COMS 4721: Machine Learning for Data Science

Lecture 11, 2/23/2017

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

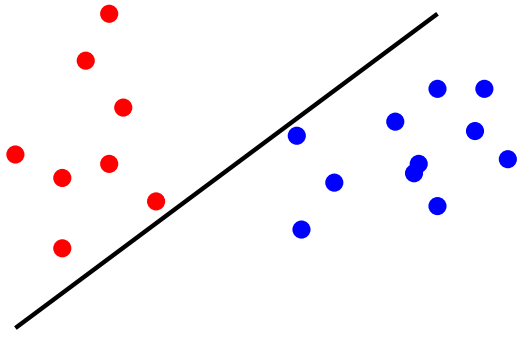
Columbia University

Maximum Margin Classifiers

Maximum margin idea

Setting

Linear classification, two linearly separable classes.



Recall Perceptron

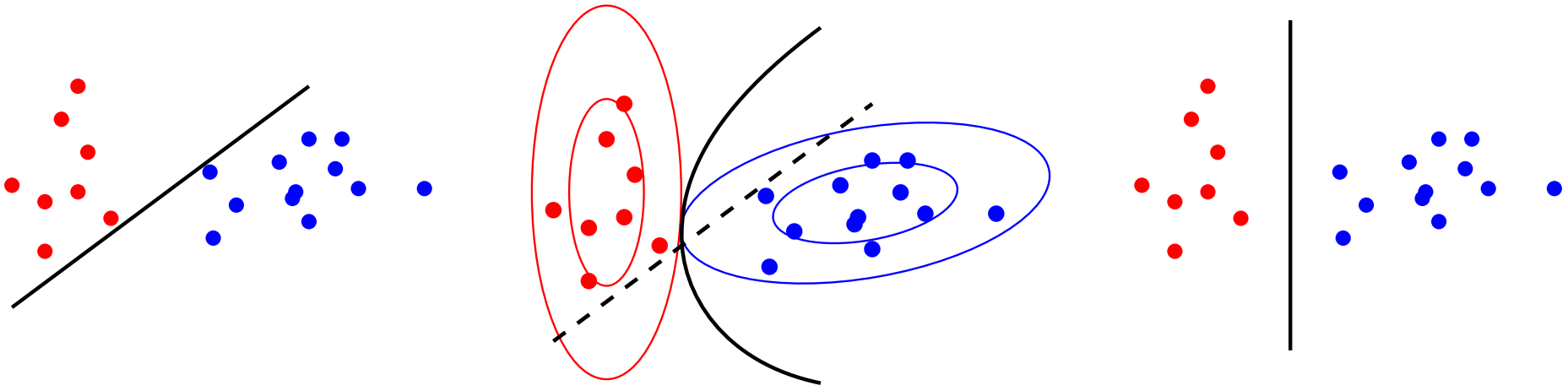
- Selects *some* hyperplane separating the classes.
- Selected hyperplane depends on several factors.

Maximum margin

To achieve good generalization (low prediction error), place the hyperplane "in the middle" between the two classes.

More precisely, choose a plane such that its distance to the closest point in each class is maximized. This distance is called the **margin**.

Generalization Error



Possible Perceptron solution

(dotted) poor generalization, (solid) better

Maximum margin solution

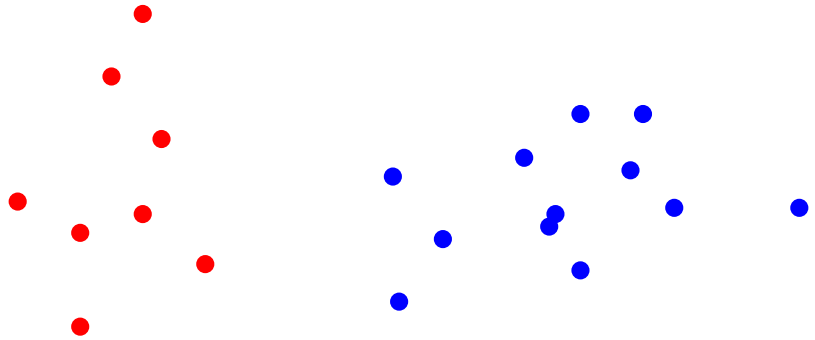
Example: Gaussian data

- Intuitively, the classifier on the left isn't good because sampling more data could lead to misclassifications.
- If we imagine the data from each class as Gaussian, we could frame the goal as to find a decision boundary that cuts into as little probability mass as possible.
- With no distribution assumptions, we can argue that max-margin is best.

