Support Vector Machines

Slides Credits : Vineeth N Balasubramanian

आई आई टी हैदराबाद

IIT Hyderabad

Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

SVM: Overview and History

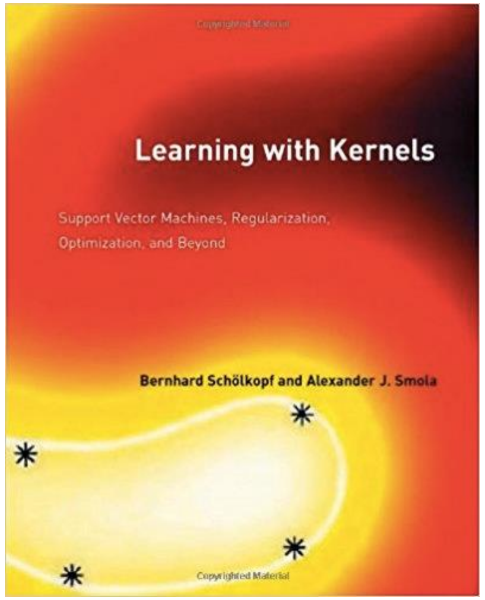
- A discriminative classifier
  - Non-parametric, Inductive
- SVM is inspired from statistical learning theory
- SVM was developed in 1992 by Vapnik, Guyon and Boser
- SVM became popular because of its success in handwritten digit recognition
- Has been one of the go-to methods in machine learning since the mid-1990s (only recently displaced by deep learning)

Papers that introduced SVM in its current form

- Boser, B. E.; Guyon, I. M.; Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory – COLT '92.
- Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297.

SVM: Overview and History

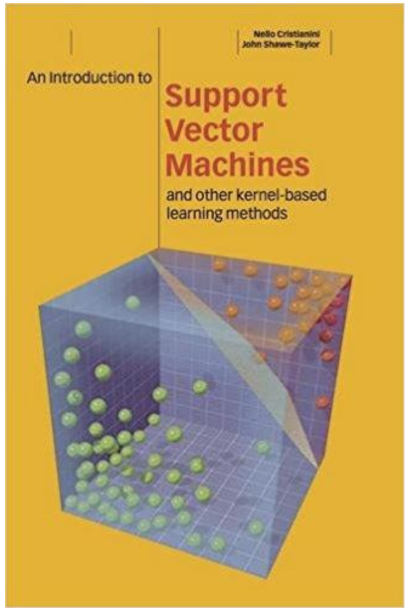
- Associated key words
  - Large-margin classifier, Max-margin classifier, Kernel methods, Reproducing kernel Hibert space, Statistical learning theory



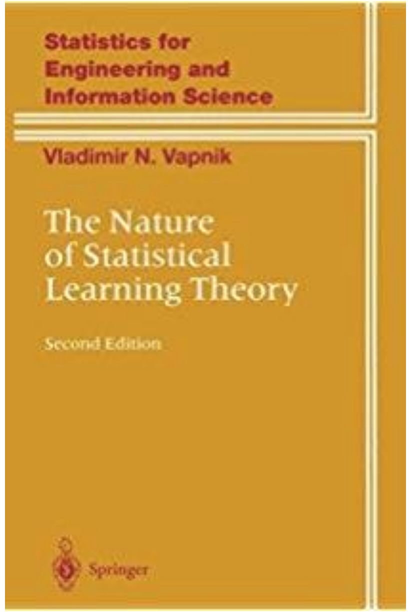
Learning with Kernels

Support Vector Machines, Regularization, Optimization, and Beyond

Bernhard Schölkopf and Alexander J. Smola


Nello Cristianini
John Shawe-Taylor

An Introduction to Support Vector Machines and other kernel-based learning methods

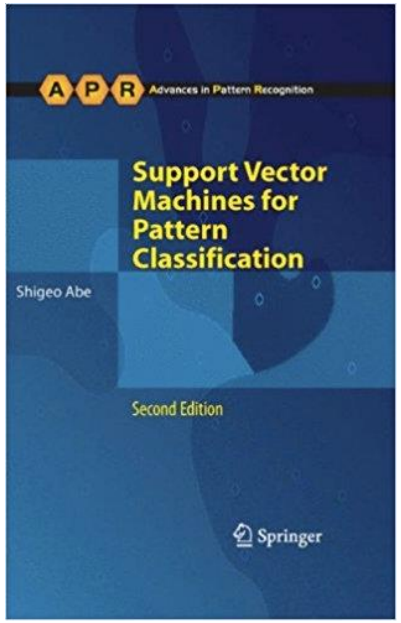

Statistics for Engineering and Information Science

Vladimir N. Vapnik

The Nature of Statistical Learning Theory

Second Edition

Springer


APR Advances in Pattern Recognition

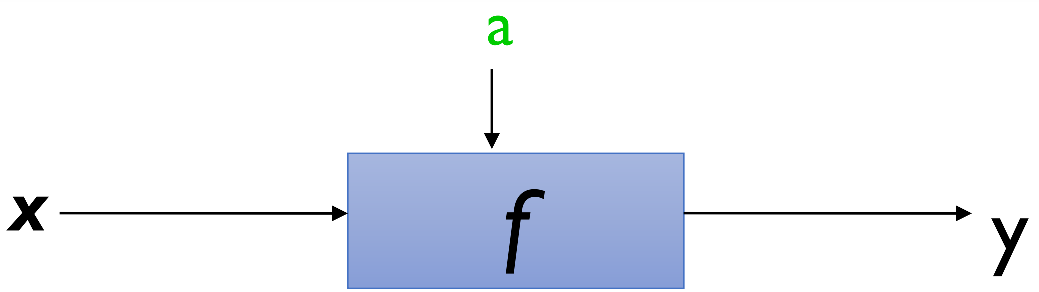
Support Vector Machines for Pattern Classification

Shigeo Abe

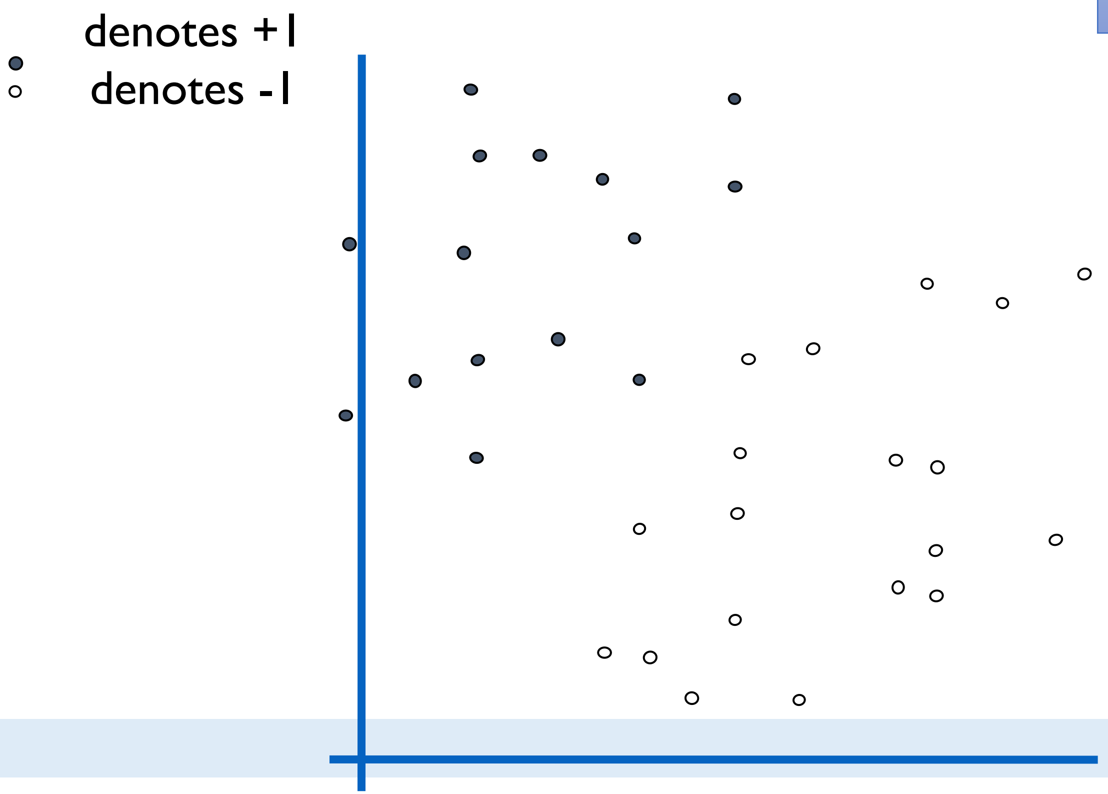
Second Edition

Springer

Linear Classifiers

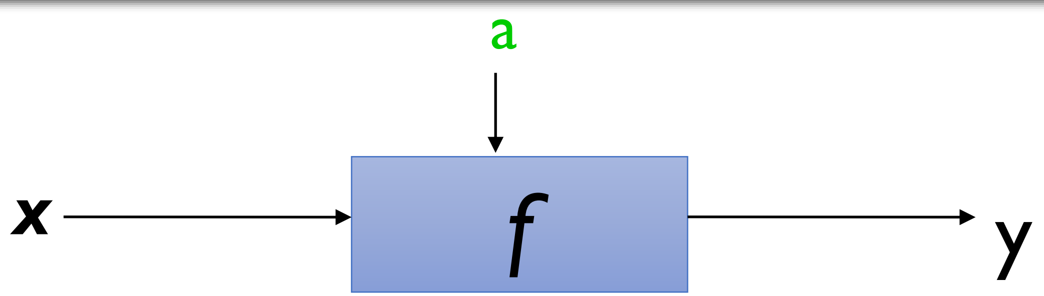


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}. \mathbf{x} + b)$

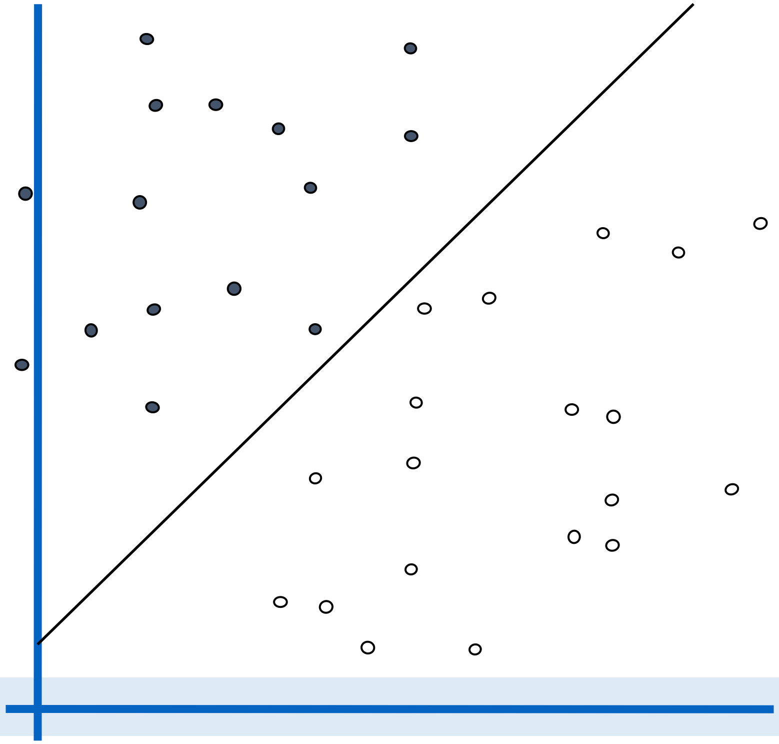


How would you classify this data?

