbb{R}^D \rightarrow O(D^3)$$
$$\text{s.t. :} \quad y^i(\mathbf{w}^T\mathbf{x}^i + b) \geq 1, \quad \forall i$$

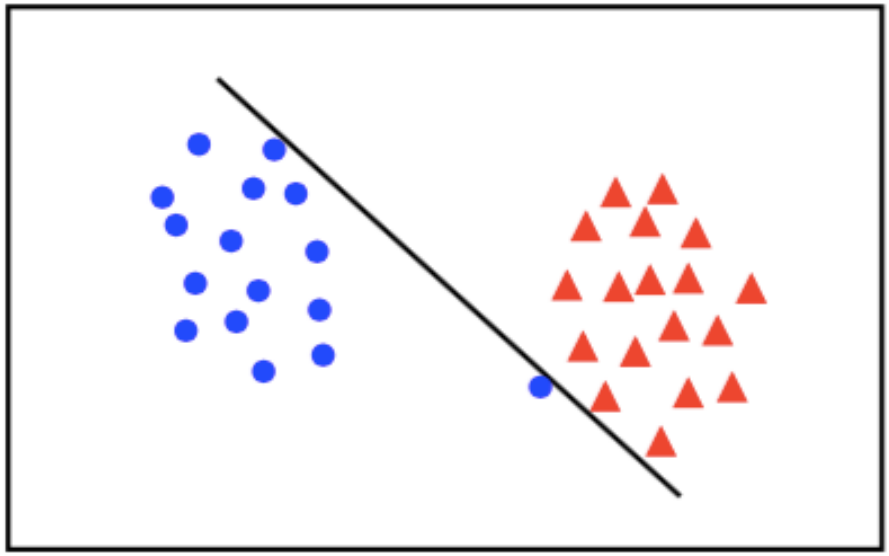
Dual:
$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha^i \alpha^j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle \qquad \boldsymbol{\alpha} \in \mathbb{R}^N \rightarrow O(N^3)$$
$$\text{s.t. :} \quad y^i \left(\sum_{j=1}^{N} \alpha^j y^j \langle \mathbf{x}^j, \mathbf{x}^i \rangle + b \right) \geq 1, \quad \forall i$$

Dual can be faster if N<D!

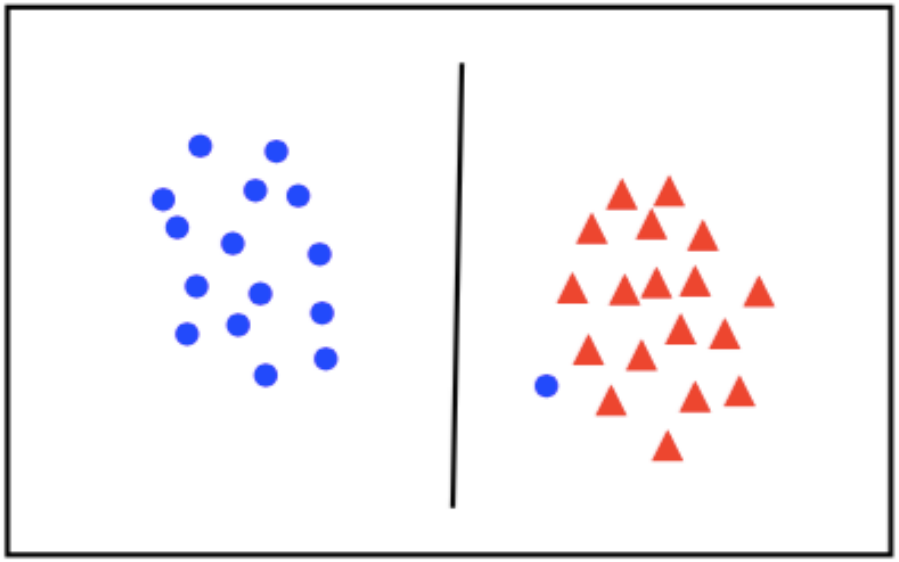
Primal and dual classifier forms:
$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \sum_{i=1}^{N} \alpha^i y^i \langle \mathbf{x}^i, \mathbf{x} \rangle + b$$

Dual form involves only inner products of features (=> kernel trick)

What is the “best” decision plane?



