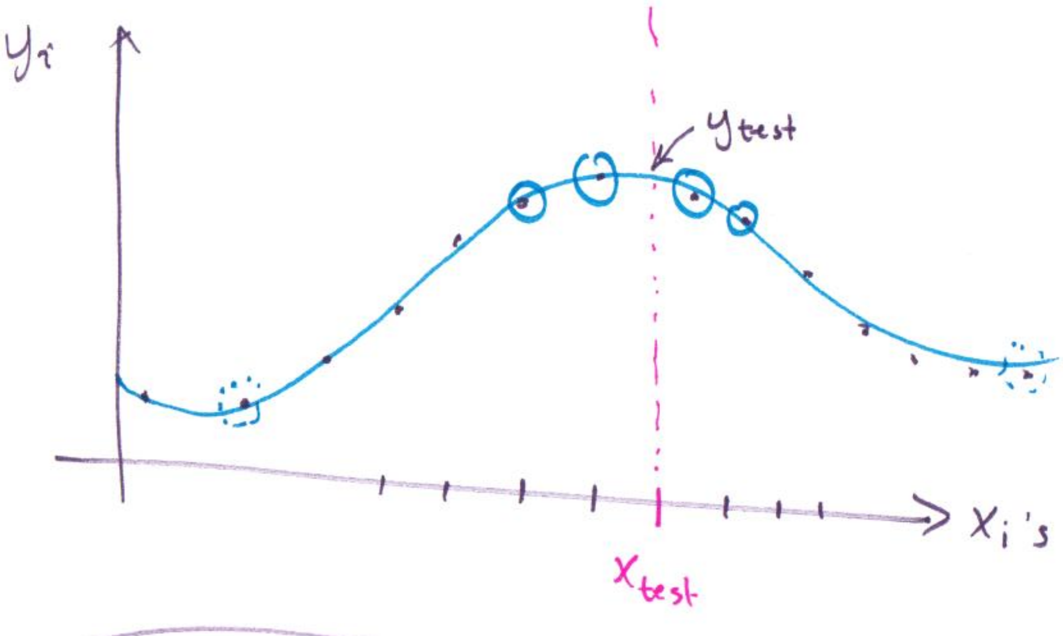
Lecture 26: Kernels and the SVM



Parametric methods

e.g. fit polynomial to data; use training data to learn parameters (poly. coefficients)

Nonparametric methods

e.g. $y = f(x)$ is "smooth"

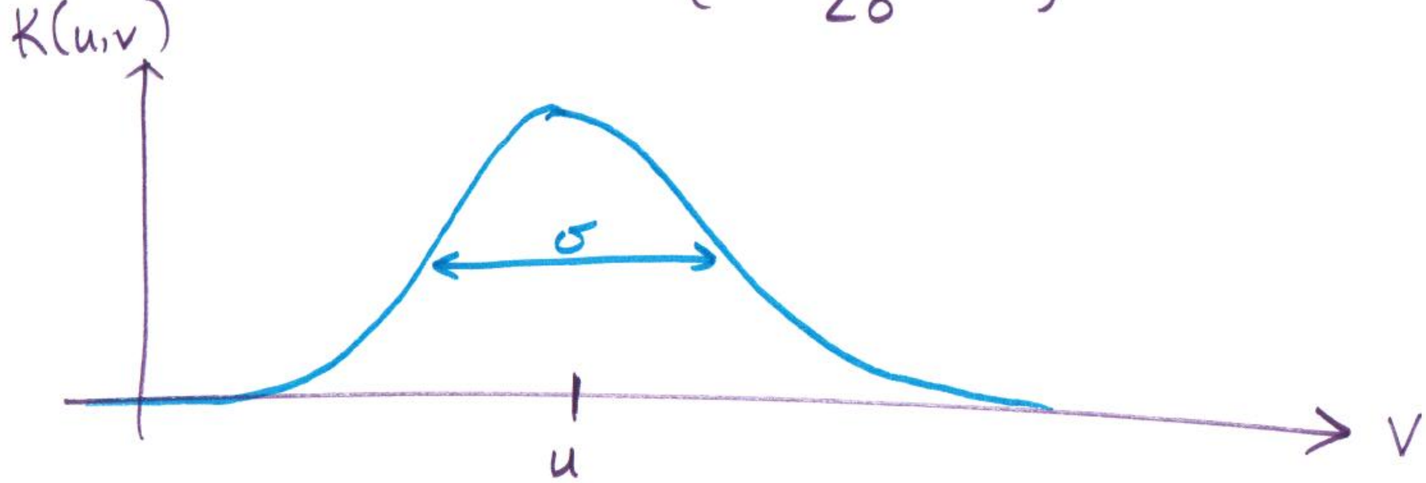
y_{test} = weighted sum y_i's.

