COMS 4721: Machine Learning for Data Science
Lecture 11, 2/23/2017

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

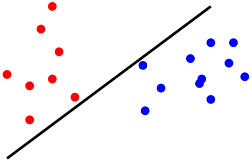
Columbia University

MAXIMUM MARGIN CLASSIFIERS

MAXIMUM MARGIN IDEA

Setting

Linear classification, two linearly separable classes.



Recall Perceptron

will find the 1st hyperplane that it comes across and then terminate.

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

doesn't really give any sort weight to 1 hyperplane over the other.

Maximum margin

not all hyperplanes are equally good.

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

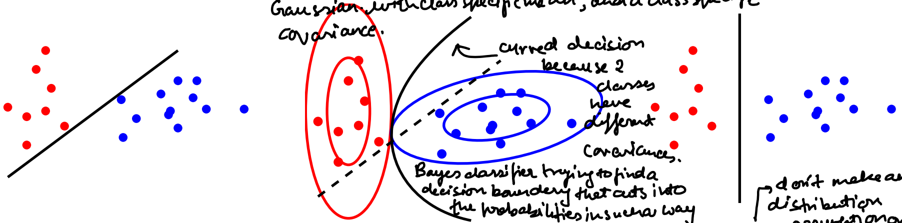
Mathematically

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

Find a hyperplane that separates the 2 classes, but it's as far away as possible from the closest data points in each of these classes.

GENERALIZATION ERROR

If we assumed some distribution on 2 classes, what I'm showing here is a bayes classifier that uses a class specific multi-variate Gaussian with class specific mean, and a class specific covariance.



curved decision because 2 classes have different covariances.

Bayes classifier trying to find a decision boundary that cuts into the probabilities in such a way that the probability of making an error under this distribution assumption is minimized.

don't make any distribution assumption on data.

Possible Perceptron solution | (dotted) poor generalization, (solid) better | Maximum margin solution

Example: Gaussian data

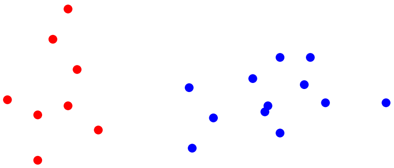
- Intuitively, the classifier on the left isn't good because sampling more data could lead to misclassifications.
- If we imagine the data from each class as Gaussian, we could frame the goal as to find a decision boundary that cuts into as little probability mass as possible.
- With no distribution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

Motivation of max. margin: simply finding the hyperplane that maximizes the distance to the nearest points the best that we can.

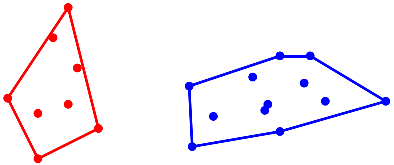
SUBSTITUTING CONVEX SETS

Observation

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



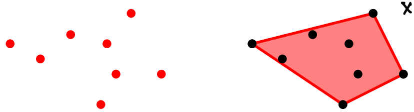
In geometric terms, we can represent each class by the smallest convex set which contains all point in the class. This is called a *convex hull*.



SUBSTITUTING CONVEX SETS

Convex hulls

The thing to remember for this lecture is that a convex hull is defined by all possible weighted averages of points in a set.



x → Any pt outside of this convex hull cannot be reached by taking a weighted avg. of all of the points in the dataset.

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

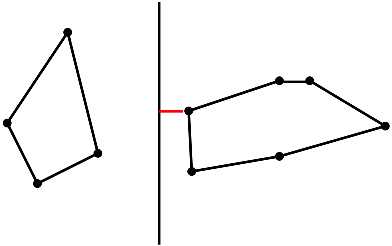
$$x_0 = \sum_{i=1}^{n} \alpha_i x_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i = 1,$$

↳ weighted combination.

for some $\underbrace{(\alpha_1, \ldots, \alpha_n)}_{\text{probability vector.}}$. No point outside this region can be reached this way.

Definition

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

Finding the hyperplane

binary classificatio

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

regression co efficient: that's our classifier that defines the angle of the hyperplane.

defines the shift of the hyperplane

always be positive if we predict correctly.

And for mathematical reasons, we're just gonna say that this is greater than or equal to 1 instead of 0.

$$\min_{w,w_0} \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad y_i(x_i^T w + w_0) \geq 1 \quad \text{for } i = 1, \ldots, n$$

(constraints)

with a linear classifier, we take the sign of this function to be the predicted label of observation x_i. $> 0 \to +1$, $< 0 \to -1$

Comments

why?

- 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 (this is useful later).
- 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.

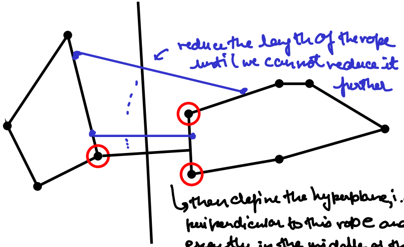
So it isn't clear from this objective function, why finding the minimum l_2 vector w such that we correctly classify all of our data points, returns the max margin hyperplane.

find the closest line that connects two convex hulls.

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.



then define the hyperplane, i.e, perpendicular to this rope and is exactly in the middle of the rope.