Linear Classifiers



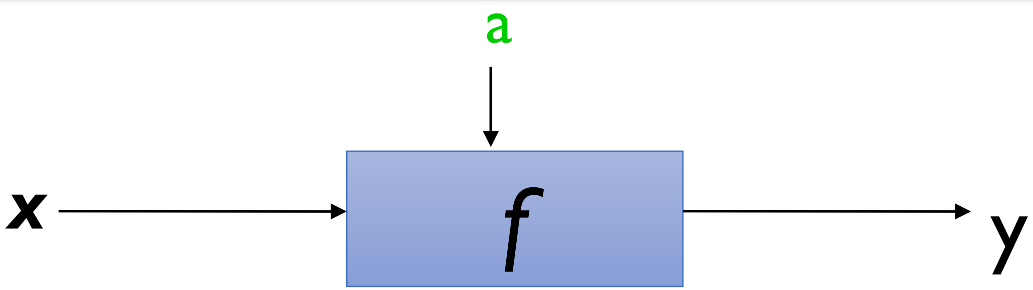
● denotes +1
○ denotes -1



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

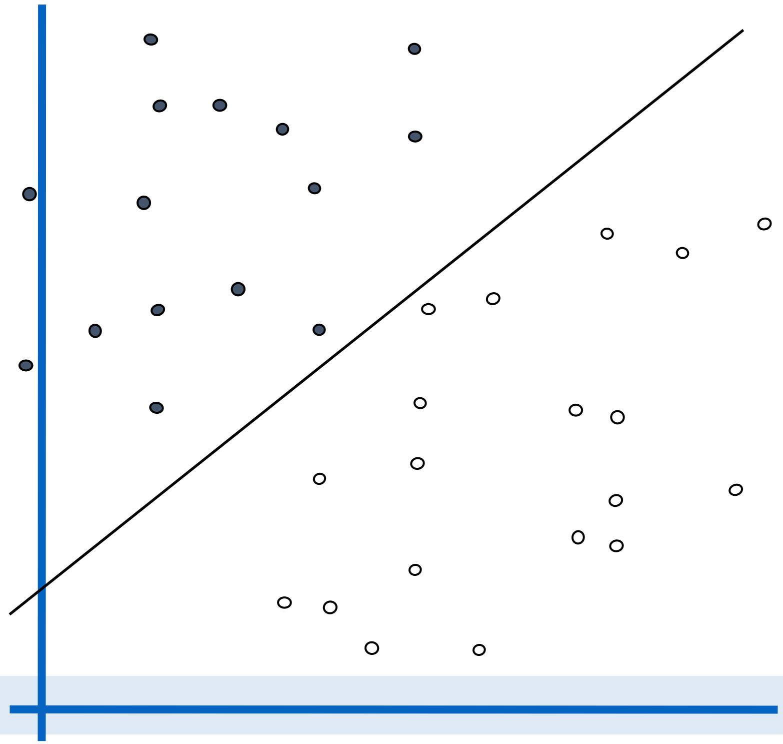
How would you classify this data?

Linear Classifiers



● denotes +1

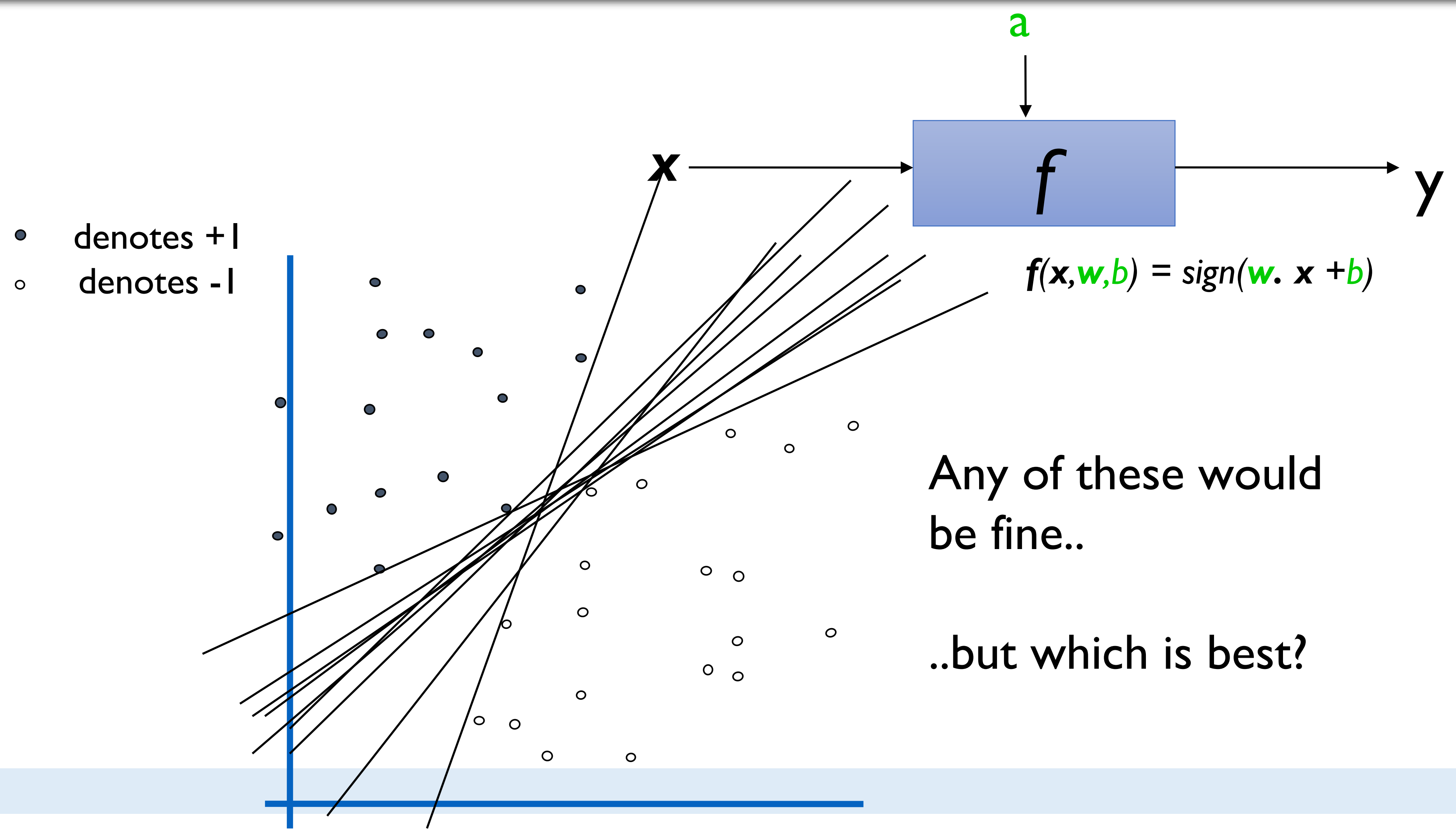
○ denotes -1



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

How would you classify this data?

Linear Classifiers

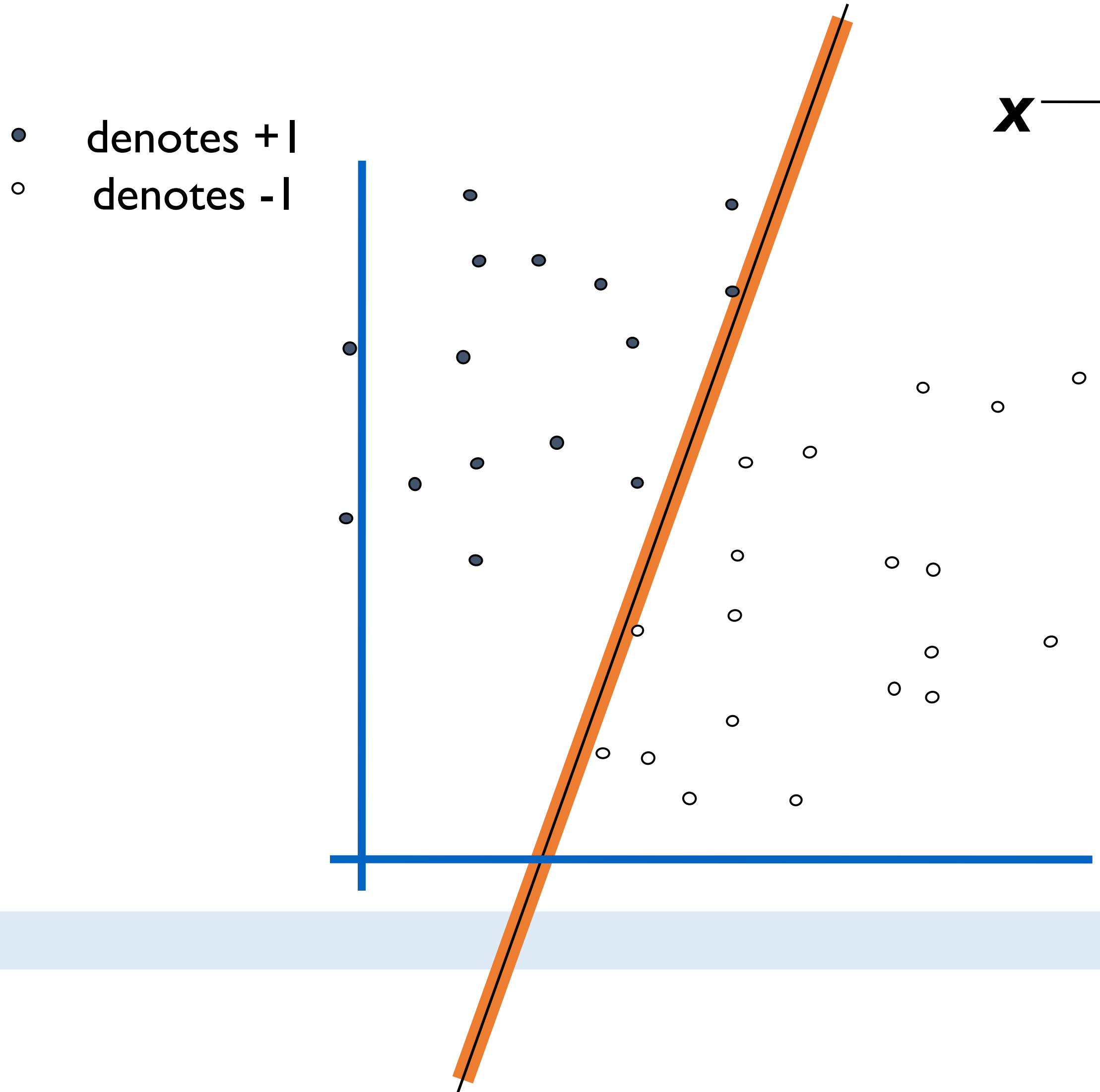


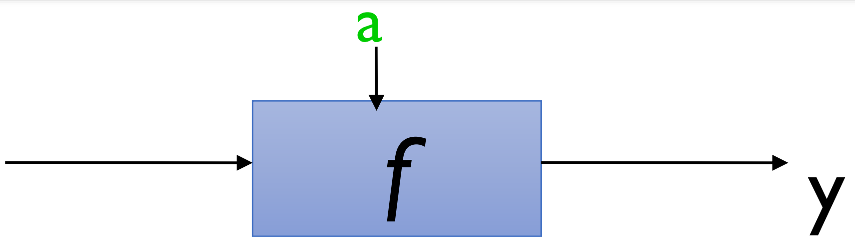
$f(\boldsymbol{x},\mathbf{w},b) = sign(\mathbf{w}.\ \boldsymbol{x} + b)$

Any of these would be fine..