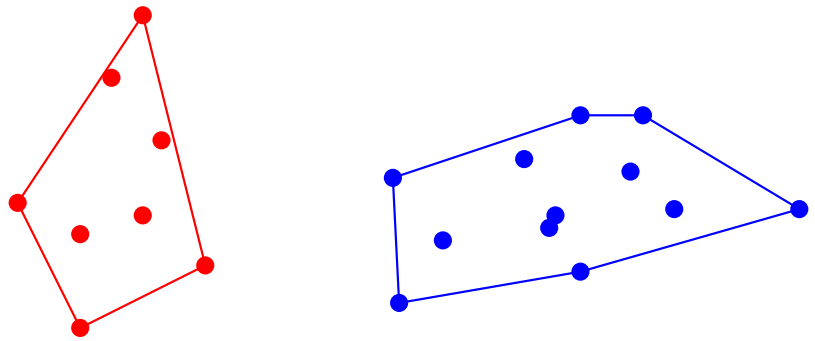
SUBSTITUTING CONVEX SETS

Observation

Where a separating hyperplane may be placed depends on the "outer" points of the sets. Points in the center do not matter.

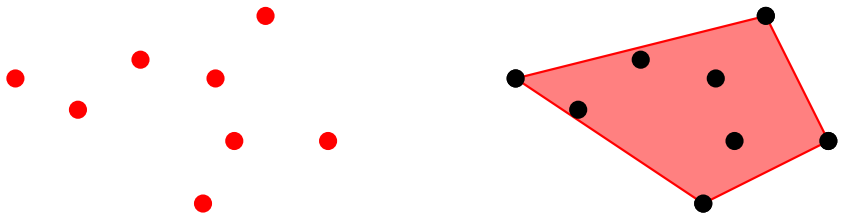


In geometric terms, we can represent each class by the smallest convex set which contains all point in the class. This is called a *convex hull*.



Convex hulls

A convex hull is defined by all possible weighted averages of points in a set.



That is, let $x_1, \ldots, x_n$ be the above data coordinates. Every point x_0 in the shaded region – i.e., the convex hull – can be reached by setting

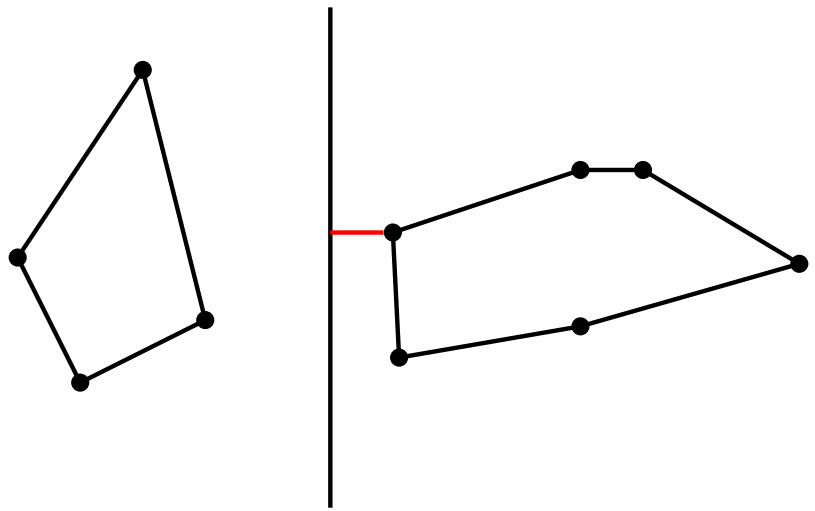
$$x_0 = \sum_{i=1}^{n} \alpha_i x_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i = 1,$$

for some $(\alpha_1, \ldots, \alpha_n)$. No point outside this region can be reached this way.

Margin

Definition

The *margin* of a classifying hyperplane H is the shortest distance between the plane and any point in either set (equivalently, in the convex hull)



When we maximize this margin, H is "exactly in the middle" of the two convex hulls. Of course, the difficult part is how do we find this H?