All points on the correct side!



But this looks better overall!

Best: understood at test time

Maybe we could sacrifice classifying some training points correctly

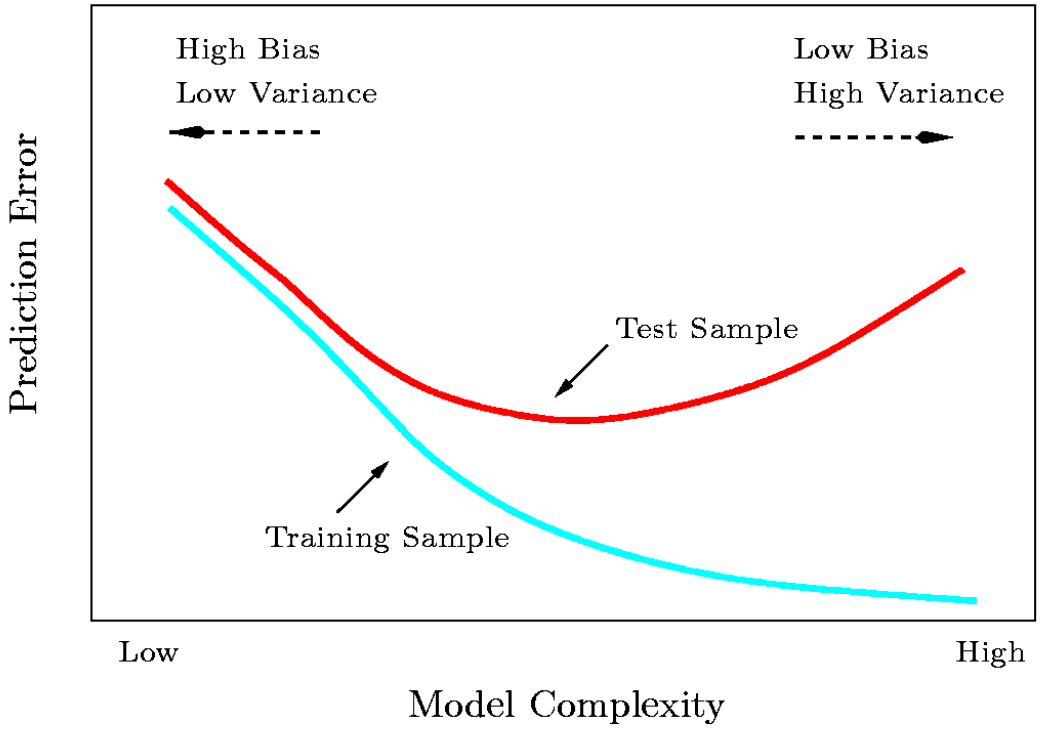
Tuning the model's complexity

A flexible model approximates the target function well in the training set

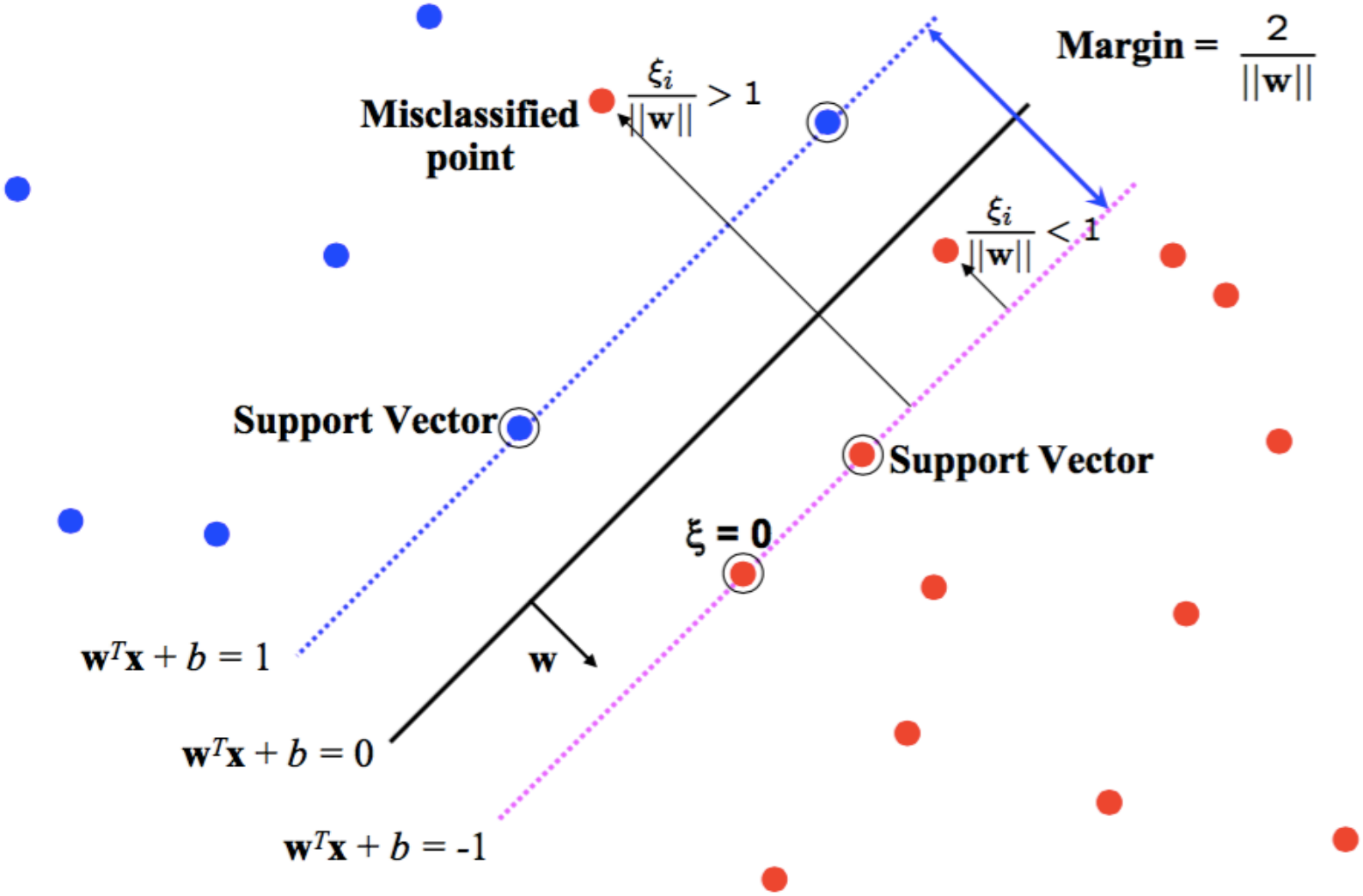
but can "overtrain" and have poor performance on the test set ("variance")

A rigid model's performance is more predictable in the test set

but the model may not be good even on the training set ("bias")



Slack variables: let us make (but also pay) some errors



Objective for non-separable data

$$\min_{\mathbf{w},\boldsymbol{\xi}} \|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi^i$$

newcomers

$$\text{s.t.}: \quad y^i(\mathbf{w}^T\mathbf{x}^i + b) \geq 1 - \xi^i, \quad \forall i$$