..but which is best?

Linear Classifiers

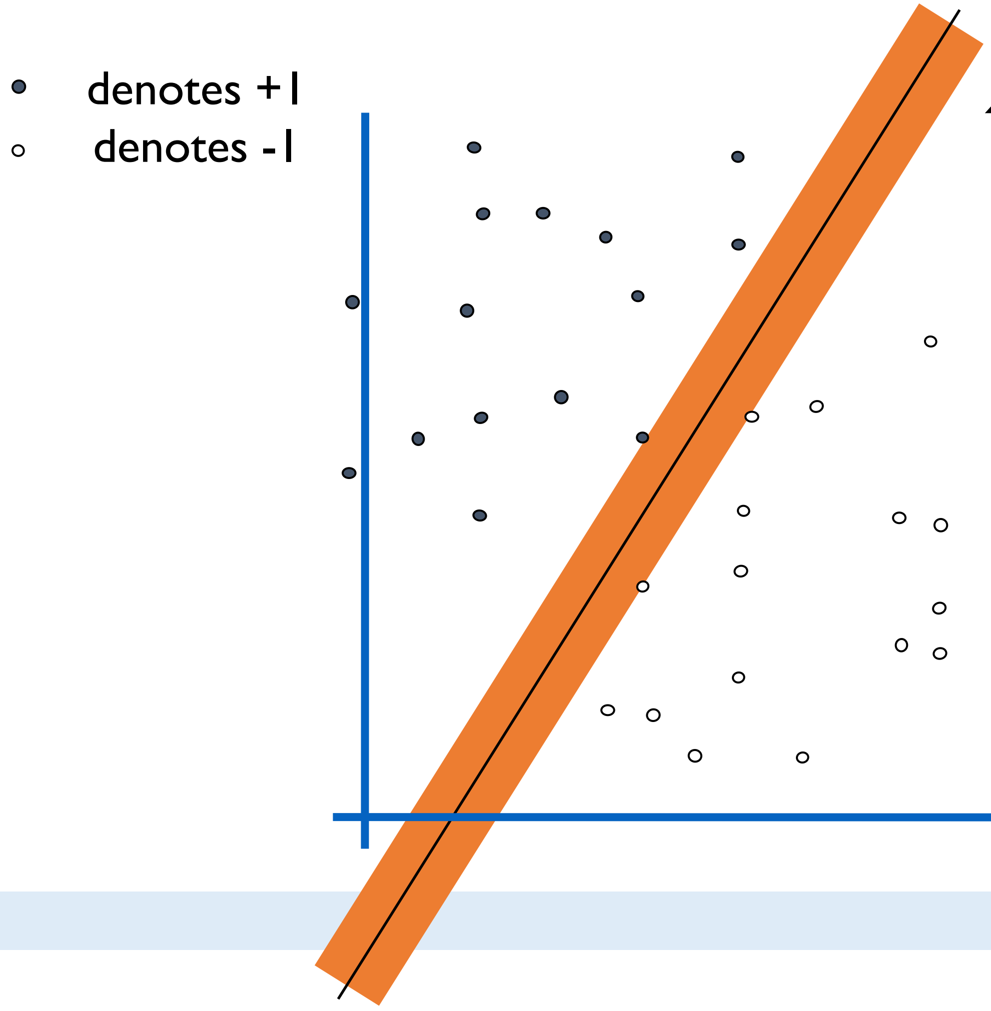


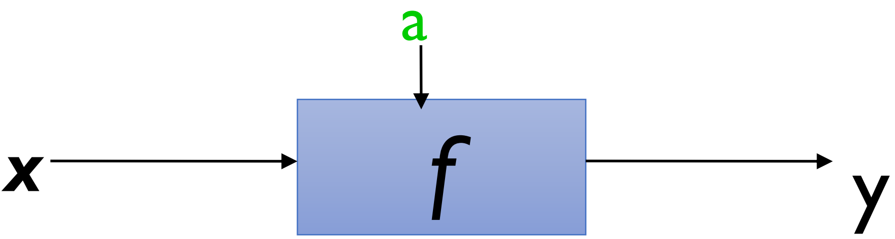


f(**x**,**w**,b) = *sign*(**w**. **x** +b)