SUPPORT VECTOR MACHINES

Support Vector Machine

Finding the hyperplane

For n linearly separable points $(x_1, y_1), \ldots, (x_n, y_n)$ with $y_i \in \{\pm 1\}$, solve:

$$\begin{aligned} \min_{w, w_0} \quad & \tfrac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i(x_i^T w + w_0) \geq 1 \quad \text{for } i = 1, \ldots, n \end{aligned}$$

Comments

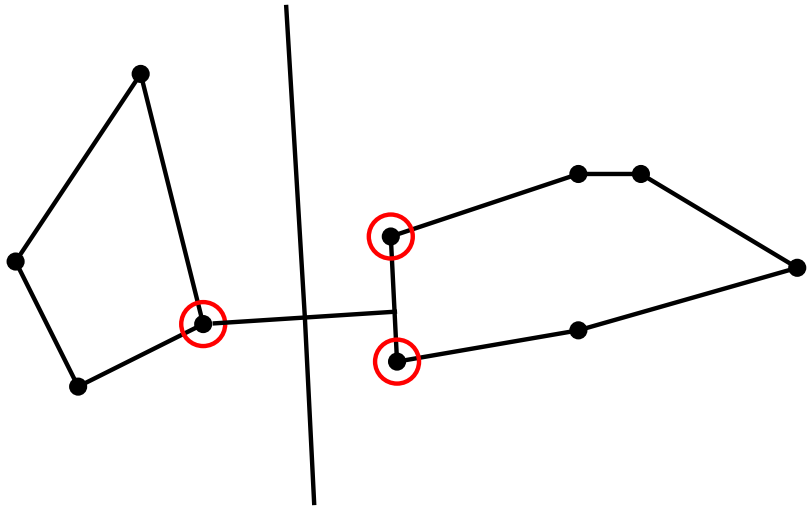
- Recall that $y_i(x_i^T w + w_0) > 0$ if $y_i = \text{sign}(x_i^T w + w_0)$.
- If there exists a hyperplane H that separates the classes, we can scale w so that $y_i(x_i^T w + w_0) > 1$ for all i.
- The resulting classifier is called a *support vector machine*. This formulation only has a solution when the classes are linearly separable.
- It is not at all obvious why this maximizes the margin. This will become more clear when we look at the solution.

SUPPORT VECTOR MACHINE

Skip to the end

Q: First, can we intuitively say what the solution should *look* like?

A: Yes, but we won't give the proof.



1. Find the closest two points *within the convex hulls* of classes $+1$ and -1.

2. Connect them with a line and put a perpendicular hyperplane in the middle.

3. If S_1 and S_0 are the sets of x in class $+1$ and -1 respectively, we're looking for two *probability* vectors α_1 and α_0 such that we minimize

$$\Big\| \underbrace{\left(\sum_{x_i \in S_1} \alpha_{1i} x_i\right)}_{\text{in conv. hull of } S_1} - \underbrace{\left(\sum_{x_i \in S_0} \alpha_{0i} x_i\right)}_{\text{in conv. hull of } S_0} \Big\|_2$$

4. Then we define the hyperplane using the two points found with α_1 and α_0.

Primal problem

The *primal* optimization problem is the one we defined:

$$\begin{aligned} \min_{w,w_0} \quad & \tfrac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i(x_i^T w + w_0) \geq 1 \quad \text{for } i = 1, \ldots, n \end{aligned}$$

This is tricky, so we use *Lagrange multipliers* to set up the "dual" problem.

Lagrange multipliers

Define Lagrange multipliers $\alpha_i > 0$ for $i = 1, \ldots, n$. The Lagrangian is

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(x_i^T w + w_0) - 1) \\ &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y_i (x_i^T w + w_0) + \sum_{i=1}^{n} \alpha_i \end{aligned}$$

We want to minimize $\mathcal{L}$ over w and w_0 and maximize over $(\alpha_1, \ldots, \alpha_n)$.

SETTING UP THE DUAL PROBLEM

First minimize over w and w_0:

$$\mathcal{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y_i (x_i^T w + w_0) + \sum_{i=1}^{n} \alpha_i$$

$$\Downarrow$$

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \qquad \Rightarrow \qquad w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \Rightarrow \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Therefore,

1. We can plug the solution for w back into the problem.
2. We know that $(\alpha_1, \ldots, \alpha_n)$ must satisfy $\sum_{i=1}^{n} \alpha_i y_i = 0$.

