Machine Learning

Prof. Barbara Caputo
Dip. Ingegneria Informatica, Automatica e Gestionale, Roma

SAPIENZA
UNIVERSITÀ DI ROMA

Parzen Windows

Density Estimation

- Observe some data x_i
- Want to estimate $p(x)$

Density Estimation

- Observe some data x_i
- Want to estimate p(x)
  - Find unusual observations (e.g. security)
  - Find typical observations (e.g. prototypes)

Density Estimation

- Observe some data x_i
- Want to estimate p(x)
  - Find unusual observations (e.g. security)
  - Find typical observations (e.g. prototypes)
  - Classifier via Bayes Rule

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{\sum_{y'} p(x|y')p(y')}$$

- Need tool for computing p(x) easily

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

25	English	Chinese	German	French	Spanish
male	5	2	3	1	0
female	6	3	2	2	1

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

25	English	Chinese	German	French	Spanish
male	0.2	0.08	0.12	0.04	0
female	0.24	0.12	0.08	0.08	0.04

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

25	English	Chinese	German	French	Spanish
male	0.2	0.08	0.12	0.04	0
female	0.24	0.12	0.08	0.08	0.04

Bin Counting

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
- Bin counting (record # of occurrences)

not enough data

25	English	Chinese	German	French	Spanish
male	0.2	0.08	0.12	0.04	0
female	0.24	0.12	0.08	0.08	0.04

Curse of dimensionality (lite)

- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
  - ZIP code
  - Day of the week
  - Operating system
  - ...

#bins grows exponentially

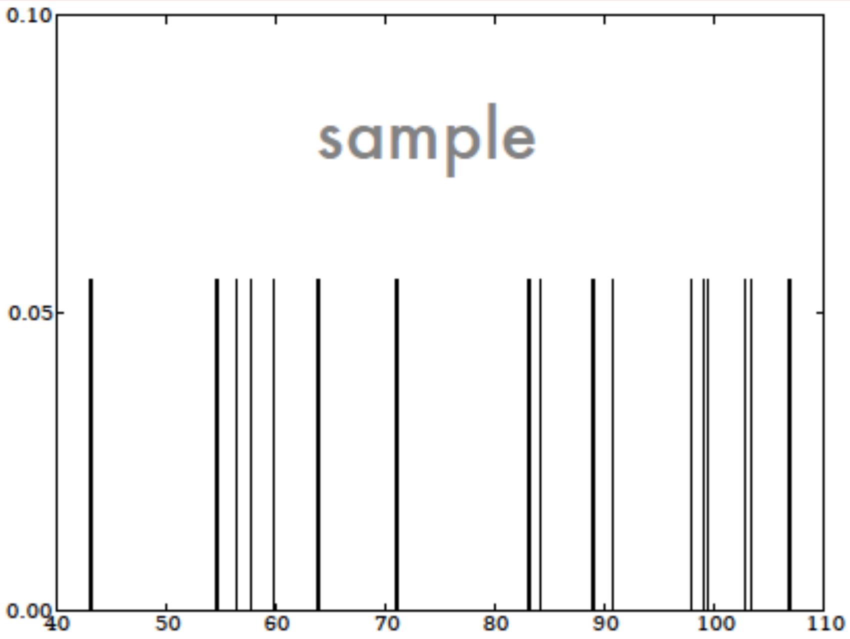
Curse of dimensionality (lite)

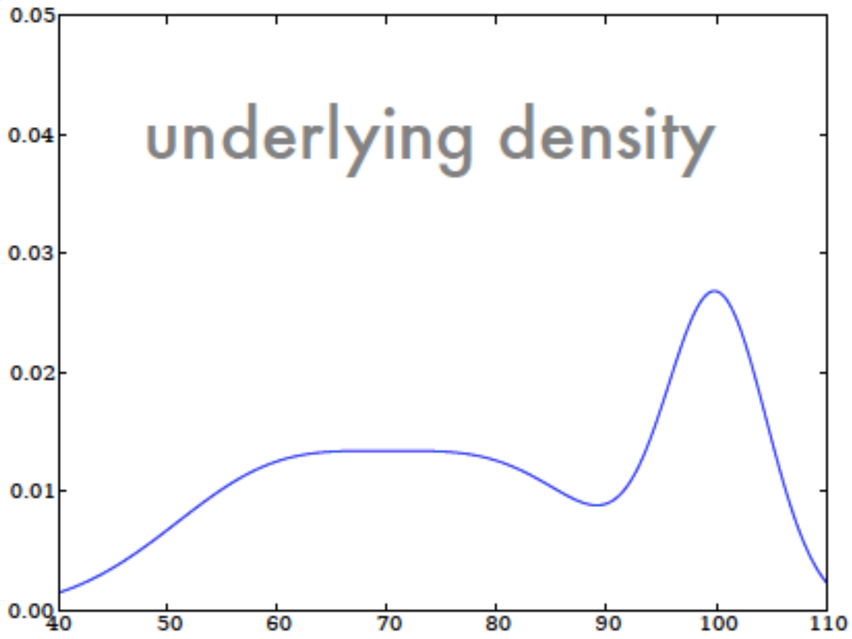
- Discrete random variables, e.g.
  - English, Chinese, German, French, ...
  - Male, Female
  - ZIP code
  - Day of the week
  - Operating system
  - ...
- Continuous random variables
  - Income
  - Bandwidth
  - Time

#bins grows exponentially

need many bins per dimension

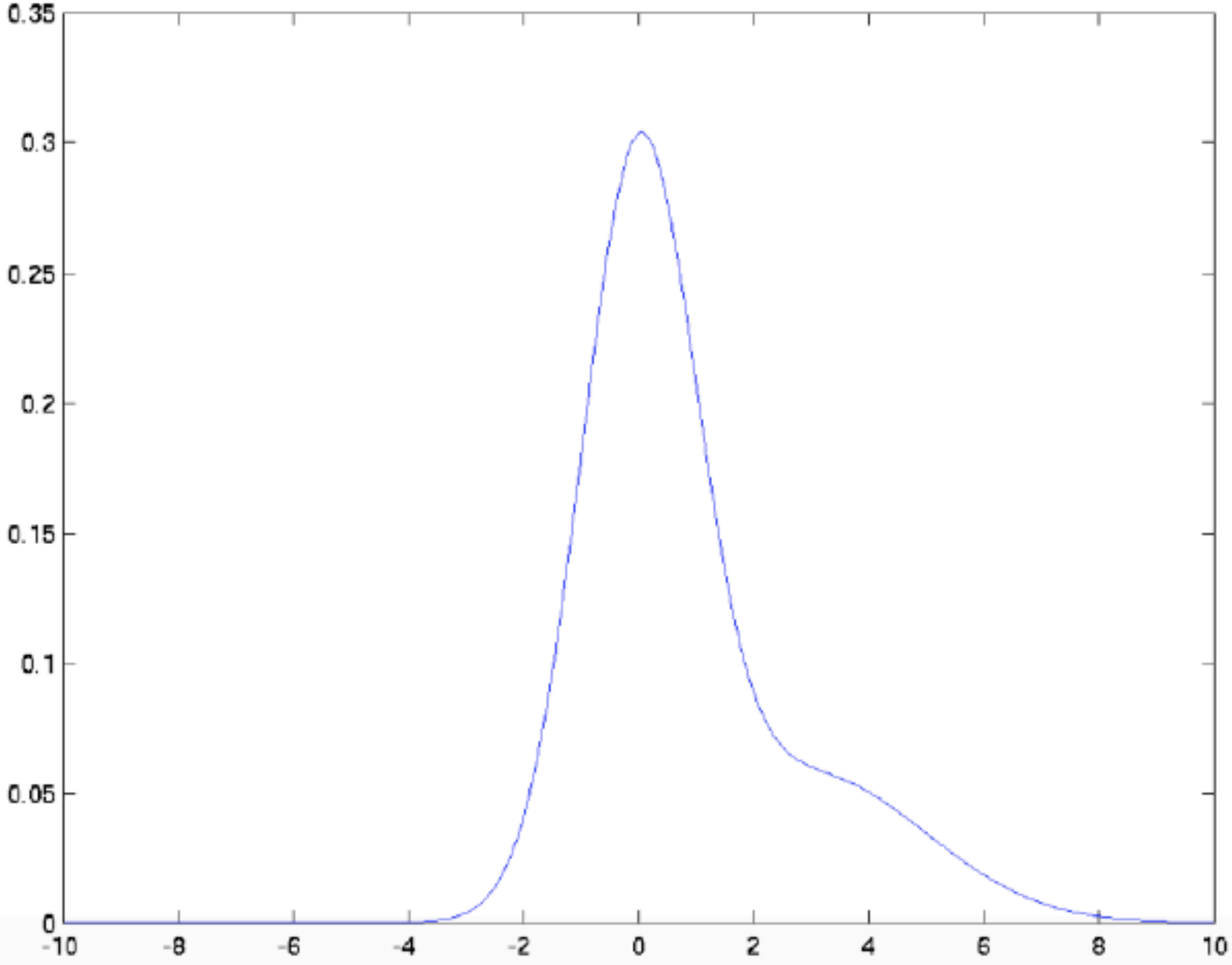
Density Estimation



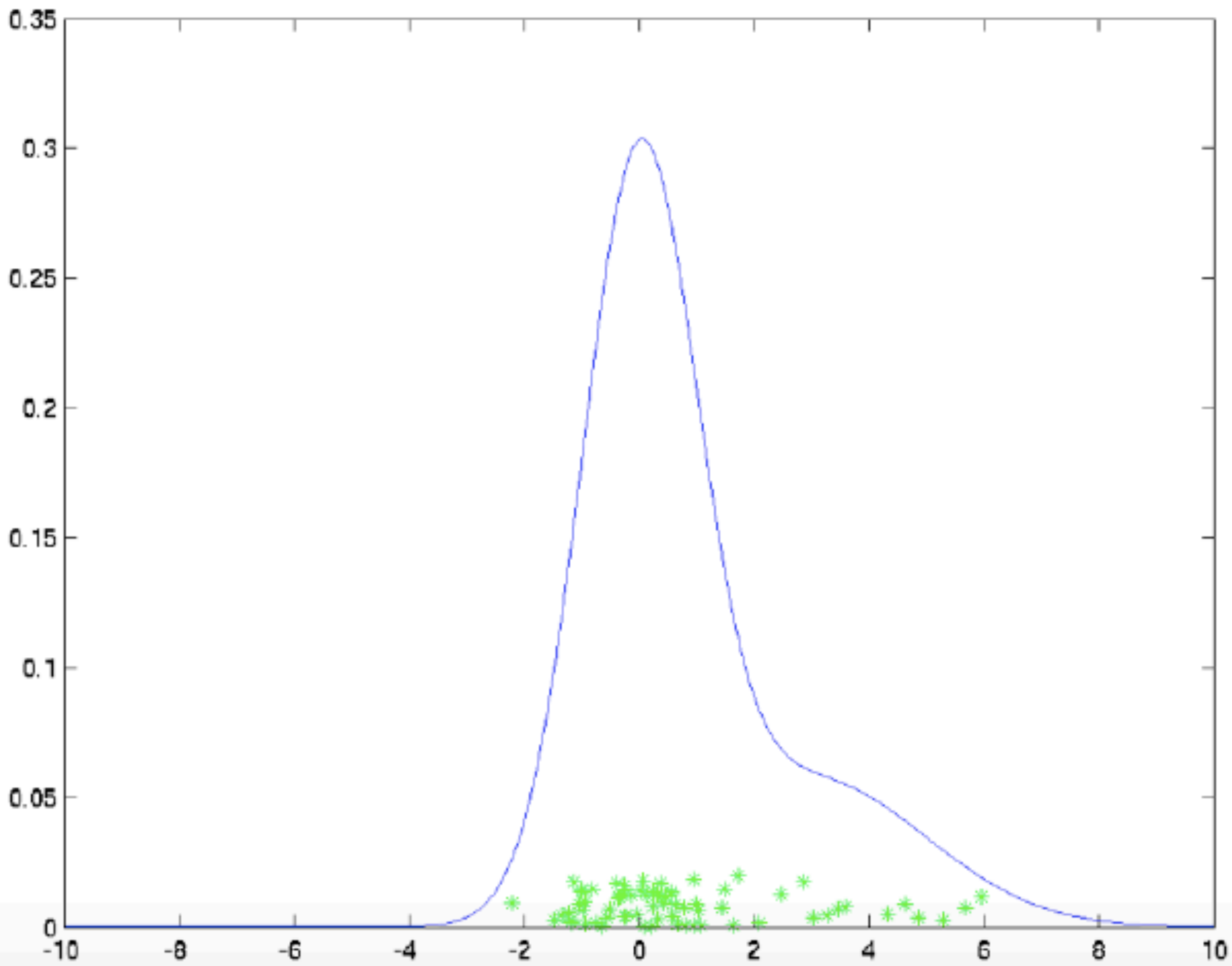


- Continuous domain = infinite number of bins
- Curse of dimensionality
  - 10 bins on [0, 1] is probably good
  - 10^{10} bins on $[0, 1]^{10}$ requires high accuracy in estimate: probability mass per cell also decreases by 10^{10}

