Kernel Based Support Vector Machines

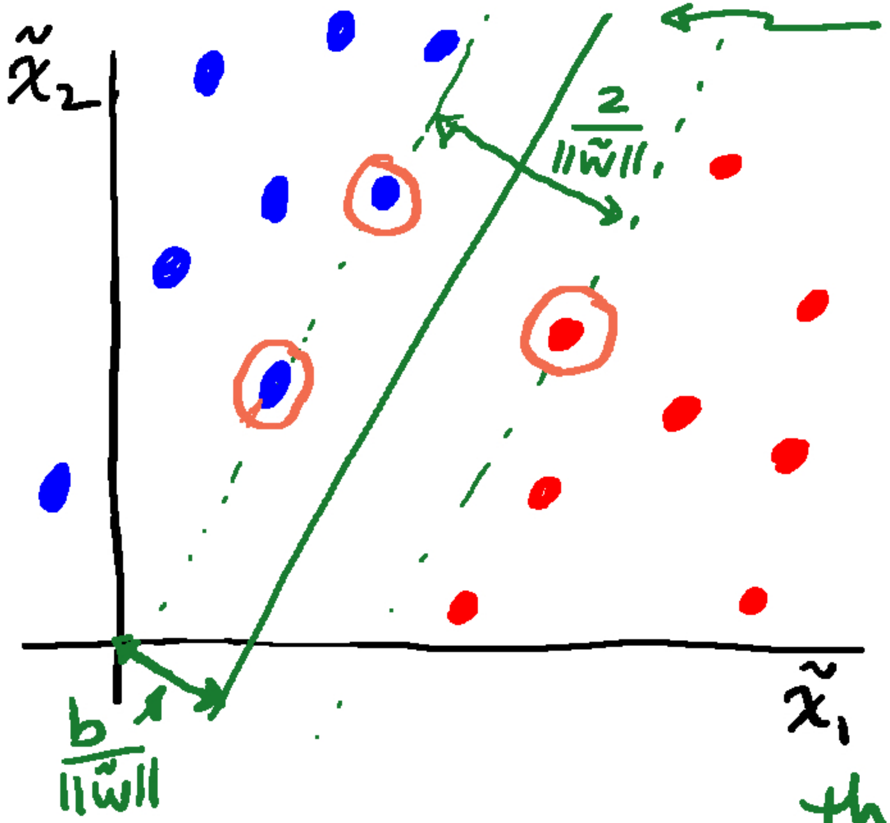
Objectives

- reformulate linear max margin classifier in terms of support vectors

- derive kernel version of hinge loss with ridge regression

- summarize features of support vector machines

Support vectors define max-margin classifier



$\underline{\tilde{x}}^T \underline{\tilde{w}} + b = 0$

$\underline{x}^T \underline{w} = 0$

$\underline{w} = [\tilde{w}^T \ b]^T$ depends only on the support vectors

Recall kernel regression

$$d(\underline{x}) = \underline{\phi}^T(\underline{x})\underline{w} = \sum_{j=1}^{N} \alpha_j K(\underline{x}, \underline{x}^j)$$

Let $\underline{\phi}(\underline{x}) = \underline{x} \Rightarrow K(\underline{x}, \underline{x}^j) = \underline{x}^T \underline{x}^j$

then $d(\underline{x}) = \sum_{j=1}^{N} \alpha_j \underline{x}^T \underline{x}^j = \underline{x}^T \underbrace{\sum_{j=1}^{N} \alpha_j \underline{x}^j}_{\underline{w}}$

So $\underline{w} = \sum_{j=1}^{N} \alpha_j \underline{x}^j$

All $\alpha_j = 0$ except support vectors!

Use kernels for nonlinear decision boundaries

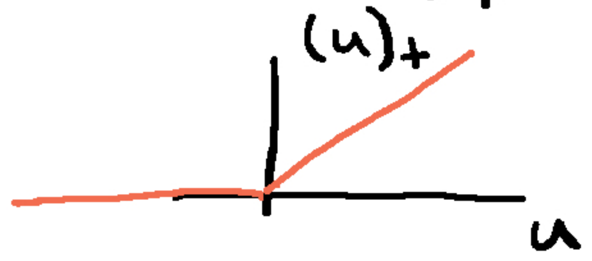
High-dimensional feature space: $\underline{x} \rightarrow \underline{\phi}(\underline{x})$

e.g., $\underline{\phi}(\underline{x}) = [\, x_1^2 \;\; x_2^2 \; \cdots \; x_2 x_4 \; \cdots \; x_{M-1} \;\; x_M \;\; 1 \,]$

$$\hat{d}(\underline{x}) = \text{sign}\left(\underline{\phi}^T(\underline{x})\,\underline{w} \right)$$

Hinge loss with ridge regression

$$\min_{\underline{w}} \sum_{i=1}^{N} \left(1 - d^i \underline{\phi}^T(\underline{x}^i)\,\underline{w}\right)_+ + \lambda \|\underline{w}\|_2^2$$



Claim: $\underline{w} = \sum_{j=1}^{N} \underline{\phi}(\underline{x}^j)\,\alpha_j$ (proof in notes)

Kernel "trick" replaces $\underline{\phi}^T(\underline{x}^i)\,\underline{\phi}(\underline{x}^i)$ with $K(\underline{x}^j, \underline{x}^i)$

Restate: $\underline{w} = \sum_{j=1}^{N} \alpha_j \underline{\phi}(\underline{x}^j)$

Hinge loss with ridge regression

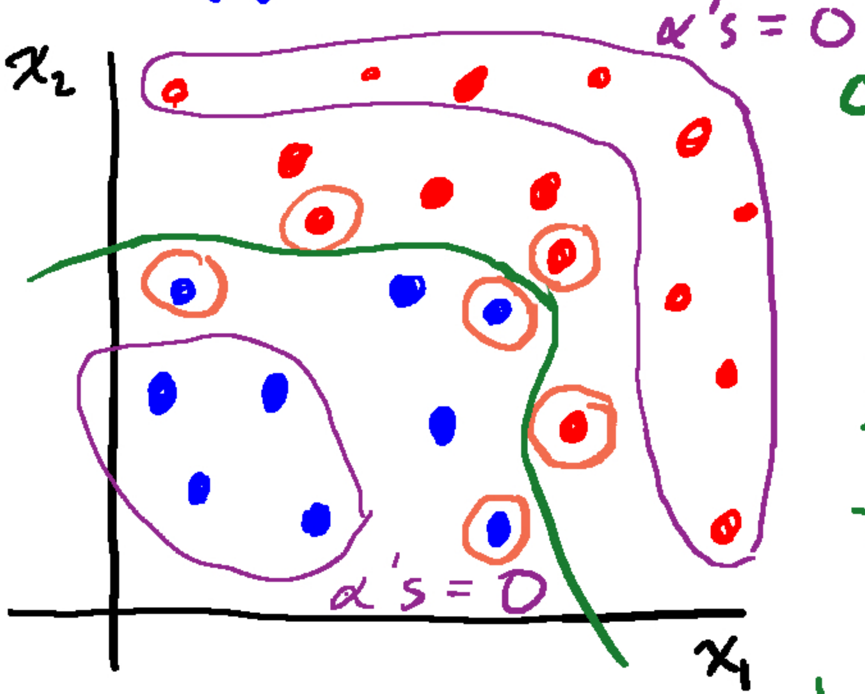
$$\min_{\underline{\alpha}} \sum_{i=1}^{N} \Big(1 - d^i \underline{\phi}^T(\underline{x}^i) \underbrace{\sum_{j=1}^{N} \alpha_j \underline{\phi}(\underline{x}^j)}_{\underline{w}}\Big)_+ + \lambda \underbrace{\sum_{i=1}^{N} \alpha_i \underline{\phi}^T(\underline{x}^i)}_{\underline{w}^T} \underbrace{\sum_{j=1}^{N} \alpha_j \underline{\phi}(\underline{x}^j)}_{\underline{w}}$$

$$\min_{\underline{\alpha}} \sum_{i=1}^{N} \Big(1 - d^i \sum_{j=1}^{N} \alpha_j \underbrace{\underline{\phi}^T(\underline{x}^i)\underline{\phi}(\underline{x}^j)}_{K(\underline{x}^i, \underline{x}^j)}\Big)_+ + \lambda \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \underbrace{\underline{\phi}^T(\underline{x}^i)\underline{\phi}(\underline{x}^j)}_{K(\underline{x}^i, \underline{x}^j)}$$

Kernel "trick"

SVM $$\min_{\underline{\alpha}} \sum_{i=1}^{N} \Big(1 - d^i \sum_{j=1}^{N} \alpha_j K(\underline{x}^i, \underline{x}^j)\Big)_+ + \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K(\underline{x}^i, \underline{x}^j)$$

Support vector machines have sparse $\underline{\alpha}$



decision boundary

$$d(\underline{x}) = 0 = \underline{\phi}^T(\underline{x})\underline{w} = \sum_{j=1}^{N} \alpha_j K(\underline{x}, \underline{x}^j)$$

Boundary (hinge loss) depends only on the support vectors

$K(\underline{u}, \underline{v})$ measures similarity/alignment of $\underline{u}, \underline{v}$

Ex: Gaussian kernels

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

Solve for $\underline{\alpha}$ using gradient descent