SVM DUAL PROBLEM

Lagrangian: $\mathcal{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^n \alpha_i y_i (x_i^T w + w_0) + \sum_{i=1}^n \alpha_i$

Dual problem

Plugging these values in from the previous slide, we get the dual problem

$$\max_{\alpha_1,\ldots,\alpha_n} \quad \mathcal{L} = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i^T x_j)$$

$$\text{subject to} \quad \sum_{i=1}^n \alpha_i y_i = 0, \qquad \alpha_i \geq 0 \quad \text{for } i = 1, \ldots, n$$

Comments

- Where did w_0 go? The condition $\sum_{i=1}^n \alpha_i y_i = 0$ gives $0 \cdot w_0$ in the dual.
- We now maximize over the α_i. This requires an algorithm that we won't discuss in class. Many good software implementations are available.

Solving the primal problem

Before discussing the solution of the dual, we ask:

After finding each α_i how do we predict a new $y_0 = sign(x_0^T w + w_0)$?

We have: $\mathcal{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i (x_i^T w + w_0) - 1)$

With conditions: $\alpha_i \geq 0, \qquad y_i (x_i^T w + w_0) - 1 \geq 0$

Solve for w.

$$\nabla_w \mathcal{L} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad \text{(just plug in the learned } \alpha_i\text{'s)}$$

What about w_0?

- We can show that at the solution, $\alpha_i (y_i (x_i^T w + w_0) - 1) = 0$ for all i.
- Therefore, pick i for which $\alpha_i > 0$ and solve $y_i (x_i^T w + w_0) - 1 = 0$ for w_0 using the solution for w (all possible i will give the same solution).

Dual problem

We can manipulate the dual problem to find out what it's trying to do.

$$\max_{\alpha_1,\ldots,\alpha_n} \quad \mathcal{L} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)$$

$$\text{subject to} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \qquad \alpha_i \geq 0 \quad \text{for } i = 1, \ldots, n$$

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

- $\sum_{i=1}^{n} \alpha_i y_i = 0 \;\Rightarrow\; C = \sum_{i \in S_1} \alpha_i = \sum_{j \in S_0} \alpha_j$
- $\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) = \left\| \sum_{i=1}^{n} \alpha_i y_i x_i \right\|^2 = C^2 \left\| \sum_{i \in S_1} \frac{\alpha_i}{C} x_i - \sum_{j \in S_0} \frac{\alpha_j}{C} x_j \right\|^2$

Dual problem

We can change notation to write the dual as

$$\begin{aligned} \max_{\alpha_1,\ldots,\alpha_n} \quad & \mathcal{L} = 2C - \frac{1}{2}C^2 \left\| \sum_{i\in S_1} \frac{\alpha_i}{C} x_i - \sum_{j\in S_0} \frac{\alpha_j}{C} x_j \right\|^2 \\ \text{subject to} \quad & C := \sum_{i\in S_1} \alpha_i = \sum_{j\in S_0} \alpha_j, \quad \alpha_i \geq 0 \end{aligned}$$

We observe that the maximum of this function satisfies

$$\min_{\alpha_1,\ldots,\alpha_n} \left\| \underbrace{\left(\textstyle\sum_{i\in S_1} \frac{\alpha_i}{C} x_i \right)}_{\text{in conv. hull of } S_1} - \underbrace{\left(\textstyle\sum_{j\in S_0} \frac{\alpha_j}{C} x_j \right)}_{\text{in conv. hull of } S_0} \right\|^2$$

Therefore, the dual problem is trying to find the closest points in the convex hulls constructed from data in class $+1$ and -1.

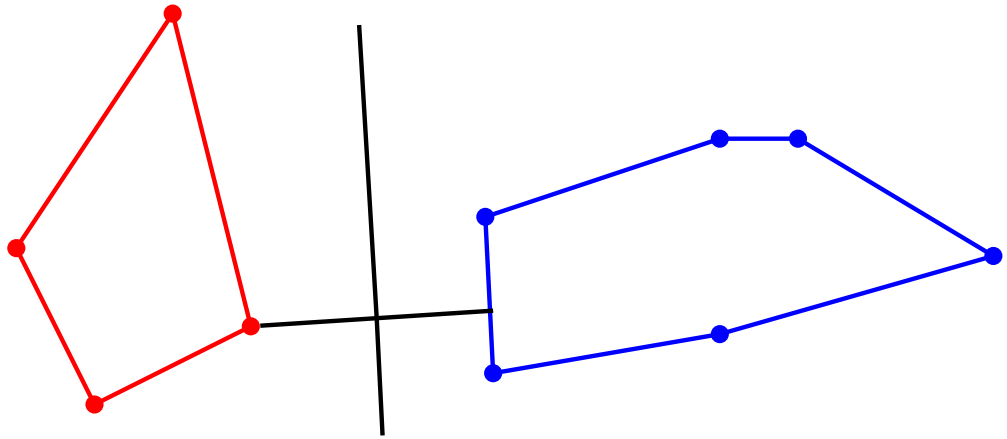
RETURNING TO THE PICTURE

Recall

We wanted to find:

$$\min_{\substack{u \in \mathcal{H}(S_1) \\ v \in \mathcal{H}(S_0)}} \|u - v\|^2$$

The direction of w is $u - v$.



We previously claimed we can find the max-margin hyperplane as follows:

1. Find shortest line connecting the convex hulls.
2. Place hyperplane orthogonal to line and exactly at the midpoint.

With the SVM we want to minimize $\|w\|^2$ and we can write this solution as

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i = C \left(\sum_{i \in S_1} \frac{\alpha_i}{C} x_i - \sum_{j \in S_0} \frac{\alpha_j}{C} x_j \right)$$

SOFT-MARGIN SVM

Question: What if the data isn't linearly separable?

Answer: Permit training data be on wrong side of hyperplane, but at a cost.

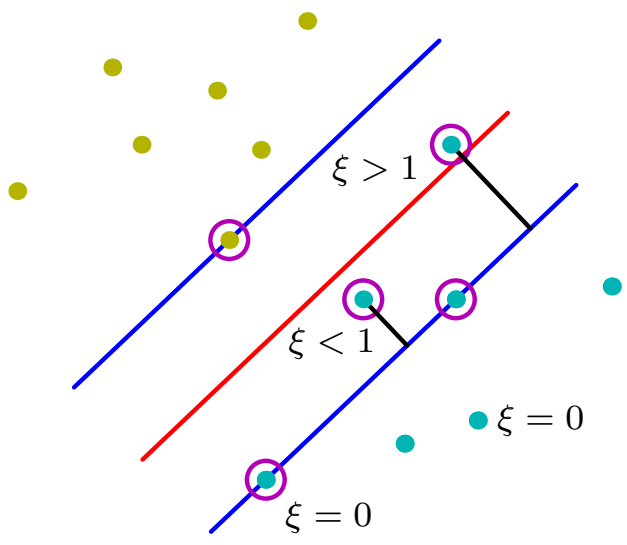
Slack variables

Replace the training rule $y_i(x_i^T w + w_0) \geq 1$ with

$$y_i(x_i^T w + w_0) \geq 1 - \xi_i,$$

with $\xi_i \geq 0$.

The ξ_i are called *slack variables*.



Soft-margin objective function

Adding the slack variables gives a new objective to optimize

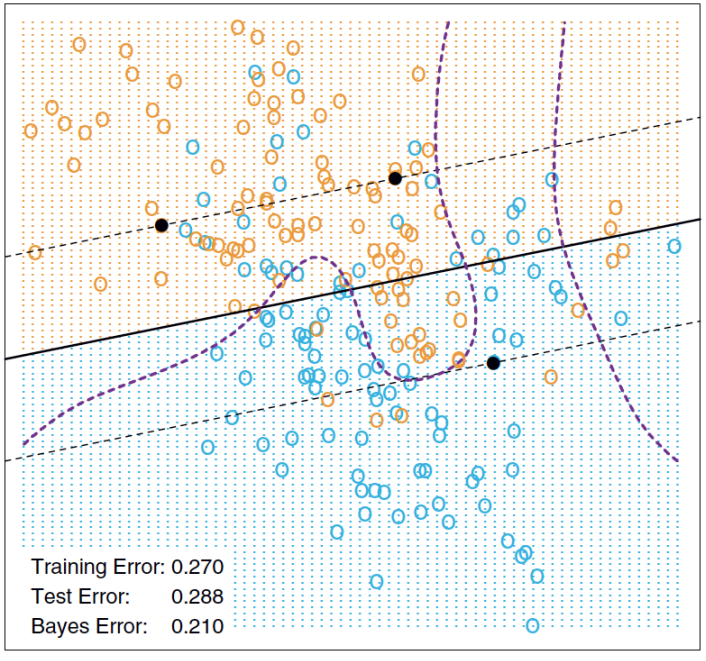
$$\begin{aligned} \min_{w, w_0, \xi_1, \ldots, \xi_n} \quad & \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(x_i^T w + w_0) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, n \\ & \xi_i \geq 0 \quad \text{for} \quad i = 1, \ldots, n \end{aligned}$$

We also have to choose the parameter $\lambda > 0$. We solve the dual as before.