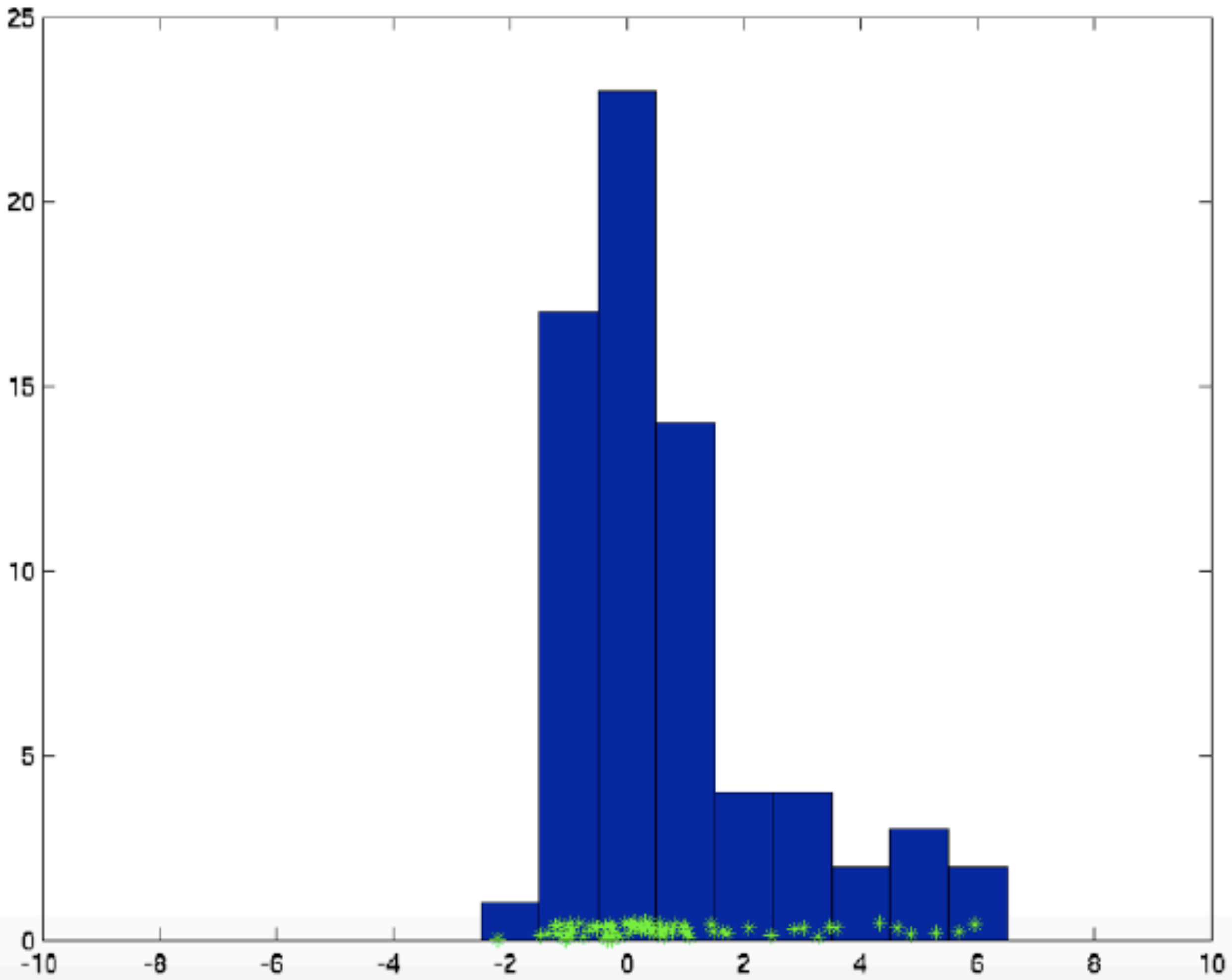
Bin Counting



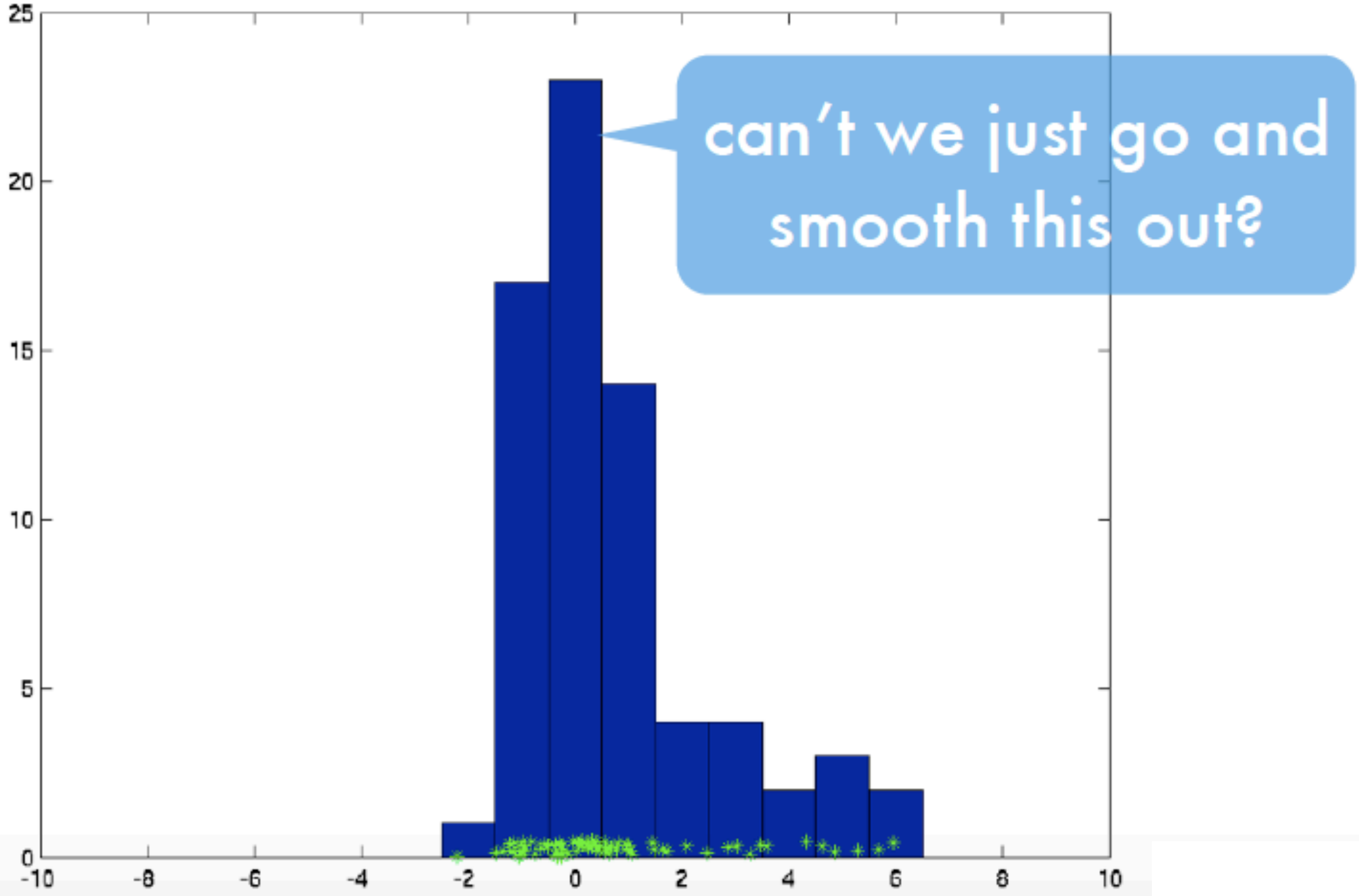
Bin Counting



Bin Counting



Bin Counting



Parzen Windows

- Naive approach
  Use empirical density (delta distributions)

$$p_{\mathrm{emp}}(x) = \frac{1}{m} \sum_{i=1}^{m} \delta_{x_i}(x)$$

Parzen Windows

- Naive approach
  Use empirical density (delta distributions)

$$p_{\text{emp}}(x) = \frac{1}{m}\sum_{i=1}^{m} \delta_{x_i}(x)$$

- This breaks if we see slightly different instances
- Kernel density estimate

Parzen Windows

- Naive approach
  Use empirical density (delta distributions)

$$p_{\text{emp}}(x) = \frac{1}{m}\sum_{i=1}^{m}\delta_{x_i}(x)$$

- This breaks if we see slightly different instances
- Kernel density estimate
  Smear out empirical density with a nonnegative smoothing kernel $k_x(x')$ satisfying

$$\int_{\mathcal{X}} k_x(x')dx' = 1 \text{ for all } x$$

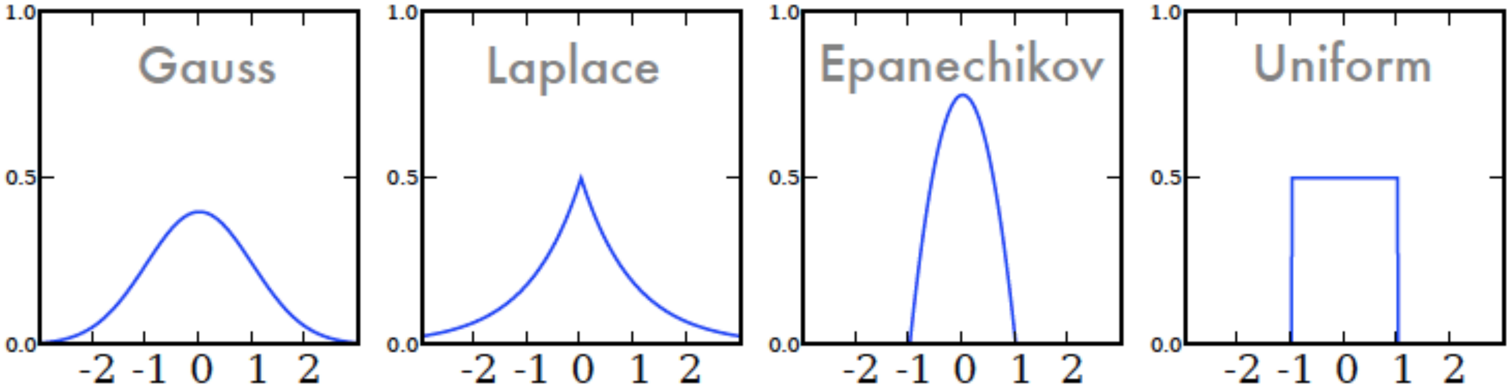
Parzen Windows

- **Density estimate**

$$p_{\mathrm{emp}}(x) = \frac{1}{m}\sum_{i=1}^{m} \delta_{x_i}(x)$$

$$\hat{p}(x) = \frac{1}{m}\sum_{i=1}^{m} k_{x_i}(x)$$

- **Smoothing kernels**



$(2\pi)^{-\frac{1}{2}} e^{-\frac{1}{2}x^2}$ $\quad$ $\frac{1}{2}e^{-|x|}$ $\quad$ $\frac{3}{4}\max(0, 1 - x^2)$ $\quad$ $\frac{1}{2}\chi_{[-1,1]}(x)$