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Linear Classifiers



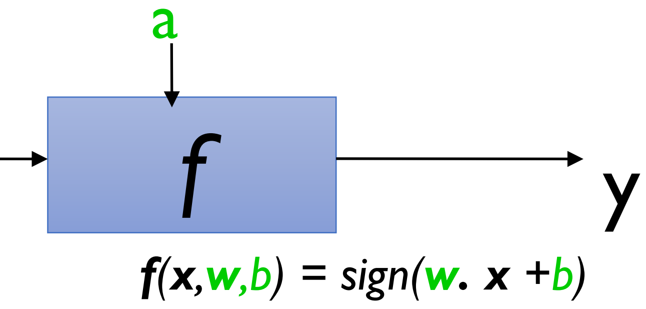


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

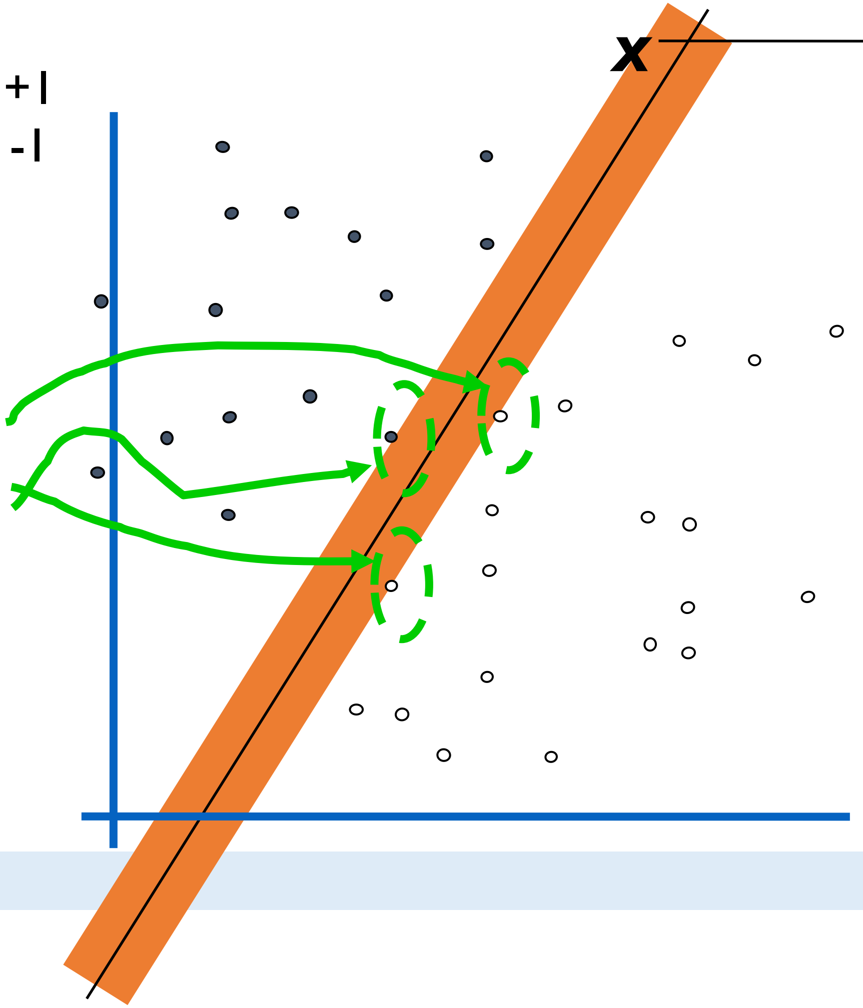
Maximum Margin Classifier



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} + b)$

● denotes +1

○ denotes -1

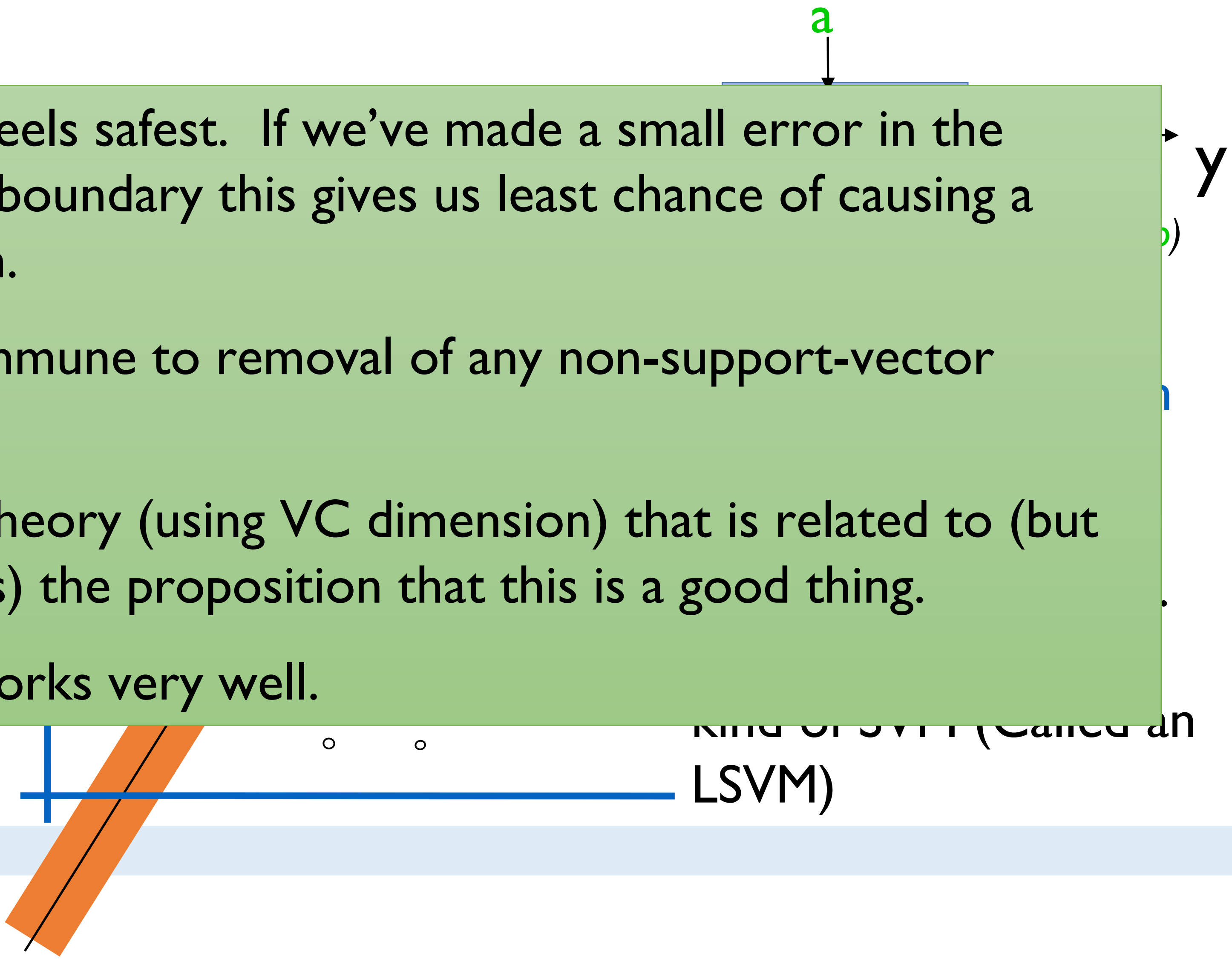


Support Vectors are those data points that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of SVM (Called an LSVM)

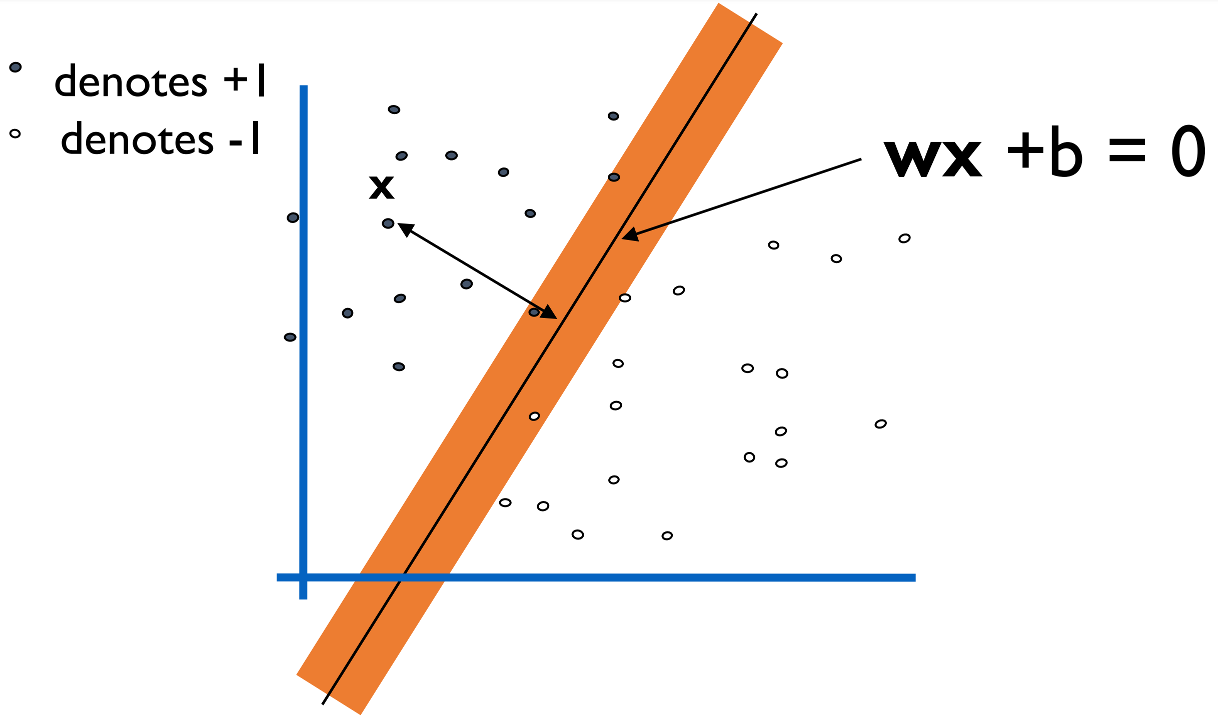
Why Maximum Margin?

- Intuitively this feels safest. If we've made a small error in the location of the boundary this gives us least chance of causing a misclassification.
- The model is immune to removal of any non-support-vector datapoints.
- There's some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing.
- Empirically it works very well.



kind of SVM (Called an LSVM)

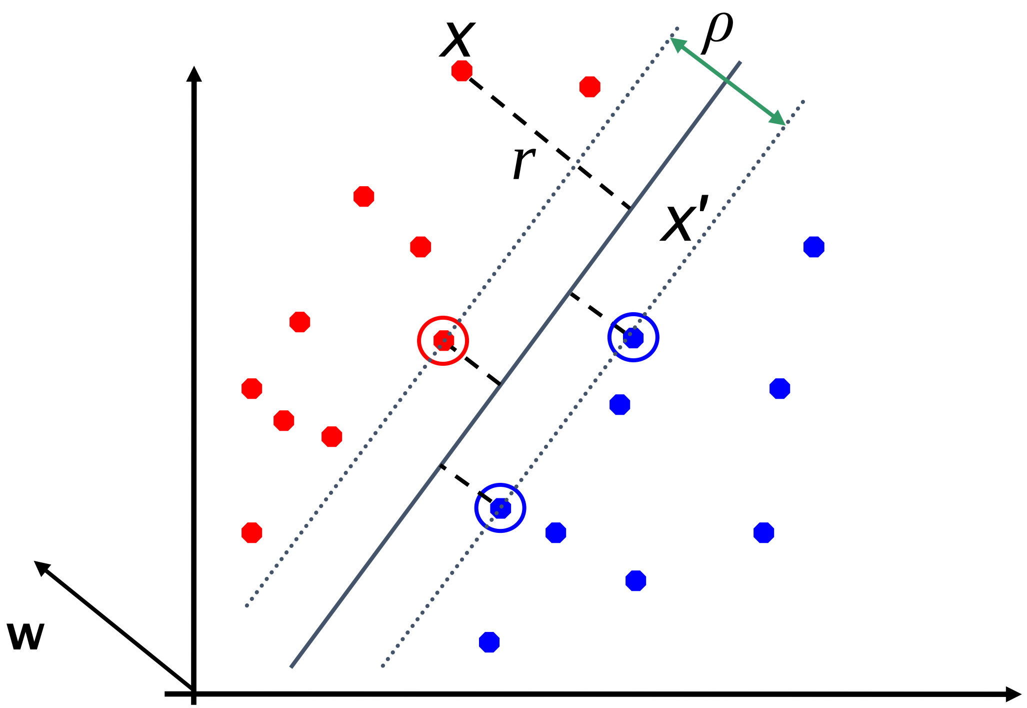
Estimating the Margin



- What is the distance expression for a point **x** to a line **wx**+b= 0?

Estimating the Margin

- Distance from example to the separator is $r = y\frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$



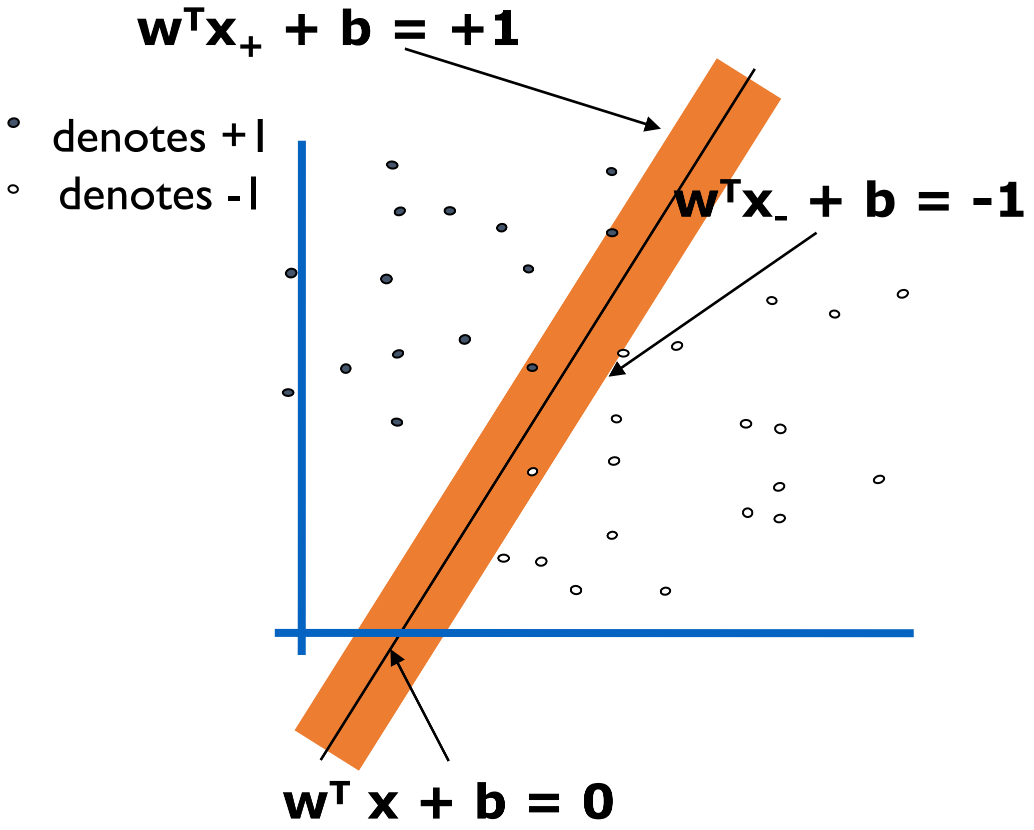
Derivation of finding *r*:

- Dotted line **x’− x** is perpendicular to decision boundary, so parallel to **w**.
- Unit vector is **w**/||**w**||, so line is r**w**/||**w**||.
- **x’** = **x** – yr**w**/||**w**||.
- **x’** satisfies $\mathbf{w}^T\mathbf{x'}$+ b = 0.
- So $\mathbf{w}^T$(**x** –yr**w**/||**w**||) + b = 0
- Recall that ||**w**|| = sqrt($\mathbf{w}^T\mathbf{w}$).
- So $\mathbf{w}^T\mathbf{x}$ –yr||**w**|| + b = 0
- So, solving for r gives: r = y($\mathbf{w}^T\mathbf{x}$ + b)/||**w**||

