Linear classifiers
Lecture 3

David Sontag
New York University

Slides adapted from Luke Zettlemoyer, Vibhav Gogate, and Carlos Guestrin

ML Methodology

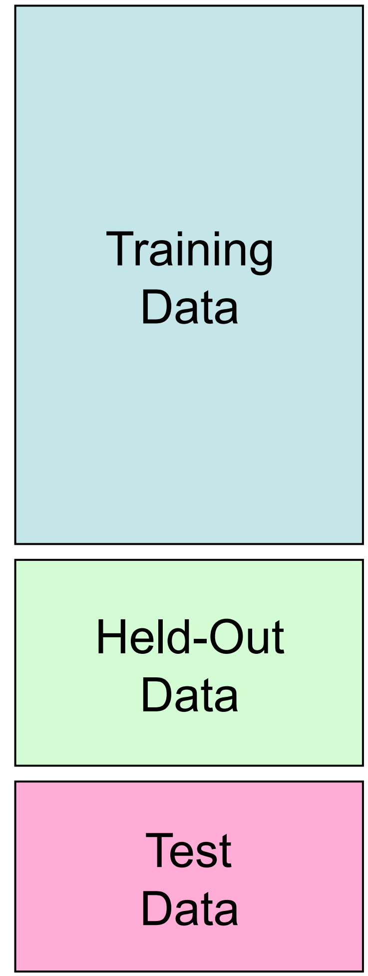
- Data: labeled instances, e.g. emails marked spam/ham
  - Training set
  - Held out set (sometimes call Validation set)
  - Test set

  Randomly allocate to these three, e.g. 60/20/20

- Features: attribute-value pairs which characterize each x

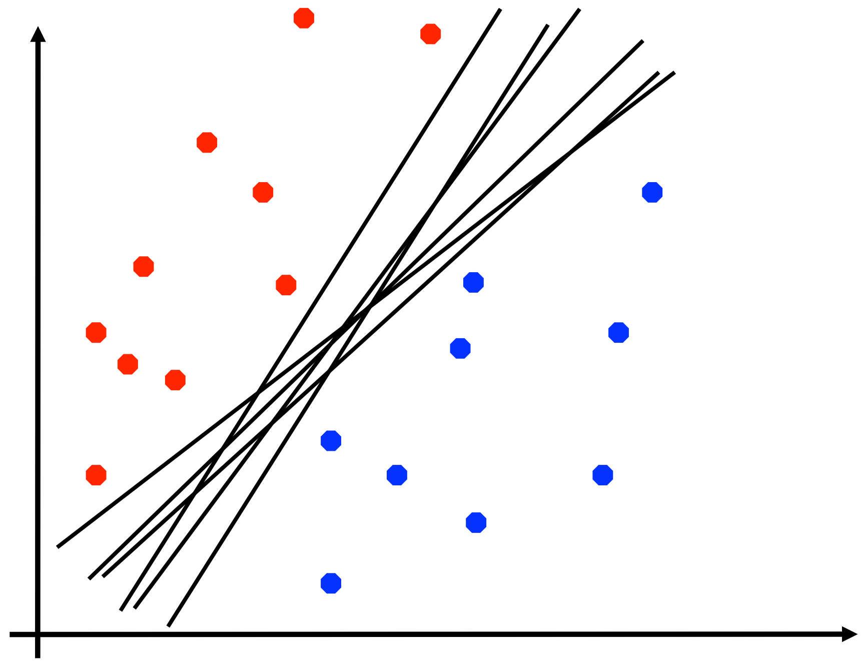
- Experimentation cycle
  - Select a hypothesis f
    (Tune hyperparameters on held-out or *validation* set)
  - Compute accuracy of test set
  - Very important: never "peek" at the test set!