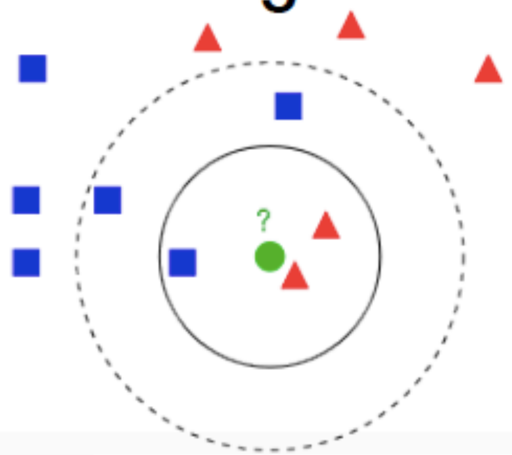
Nearest Neighbor

Nearest Neighbors

- Table lookup
  For previously seen instance remember label

Nearest Neighbors

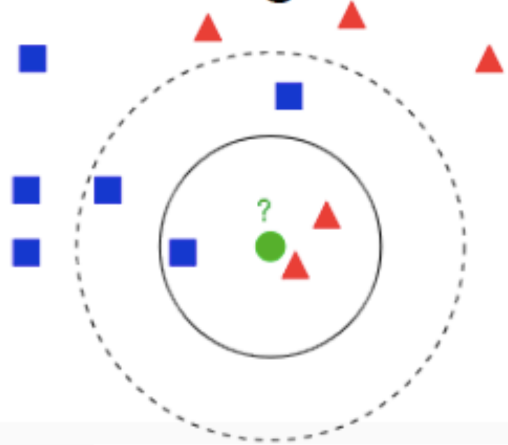
- Table lookup
  For previously seen instance remember label
- Nearest neighbor
  - Pick label of most similar neighbor
  - Slight improvement - use k-nearest neighbors



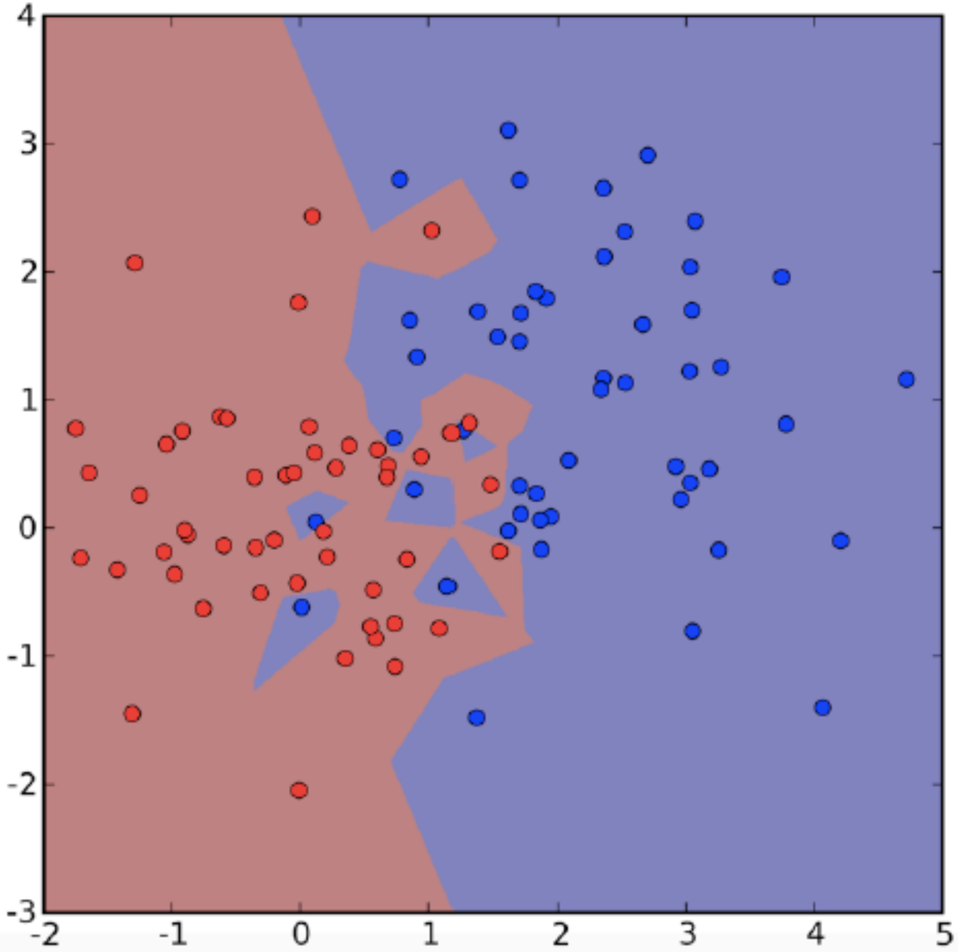
Nearest Neighbors

- Table lookup
  For previously seen instance remember label
- Nearest neighbor
  - Pick label of most similar neighbor
  - Slight improvement - use k-nearest neighbors

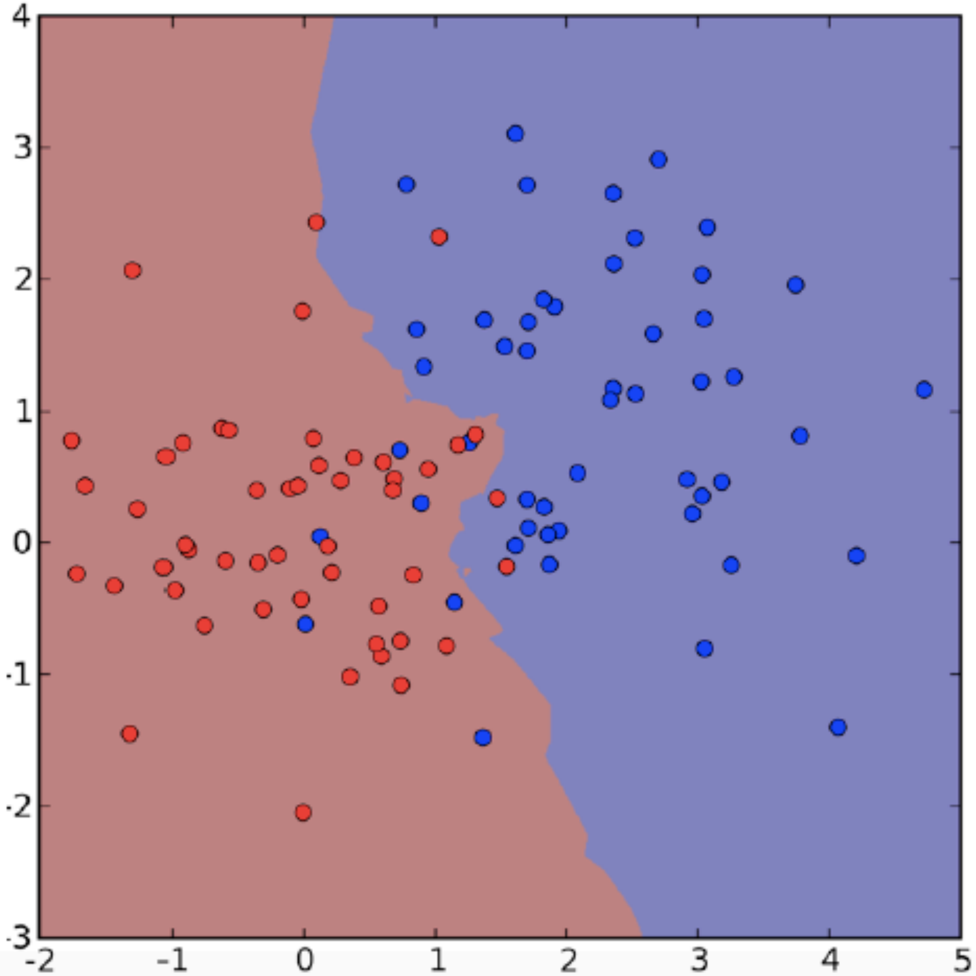
  - Really useful baseline!
  - Easy to implement for small amounts of data.



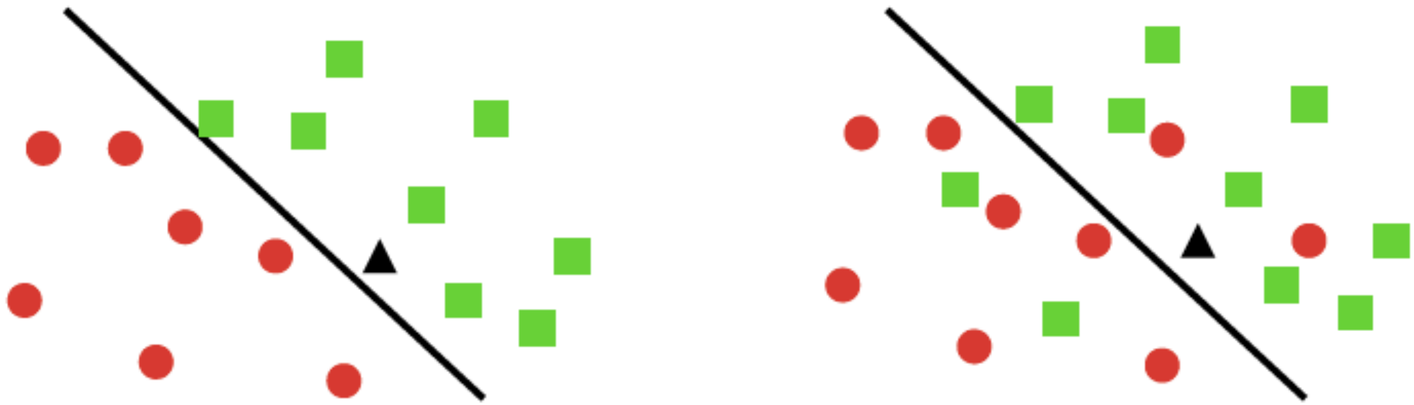
1-Nearest Neighbor



4-Nearest Neighbors Sign

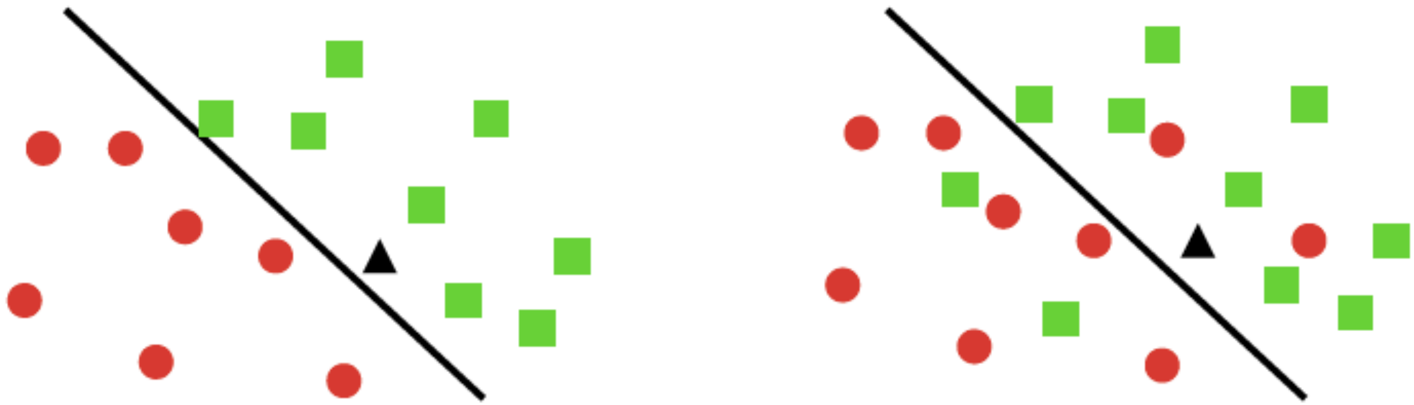


If we get more data



- 1 Nearest Neighbor
  - Converges to perfect solution if separation
  - Twice the minimal error rate 2p(1-p) for noisy problems