assign bigger weight to y_i if x_i is close to x_{test}

Recall: Kernel function measures similarity or alignment ~~of~~ of two feature vectors

e.g. Gaussian Kernel

$$K(u,v) = \exp\left\{ \frac{-\|u-v\|_2^2}{2\sigma^2} \right\}$$



Nadaraya-Watson Kernel Regression

$$\hat{y}_{test} = \frac{\sum_{i=1}^{n} y_i K(x_i, x_{test})}{\sum_{j=1}^{n} K(x_j, x_{test})}$$

e.g. Kernel Ridge Regression

$$\hat{y}_{test} = \sum_i \alpha_i \, K(x_i, x_{test})$$

$$\text{w/} \quad \alpha = (K + \lambda I)^{-1} y$$

equivalent to ridge regression in a high-dimensional feature space

where $K(\cancel{x,y}\, u, v) = \langle \phi(u), \underbrace{\phi(v)}_{} \rangle$

high dimensional feature vectors.

Kernels and Support Vector Machines for classification

labels $y_i \in \{+1, -1\}$

First, consider regularized least squares:

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n} \left(1 - y_i x_i^T w\right)^2 + \lambda \|w\|_2^2$$

Last time, showed $\hat{w} = X^T \alpha$ for some α

$$= \sum_{i=1}^{n} \alpha_i x_i$$

$$\Rightarrow \hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} \left(1 - y_i x_i^T \left(\sum_{j=1}^{n} \alpha_j x_j\right)\right)^2 + \lambda \left\|\sum_j \alpha_j x_j\right\|_2^2$$

$$= \arg\min_{\alpha} \sum_{i=1}^{n} \left(1 - y_i \sum_{j=1}^{n} \alpha_j \underline{\langle x_i, x_j \rangle}\right)^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \underline{\langle x_i, x_j \rangle}$$

"Kernel trick" — replace inner products $\langle x_i, x_j \rangle$ with the kernel function $K(x_i, x_j)$

$$\Rightarrow \hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{n} \Big(1 - y_i \sum_{j=1}^{n} \alpha_j K(x_i, x_j)\Big)^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$$

$$\hat{\alpha} = (K + \lambda I)^{-1} y.$$

Now consider hinge loss:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \big(1 - y_i x_i^T \underline{w}\big)_+ + \lambda \|\underline{w}\|_2^2$$

- show $\hat{w} = X^T \alpha = \sum_{j=1}^{n} \alpha_j x_j$ (like before, but different α's)

Imagine: $\underline{\hat{w} = X^T \alpha + x^{\perp}}$ ($x^{\perp}$ some vector orthogonal to the x_i's)

$$\min_{\alpha, x^{\perp}} \sum_{i=1}^{n} \left(1 - y_i x_i^T \left(\sum_{j=1}^{n} \alpha_j x_j + x^{\perp}\right)\right)_+ + \lambda \left\| \sum_{j=1}^{n} \alpha_j x_j + x^{\perp} \right\|_2^2$$

$$= \min_{\alpha, x^{\perp}} \sum_{i} \left(1 - y_i \left[\sum_{j=1}^{n} \alpha_j \langle x_i, x_j \rangle + \underbrace{x_i^T x^{\perp}}_{\text{always } 0!}\right]\right)_+ + \lambda \left[\left\| \sum_{j=1}^{n} \alpha_j x_j \right\|_2^2 + \| x^{\perp} \|_2^2\right]$$

$$= \min_{\alpha, x^{\perp}} \sum_{i} \left(1 - y_i \sum_{j=1}^{n} \alpha_j \langle x_i, x_j \rangle\right)_+ + \lambda \left[\left\| \sum_{j=1}^{n} \alpha_j x_j \right\|_2^2 + \| x^{\perp} \|_2^2\right]$$

this must be zero at optimum

$$\Rightarrow x^{\perp} = 0$$

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} \left(1 - y_i \sum_{j=1}^{n} \alpha_j \underline{\langle x_i, x_j \rangle}\right)_+ + \lambda \underbrace{\left\| \sum_{j=1}^{n} \alpha_j x_j \right\|_2^2}_{= \sum_j \sum_i \alpha_i \alpha_j \langle x_i, x_j \rangle}$$

