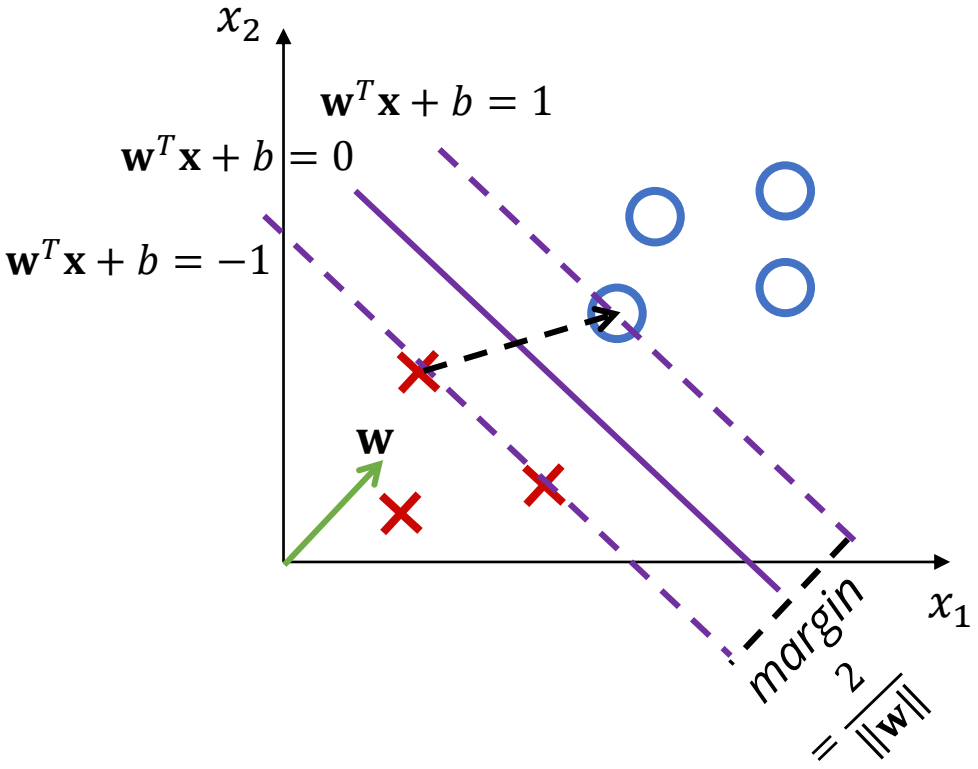
CSE 575
Statistical Machine Learning

Lecture 10
YooJung Choi
Fall 2022

Recap: Maximum margin classifier



- Binary class: $t \in \{-1, +1\}$
- Training data is linearly separable
- Classification function:

$$\begin{cases} +1 & \text{if } \mathbf{w}^T\mathbf{x}+b \geq 0 \\ -1 & \text{if } \mathbf{w}^T\mathbf{x}+b < 0 \end{cases}$$

- Maximum margin classifier:

$$\text{argmax}_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|} \quad \text{s.t. } t_n(\mathbf{w}^T\mathbf{x}_n+b) \geq 1 \quad \forall n$$

$$\text{argmin}_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t. } t_n(\mathbf{w}^T\mathbf{x}_n+b)-1 \geq 0 \quad \forall n$$

Maximum margin: dual formulation

From constrained optimization…

$$\text{argmin}_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t. } t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0 \quad \forall n$$

…to “unconstrained” optimization, using Lagrange multipliers

$$\text{argmax}_{\mathbf{a}} \ \min_{\mathbf{w},b} \underbrace{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n[t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1]}_{L(\mathbf{w},b,\mathbf{a})} \quad \text{s.t. } a_n \geq 0 \quad \forall n$$

Maximum margin: dual formulation

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n[t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1]$$

Differentiate and set to zero:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} a_n t_n \mathbf{x}_n = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} a_n t_n = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} a_n t_n = 0$$

Using $\frac{\partial(\mathbf{w}^T\mathbf{w})}{\partial \mathbf{w}} = 2\mathbf{w}$, $\frac{\partial(\mathbf{w}^T\mathbf{x})}{\partial \mathbf{w}} = \mathbf{x}$