If we get more data



- 1 Nearest Neighbor
  - Converges to perfect solution if separation
  - Twice the minimal error rate 2p(1-p) for noisy problems
- k-Nearest Neighbor
  - Converges to perfect solution if separation (but needs more data)
  - Converges to minimal error min(p,1-p) for noisy problems (use increasing k)

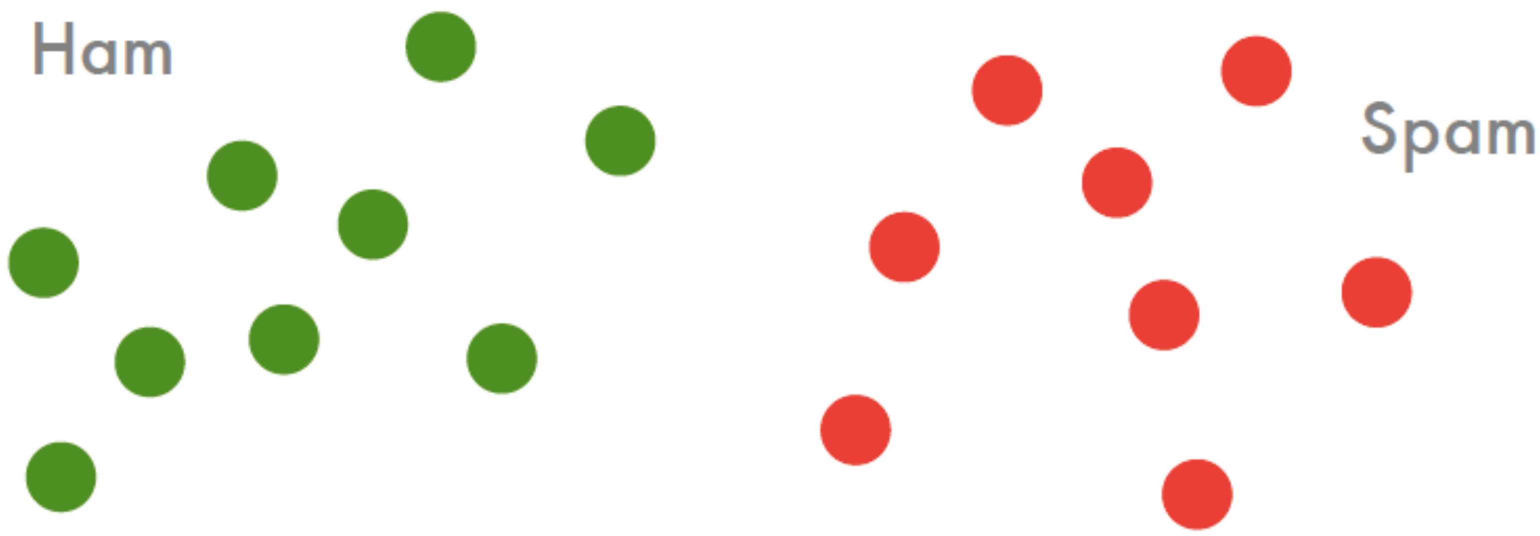
5 minutes break

Support Vector Machines

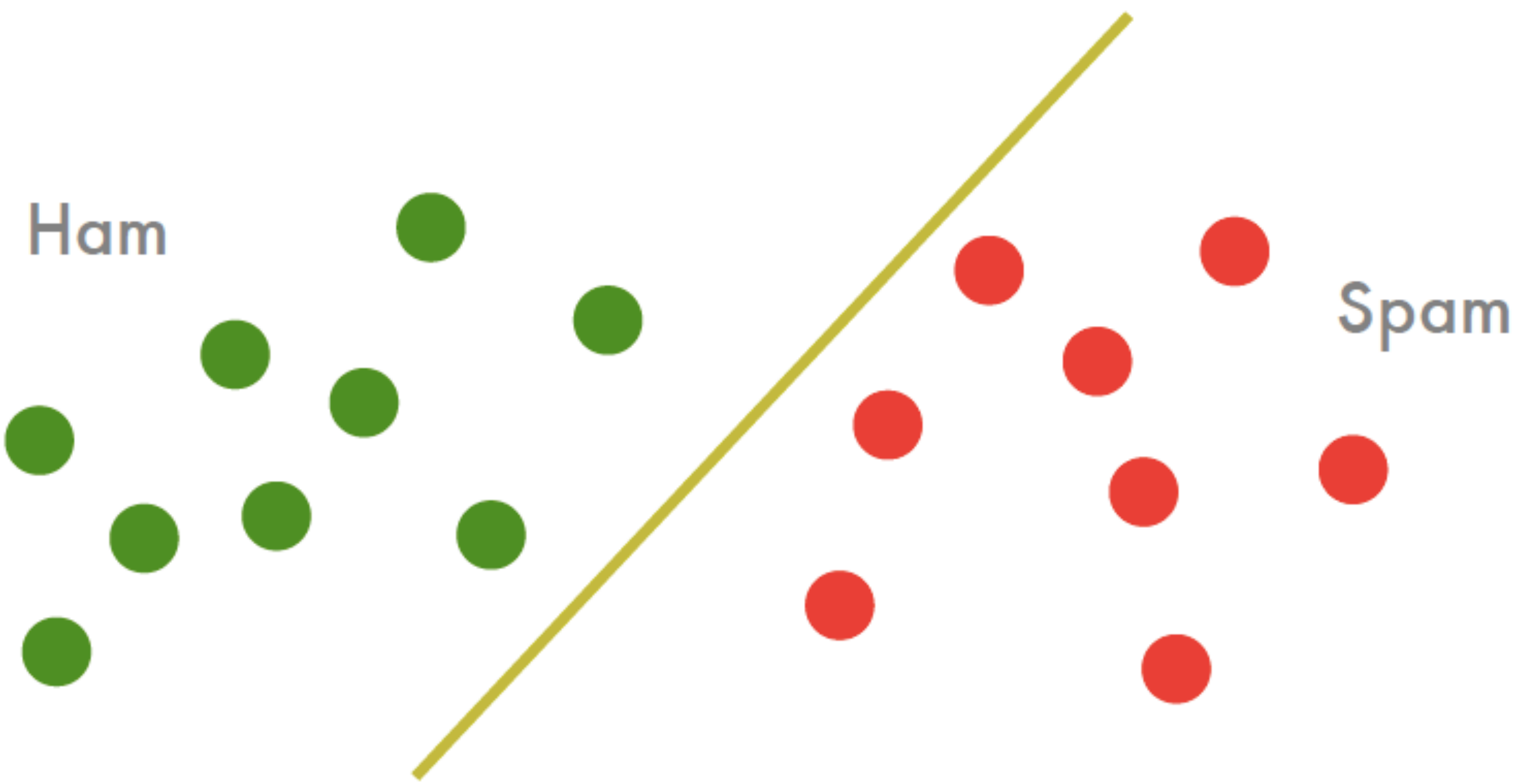
Outline

- Support Vector Classification
  Large Margin Separation, optimization problem
- Properties
  Support Vectors, kernel expansion
- Soft margin classifier
  Dual problem, robustness