- Evaluation
  - Accuracy: fraction of instances predicted correctly



Linear Separators

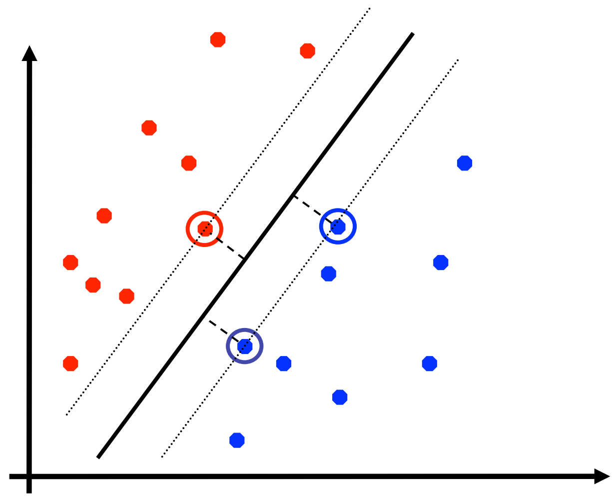
- Which of these linear separators is optimal?



Support Vector Machine (SVM)

- SVMs (Vapnik, 1990's) choose the linear separator with the **largest margin**

Robust to outliers!



V. Vapnik

- Good according to intuition, theory, practice

- SVM became famous when, using images as input, it gave accuracy comparable to neural-network with hand-designed features in a handwriting recognition task

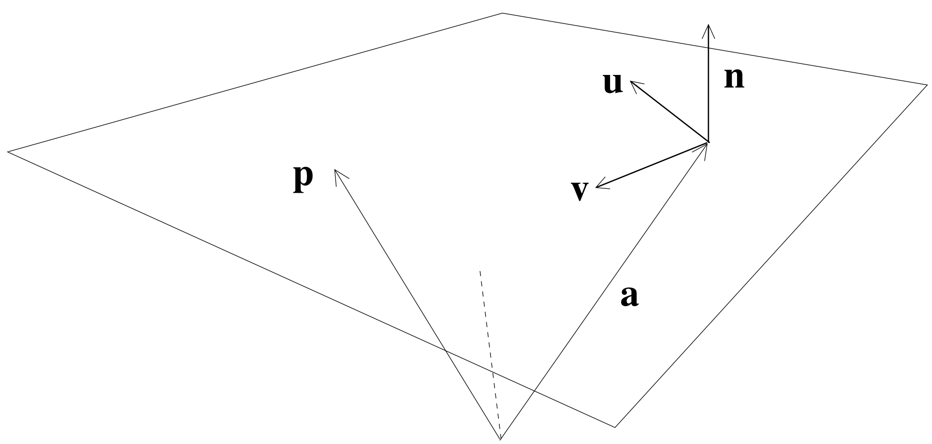
Planes and Hyperplanes

A plane can be specified as the set of all points given by:

$$\mathbf{p} = \mathbf{a} + s\mathbf{u} + t\mathbf{v}, \qquad (s,t) \in \mathcal{R}.$$

Vector from origin to a point in the plane

Two non-parallel directions in the plane



Alternatively, it can be specified as:

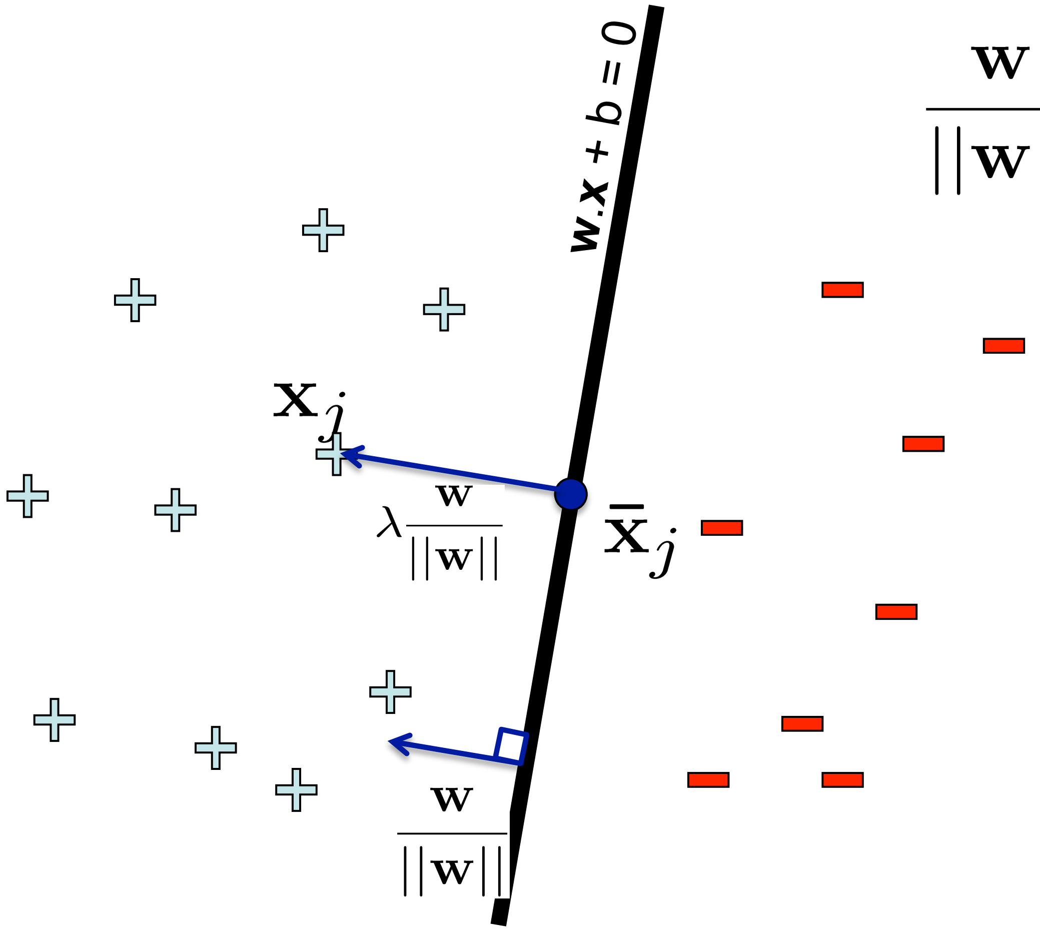
$$(\mathbf{p} - \mathbf{a}) \cdot \mathbf{n} = 0 \Leftrightarrow \mathbf{p} \cdot \mathbf{n} = \mathbf{a} \cdot \mathbf{n}$$

Normal vector
(we will call this w)

Only need to specify this dot product,
a scalar (we will call this the offset)

Barber, Section 29.1.1-4

Normal to a plane



w: normal vector for the plane

$\frac{\mathbf{w}}{||\mathbf{w}||}$ -- *unit* vector parallel to w

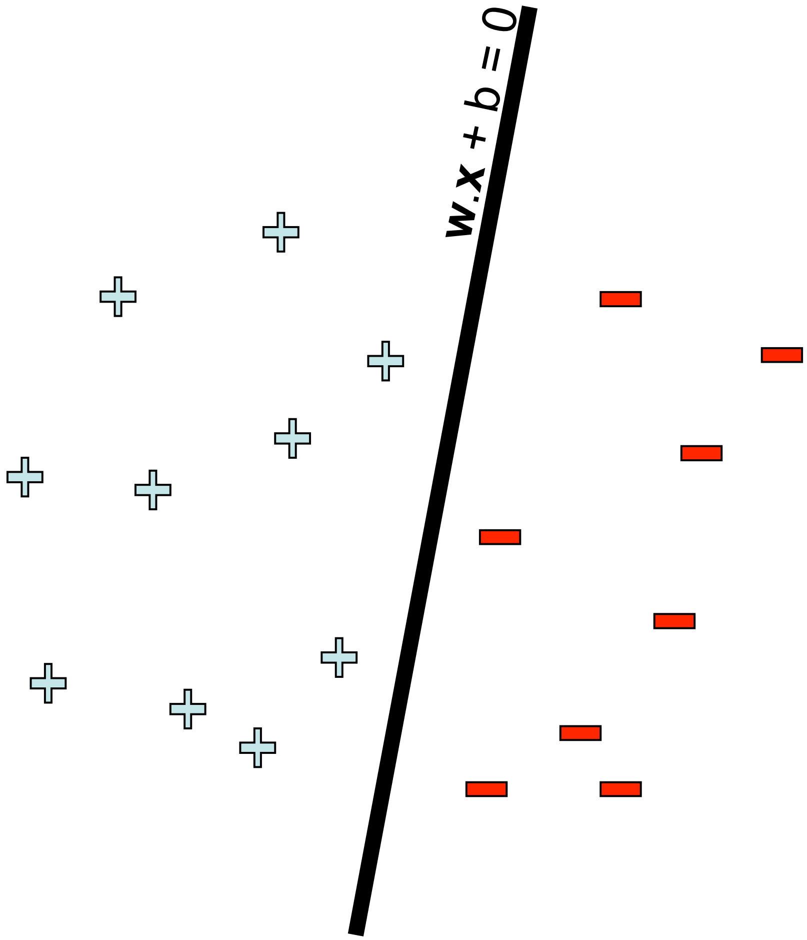
$\bar{\mathbf{x}}_j$ -- projection of x_j onto the plane

$$x_j - \bar{x}_j = \lambda \frac{w}{||w||}$$

λ is the length of the vector, i.e.

$$||x_j - \bar{x}_j|| = \frac{\lambda}{||w||}||w|| = \lambda$$

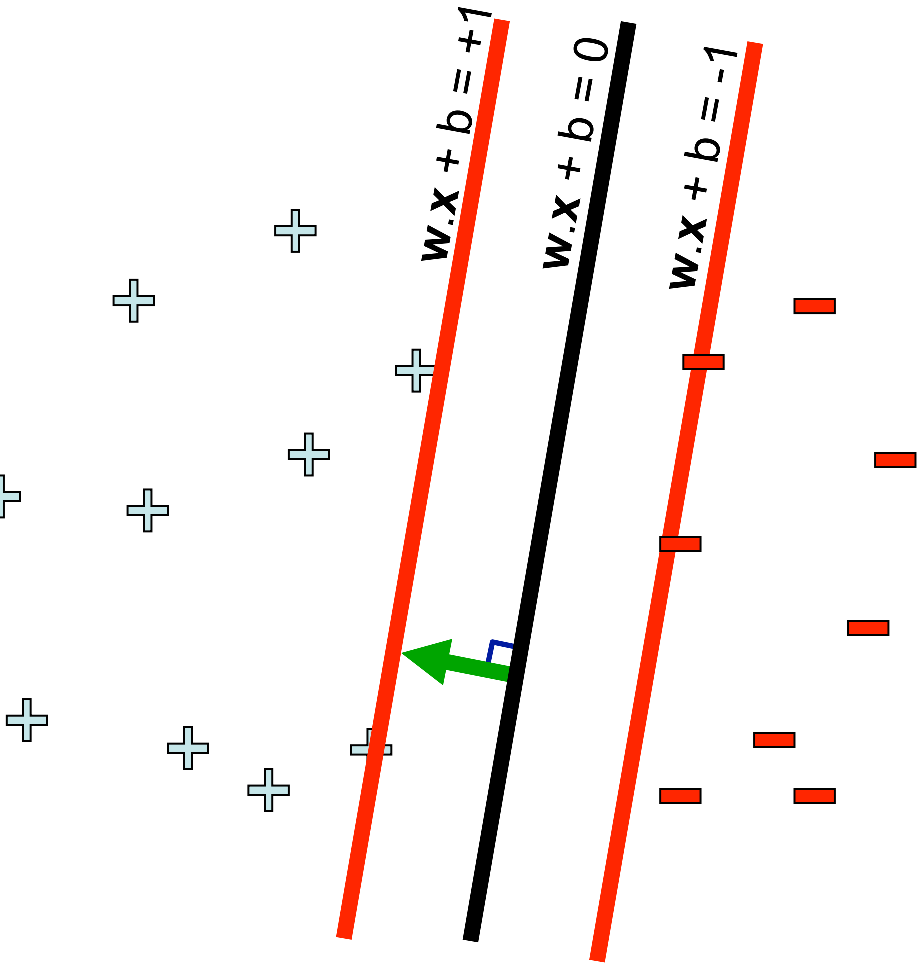
Scale invariance



Any other ways of writing the same dividing line?

- **w.x** + b = 0
- 2**w.x** + 2b = 0
- 1000**w.x** + 1000b = 0
- ….

Scale invariance



During learning, we set the scale by asking that, for all *t*,

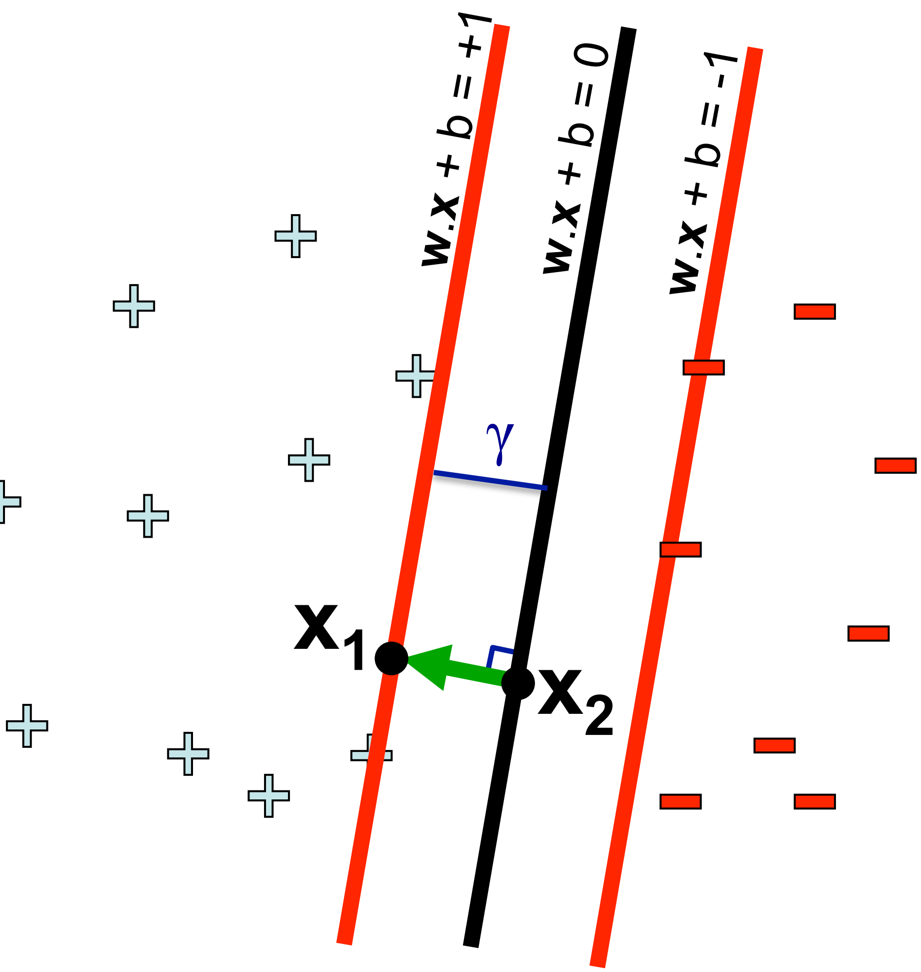
for y_t = +1, $w \cdot x_t + b \geq 1$

and for y_t = -1, $w \cdot x_t + b \leq -1$

That is, we want to satisfy all of the **linear** constraints

$$y_t \left(w \cdot x_t + b \right) \geq 1 \quad \forall t$$

What is γ as a function of **w**?



$$w \cdot x_1 + b = 1$$

$$-$$

$$w \cdot x_2 + b = 0$$

$$w \cdot (x_1 - x_2) = 1$$

Plug in

We also know that:

$$x_1 - x_2 = \gamma \frac{w}{||w||}$$

$$1 = w \cdot \left(\gamma \frac{w}{||w||}\right) = \frac{\gamma}{||w||} w \cdot w = \gamma ||w||$$

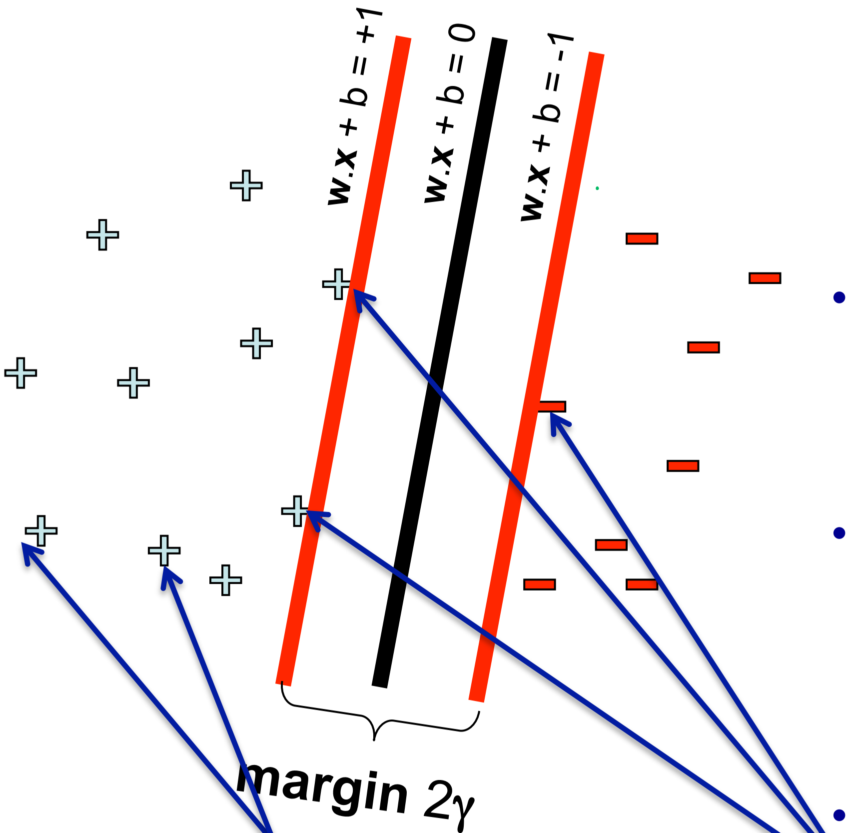
So, $\gamma = \dfrac{1}{||w||}$

Final result: can maximize margin by minimizing $||w||_2$!!!

Support vector machines (SVMs)

$$\text{minimize}_{\mathbf{w},b} \quad \mathbf{w}.\mathbf{w}$$
$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1, \ \ \forall j$$

- Example of a **convex optimization** problem
  - A quadratic program
  - Polynomial-time algorithms to solve!
- Hyperplane defined by **support vectors**
  - Could use them as a lower-dimension basis to write down line, although we haven't seen how yet
- More on these later



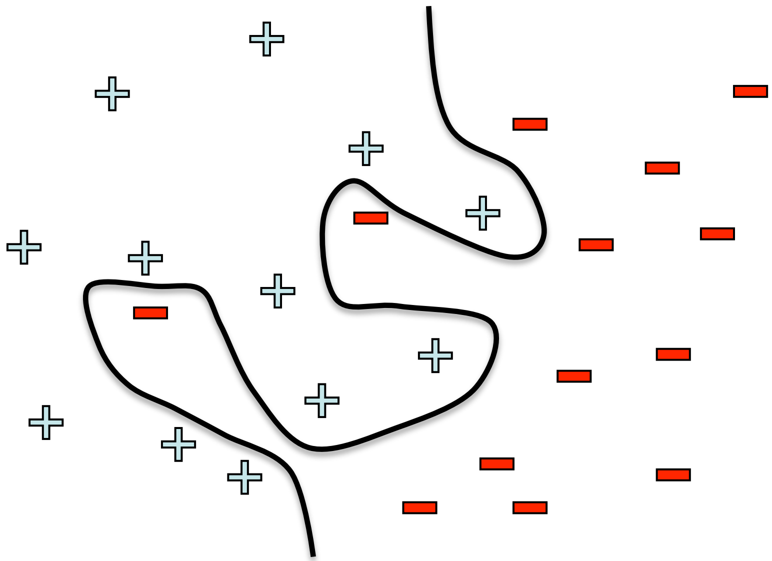
Non-support Vectors:
- everything else
- moving them will not change **w**

Support Vectors:
- data points on the canonical lines

What if the data is not linearly separable?

$$\left\langle x_i^{(1)}, \ldots, x_i^{(m)} \right\rangle \quad \text{— } m \text{ features}$$

$$y_i \in \{-1, +1\} \quad \text{— class}$$



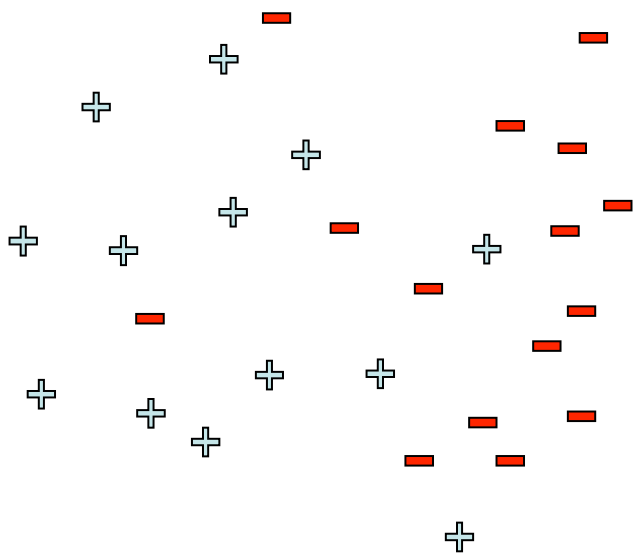
Add More Features!!!

$$\phi(x) = \begin{pmatrix} x^{(1)} \\ \ldots \\ x^{(n)} \\ x^{(1)}x^{(2)} \\ x^{(1)}x^{(3)} \\ \ldots \\ e^{x^{(1)}} \\ \ldots \end{pmatrix}$$

What about overfitting?

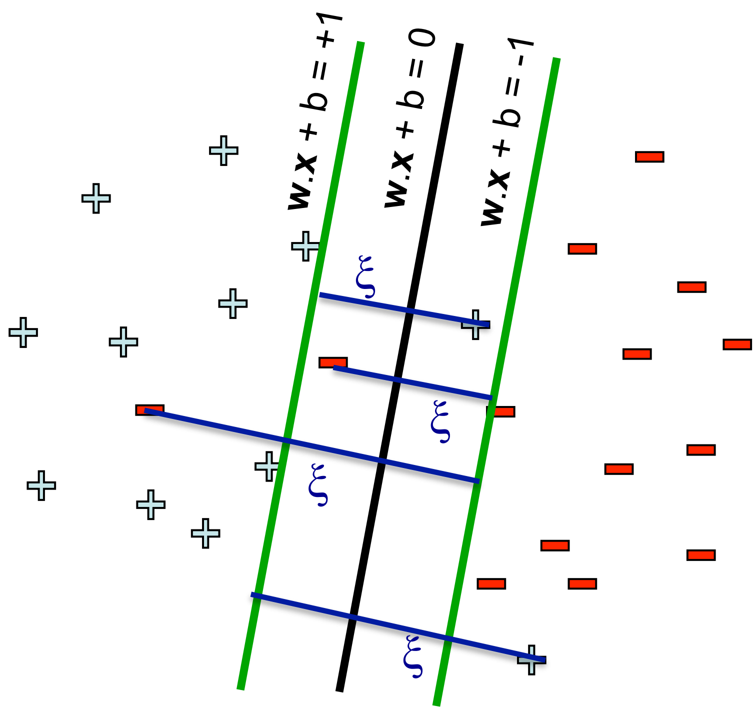
What if the data is not linearly separable?

$$\text{minimize}_{\mathbf{w},b} \quad \mathbf{w}.\mathbf{w} + \text{C \#(mistakes)}$$
$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 \qquad , \forall j$$



- First Idea: Jointly minimize **w.w** and number of training mistakes
  - How to tradeoff two criteria?
  - Pick C using validation data
- Tradeoff #(mistakes) and **w.w**
  - 0/1 loss
  - Not QP anymore
  - Also doesn't distinguish near misses and really bad mistakes
  - NP hard to find optimal solution!!!

Allowing for slack: "Soft margin SVM"



$$\text{minimize}_{\mathbf{w},b} \quad \mathbf{w}.\mathbf{w} + C \Sigma_j \xi_j$$

$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 - \xi_j \quad , \forall j \quad \xi_j \geq 0$$

"slack variables"

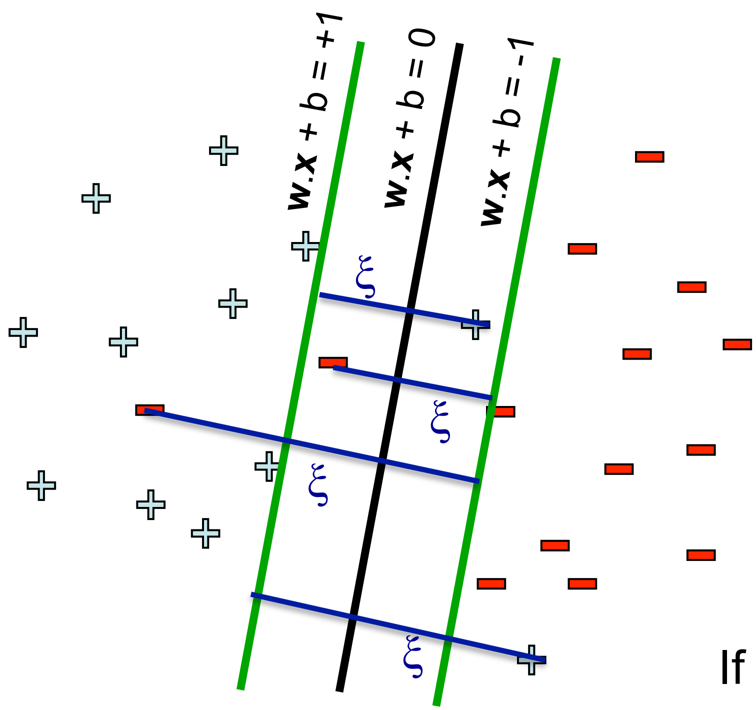
Slack penalty $C > 0$:

- $C=\infty$ → have to separate the data!
- $C=0$ → ignores the data entirely!
- Select using validation data

For each data point:

- If margin ≥ 1, don't care
- If margin < 1, pay linear penalty

Allowing for slack: "Soft margin SVM"



$$\text{minimize}_{\mathbf{w},b} \quad \mathbf{w}.\mathbf{w} + C\,\Sigma_j\,\xi_j$$

$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 - \xi_j \quad ,\forall j \quad \xi_j \geq 0$$

"slack variables"

What is the (optimal) value of ξ_j as a function of **w** and b?

If $(w \cdot x_j + b)\, y_j \geq 1$, then $\xi_j = 0$

If $(w \cdot x_j + b)\, y_j < 1$, then $\xi_j = 1 - (w \cdot x_j + b)\, y_j$

$$\xi_j = \max\left(0, 1 - (w \cdot x_j + b)\, y_j\right)$$