आई आई टी हैदराबाद
IIT Hyderabad

Estimating the Margin

- Since $\mathbf{w}^T\mathbf{x} + b = 0$ and $c(\mathbf{w}^T\mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization of $\mathbf{w}$ (i.e. c)
- Let us choose normalization such that $\mathbf{w}^T\mathbf{x}_+ + b = +1$ and $\mathbf{w}^T\mathbf{x}_- + b = -1$ for the positive and negative support vectors respectively

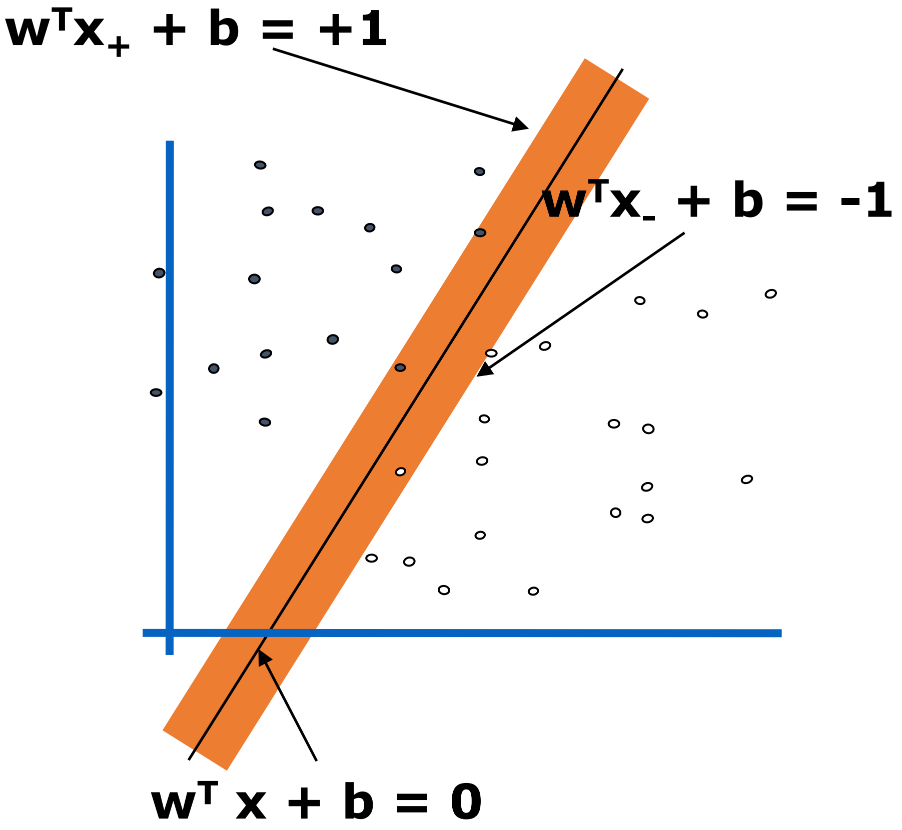


$\mathbf{w}^T\mathbf{x}_+ + \mathbf{b} = +1$
● denotes +1
○ denotes -1
$\mathbf{w}^T\mathbf{x}_- + \mathbf{b} = -1$
$\mathbf{w}^T\ \mathbf{x} + \mathbf{b} = 0$

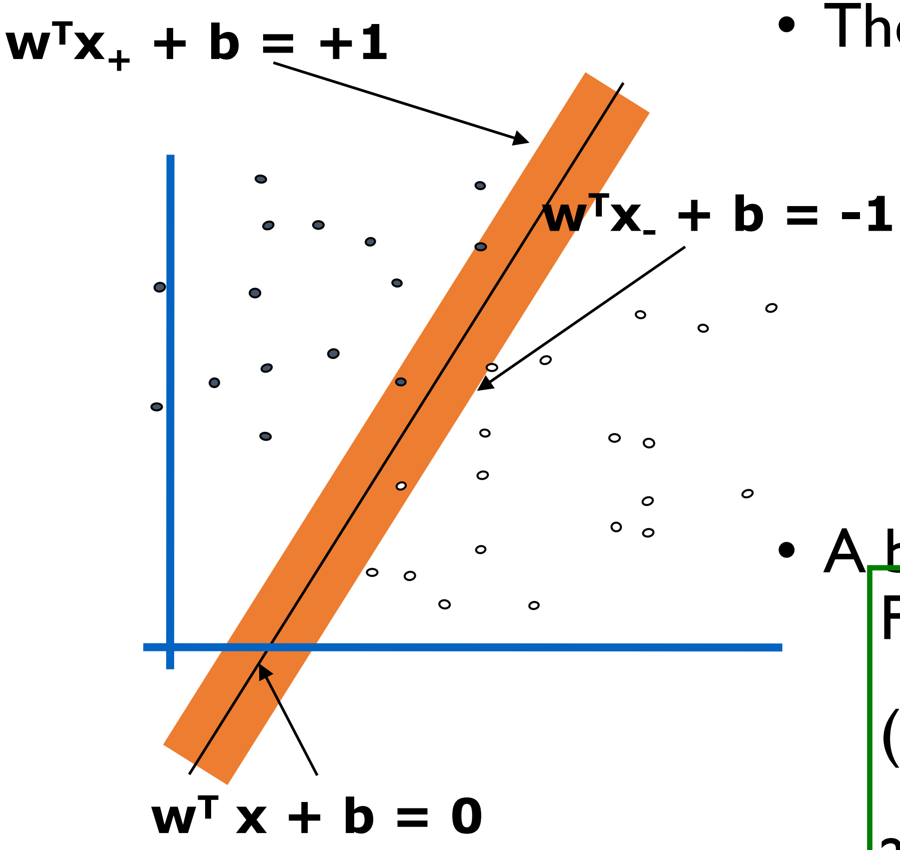
Estimating the Margin

- Since $\mathbf{w}^T\mathbf{x} + \mathbf{b} = 0$ and $c(\mathbf{w}^T\mathbf{x} + \mathbf{b}) = 0$ define the same plane, we have the freedom to choose the normalization of **w** (i.e. c)
- Let us choose normalization such that $\mathbf{w}^T\mathbf{x}_+ + \mathbf{b} = +1$ and $\mathbf{w}^T\mathbf{x}_- + \mathbf{b} = -1$ for the positive and negative support vectors respectively
- Hence, margin now is:

$$(+1)*\frac{\mathbf{w}^T\mathbf{x}_+ + b}{\|\mathbf{w}\|} + (-1).\frac{\mathbf{w}^T\mathbf{x}_- + b}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$


$\mathbf{w}^T\mathbf{x}_+ + \mathbf{b} = +1$
$\mathbf{w}^T\mathbf{x}_- + \mathbf{b} = -1$
$\mathbf{w}^T\mathbf{x} + \mathbf{b} = 0$

Maximizing the Margin



$\mathbf{w}^T\mathbf{x}_+ + \mathbf{b} = +1$
$\mathbf{w}^T\mathbf{x}_- + \mathbf{b} = -1$
$\mathbf{w}^T\ \mathbf{x} + \mathbf{b} = 0$

- Then we can formulate the *quadratic optimization problem:*

$\rho = \frac{2}{\|\mathbf{w}\|}$

Find **w** and *b* such that

is maximized; and for all $\{(\mathbf{x}_i\ ,\ y_i)\}$

$\mathbf{w}^T\mathbf{x}_i + b \geq 1$ if y_i=+1; $\mathbf{w}^T\mathbf{x}_i + b \leq$ **-1** if y_i = -1

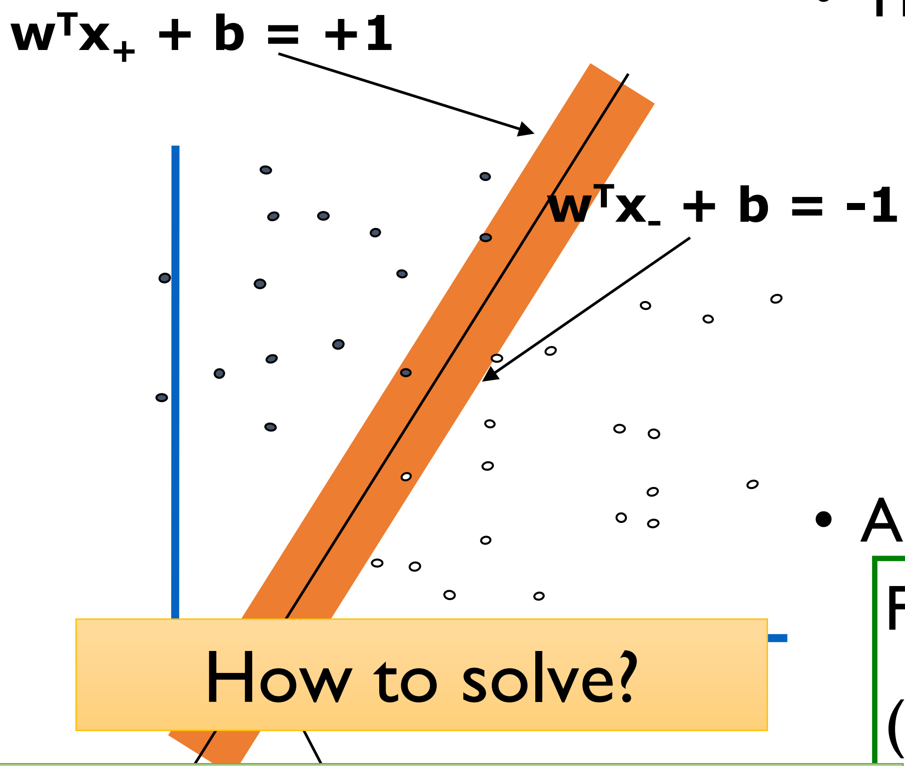
- A better formulation (min **||w||** = max 1/ **||w||**):

Find **w** and *b* such that

$(\frac{1}{2}\ \mathbf{w}^T\mathbf{w})$ is minimized

and for all $\{(\mathbf{x}_i\ ,y_i)\}$: $y_i\ (\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

Maximizing the Margin


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

- Then we can formulate the *quadratic optimization problem:*

Find **w** and *b* such that

$\rho = \dfrac{2}{\|\mathbf{w}\|}$ is maximized; and for all $\{(\mathbf{x}_i , y_i)\}$

$\mathbf{w}^T\mathbf{x}_i + b \geq 1$ if y_i=+1; $\mathbf{w}^T\mathbf{x}_i + b \leq$ **-**1 if y_i = -1

- A better formulation (min $\|\mathbf{w}\|$ = max 1/ $\|\mathbf{w}\|$):

Find **w** and *b* such that

($\frac{1}{2}\,\mathbf{w}^T\mathbf{w}$) is minimized

nd for all $\{(\mathbf{x}_i ,y_i)\}$: $y_i\,(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

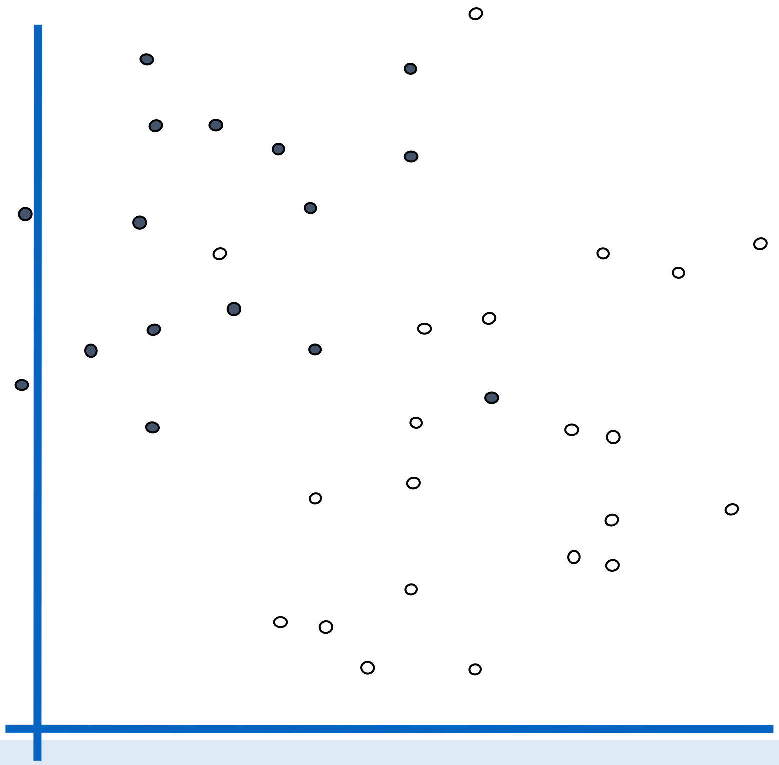
How to solve?