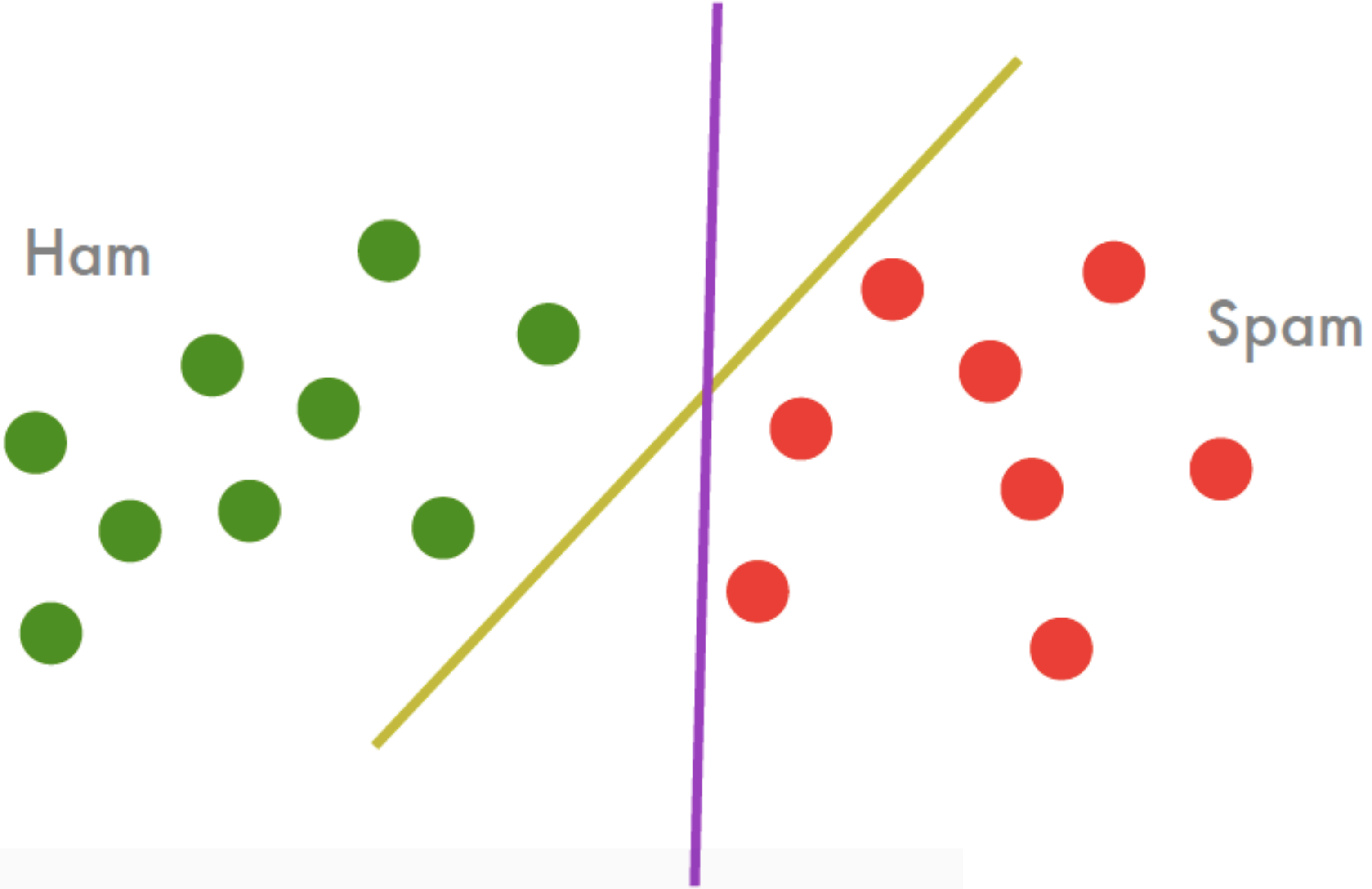
Linear Separator



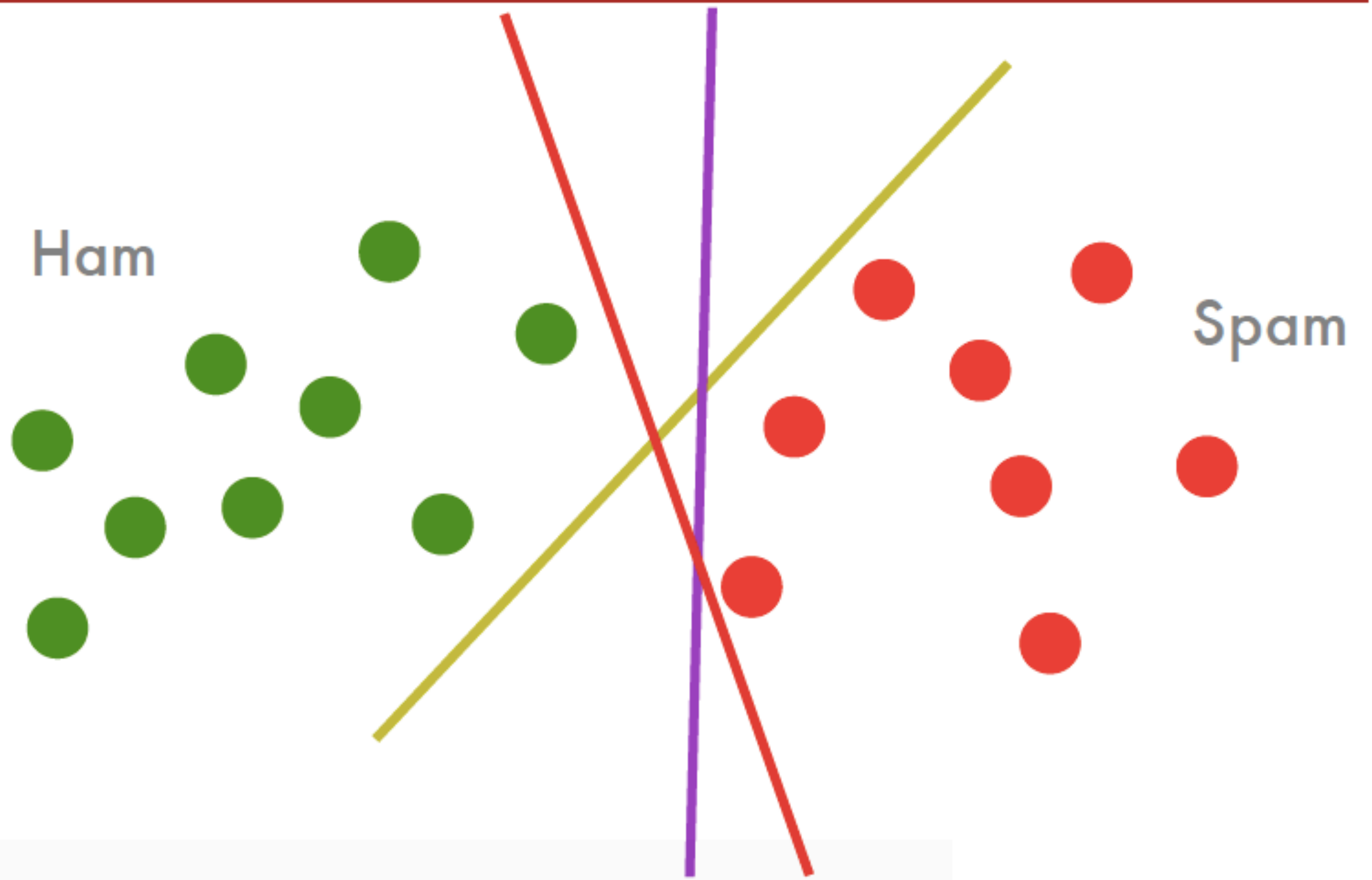
Linear Separator



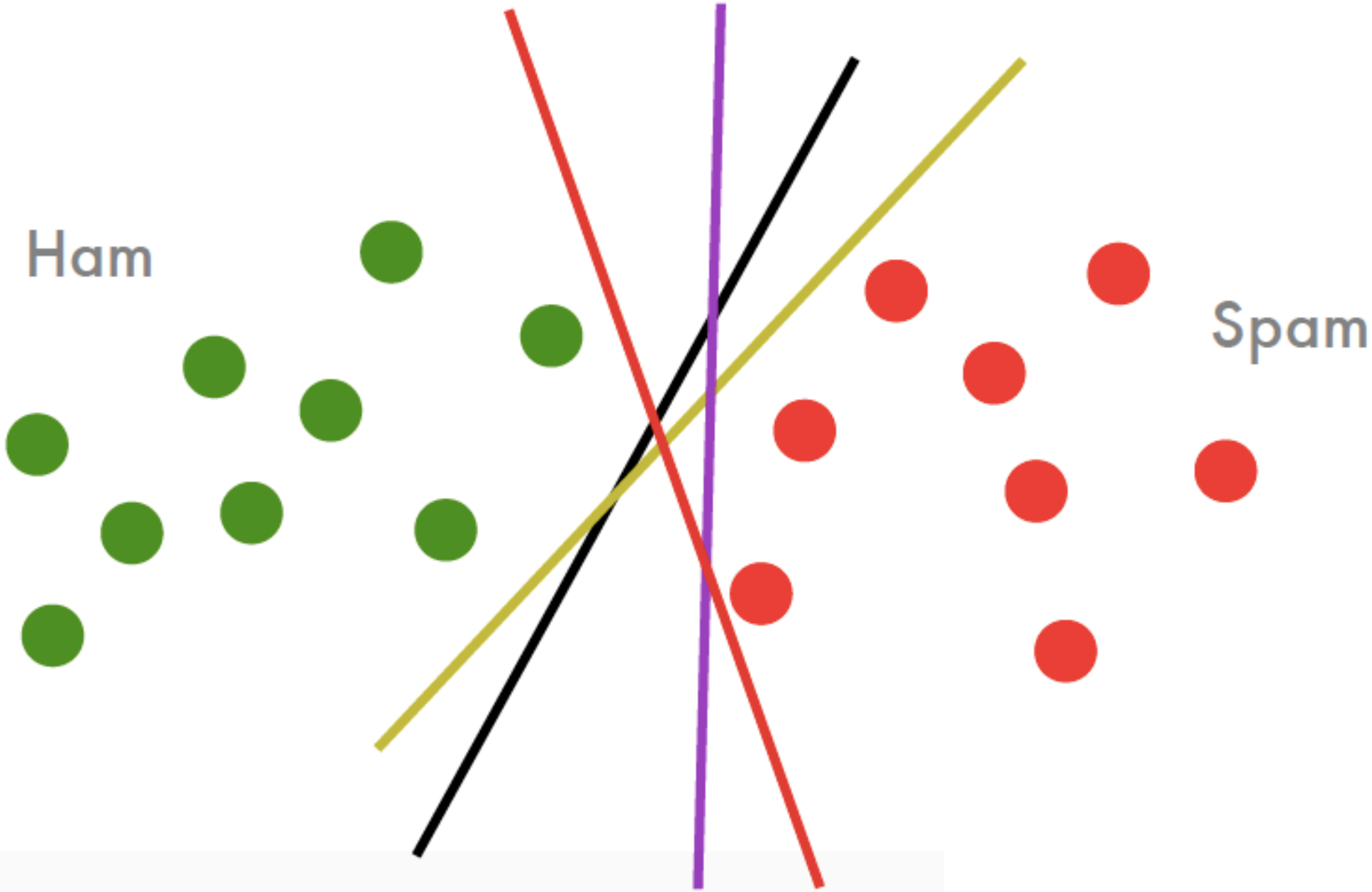
Linear Separator



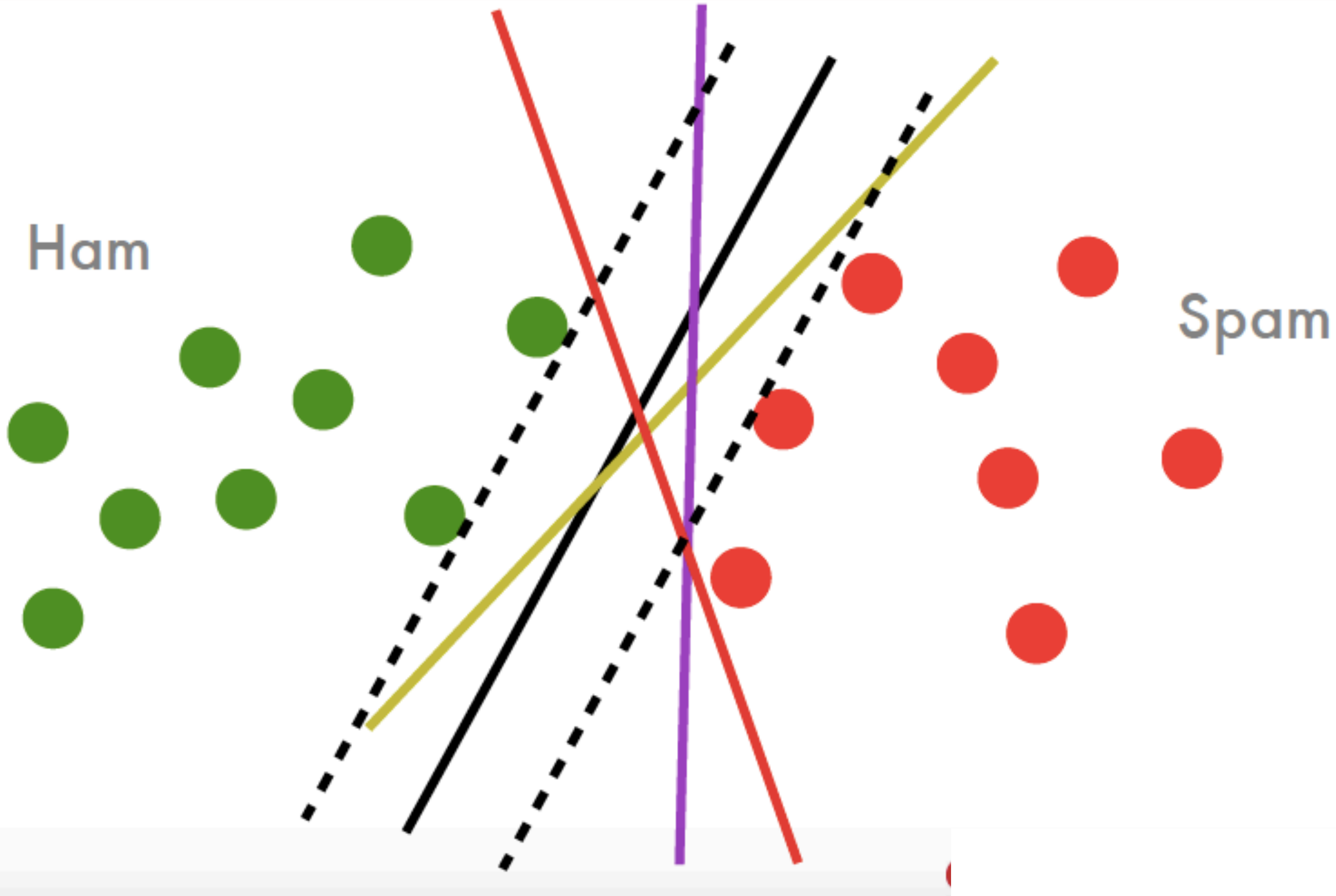
Linear Separator