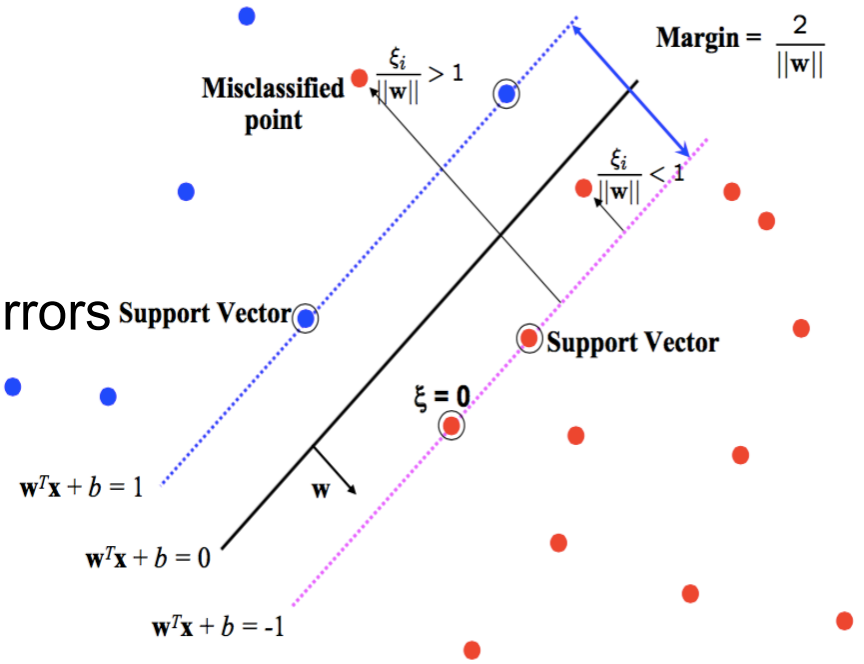
$$\xi^i \geq 0, \quad \forall i$$

misclassification when ξ>1

$\sum_i \xi^i$:upper bound on number of errors

C: hyperparameter

(cross-validation!)



Appendix

- Primal and Dual form of SVMs: the full story

References:

S. Boyd and L. Vandeberghe: Convex Optimization (textbook)

C. Burges: A tutorial on SVMs for pattern recognition

Duality

- Constrained optimization problem:

$$\begin{aligned} \min_{w} \quad & f(w) \\ s.t. \quad & h_i(w) = 0, \quad i = 1 \dots l \\ & g_i(w) \leq 0, \quad i = 1 \dots m \end{aligned}$$

- Equivalent to unconstrained problem:

$$\min_{w} \quad f_{uc}(w) = f(w) + \sum_{i=1}^{l} I_0(h_i(w)) + \sum_{i=1}^{m} I_+(g_i(w))$$

$$I_0(x) = \begin{cases} 0, & x = 0 \\ \infty, & x \neq 0 \end{cases}, \qquad I_+(x) = \begin{cases} 0, & x \leq 0 \\ \infty, & x > 0 \end{cases}$$

- Soften constraint terms $I_0(x_i) \rightarrow \lambda_i x_i \qquad I_+(x_i) \rightarrow \mu_i x_i, \mu > 0$

Lagrangian

- Replace hard constraints with soft ones

$$\min_w \quad f_{uc}(w) = f(w) + \sum_{i=1}^{l} I_0(h_i(w)) + \sum_{i=1}^{m} I_+(g_i(w))$$

$$L(w, \lambda, \mu) = f(w) + \sum_{i=1}^{l} \lambda_i h_i(w) + \sum_{i=1}^{m} \mu_i g_i(w), \quad \mu_i > 0 \forall i$$

- Observe that

$$f_{uc}(w) = \max_{\lambda, \mu : \mu_i > 0} L(w, \lambda, \mu)$$

- At an optimum:

$$f(w^*) = \min_w \max_{\lambda, \mu : \mu_i > 0} L(w, \lambda, \mu)$$

Lagrange Dual Function

- Form $$\theta(\lambda, \mu) = \inf_{w} L(w, \lambda, \mu)$$

- θ : lower bound on optimal value of the original problem

$$\begin{aligned} L(w^*, \lambda, \mu) &= f(w^*) + \sum_{i=1}^{l} \lambda_i h_i(w^*) + \sum_{i=1}^{m} \mu_i g_i(w^*) = \\ &\overset{w^*:feasible}{=} f(w^*) + \sum_{i=1}^{l} \lambda_i 0 + \sum_{i=1}^{m} \mu_i \underbrace{g_i(w^*)}_{<0} = \\ &\overset{\mu_i > 0}{\leq} f(w^*) \end{aligned}$$