Quadratic Programming

Non-separable Data

● denotes +1

○ denotes -1



This is going to be a problem!

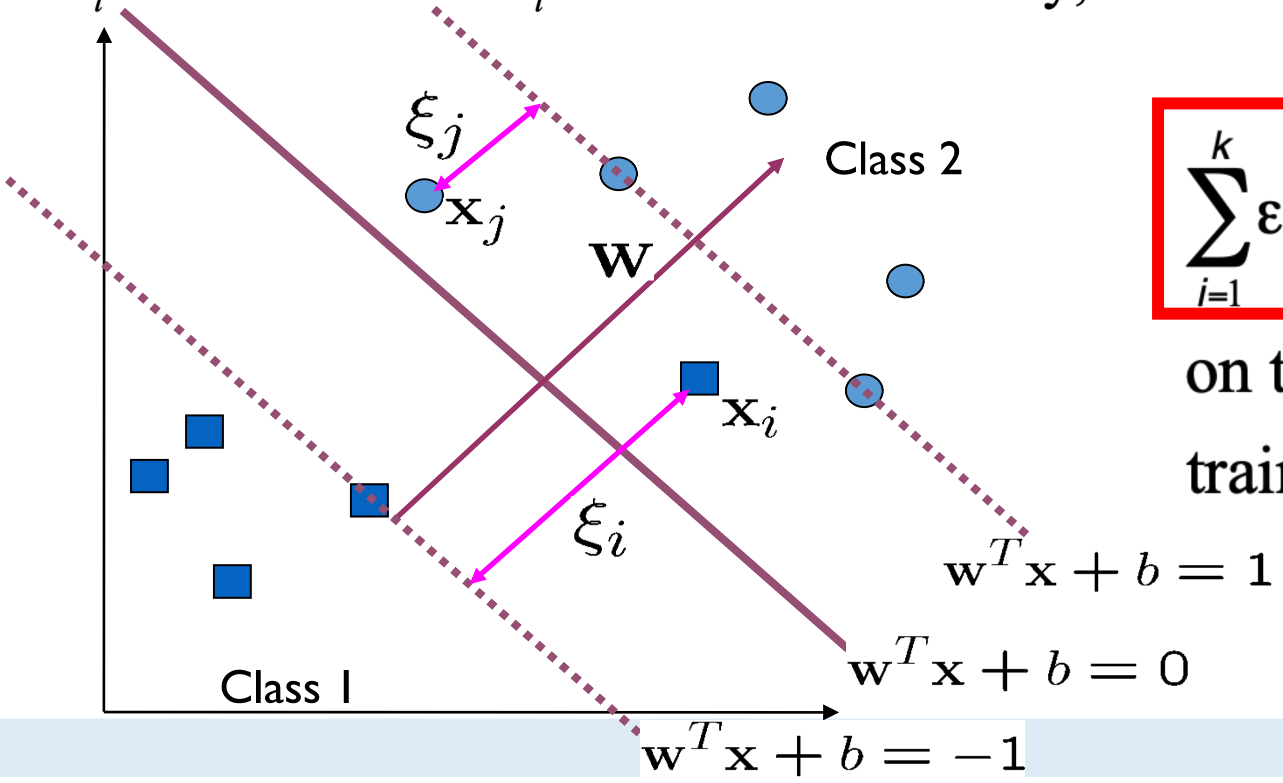
What should we do?

SVM for Noisy Data

$\varepsilon_i \geq 1 \quad \Leftrightarrow \quad y_i(wx_i + b) < 0,$ i.e., misclassification

slack parameter $\quad 0 \boxed{?}\varepsilon_i \boxed{?} \quad \Leftrightarrow \quad x_i$ is correctly classified, but lies inside the margin

$\varepsilon_i = 0 \quad \Leftrightarrow \quad x_i$ is classified correctly, and lies outside the margin



ξ_j
$\mathbf{x}_j$
$\mathbf{w}$
Class 2
$\mathbf{x}_i$
ξ_i
Class 1
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

$\boxed{\sum_{i=1}^{k} \varepsilon_i}$ is an upper bound on the number of training errors.

SVM for Noisy Data

$$\{\vec{w}^*, b^*\} = \min_{\vec{w},b} \sum_{i=1}^{d} w_i^2 + c\sum_{j=1}^{N} \varepsilon_j$$

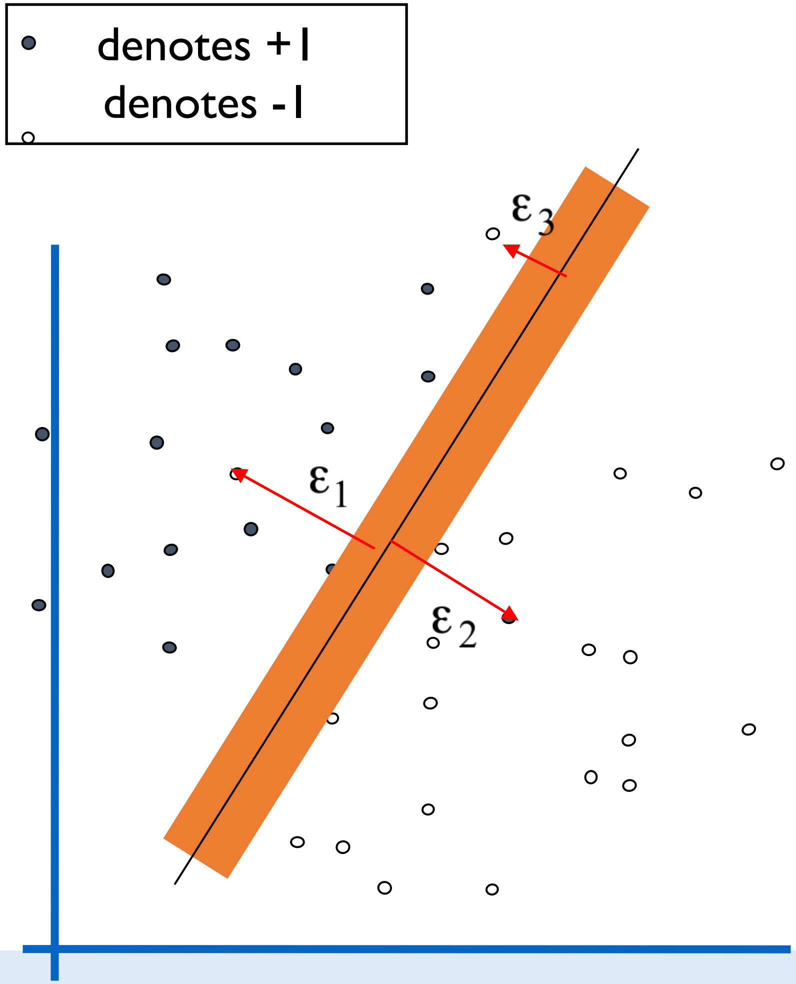
$$y_1\left(\vec{w}\cdot\vec{x}_1 + b\right) \geq 1 - \varepsilon_1, \varepsilon_1 \geq 0$$

$$y_2\left(\vec{w}\cdot\vec{x}_2 + b\right) \geq 1 - \varepsilon_2, \varepsilon_2 \geq 0$$

$$\ldots$$

$$y_N\left(\vec{w}\cdot\vec{x}_N + b\right) \geq 1 - \varepsilon_N, \varepsilon_N \geq 0$$

Balance the trade off between margin and classification errors



● denotes +1

○ denotes -1

Soft-Margin SVM : SVM for Noisy Data

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}||w||^2 + C\sum_n \xi_n \\ \text{subj. to} \quad & y_n\left(w \cdot x_n + b\right) \geq 1 - \xi_n && (\forall n) \\ & \xi_n \geq 0 && (\forall n) \end{aligned}$$

$$y_n\left(w \cdot x_n + b\right) - 1 + \xi_n \geq 0.$$

IIT Hyderabad

Soft-Margin SVM : SVM for Noisy Data

$$\begin{aligned} \min_{w,b,\boldsymbol{\xi}} \quad & \frac{1}{2}||w||^2 + C\sum_n \xi_n \\ \text{subj. to} \quad & y_n\left(w \cdot x_n + b\right) \geq 1 - \xi_n \qquad (\forall n) \\ & \xi_n \geq 0 \qquad (\forall n) \end{aligned}$$

$$y_n\left(w \cdot x_n + b\right) - 1 + \xi_n \geq 0.$$

$$\begin{aligned} \mathcal{L}(w, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = & \frac{1}{2}||w||^2 + C\sum_n \xi_n - \sum_n \beta_n \xi_n \\ & - \sum_n \alpha_n \left[y_n\left(w \cdot x_n + b\right) - 1 + \xi_n\right] \end{aligned}$$

$$\min_{w,b,\boldsymbol{\xi}} \max_{\boldsymbol{\alpha} \geq 0} \max_{\boldsymbol{\beta} \geq 0} \mathcal{L}(w, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Soft-Margin SVM : SVM for Noisy Data

$$\mathcal{L}(w, b, \xi, \alpha, \beta) = \frac{1}{2}\,||w||^2 + C\sum_n \xi_n - \sum_n \beta_n \xi_n \\ - \sum_n \alpha_n \left[y_n \left(w \cdot x_n + b\right) - 1 + \xi_n\right]$$

$$\min_{w,b,\xi}\ \max_{\alpha \geq 0}\ \max_{\beta \geq 0} \mathcal{L}(w, b, \xi, \alpha, \beta)$$

$$\nabla_w \mathcal{L} = w - \sum_n \alpha_n y_n x_n = 0 \quad \iff \quad w = \sum_n \alpha_n y_n x_n$$

$$\mathcal{L}(b, \xi, \alpha, \beta) = \frac{1}{2}\left|\left|\sum_m \alpha_m y_m x_m\right|\right|^2 + C\sum_n \xi_n - \sum_n \beta_n \xi_n \qquad (11 \\ - \sum_n \alpha_n \left[y_n \left(\left[\sum_m \alpha_m y_m x_m\right] \cdot x_n + b\right) - 1 + \xi_n\right]$$