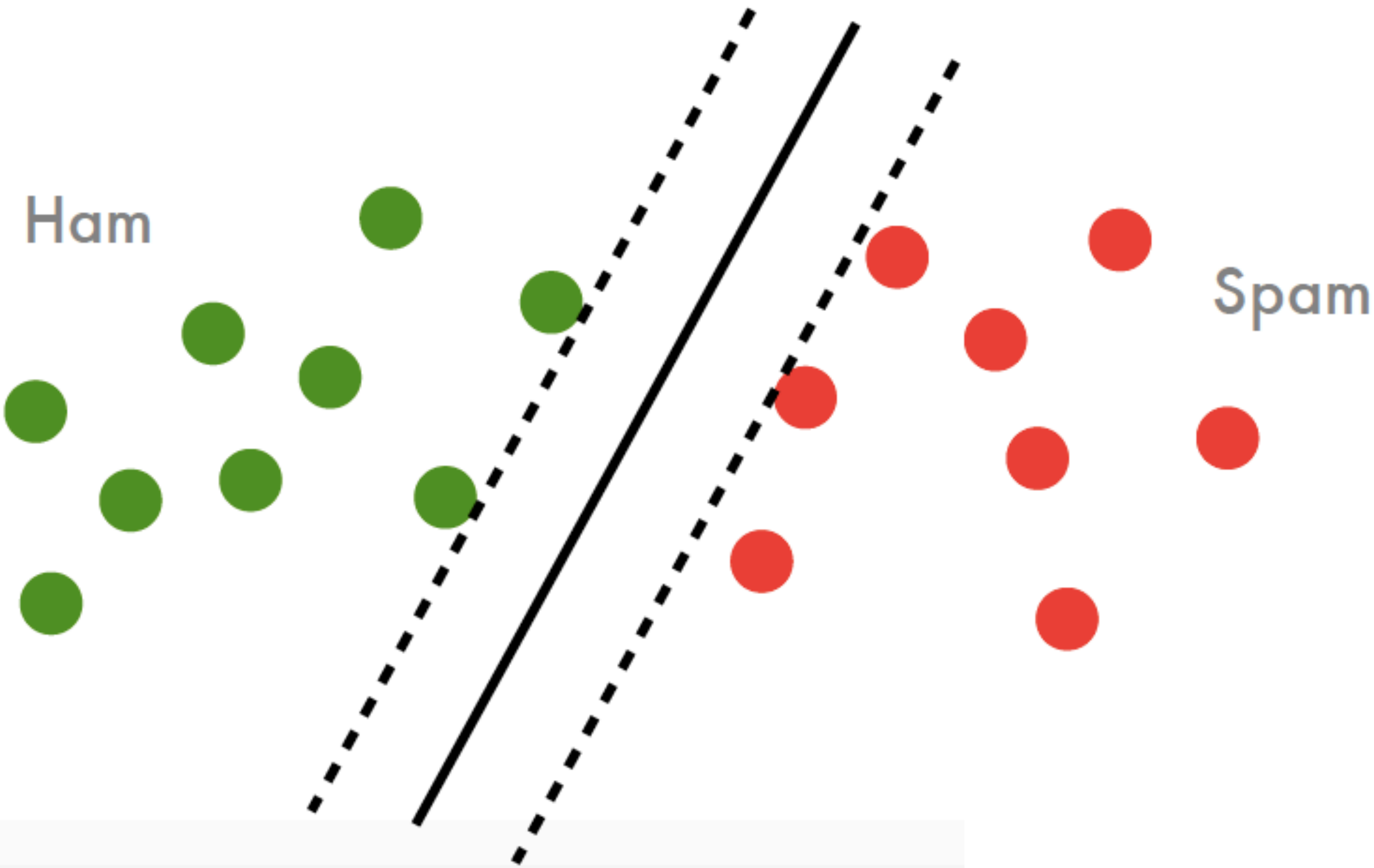
Linear Separator



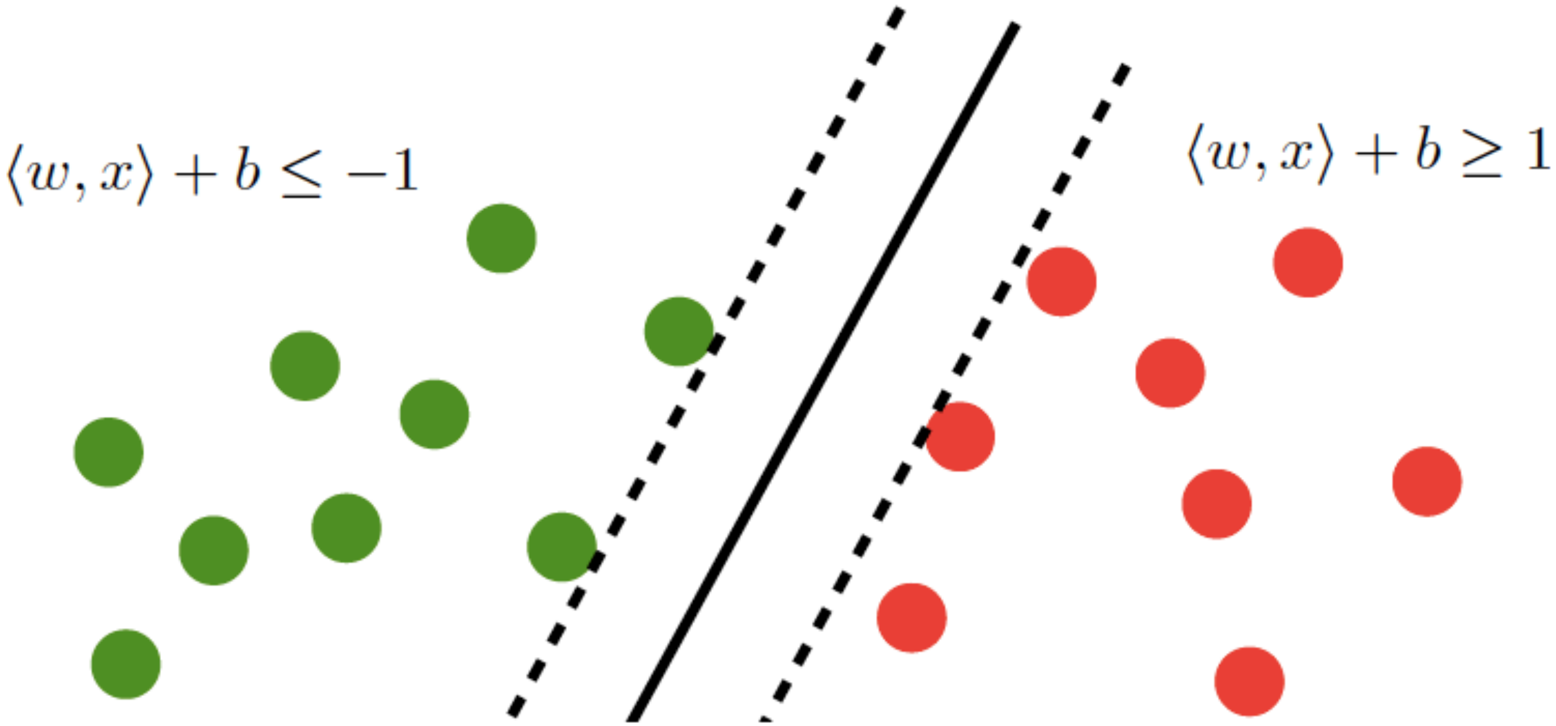
Linear Separator



Linear Separator



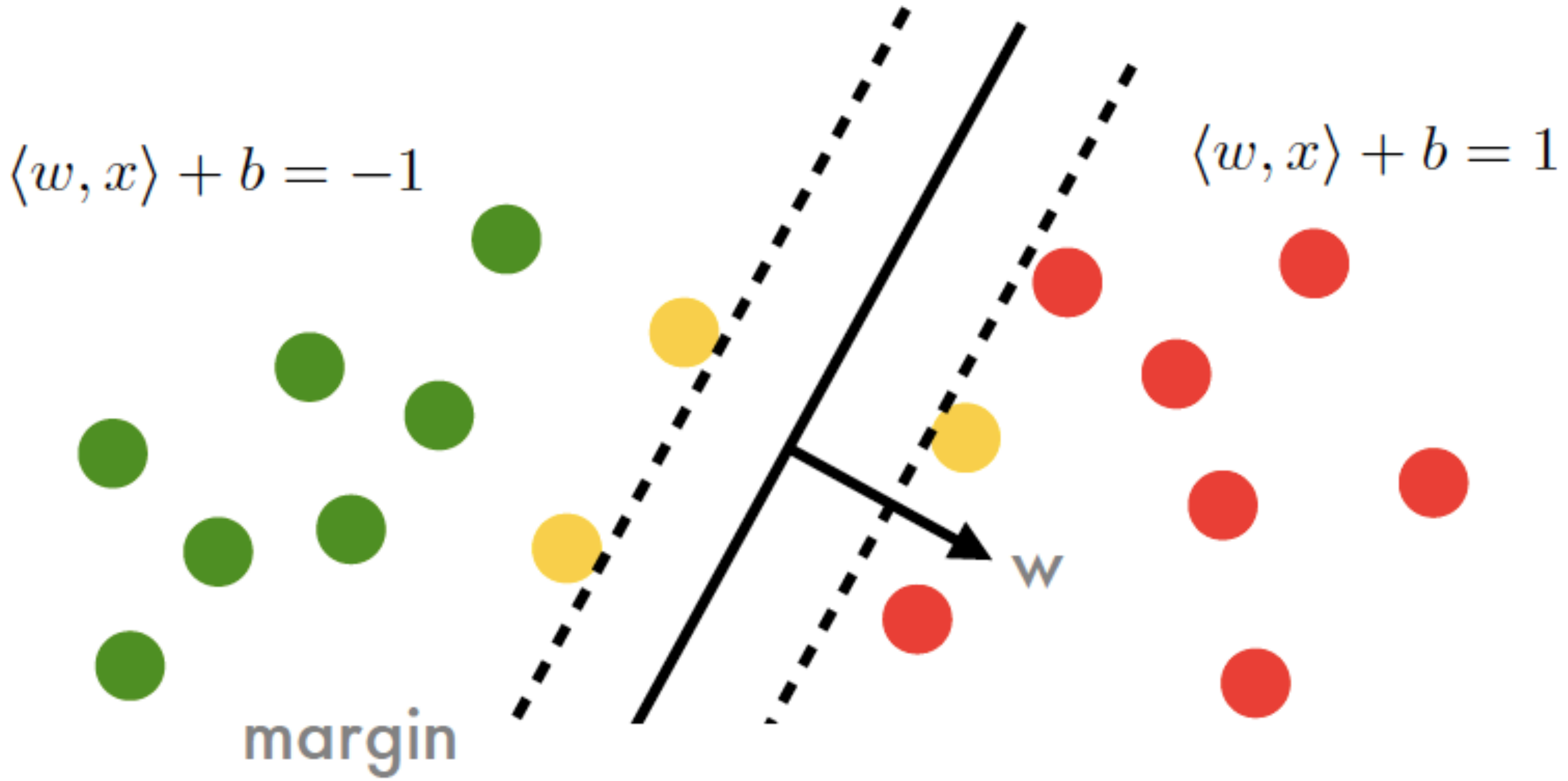
Large Margin Classifier



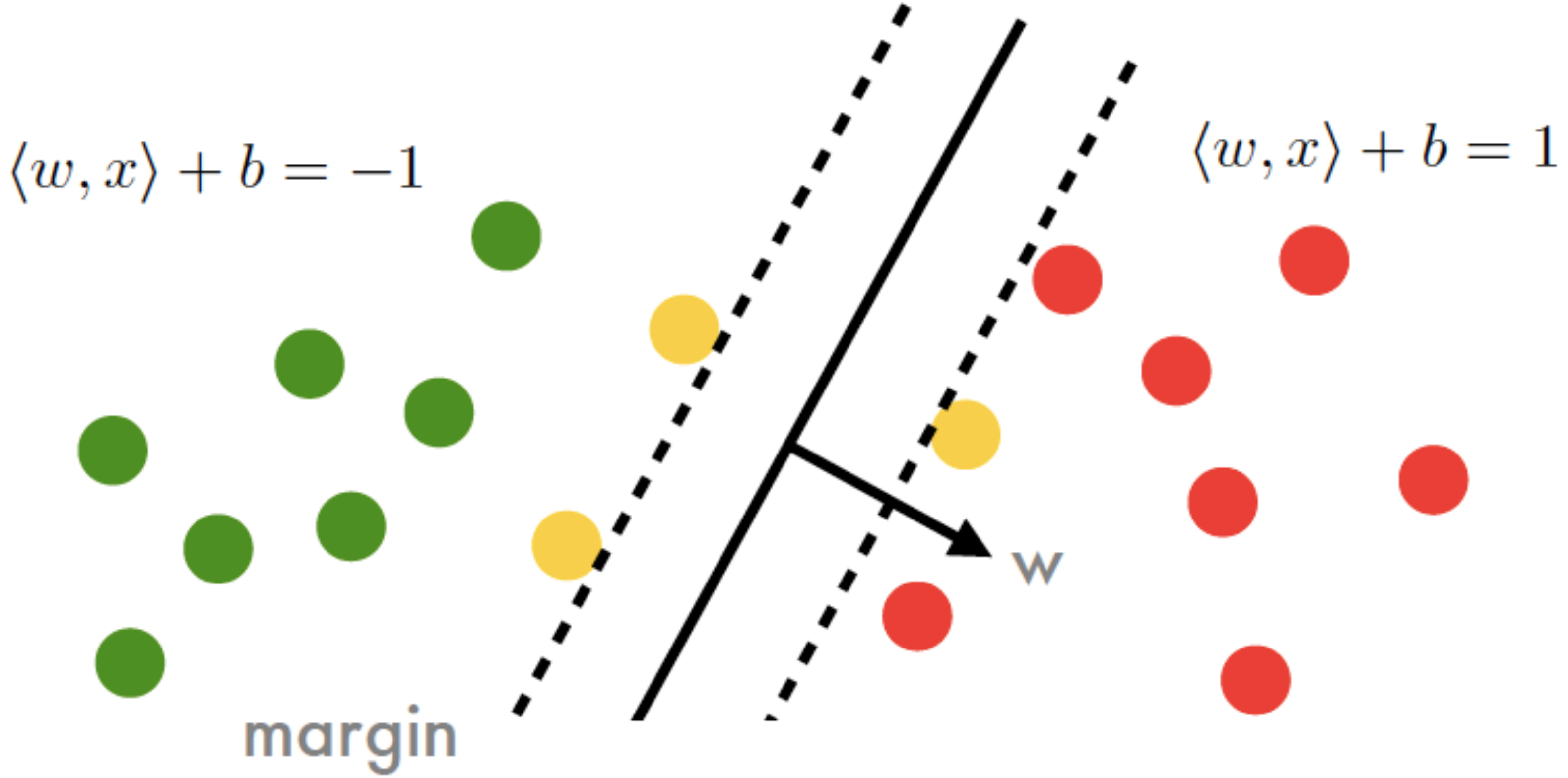
linear function

$$f(x) = \langle w, x \rangle + b$$

Large Margin Classifier

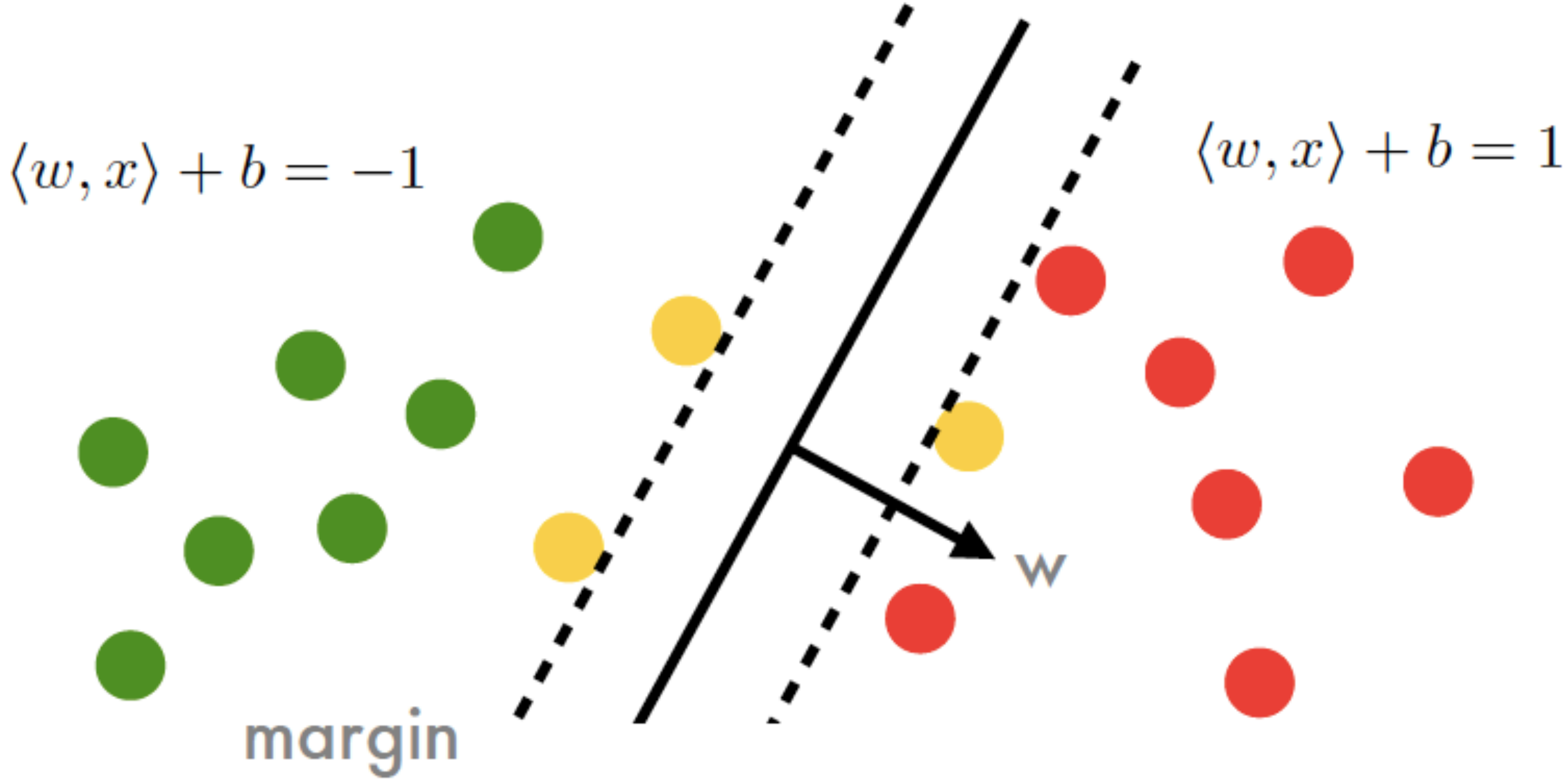


Large Margin Classifier



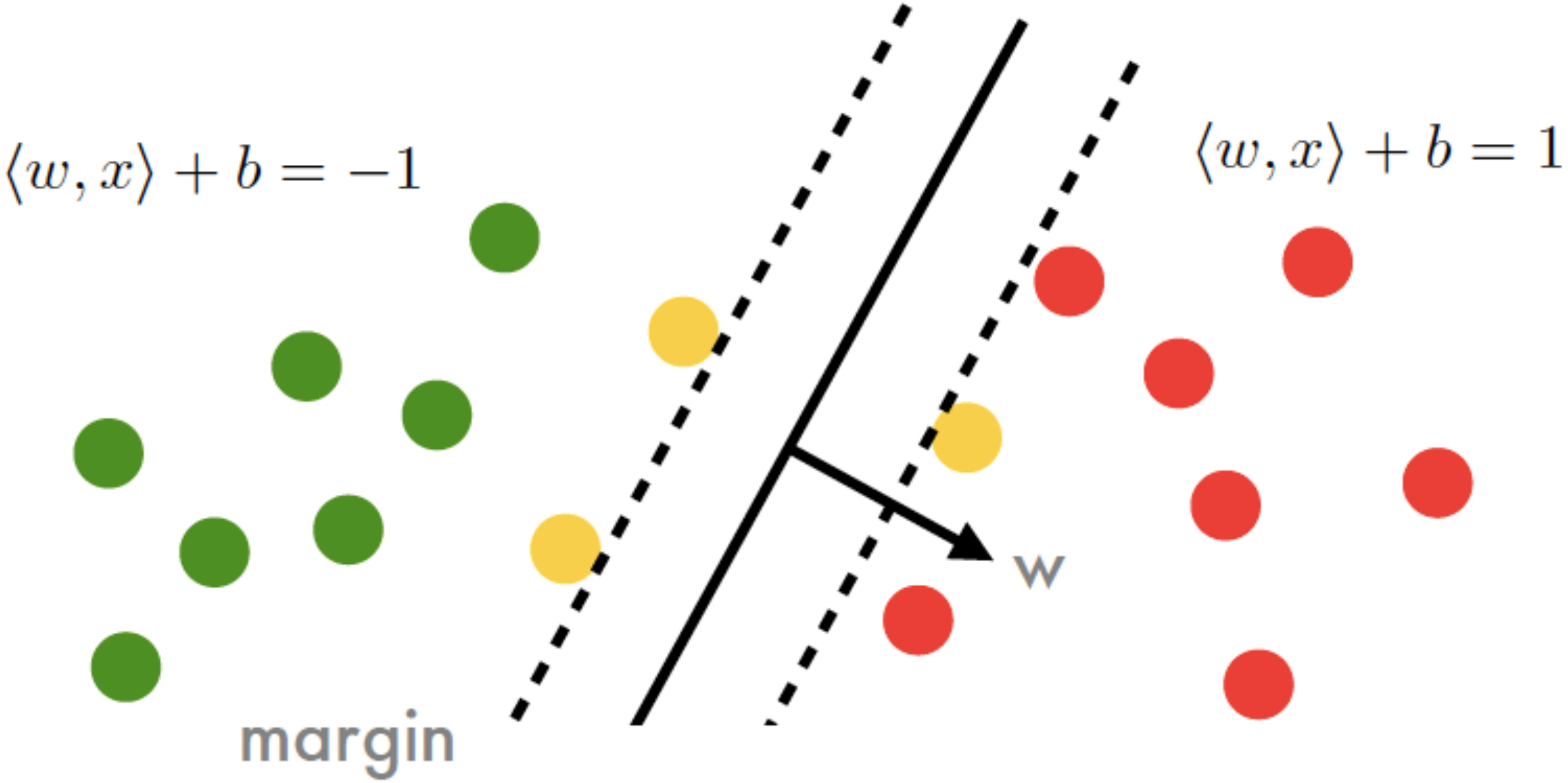
$$\frac{\langle x_+ - x_-, w \rangle}{2\,\|w\|} =$$

Large Margin Classifier



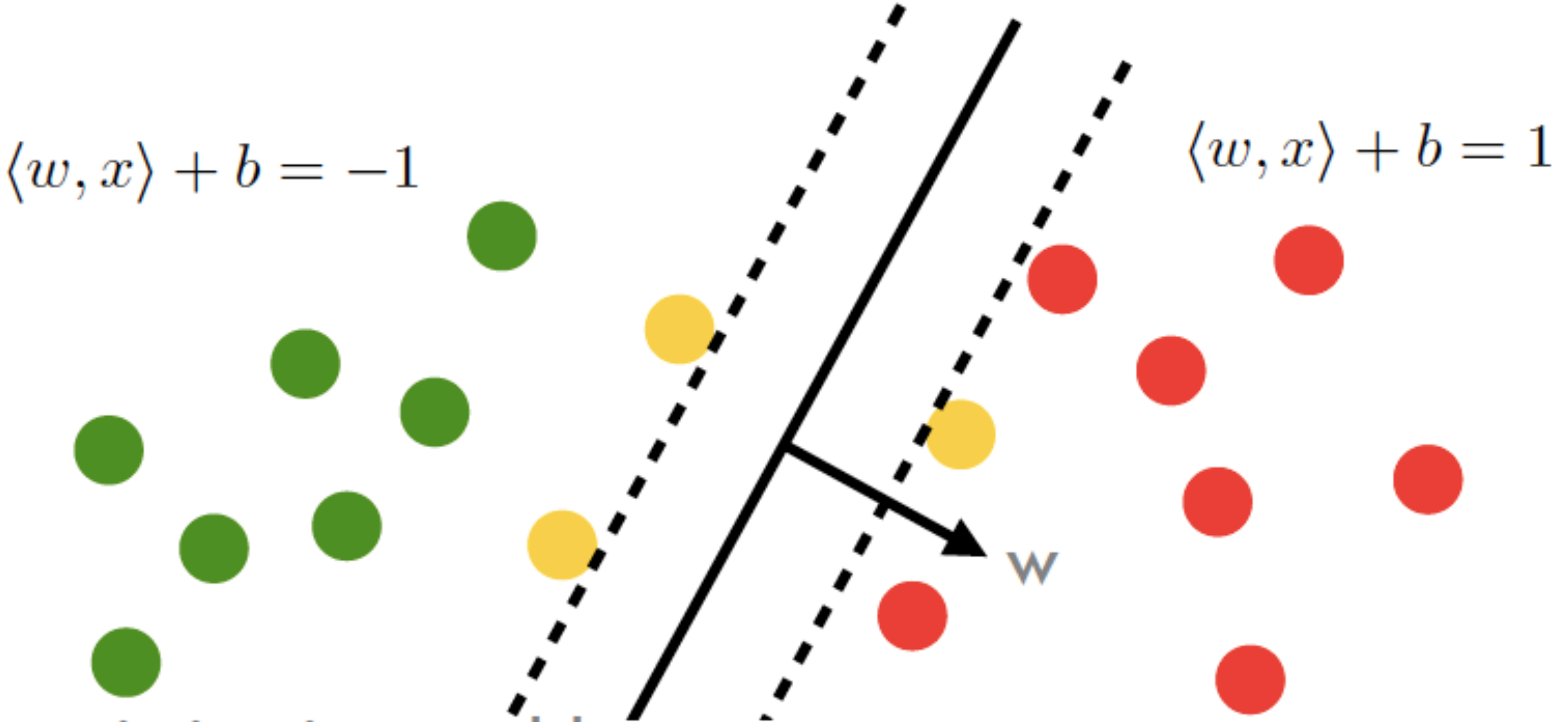
$$\frac{\langle x_+ - x_-, w \rangle}{2\,\|w\|} = \frac{1}{2\,\|w\|}\,[[\langle x_+, w \rangle + b] - [\langle x_-, w \rangle + b]] =$$

Large Margin Classifier



$$\frac{\langle x_+ - x_-, w \rangle}{2\|w\|} = \frac{1}{2\|w\|}\left[\left[\langle x_+, w \rangle + b\right] - \left[\langle x_-, w \rangle + b\right]\right] = \frac{1}{\|w\|}$$

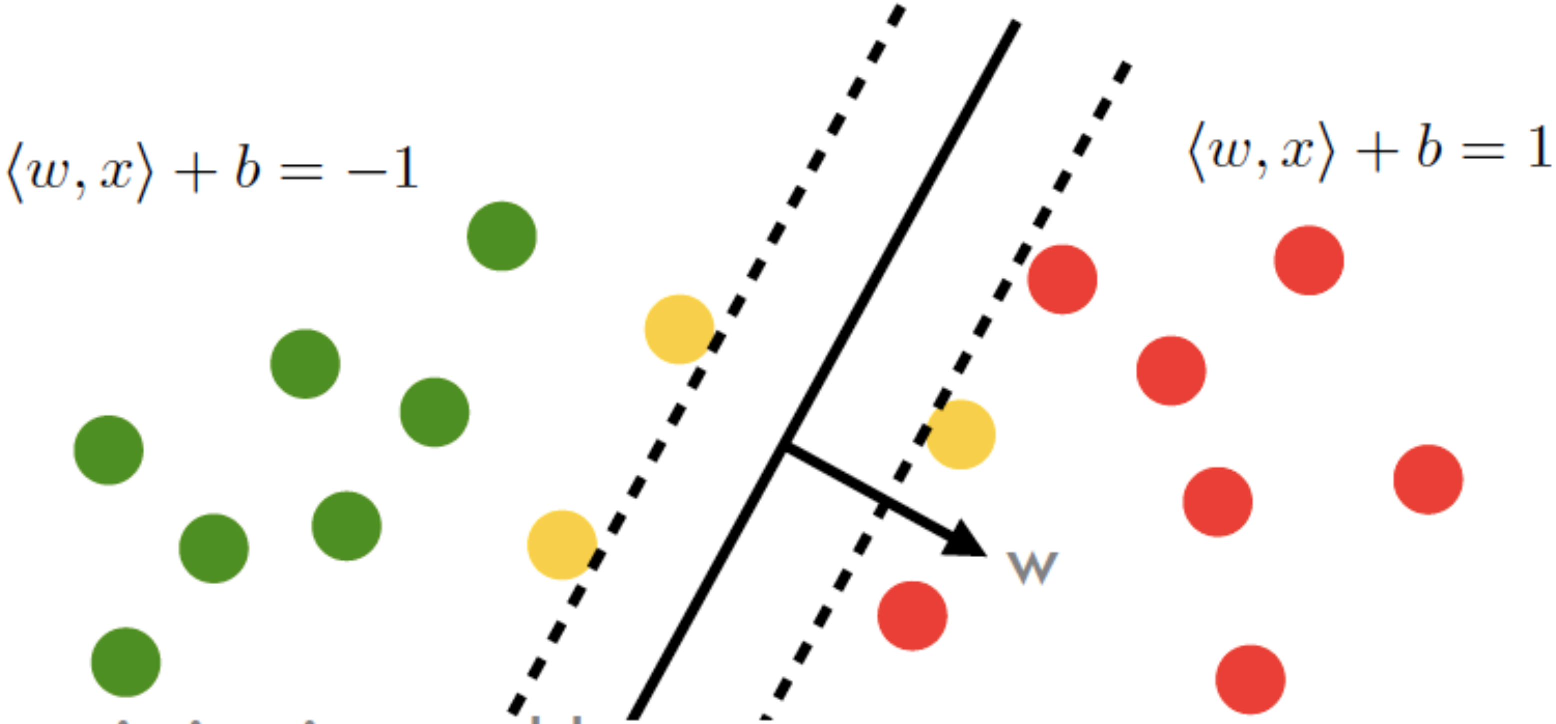
Large Margin Classifier



optimization problem

$$\underset{w,b}{\text{maximize}} \frac{1}{\|w\|} \text{ subject to } y_i \left[\langle x_i, w \rangle + b\right] \geq 1$$

Large Margin Classifier



optimization problem

$$\underset{w,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \text{ subject to } y_i \left[\langle x_i, w \rangle + b \right] \geq 1$$

Dual Problem

- **Primal optimization problem**

$$\underset{w,b}{\text{minimize}}\ \frac{1}{2}\|w\|^2 \text{ subject to } y_i\left[\langle x_i, w\rangle + b\right] \geq 1$$

Dual Problem

- **Primal optimization problem**

$$\underset{w,b}{\text{minimize}} \frac{1}{2}\|w\|^2 \text{ subject to } y_i\left[\langle x_i, w\rangle + b\right] \geq 1$$

- **Lagrange function**

constraint

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left[y_i\left[\langle x_i, w\rangle + b\right] - 1\right]$$

Dual Problem

- Primal optimization problem

$$\underset{w,b}{\text{minimize}}\ \frac{1}{2}\|w\|^2 \text{ subject to } y_i\left[\langle x_i, w\rangle + b\right] \geq 1$$

- Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \overbrace{\left[y_i\left[\langle x_i, w\rangle + b\right] - 1\right]}^{\text{constraint}}$$

  Optimality in w, b is at saddle point with α
- Derivatives in w, b need to vanish

Dual Problem

- **Lagrange function**

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum \alpha_i \left[y_i \left[\langle x_i, w\rangle + b\right] - 1\right]$$

Dual Problem

- Lagrange function

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i \left[y_i \left[\langle x_i, w \rangle + b \right] - 1 \right]$$

- Derivatives in w, b need to vanish

$$\partial_w L(w, b, a) = w - \sum_i \alpha_i y_i x_i = 0$$

Dual Problem

- Lagrange function

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left[y_i \left[\langle x_i, w\rangle + b\right] - 1\right]$$

- Derivatives in w, b need to vanish

$$\partial_w L(w,b,a) = w - \sum_i \alpha_i y_i x_i = 0$$

$$\partial_b L(w,b,a) = \sum_i \alpha_i y_i = 0$$

Dual Problem

- Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left[y_i \left[\langle x_i, w \rangle + b\right] - 1\right]$$

- Derivatives in w, b need to vanish

$$\partial_w L(w, b, a) = w - \sum_i \alpha_i y_i x_i = 0$$

$$\partial_b L(w, b, a) = \sum_i \alpha_i y_i = 0$$

- Plugging terms back into L yields

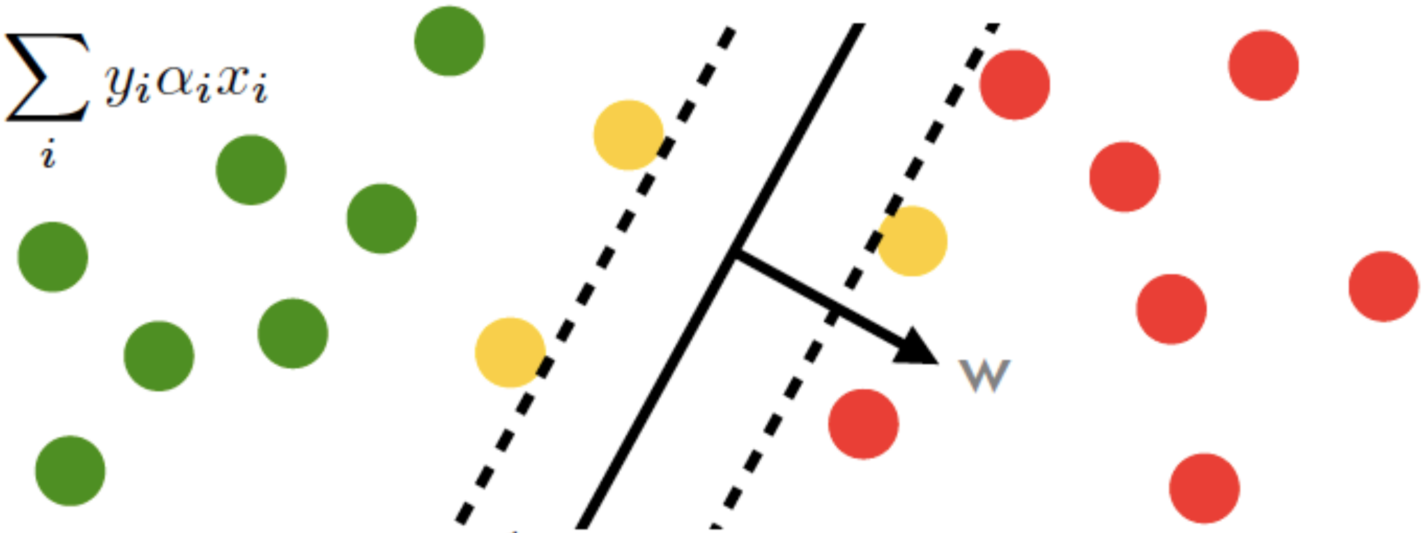
$$\underset{\alpha}{\text{maximize}} - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

$$\text{subject to } \sum \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0$$

Support Vector Machines

$$\underset{w,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \text{ subject to } y_i \left[\langle x_i, w \rangle + b\right] \geq 1$$

$$w = \sum_i y_i \alpha_i x_i$$



$$\underset{\alpha}{\text{maximize}} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

$$\text{subject to } \sum \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0$$

That’s all!