- Therefore: $$\theta(\lambda, \mu) = \inf_{w} L(w, \lambda, \mu) \leq L(w^*, \lambda, \mu) \leq f(w^*)$$

Dual Problem

- Maximize the lower bound on the cost of the primal

$$\max_{\lambda,\mu} \quad \theta(\lambda,\mu)$$
$$s.t. \quad \mu_i > 0 \quad \forall i$$

- In general:

$$\begin{aligned} d^* &= \max_{\lambda,\mu} \theta(\lambda,\mu) \\ &= \max_{\lambda,\mu:\mu_i>0} \min_{w} L(w,\lambda,\mu) \\ &\leq \min_{w} \max_{\lambda,\mu:\mu_i>0} L(w,\lambda,\mu) \\ &= \min_{w} f_{uc}(w) = p^* \end{aligned}$$

- For convex cost and convex constraints (SVM case): $d^* = p^*$

Complementary Slackness

- Assume $d^* = p^*$
- There exists a feasible solution w^*, λ^*, μ^* to the primal and dual problems, such that $f(w^*) = \theta(\lambda^*, \mu^*)$
- We will have

$$\begin{aligned} f(w^*) &= \theta(\lambda^*, \mu^*) \\ &= \inf_w f(w) + \sum_{i=1}^{l} \lambda_i^* h_i(w) + \sum_{i=1}^{m} \mu_i^* f_i(w) \\ &\leq f(w^*) + \sum_{i=1}^{M} \lambda_i^* h_i(w^*) + \sum_{i=1}^{m} \mu_i^* f_i(w^*) \\ &\leq f(w^*) \end{aligned}$$

- This means $\mu_i^* f_i(w^*) = 0, \quad \forall i$

Karush-Kuhn Tucker (KKT) Conditions

- Solution of the primal problem:
  - minimum of the Lagrangan w.r.t. the primal variables
  - therefore $$\nabla f(w^*) + \sum_{i=1}^{l} \lambda_i \nabla h_i(w^*) + \sum_{i=1}^{m} \nabla f_i(w^*) = 0$$

- Putting all constraints together: KKT conditions

$$\begin{aligned} h_i(w^*) &= 0 \\ f_i(w^*) &\leq 0 \\ \mu_i f_i(w^*) &= 0 \\ \mu_i &\geq 0 \\ \nabla f(w^*) + \sum_{i=1}^{l} \lambda_i \nabla h_i(w^*) + \sum_{i=1}^{m} \nabla f_i(w^*) &= 0 \end{aligned}$$

Problem Lagrangian

Primal:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2$$

$$s.t. \quad -y^i(\mathbf{w}^T x^i + b) + 1 \leq 0, \quad i = 1 \ldots M$$

Lagrangian: $L(\mathbf{w}, b, \mu) = \frac{1}{2}|\mathbf{w}|^2 - \sum_{i=1}^{M} \mu_i [y^i(\mathbf{w}^T x^i + b) - 1] \qquad \mu_i \geq 0$

Optimum w.r.t. $\mathbf{w}$: $0 = \mathbf{w}^* - \sum_{i=1}^{M} \mu_i [y^i x^i] \qquad \mathbf{w}^* = \sum_{i=1}^{M} \mu_i y^i x^i$

Optimum w.r.t. b: $0 = \sum_{i=1}^{M} \mu_i y^i$

Dual for Large-Margin Classifier-I

Plug optimal values into Lagrangian:

$$
\begin{aligned}
\theta(\mu) &= L(\mathbf{w}^*, b^*, \mu) \\
&= \frac{1}{2}|\mathbf{w}^*|^2 - \sum_{i=1}^{M} \mu_i [y^i(\mathbf{w}^{*T} x^i + b) - 1] \\
&= \frac{1}{2}(\sum_{i=1}^{M} \mu_i y^i x^i)^T (\sum_{j=1}^{M} \mu_j y^j x^j) - \sum_{i=1}^{M} \mu_i [y^i((\sum_{j=1}^{M} \mu_j y^j x^j)^T x^i + b) - 1] \\
&= \sum_{i=1}^{M} \mu_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \mu_i \mu_j y^i y^j (x^i)^T (x^j) - b \sum_{i=1}^{M} \mu_i y^i
\end{aligned}
$$