Apply Kernel trick:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{n} \left(1 - y_i \sum_j \alpha_j \underline{K(x_i, x_j)}\right)_+ + \lambda \sum_i \sum_j \alpha_i \alpha_j \underline{K(x_i, x_j)}$$

no closed-form expression for $\hat{\alpha}$

find $\hat{\alpha}$ using optimization (GD)

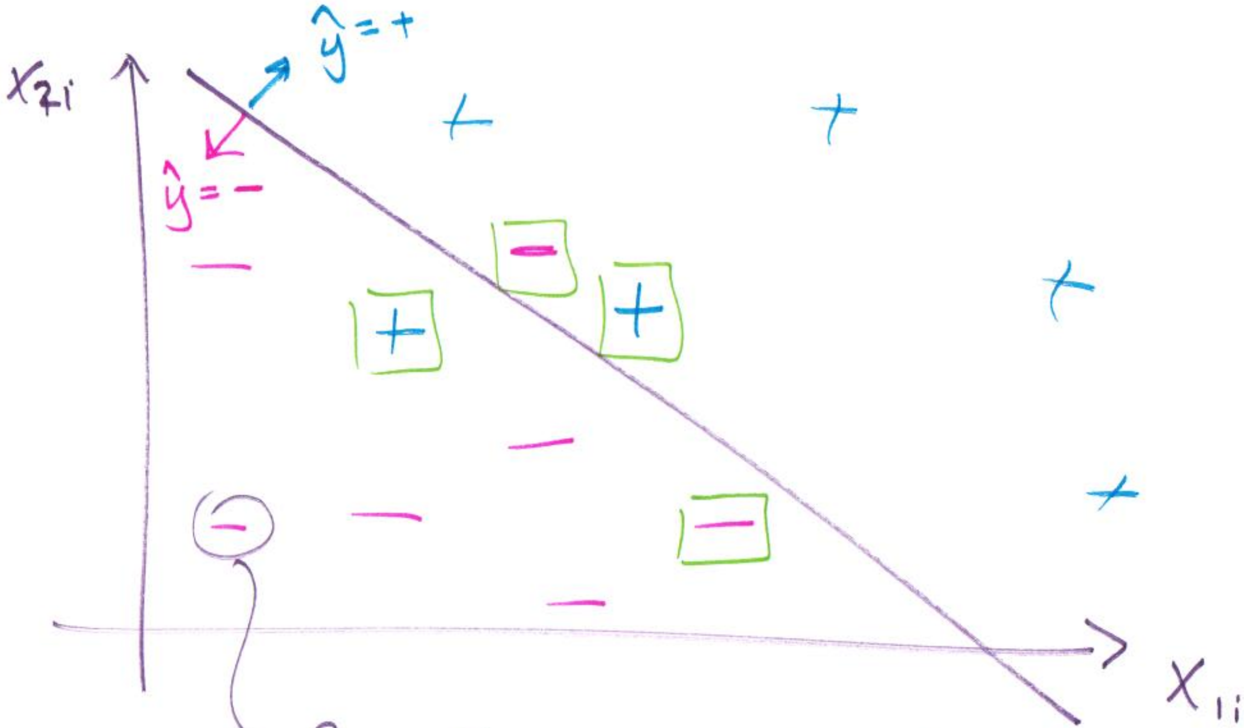
Why is this called a "support vector machine"?

$\hat{\alpha}$ is sparse — most $\hat{\alpha}_j = 0$

recall $\hat{w} = \sum_{j=1}^{n} \hat{\alpha}_j x_j$

$\Rightarrow$ $\hat{w}$ = lin. combo of only a few training x_i's

these x_i's are called "support vectors"



far from decision boundary
AND correctly classified
$\Rightarrow$ does not affect hinge loss
$\Rightarrow$ receive zero weight in opt solution
(corresponding $\alpha = 0$)

At start of semester:

- no background in lin. alg.
- no background in ML

Now

- classification (face emotions, irises)
- PCA (dimension reduction)
- Matrix completion (recommeder system)
- PageRank
- Optimization, Neural networks
- Kernels + SVM