Soft-Margin SVM : SVM for Noisy Data

$$\mathcal{L}(b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\left\|\sum_m \alpha_m y_m x_m\right\|^2 + C\sum_n \xi_n - \sum_n \beta_n \xi_n \qquad (11$$

$$-\sum_n \alpha_n \left[y_n \left(\left[\sum_m \alpha_m y_m x_m\right] \cdot x_n + b\right) - 1 + \xi_n\right]$$

$$\mathcal{L}(b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\sum_n \sum_m \alpha_n \alpha_m y_n y_m x_n \cdot x_m + \sum_n (C - \beta_n)\xi_n \qquad (11$$

$$-\sum_n \sum_m \alpha_n \alpha_m y_n y_m x_n \cdot x_m - \sum_n \alpha_n \left(y_n b - 1 + \xi_n\right)$$

$$(11$$

$$= -\frac{1}{2}\sum_n \sum_m \alpha_n \alpha_m y_n y_m x_n \cdot x_m + \sum_n (C - \beta_n)\xi_n \qquad (11$$

$$-b\sum_n \alpha_n y_n - \sum_n \alpha_n (\xi_n - 1) \qquad (11$$

Soft-Margin SVM : SVM for Noisy Data

$$\mathcal{L}(b,\xi,\alpha,\beta) = \frac{1}{2}\sum_n\sum_m \alpha_n\alpha_m y_n y_m x_n \cdot x_m + \sum_n (C-\beta_n)\xi_n \qquad (11$$

$$-\sum_n\sum_m \alpha_n\alpha_m y_n y_m x_n \cdot x_m - \sum_n \alpha_n \left(y_n b - 1 + \xi_n\right)$$

$$(11$$

$$= -\frac{1}{2}\sum_n\sum_m \alpha_n\alpha_m y_n y_m x_n \cdot x_m + \sum_n (C-\beta_n)\xi_n \qquad (11$$

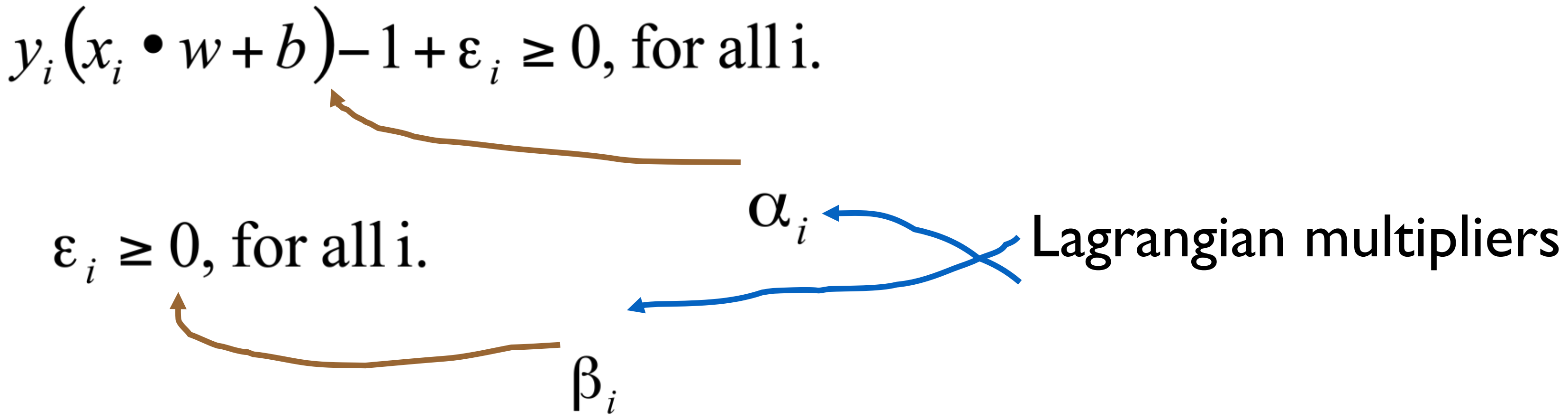
$$-b\sum_n \alpha_n y_n - \sum_n \alpha_n(\xi_n - 1) \qquad (11$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_n \alpha_n y_n = 0 \qquad \frac{\partial \mathcal{L}}{\partial \xi_n} = C - \beta_n - \alpha_n \iff C - \beta_n = \alpha_n$$

$\sum_n (C-\beta_n)\xi_n$ as $\sum_n \alpha_n\xi_n$. $\qquad \alpha_n \leq C,$

SVM for Noisy Data

- Use the Lagrangian formulation for the optimization problem.
- Introduce a positive Lagrangian multiplier for each inequality constraint.



$$y_i(x_i \bullet w + b) - 1 + \varepsilon_i \geq 0, \text{ for all i.}$$

$$\alpha_i$$

Lagrangian multipliers

$$\varepsilon_i \geq 0, \text{ for all i.}$$

$$\beta_i$$

Get the following Lagrangian: $L_p = \|w\|^2 + c\sum_i \varepsilon_i - \sum_i \alpha_i \{y_i(x_i \bullet w + b) - 1 + \varepsilon_i\} - \sum_i \beta_i \varepsilon_i$

SVM for Noisy Data

$$L_p = \|w\|^2 + c\sum_i \varepsilon_i - \sum_i \alpha_i \{y_i(x_i \bullet w + b) - 1 + \varepsilon_i\} - \sum_i \beta_i \varepsilon_i$$

Take the derivatives of L_p with respect to w, b, and ε_i.

$$\frac{\partial L_p}{\partial w} = 2w - \sum_i \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \frac{1}{2}\sum_i \alpha_i y_i x_i$$

$$\frac{\partial L_p}{\partial b} = -\frac{1}{2}\sum_i \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_i \alpha_i y_i = 0$$

$$\frac{\partial L_p}{\partial \varepsilon_i} = c - \beta_i - \alpha_i = 0 \quad \Rightarrow \quad c = \beta_i + \alpha_i$$

Karush-Kuhn-Tucker Conditions

$$0 \le \alpha_i \le c \quad \forall i$$

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \bullet x_j)$$

Both ε_i and its multiplier β_i are not involved in the function.

SVM Lagrangian Dual

Maximize $\sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l (\mathbf{x}_k \cdot \mathbf{x}_l)$

subject to constraints: $0 \le \alpha_k \le c \quad \forall k \qquad \sum_{k=1}^{R} \alpha_k y_k = 0$

Once solved, we obtain w and b using:

$$\mathbf{w} = \frac{1}{2} \sum_{k=1}^{R} \alpha_k y_k \mathbf{x}_k$$

$$y_i (x_i \bullet w + b) - 1 = 0$$

$$b = -y_i (y_i (x_i \bullet w) - 1)$$

Then classify with:

f(***x***,***w***,*b*) = *sign*(***w***. ***x*** + *b*)

SVM Lagrangian Dual

Maximize $\sum_{k=1}^{R} \alpha_k - \frac{1}{2}\sum_{k=1}^{R}\sum_{l=1}^{R} \alpha_k \alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l (\mathbf{x}_k \cdot \mathbf{x}_l)$

subject to constraints: $0 \le \alpha_k \le c \quad \forall k \qquad \sum_{k=1}^{R} \alpha_k y_k = 0$

Datapoints with $\alpha_k > 0$ will be the support vectors

Once solved, we obtain w and b using:

$$\frac{1}{2}\sum_{k=1}^{R} \alpha_k y_k \mathbf{x}_k$$

$$y_i\left(x_i \bullet w + b\right) - 1 = 0$$

$$b = -y_i\left(y_i\left(x_i \bullet w\right) - 1\right)$$

..so this sum only needs to be over the support vectors.

Then classify with:

$\boldsymbol{f}(\mathbf{x},\mathbf{w},b) = \mathit{sign}(\mathbf{w}.\ \mathbf{x} + b)$

A bit more on the SVM: The Lagrange Multiplier Method

Optimization problem:

Minimize: $f(\vec{x})$

Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Consider the augmented function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^{n} \lambda_i g(\vec{x})$$

(Lagrange function)

(Lagrange variables, or dual variables)

Observation:

For ***any*** feasible *x* and ***all*** $\lambda_i \geq 0$, we have $L(\vec{x}, \vec{\lambda}) \leq f(\vec{x})$

$$\implies \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda}) \leq f(\vec{x})$$

So, the optimal value to the constrained optimization:

$$p^* := \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$$

The problem becomes unconstrained in x!

Convex Optimization

Observations:

- object function is convex
- the constraints are affine, inducing a polytope constraint set.

So, SVM is a convex optimization problem (in fact a **quadratic program**)

Moreover, **strong duality holds**.

Let's examine the dual… the Lagrangian is:

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\|\vec{w}\|^2 + \sum_{i=1}^{n} \alpha_i \big(1 - y_i(\vec{w} \cdot \vec{x}_i - b)\big)$$

SVM standard (primal) form:

Minimize: $\frac{1}{2}\|\vec{w}\|^2$
(w,b)

Such that: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$
(for all i)

Back to SVM

SVM standard (primal) form:

Minimize: $\frac{1}{2}\|\vec{w}\|^2$
(w,b)

Such that: $y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1$
(for all i)

Maximize γ = 2/||w||

SVM standard (dual) form:

Maximize: $\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j)$
(α_i)

Such that: $\sum_{i=1}^{n}\alpha_i y_i = 0 \qquad \alpha_i \geq 0$
(for all i)

Both yield the same solution

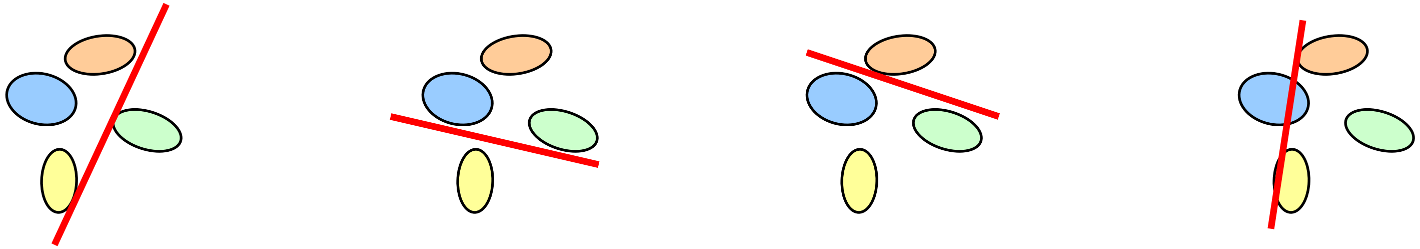
Only a function of "support vectors"