Maximum margin: dual formulation

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n [t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1]$$

$$= \frac{1}{2}\left(\sum_{i=1}^{N} a_i t_i \mathbf{x}_i\right)^T \left(\sum_{j=1}^{N} a_j t_j \mathbf{x}_j\right) - \sum_{j=1}^{N} a_j t_j \left(\sum_{i=1}^{N} a_i t_i \mathbf{x}_i\right)^T \mathbf{x}_j - \left(\sum_{n=1}^{N} a_n t_n\right) b + \sum_{n=1}^{N} a_n$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j t_i t_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{n=1}^{N} a_n$$

$$\text{argmax}_{\boldsymbol{a}} \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j t_i t_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{s.t.} \quad a_n \geq 0 \quad \forall n; \quad \sum_{n=1}^{N} a_n t_n = 0$$

Important: optimal classifier depends only on the training examples only through the *inner products* $\{\mathbf{x}_i^T \mathbf{x}_j\}_{i,j}$

Support vectors

$$\text{argmin}_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{s.t. } t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0 \quad \forall n$$

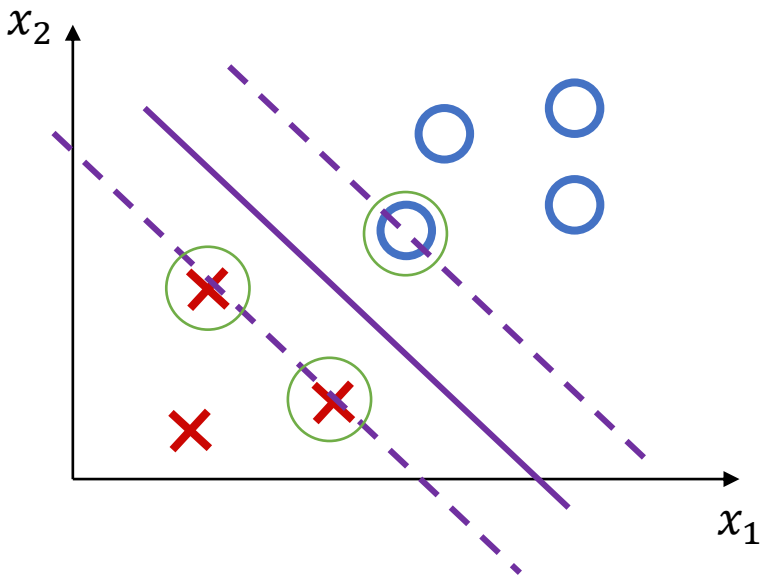
$$\text{argmax}_{\boldsymbol{a}} \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j t_i t_j \mathbf{x}_i^T \mathbf{x}_j$$
$$\text{s.t. } a_n \geq 0 \quad \forall n; \sum_{n=1}^{N} a_n t_n = 0$$

- Property: satisfies the following

$$a_n[t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1] = 0 \quad \forall n$$

- i.e., if $t_n(\mathbf{w}^T\mathbf{x}_n + b) > 1$, then $a_n = 0$
- Also, if $a_n > 0$, then $t_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$ *Support vectors*
- Thus, $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n = \sum_{m\in\mathcal{S}} a_m t_m \mathbf{x}_m$

  a linear combination of support vectors



Support vectors

- Given the optimal $\mathbf{a}$, classify a new example $\mathbf{x}$ as +1 if:

$$\mathbf{w}^T\mathbf{x} + b = \sum_{m\in\mathcal{S}} a_m t_m \mathbf{x}_m^T \mathbf{x} + b \geq 0$$

- We can solve for b using the fact that for any support vector $\mathbf{x}_n$,

$$t_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \quad \Rightarrow \quad b = t_n - \sum_{m\in\mathcal{S}} a_m t_m \mathbf{x}_m^T \mathbf{x}_n$$

- Or by averaging over all support vectors:

$$b = \frac{1}{|\mathcal{S}|}\sum_{n\in\mathcal{S}}\left(t_n - \sum_{m\in\mathcal{S}} a_m t_m \mathbf{x}_m^T \mathbf{x}_n\right)$$

Kernel trick

- Putting everything together, the SVM classifier is

$$y(\mathbf{x}) = \begin{cases} +1 & \text{if } \sum_{m \in \mathcal{S}} a_m t_m \mathbf{x}_m^T \mathbf{x} + b \geq 0 \\ -1 & \text{otherwise} \end{cases} \text{ where } b = \frac{1}{|\mathcal{S}|} \sum_{n \in \mathcal{S}} (t_n - \sum_{m \in \mathcal{S}} a_m t_m \mathbf{x}_m^T \mathbf{x}_n)$$