Dual for Large-Margin Classifier-II

Equivalent optimization problem:

$$\max_{\mu} \quad \theta(\mu) = \sum_{i=1}^{M} \mu_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \mu_i \mu_j y^i y^j < x^i, x^j >$$

$$s.t. \quad \mu_i > 0, \quad \forall i$$

$$\sum_{i=1}^{M} \mu_i y^i = 0$$

Support Vectors

From complementary slackness (KKT) $\mu_i g_i(x) = 0$

where $g_i(x) = 1 - y^i(\mathbf{w}^T x^i + b) \ (\leq 0)$

Therefore: $\mu_i \neq 0 \rightarrow y^i(\mathbf{w}^T x^i + b) = 1$

Interpretation: μ is nonzero only for points on the margin (hardest points)

From mimimum w.r.t. w: $\mathbf{w}^* = \sum_{i=1}^{M} \mu_i y^i x^i$

Interpretation: only points on the margin contribute to the solution

- **`Support Vectors'**

Intuitively ok: we want to maximize the margins of the hardest cases

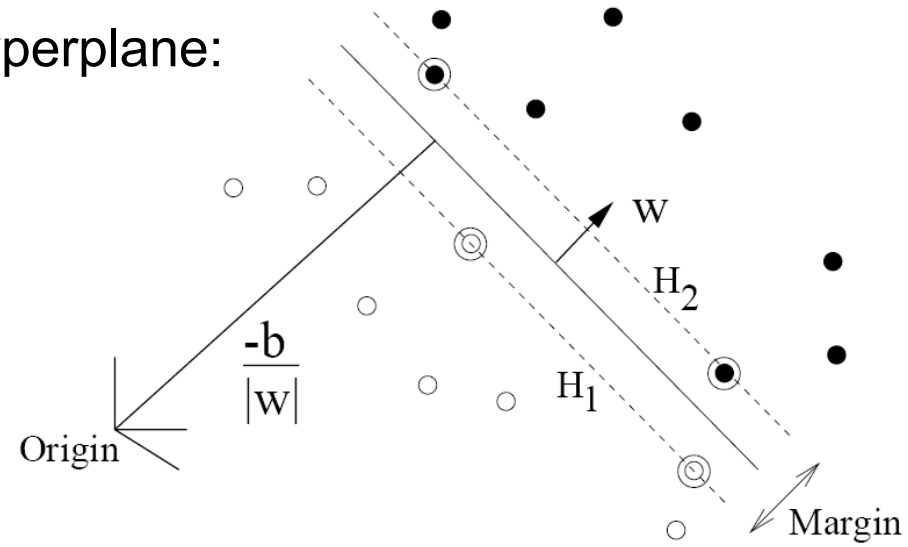
Decision Hyperplanes & Support Vectors

Use support vectors to determine b^*:

$$y^i(\mathbf{w}^T x^i + b) = 1, \quad \forall i \in S$$

$$b^* = \frac{1}{N_S} \sum_{i \in S} (y^i - \mathbf{w}^T x^i)$$

Support Vector Machine decision hyperplane:



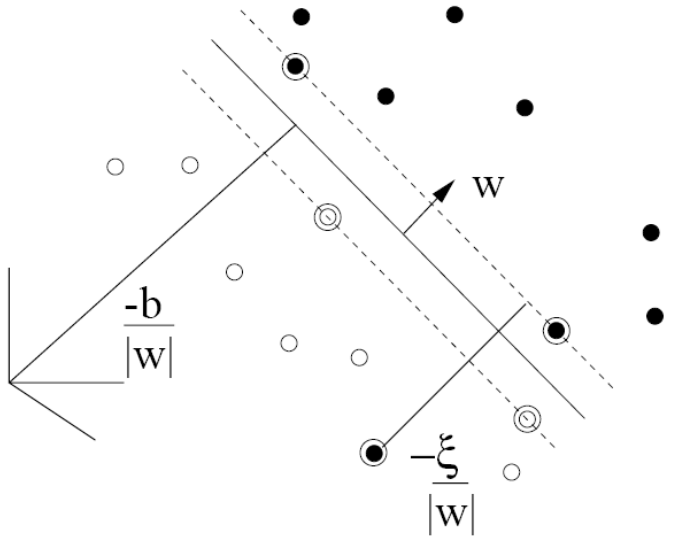
Non-separable data

Primal:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}\xi^i$$

$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i$$

$$\xi^i \geq 0$$



Lagrangian:

$$L(\mathbf{w}, b, \xi, \mu, \nu) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{M}\xi^i - \sum_{i=1}^{M}\mu_i[y^i(\mathbf{w}^T x + b) - 1 + \xi] - \sum_{i=1}^{M}\nu_i\xi_i$$

Dual:

$$\max_{\mu} \quad \sum_{i=1}^{M}\mu_i - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} y^i y^j \mu_i \mu_j < x^i, x^j >$$

$$s.t. \quad 0 \leq \mu_i \leq C$$

$$\sum_{i=1}^{M}\mu_i y^i = 0$$

$$\mu_i \geq 0, \quad \forall i$$

$$\nu_i \geq 0, \quad \forall i$$

KKT conditions – nonseparable case

$$
\begin{aligned}
C - \mu^i - \nu^i &= 0 && (1)\\
y^i(<\mathbf{x}^i, \mathbf{w}> + b) - 1 + \xi^i &\geq 0 && (2)\\
\mu^i \left[y^i(<\mathbf{x}^i, \mathbf{w}> + b) - 1 + \xi^i\right] &= 0 && (3)\\
\nu^i \xi^i &= 0 && (4)\\
\xi^i &\geq 0 && (5)\\
\mu^i &\geq 0 && (6)\\
\nu^i &\geq 0 && (7)
\end{aligned}
$$

Complementary slackness

Complementary slackness

Case analysis:

$$y^i(<\mathbf{x}^i, \mathbf{w}> + b) > 1 \xrightarrow{3:\xi^i \geq 0} \mu_i = 0$$

$$y^i(<\mathbf{x}^i, \mathbf{w}> + b) < 1 \xrightarrow{2:\xi^i > 0 \rightarrow 4:\nu^i = 0 \rightarrow 1:} \mu_i = C$$

$$y^i(<\mathbf{x}^i, \mathbf{w}> + b) = 1 \xrightarrow{3:\mu^i \xi^i = 0 \rightarrow 1:} \mu_i \in [0, C]$$

Interpretation: influence, μ, of any training point is bounded in [0,C]

Hinge Loss

$$
\begin{aligned}
C - \mu^i - \nu^i &= 0 \qquad (1)\\
\mu^i \left[y^i (< \mathbf{x}^i, \mathbf{w} > + b) - 1 + \xi^i \right] &= 0 \qquad (3)\\
\nu^i \xi^i &= 0 \qquad (4)\\
\xi^i &\geq 0 \qquad (5)
\end{aligned}
$$

$$\mu^i \neq 0 \overset{(3)}{\rightarrow} \xi^i = 1 - y^i (< \mathbf{x}^i, \mathbf{w} > + b)$$

$$\mu^i = 0 \overset{(1)}{\rightarrow} \nu^i = C \quad \overset{(4)}{\rightarrow} \xi^i = 0$$

$$\xi^i = \max(0, 1 - y^i (< \mathbf{x}^i, \mathbf{w} > + b))$$

$$
\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{M} \xi^i \\
s.t. \quad & y^i (\mathbf{w}^T x^i + b) \geq 1 - \xi^i \\
& \xi^i \geq 0
\end{aligned}
\quad \Longleftrightarrow \quad
L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y^i h_{\mathbf{w},b}(x^i))
$$

Loss function for SVM training

Optimization problem:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\max(0, 1 - y^i h_{\mathbf{w},b}(x^i))$$

$$\propto \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\underbrace{\max(0, 1 - y^i h_{\mathbf{w},b}(x^i))}_{l(y^i, x^i)}$$

Hinge loss: