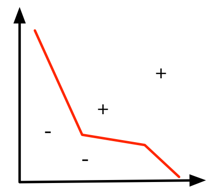
6.034
Support Vector Machines

Peter Szolovits

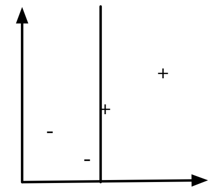
ai6034.mit.edu

October 21, 2019

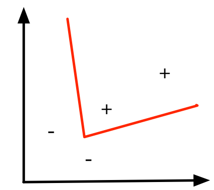
Many Possible Classifiers Fit Data



NN
(Voronoi Diagram)

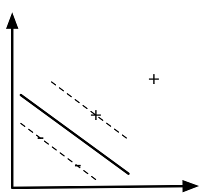


Classification
(ID) Tree



Neural
Network

Maximum Margin Classifier



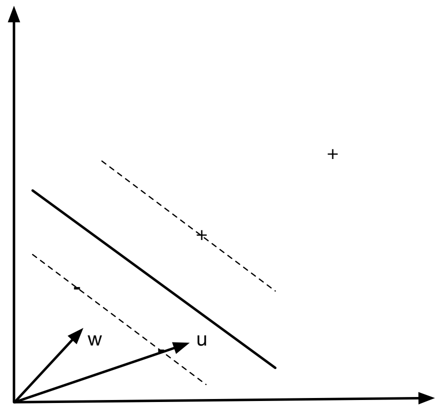
Vapnik's Idea

- Find the separator between classes that maximizes the *margin*; i.e., is farthest from the nearest points on opposite sides of the separator

Boser, B. E., Guyon, I., & Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers. Colt, 144-152. http://doi.org/

Decision Rule

- $\bar{w} \cdot \bar{u} \geq c$
  - or
  - $(c = -b)$
- If $\bar{w} \cdot \bar{u} + b \geq 0$ then +
- If $\bar{w} \cdot \bar{u} + b < 0$ then -
- But what are w, b?



SVM Constraints

- + points (definitely on + side of margin)
  - $\bar{w} \cdot \bar{x}_+ + b \geq 1$ ($\neq$ if not on the boundary)
- - points (definitely on - side of margin)
  - $\bar{w} \cdot \bar{x}_- + b \leq -1$
- Introduce new "outcome" variable
  - $y_i = +1 \text{ if } +, -1 \text{ if } -$
- Then, we can simplify both by multiplying constraints by y_i
  - $y_i(\bar{w} \cdot \bar{x}_i + b) \geq 1$
  - or $y_i(\bar{w} \cdot \bar{x}_i + b) - 1 \geq 0$
  - and $y_i(\bar{w} \cdot \bar{x}_i + b) - 1 = 0$ for $\bar{x}_i$ in the "street"

How wide is the "street"?

$\text{width} = (\bar{x}_+ - \bar{x}_-) \cdot \frac{\bar{w}}{\|\bar{w}\|}$

$y_i(\bar{w} \cdot \bar{x}_i + b) - 1 \geq 0$

so for $\bar{x}_+$, $y_i = +1$, so

$\bar{w} \cdot \bar{x}_+ + b - 1 \geq 0$ or $\bar{w} \cdot \bar{x}_+ = 1 - b$

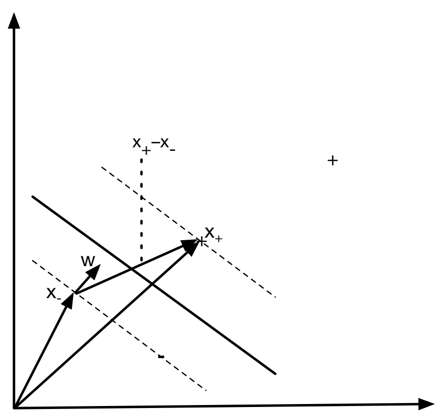
and for $\bar{x}_-$, $y_i = -1$, so

$-\bar{w} \cdot \bar{x}_i - b - 1 \geq 0$ or $-\bar{w} \cdot \bar{x}_- = 1 + b$

so $\bar{w} \cdot \bar{x}_+ = 1 - b$ and $-\bar{w} \cdot \bar{x}_- = 1 + b$ by the constraint that points in the street are at 0

$(1 - b) + (1 + b) = 2$

Thus, $\text{width} = \frac{2}{\|\bar{w}\|}$



How wide is the "street"?

$\text{width} = (\bar{x}_+ - \bar{x}_-) \cdot \frac{\bar{w}}{\|\bar{w}\|} = \frac{\bar{w} \cdot \bar{x}_+ - \bar{w} \cdot x_-}{\|\bar{w}\|}$

$y_i(\bar{w} \cdot \bar{x}_i + b) - 1 \geq 0$

so for $\bar{x}_+$, $y_i = +1$, so

$\bar{w} \cdot \bar{x}_+ + b - 1 \geq 0$ or $\bar{w} \cdot \bar{x}_+ = 1 - b$

and for $\bar{x}_-$, $y_i = -1$, so

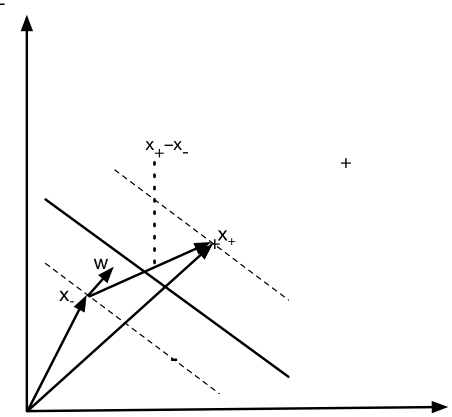
$-\bar{w} \cdot \bar{x}_i - b - 1 \geq 0$

or $-\bar{w} \cdot \bar{x}_- = 1 + b$ or $\bar{w} \cdot \bar{x}_+ = -1 - b$

by the constraint that points in the "gutter" are at 0

$(1 - b) - (-1 - b) = 2$

Thus, $\text{width} = \frac{2}{\|\bar{w}\|}$



Maximize width of street

- $\text{width} = \dfrac{2}{\|\bar{w}\|}$,
  - thus want to maximize $\dfrac{1}{\|\bar{w}\|}$,
  - or minimize $\|\bar{w}\|$,
  - or minimize $\dfrac{1}{2}\|\bar{w}\|^2$

LaGrange Multipliers

- Useful to find extremum of a function under constraints

$$L = \frac{1}{2}\|\bar{w}\|^2 - \sum_i \lambda_i \left[y_i(\bar{w} \cdot \bar{x}_i + b) - 1\right]$$

$$\frac{\partial L}{\partial \bar{w}} = \bar{w} - \sum_i \lambda_i y_i \bar{x}_i = 0, \text{ thus } \boxed{\bar{w} = \sum_i \lambda_i y_i \bar{x}_i}$$

$$\frac{\partial L}{\partial b} = \boxed{-\sum_i \lambda_i y_i = 0}$$

Plugging in What We Have Learned

$$L = \frac{1}{2}\left(\sum_i \lambda_i y_i \bar{x}_i\right) \cdot \left(\sum_j \lambda_j y_j \bar{x}_j\right) - \left(\sum_i \lambda_i y_i \bar{x}_i\right) \cdot \left(\sum_j \lambda_j y_j \bar{x}_j\right) - \sum_i \lambda_i y_i b + \sum_i \lambda_i$$

$$= \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \bar{x}_i \cdot \bar{x}_j$$

- Now, “all we need do” is to find the minimum of L wrt λ_i
- Call a numerical analyst! — quadratic optimization problem
  - Convex, thus no local extrema
- ***Optimum depends only on dot products between pairs of vectors***
- Decision rule becomes:

If $\sum_i \lambda_i y_i \bar{x}_i \cdot \bar{u} + b \geq 0$ then +, else -

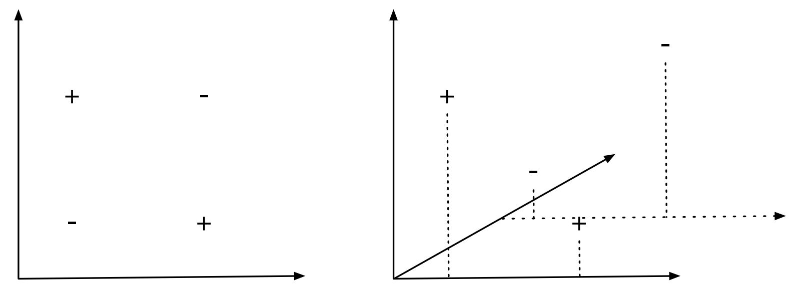
Vapnick’s Next Nice Idea

- Not all data are linearly separable!
- *Kernel* trick handles non-linearity
- Transform data into a higher-dimensional space via $\phi(\bar{x}_i)$
  - instead of dot products $\bar{x}_i \cdot \bar{x}_j$ and $\bar{x}_i \cdot \bar{u}$, define $K(\bar{x}_i, \bar{x}_j) = \phi(\bar{x}_i) \cdot \phi(\bar{x}_j)$
  - We don’t need explicit definition of ϕ!

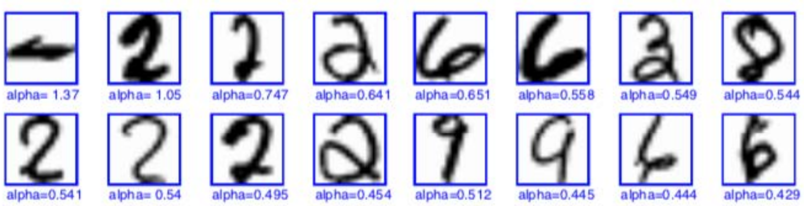
Common Kernels

- **Linear:** $\bar{x}_i \cdot \bar{x}_j$
- **Polynomial:** $(\gamma\, \bar{x}_i \cdot \bar{x}_j + c)^n$
- **Radial Basis:** $e^{-\frac{\|\bar{x}_i - \bar{x}_j\|^2}{\sigma}}$
- **Sigmoid:** $\tanh(\gamma\, \bar{x}_i \cdot \bar{x}_j + c)$
- ...

Kernels Can Raise Dimensionality of the Data



Early Competition with ANN



	DB1		DB2		
q	error	$<m>$	error	$<m>$	N
1 (linear)	3.2 %	36	10.5 %	97	256
2	1.5 %	44	5.8 %	89	$3 \cdot 10^4$
3	1.7 %	50	5.2 %	79	$8 \cdot 10^7$
4			4.9 %	72	$4 \cdot 10^9$
5			5.2 %	69	$1 \cdot 10^{12}$

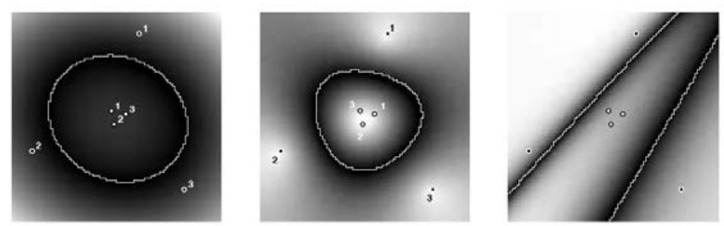


Figure 4: Decision boundaries for maximum margin classifiers with second order polynomial decision rule $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (left) and an exponential RBF $K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|/2)$ (middle). The rightmost picture shows the decision boundary of a two layer neural network with two hidden units trained with backpropagation.

Vapnick's Ideas

- Find the separator between classes that maximizes the *margin*; i.e., is farthest from the nearest points on opposite sides of the separator
- *Kernel* trick introduces non-linearity
- *Soft margins* allow fitting data that are not fully consistent
- Regression estimates *how much* does an item fit a category