- We can again use feature mappings $\boldsymbol{\phi}(\mathbf{x})$ to get non-linear decision boundaries
- *Only the inner products are relevant*: let $k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\phi}(\mathbf{x}_n)^T \boldsymbol{\phi}(\mathbf{x}_m)$

$$y(\mathbf{x}) = \begin{cases} +1 & \text{if } \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_m, \mathbf{x}) + b \geq 0 \\ -1 & \text{otherwise} \end{cases} \text{ where } b = \frac{1}{|\mathcal{S}|} \sum_{n \in \mathcal{S}} (t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m))$$

Maximum margin classifier + kernel trick => support vector machines

Example kernels

- Some feature mappings allow *efficient evaluation of the kernel* $k(\mathbf{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z})$ without explicitly constructing the features $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{z})$
- *Linear kernel:* using $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, $k(\mathbf{x}, \boldsymbol{z}) = \mathbf{x}^T \mathbf{z}$
- *Quadratic kernel:* $k(\mathbf{x}, \boldsymbol{z}) = (\mathbf{x}^T \mathbf{z})^2$

d-dim $\qquad\qquad$ d^2-dim

Suppose $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ gets mapped to $\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_3 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \end{bmatrix}$

Then $k(\mathbf{x}, \boldsymbol{z}) = (\mathbf{x}^T \mathbf{z})^2$

Proof:

$(\mathbf{x}^T \mathbf{z})^2 = \left(\sum_i x_i z_i\right)\left(\sum_j x_j z_j\right)$

$= \sum_i \sum_j x_i z_i x_j z_j$

$= \sum_{i,j} (x_i x_j)(z_i z_j) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z})$

Example kernels

- Some feature mappings allow *efficient evaluation of the kernel* $k(\mathbf{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z})$ without explicitly constructing the features $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{z})$
- *Linear kernel:* using $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, $k(\mathbf{x}, \boldsymbol{z}) = \mathbf{x}^T \mathbf{z}$
- *Quadratic kernel:* $k(\mathbf{x}, \boldsymbol{z}) = (\mathbf{x}^T \mathbf{z} + c)^2$

  Takes only $O(d)$ time to compute, whereas constructing $\boldsymbol{\phi}(\mathbf{x})$ takes $O(d^2)$ time

Corresponds to $\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_3 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \\ \sqrt{2c} x_1 \\ \sqrt{2c} x_2 \\ \sqrt{2c} x_3 \\ c \end{bmatrix}$

Example kernels

- Some feature mappings allow *efficient evaluation of the kernel* $k(\mathbf{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z})$ without explicitly constructing the features $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{z})$
- *Linear kernel:* using $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, $k(\mathbf{x}, \boldsymbol{z}) = \mathbf{x}^T \mathbf{z}$
- *Quadratic kernel:* $k(\mathbf{x}, \boldsymbol{z}) = (\mathbf{x}^T \mathbf{z} + c)^2$
- *Polynomial kernel:* $k(\mathbf{x}, \boldsymbol{z}) = (\mathbf{x}^T \mathbf{z} + c)^m$

$\boldsymbol{\phi}(\mathbf{x})$ contains monomials of degree $\leq m$

e.g. for $m = 5$, $\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \alpha_1 x_1^5 \\ \alpha_2 x_1^4 x_2 \\ \vdots \\ \alpha_k x_1 x_2 x_3 \\ \vdots \end{bmatrix}$

$\binom{d+m}{m}$-dimensional

Example kernels

- Some feature mappings allow *efficient evaluation of the kernel* $k(\mathbf{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z})$ without explicitly constructing the features $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{z})$
- *Linear kernel:* using $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, $k(\mathbf{x}, \boldsymbol{z}) = \mathbf{x}^T \mathbf{z}$
- *Quadratic kernel:* $k(\mathbf{x}, \boldsymbol{z}) = (\mathbf{x}^T \mathbf{z} + c)^2$
- *Polynomial kernel:* $k(\mathbf{x}, \boldsymbol{z}) = (\mathbf{x}^T \mathbf{z} + c)^m$
- *Gaussian (Radial Basis) kernel:* $k(\mathbf{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\mathbf{x}-\boldsymbol{z}\|^2}{2\sigma^2}\right)$

Corresponding $\boldsymbol{\phi}(\mathbf{x})$ is ∞-dimensional!

contains monomials of arbitrary degree

Valid kernels

- How do we construct kernels / check whether it is valid?
- Approach 1: choose a feature mapping then derive the corresponding kernel
- Approach 2: directly construct kernel functions