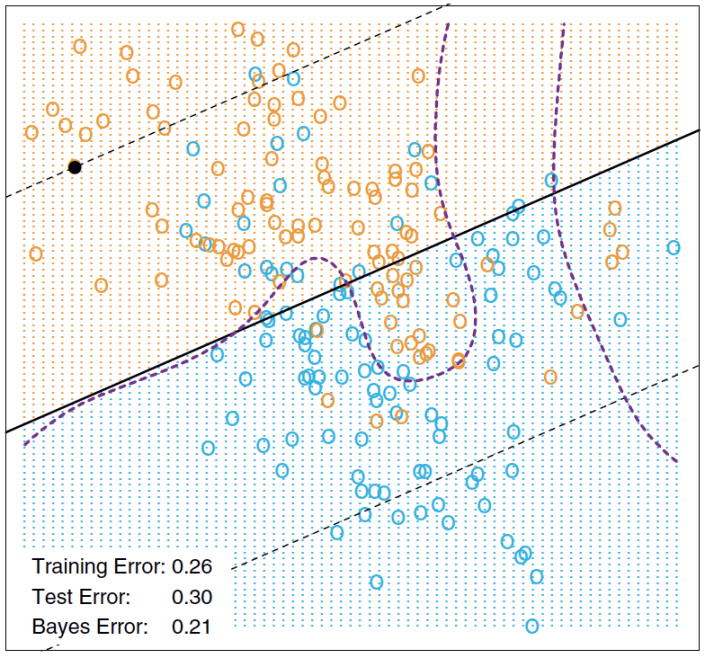
Role of λ

- Specifies the "cost" of allowing a point on the wrong side.
- If λ is very small, we're happy to misclassify.
- For $\lambda \to \infty$, we recover the original SVM because we want $\xi_i = 0$.
- We can use cross-validation to choose it.

Influence of Margin Parameter



$\lambda = 100000$



$\lambda = 0.01$

Hyperplane is sensitive to λ. Either way, a linear classifier isn't ideal . . .

Primal problem with slack variables

Let's map the data into higher dimensions using the function $\phi(x_i)$,

$$\begin{aligned} \min_{w,w_0,\xi_1,\ldots,\xi_n} \quad & \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\phi(x_i)^T w + w_0) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, n \\ & \xi_i \geq 0 \quad \text{for} \quad i = 1, \ldots, n \end{aligned}$$

Dual problem

Maximize over each (α_i, μ_i) and minimize over $w, w_0, \xi_1, \ldots, \xi_n$

$$\mathcal{L} = \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i(\phi(x_i)^T w + w_0) - 1 + \xi_i) - \sum_{i=1}^{n} \mu_i \xi_i$$

subject to $\alpha_i \geq 0, \quad \mu_i \geq 0, \quad y_i(\phi(x_i)^T w + w_0) - 1 + \xi_i \geq 0$

KERNELIZING THE SVM

Dual problem

Minimizing for w, w_0 and each ξ_i, we find

$$w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i), \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \lambda - \alpha_i - \mu_i = 0$$

If we plug w and $\mu_i = \lambda - \alpha_i$ back into the $\mathcal{L}$, we have the dual problem