Multi-class Classification with SVMs

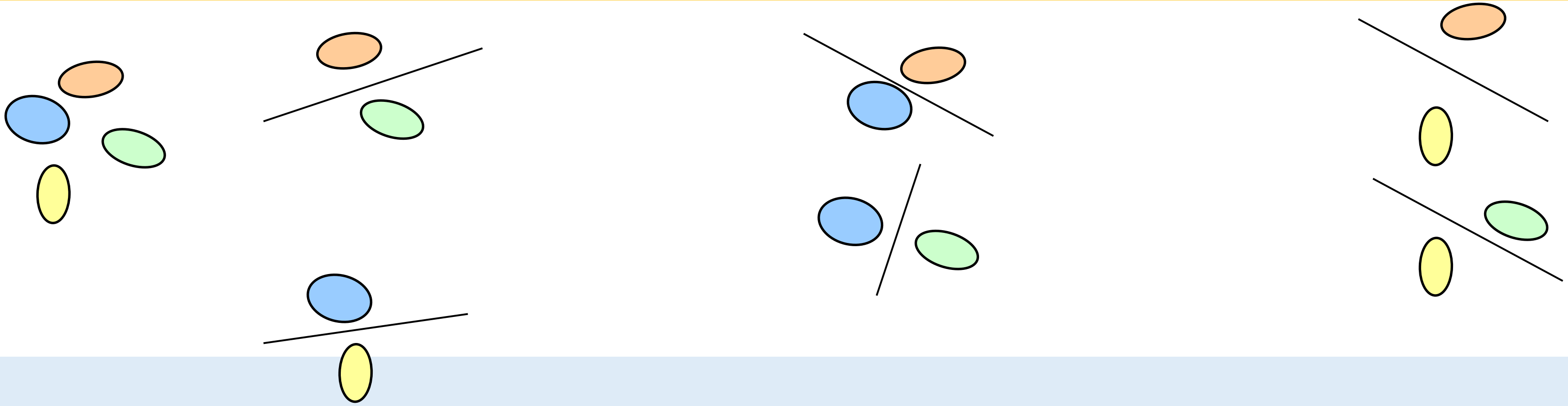
- SVMs can only handle two-class outputs.
- What can be done?
- Answer: with output arity N, learn N SVM’s
  - SVM 1 learns “Output==1” vs “Output != 1”
  - SVM 2 learns “Output==2” vs “Output != 2”
  - :
  - SVM N learns “Output==N” vs “Output != N”
- Then to predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

Multi-class Classification using SVM

One- versus-all



One- versus-one



Soft-margin SVM objective:

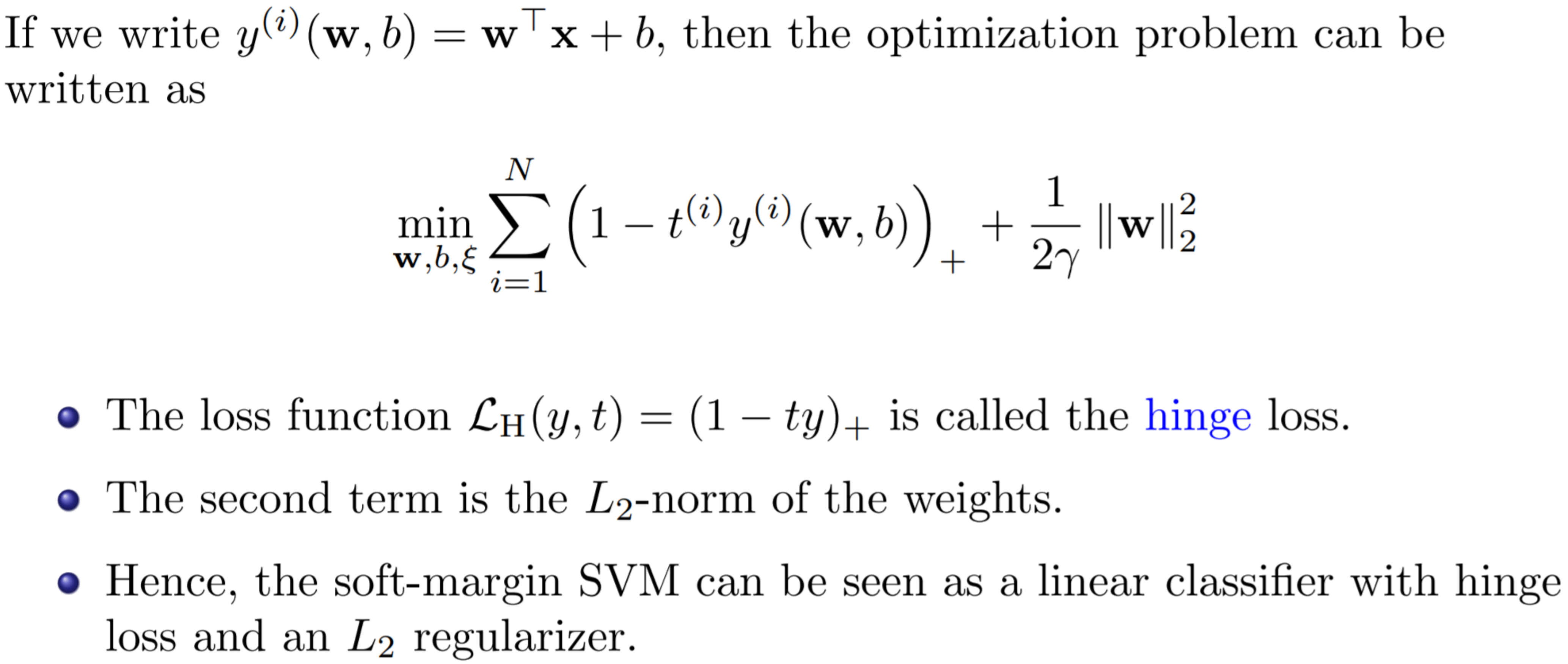
$$\begin{aligned} \min_{\mathbf{w},b,\xi} \frac{1}{2} \|\mathbf{w}\|_2^2 + \gamma \sum_{i=1}^{N} \xi_i & \\ \text{s.t.} \quad t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i & \qquad i = 1, \dots, N \\ \xi_i \geq 0 & \qquad i = 1, \dots, N \end{aligned}$$

$$\xi_i = \max\{0, 1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)\}.$$

$$\sum_{i=1}^{N} \xi_i = \sum_{i=1}^{N} \max\{0, 1 - t^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)\}.$$

vrite $\max\{0, y\} = (y)_+$

Soft Margin SVMs and Hinge Loss



If we write $y^{(i)}(\mathbf{w}, b) = \mathbf{w}^\top \mathbf{x} + b$, then the optimization problem can be written as

$$\min_{\mathbf{w},b,\xi} \sum_{i=1}^{N} \left(1 - t^{(i)} y^{(i)}(\mathbf{w}, b)\right)_+ + \frac{1}{2\gamma} \|\mathbf{w}\|_2^2$$

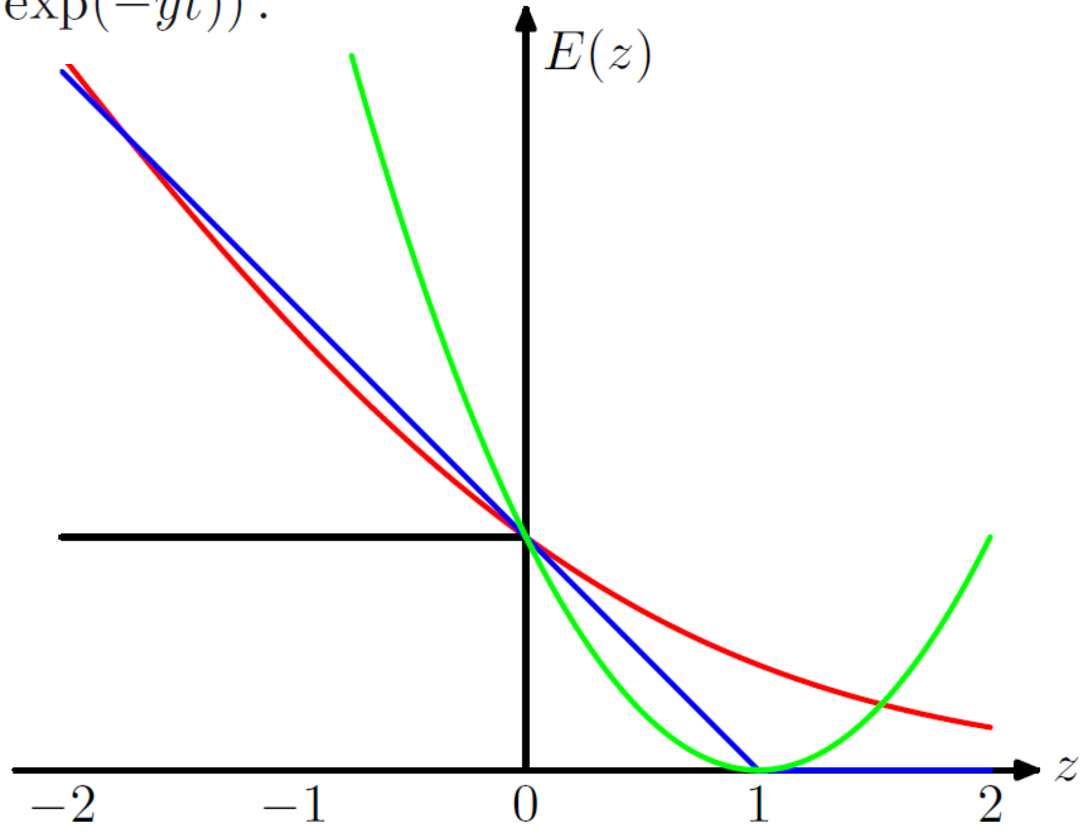
- The loss function $\mathcal{L}_{\text{H}}(y, t) = (1 - ty)_+$ is called the hinge loss.
- The second term is the L_2-norm of the weights.
- Hence, the soft-margin SVM can be seen as a linear classifier with hinge loss and an L_2 regularizer.

Hinge Loss vs other losses

- Blue : hinge loss
- Red : logistic loss
- Green : squared error
- Black : 0/1 loss

$$E_{\mathrm{SV}}(y_n t_n) = [1 - y_n t_n]_+$$

$$E_{\mathrm{LR}}(yt) = \ln\left(1 + \exp(-yt)\right).$$



Readings

- PRML, Bishop, Chapter 7 (7.1-7.3)
- "Introduction to Machine Learning" by Ethem Alpaydin, 2nd edition, Chapters 3 (3.1-3.4), Chapter 13 (13.1-13.9)
- Do read these!
  - https://www.svm-tutorial.com/2017/02/svms-overview-support-vector-machines/
  - https://www.svm-tutorial.com/2016/09/duality-lagrange-multipliers/
  - https://www.svm-tutorial.com/2017/10/support-vector-machines-succinctly-released/