$k(\mathbf{x}, \boldsymbol{z})$ is a valid kernel $\Leftrightarrow$

For any set of data points $\{\mathbf{x}_n\}_{n=1,\ldots,N}$,

Let $\mathbf{K}$ be an $N \times N$ matrix s.t. $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Then $\mathbf{K}$ is *positive semidefinite:*

For any $\boldsymbol{z}$, $\boldsymbol{z}^T \mathbf{K} \mathbf{z} \geq 0$

If $k(\mathbf{x}, \boldsymbol{z})$ is a valid kernel, there exists a feature mapping $\boldsymbol{\phi}$ such that $k(\mathbf{x}, \mathbf{z}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z})$

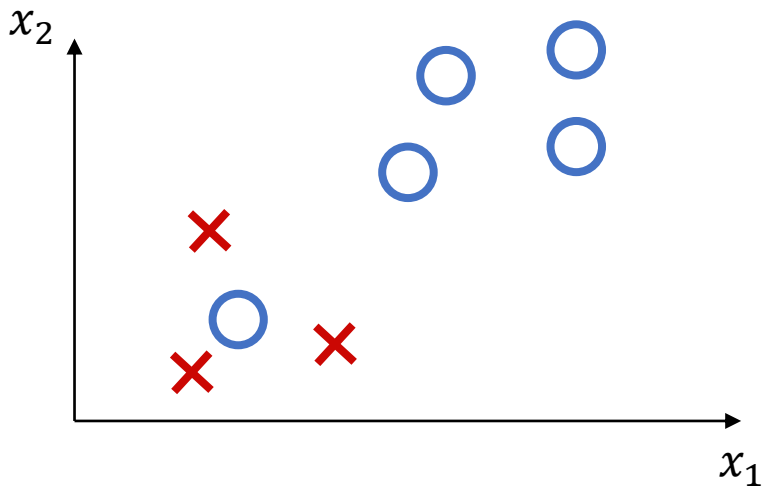
Given a set of data points $\{\mathbf{x}_n\}_{n=1,\ldots,N}$, let $\mathbf{K}$ be an $N \times N$ matrix s.t. $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Then, for any N-dim vector $\boldsymbol{z}$, the following holds:

$$\boldsymbol{z}^T \mathbf{K} \mathbf{z} = \sum_{i=1}^{N} \sum_{j=1}^{N} z_i \mathbf{K}_{ij} z_j = \sum_{i=1}^{N} \sum_{j=1}^{N} z_i k(\mathbf{x}_i, \mathbf{x}_j) z_j = \sum_{i=1}^{N} \sum_{j=1}^{N} z_i \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j) z_j$$

$$= \left(\sum_{i=1}^{N} z_i \boldsymbol{\phi}(\mathbf{x}_i) \right)^T \left(\sum_{j=1}^{N} z_j \boldsymbol{\phi}(\mathbf{x}_j) \right) = \left\| \sum_{i=1}^{N} z_i \boldsymbol{\phi}(\mathbf{x}_i) \right\|^2 \geq 0$$

Non-separable case

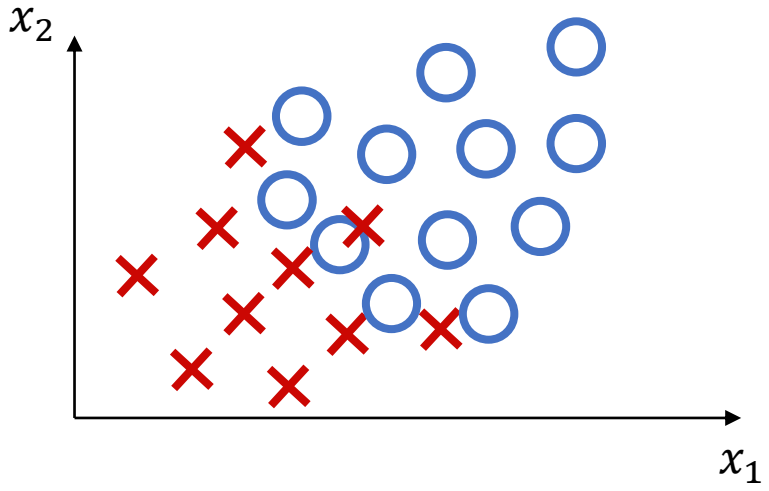


- What happens if the data is not linearly separable?

$$\operatorname{argmin}_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{s.t. } t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0 \quad \forall n$$

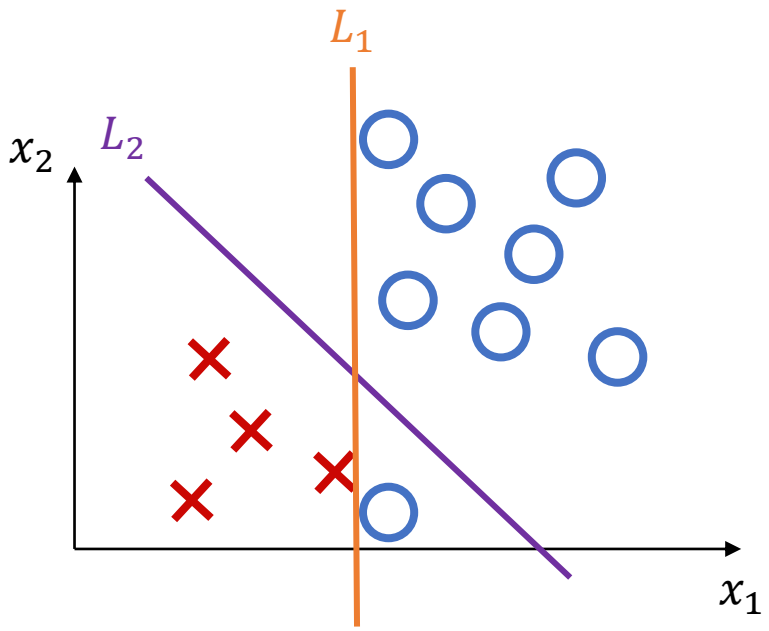
- Impossible to satisfy the constraint: by convention, *infinite error*
- We could use higher dimensional kernels to separate using a non-linear decision boundary

Overlapping distributions



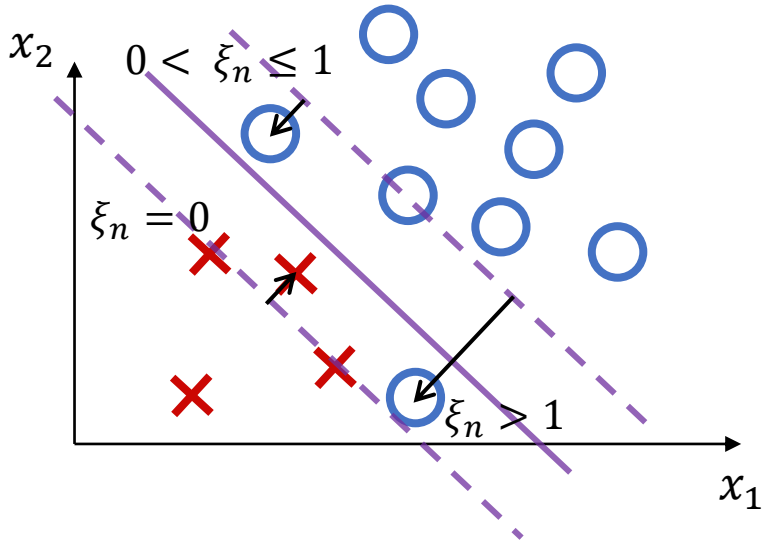
- If class-conditional distributions overlap, forcing a perfect separation may lead to *poor generalization*
- Need a different objective function to still learn a reasonable classifier that is *allowed to make mistakes*

Outliers



- You may prefer a classifier that is allowed to make mistakes, *even when data is linearly separable*
- L_1: linear separation but very narrow margin
- L_2: one mistake but otherwise a large margin
- Can we trade off some small mistakes for a large margin?

Soft-margin SVM



- Introducing slack variables $\xi_n \geq 0$:
  - $\xi_n = 0$: $\mathbf{x}_n$ is correctly classified on the right side of margin boundary
  - $0 < \xi_n \leq 1$: $\mathbf{x}_n$ is correctly classified but on the wrong side of margin boundary (*margin violation)*
  - $\xi_n > 1$: $\mathbf{x}_n$ is misclassified
- Intuition: we want the total slack to be small

Midterm logistics

- Written exam, Wednesday 10/5, in-class
- Closed book. You may bring a single-page letter-sized cheat sheet (with your name on it).
- Topics:
  - Probability basics
  - MLE vs MAP parameters
  - Discrete / continuous Bayes classifier
  - Naïve Bayes classifier
  - Brief questions about regression and classification algorithms