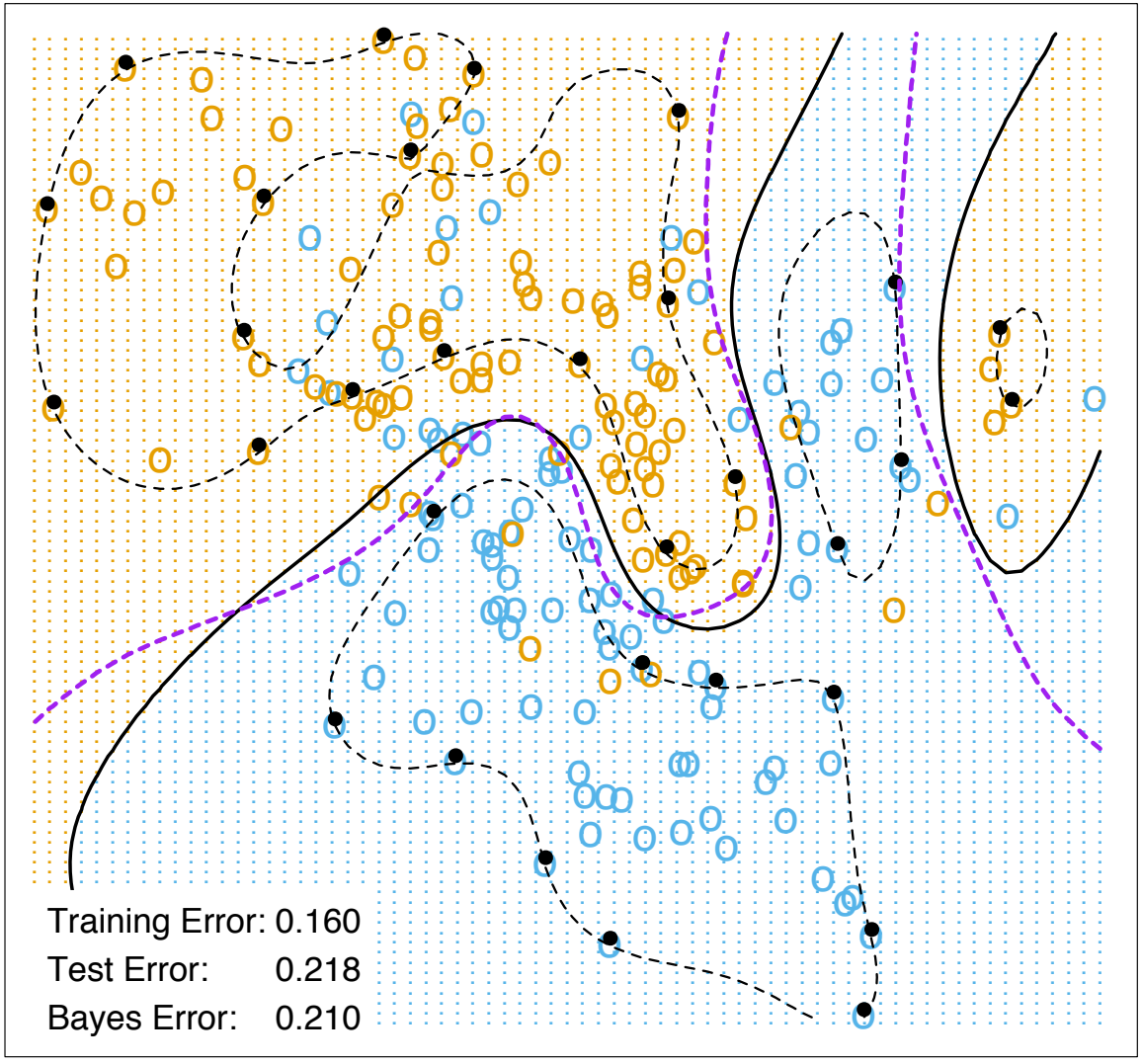
$$\max_{\alpha_1,\ldots,\alpha_n} \quad \mathcal{L} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underbrace{\phi(x_i)^T \phi(x_j)}_{\boldsymbol{K(x_i, x_j)}}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq \lambda$$

Classification: Using the solution $w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i)$, declare

$$y_0 = \text{sign}\Big(\sum_{i=1}^{n} \alpha_i y_i \phi(x_0)^T \phi(x_i) + w_0 \Big) = \text{sign}\Big(\sum_{i=1}^{n} \alpha_i y_i K(x_0, x_i) + w_0 \Big)$$

KERNELIZING THE SVM



Black solid line

SVM decision boundary

Classification rule

$$\text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x_0, x_i) + w_0\right)$$

Dots

Support vectors ($\alpha_i > 0$)

Purple line

A Bayes classifier.

Summary: Support Vector Machine

Basic SVM

- Linear classifier for linearly separable data.
- Position of affine hyperplane is determined to maximize the margin.
- The dual is a convex, so we can find exact solution with optimization.

Full-fledged SVM

Ingredient	Purpose
Maximum margin	Good generalization properties
Slack variables	Overlapping classes, robust against outliers
Kernel	Nonlinear decision boundary

Use in practice

- Software packages (many options)
- Choose a kernel function (e.g., RBF)
- Cross-validate λ parameter and RBF kernel width parameter