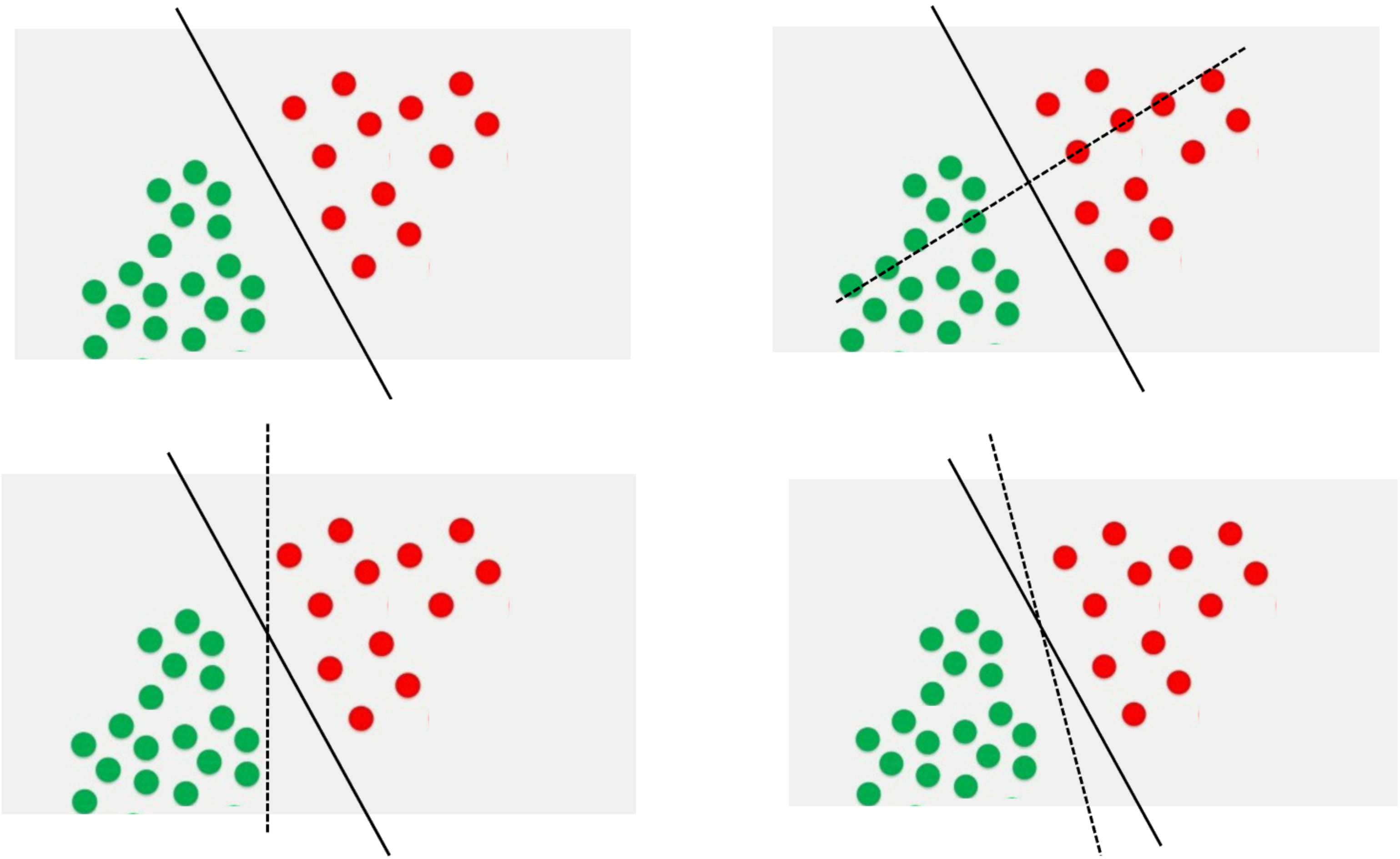
ISE 599
Special Topics Applied Predictive Analytics

Mayank Kejriwal
Research Assistant Professor/Research Lead
Department of Industrial and Systems Engineering
Information Sciences Institute
USC Viterbi School of Engineering
kejriwal@isi.edu

Support Vector Machines

Last time

I showed you a bunch of 'linear' classifiers and asked which one was best

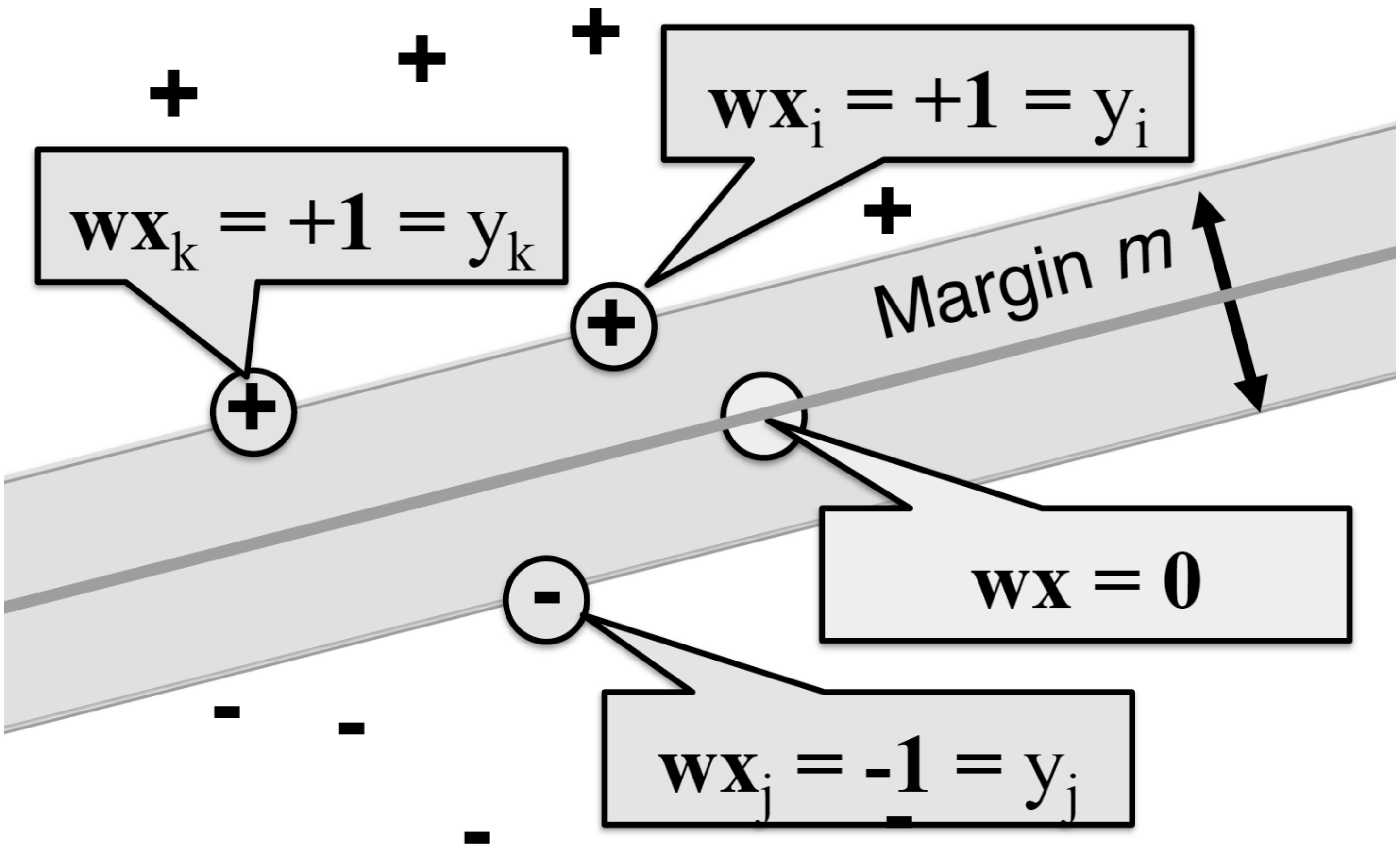


Intuition

- We want the classifier with the decision boundary furthest away from any data point
  - This classifier has **the largest margin**
- This additional requirement (bias) reduces the variance and consequently reduces overfitting

In short, we want the ‘maximum margin classifier’

Maximum margin decision boundary



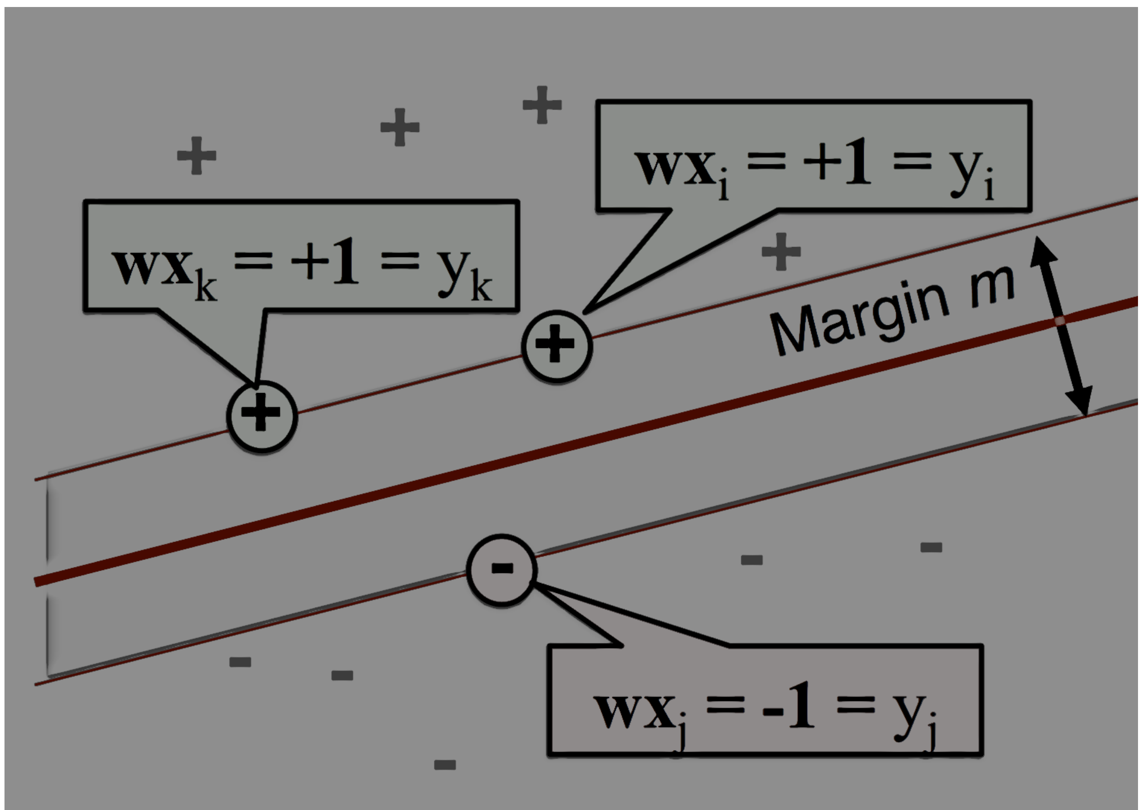
How is this decision boundary defined?

- Two parallel hyperplanes:

- one that goes through the positive data points (y_j = +1) that are closest to the decision boundary, and
- one that goes through the negative data points (y_j = -1) that are closest to the decision boundary

Support vectors

- express the separating hyperplane in terms of the data points xj that are closest to the decision boundary
- such data points are known as ‘support vectors’



Primal vs. dual representation (advanced)

- Primal
  - The data items x = (x1…xn) have n features
  - The weight vector w = (w1…wn) has n elements
  - Learning:
    - Find a weight wj for each feature xj
  - Classification:
    - Evaluate **wx**

- Dual
  - We can represent **w** as a linear combination of the items in the training data:

$$\mathbf{w} = \sum_j \alpha_j \mathbf{x}_j$$

  - Learning:
    - Find a weight αj (≥ 0) for each data point xj
    - This requires computing the inner product xixj = 〈xi xj〉 between all data items xi xj
  - Support vectors
    - Set of data points xj with non-zero weights αj

Classifying test data (primal vs. dual)

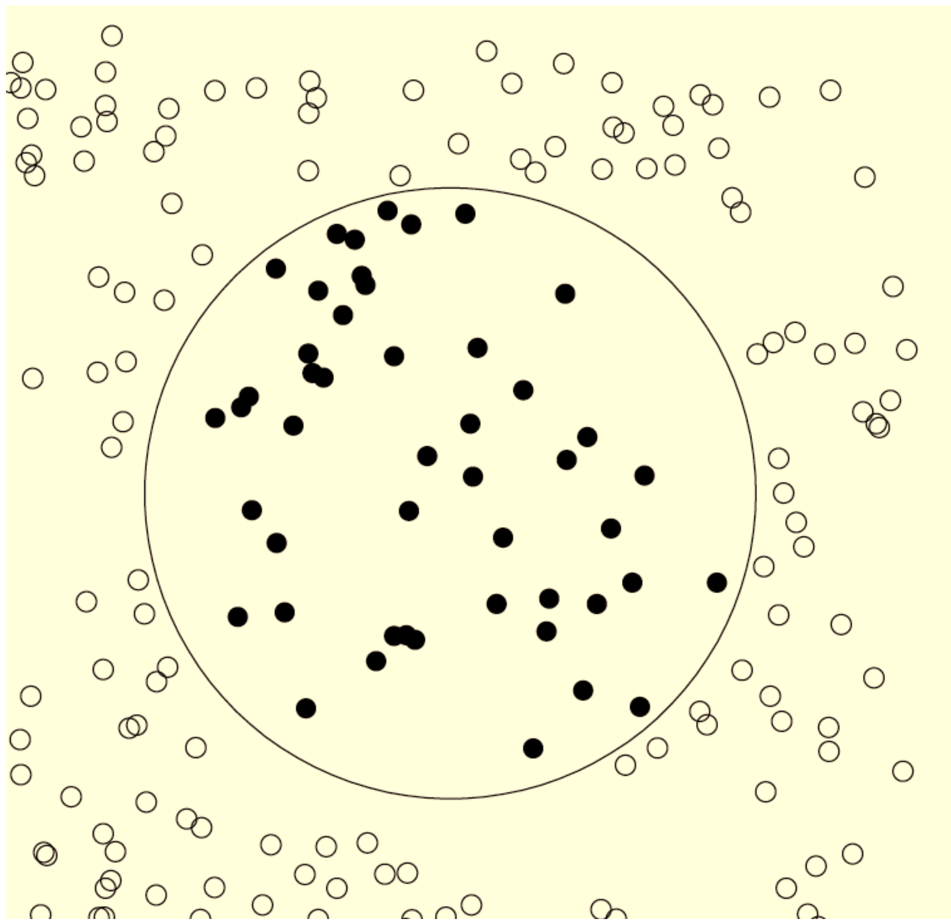
- Primal:
  - compute inner product between weight vector and test item

$$\mathbf{wx} = \langle \mathbf{w}, \mathbf{x} \rangle$$

- Dual:
  - compute inner product between support vectors and test item

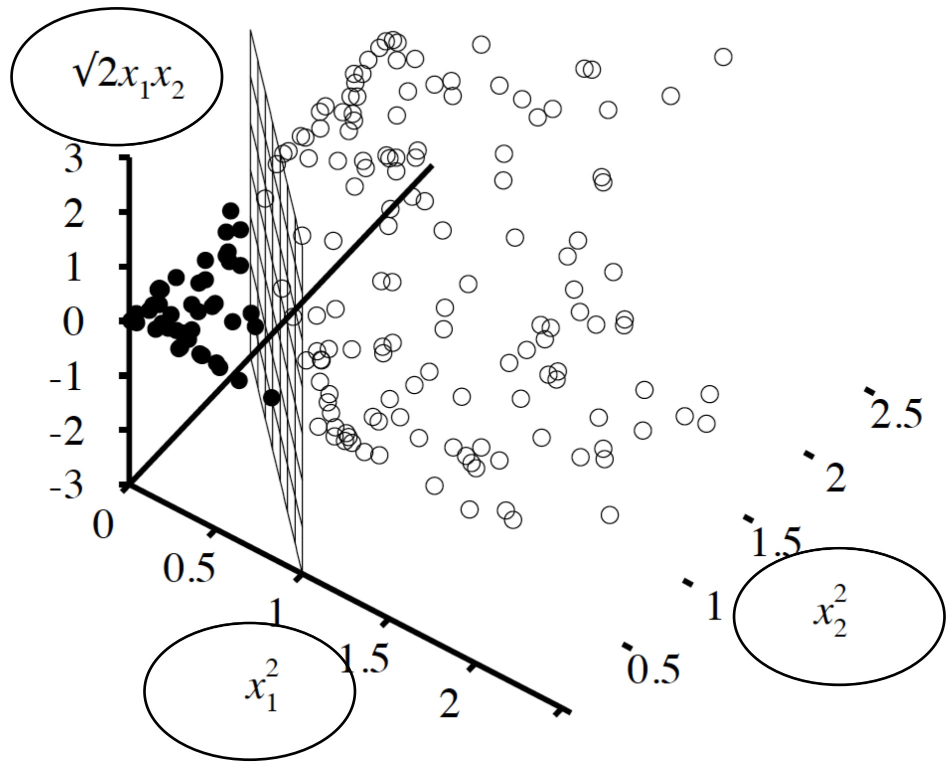
$$\mathbf{wx} = \langle \mathbf{w}, \mathbf{x} \rangle = \langle \sum_j \alpha_j x_j, \mathbf{x} \rangle = \sum_j \alpha_j \langle x_j, \mathbf{x} \rangle$$

We’ve been assuming linear separability so far, but what if it is violated?



Kernel trick

- Map data items to new feature space that will make them linearly separable
- In this example, we've taken the original variables (x1 and x2) and mapped them to functions of themselves to make the space amenable to linear classifiers



Kernel trick (cont'd)

- It turns out that N (independent) data points will always be linearly

separable in N-1 dimensions.

- Intuition 1: if we map each x to a point G(x) in a higher-dimensional feature space, the data become linearly separable!

- Intuition 2: in the dual, we compute 〈G(xi) G(xj)〉. This can often be computed directly as a 'kernel' function K(xi, xj)