1. Find the closest two points from the convex hulls of class $+1$ and -1.

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

All we need to know is the left and the right point. Given those 2 points we have everything that we need to be able to define the max. margin hyperplane.

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 the distance btw these 2 vectors.

length equal to the no. of points in S_1

length equal to the no. of points in S_0

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

distance between some pt. in the convex hull of class 1 and some point in the convex hull of class -1.

a point in the convex hull of -1 class.

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

By trying to minimize this *, we trying to find the points in the respective convex hull that are closest together.

Primal and Dual problems

Primal problem

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

$$\begin{array}{ll} \min\limits_{w,w_0} & \frac{1}{2}\|w\|^2 \\ \text{subject to} & y_i(x_i^T w + w_0) \geq 1 \quad \text{for } i = 1, \ldots, n \end{array}$$

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:

• setup an objective function that incorporates our original constraints by introducing multipliers. We want to minimise that function L over w and maximise it over α.

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

I.) Minimize over w for a particular setting of these α_is, and minimize over w_0. And then plug those solutions back into the original objective.

• we have written the objective function now in this way by multiplying through α.

Minimize over w $\nabla_w \mathcal{L} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w \overset{*}{=} \sum_{i=1}^{n} \alpha_i y_i x_i$

For a particular setting of the Lagrange multiplier $\alpha_1 \ldots \alpha_n$, the $\vec{w}$ minimizes this is equal to this

Similarly,

Derivative with respect to w_0 $\frac{\partial \mathcal{L}}{\partial w_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$

[w_0 is eliminated but we get this additional constraint]

is saying the difference of sum of alphas in class −1, and the sum of alphas in class 1 has to be equal to 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 $\sum_{i=1}^{n} \alpha_i y_i = 0$.

* We're still constructing our hyperplane as just a sum of data points weighted (pre-multiplied) by which class they are +1 or −1. The only difference is going to be how we learn these values α_i.

SVM DUAL PROBLEM

We have taken our primal problem, which is minimizing over some parameters. We've constructed the Lagrangian, which takes into consideration the constraints of the primal problem, adds Lagrange multipliers. Then we minimized the Lagrange over the originial parameters to get the dual problem. So when we minimize over w and plug back in, we obtain the dual problem which is equal to this

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

what about this?

what happens to w?

Dual problem

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

?

Now, have this dual problem, to maximise over α

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

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

or $\alpha_i > 0$?

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.

All that we require to use in order to solve this dual problem are the dot products between our data points. So when we see these dot products here we immediately think that we can replace that with a kernel.

AFTER SOLVING THE DUAL

Solving the primal problem

Before discussing the solution of the dual, we ask:

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

≥ 0

We have: $\mathcal{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(x_i^T w + w_0) \overset{*}{-} 1)$ ≥ 0

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

if non-zero, then $\alpha_i = 0$.

if $\alpha = +ve$, * has to be 0.

So these two terms are conflicting with eachother, they can't both be positive. If one of them is +ve, other is 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).

why?

 istrying to do.

Here's where we can see that maximizing the dual is doing something that we originally said we wanted to do, which finding 2 points in the respective convex hulls of 2 classes, that have minimum distance to each other.

Dual problem

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

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

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

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

y is +1 for these data points

y is -1 for these data points.

- $$\sum_{i=1}^{n} \alpha_i y_i = 0 \;\; \Rightarrow \;\; C = \sum_{i \in S_1} \alpha_i = \sum_{j \in S_0} \alpha_j$$

Rewrite

- $$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) = \Big\| \underbrace{\sum_{i=1}^{n} \alpha_i y_i x_i}_{= w} \Big\|^2 = C^2 \Big\| \sum_{i \in S_1} \frac{\alpha_i}{C} x_i - \sum_{j \in S_0} \frac{\alpha_j}{C} x_j \Big\|^2$$

equal to the magnitude of w squared

C → sum of αs corresponding to one of the classes, because it doesn't matter which class I pick.

Dual problem

We can change notation to write the dual as

equivalently we trying to minimize this term.

$$\max_{\alpha_1,\dots,\alpha_n} \qquad \mathcal{L} = 2C - \frac{1}{2}C^2 \Big\| \sum_{i\in S_1} \frac{\alpha_i}{C} x_i - \sum_{j\in S_0} \frac{\alpha_j}{C} x_j \Big\|^2$$

$$\text{subject to} \qquad C := \sum_{i\in S_1} \alpha_i = \sum_{j\in S_0} \alpha_j, \quad \alpha_i \geq 0$$

We take all points for class 1 and construct a prob. distribution by normalizing them. And then we take all points for class -1 and construct a prob. distribution by normalising them (which is what C is doing.)

We observe that the maximum of this function satisfies

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

Then we trying to minimize the distance 2 points within the respective convex hulls of each of the 2 classes.

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

① what does this prob. distribution represent? Why do weigh points differently?
② why will it sum to 1?

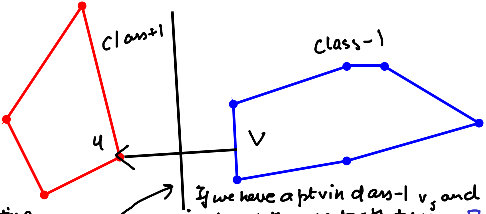
RETURNING TO THE PICTURE

If we could find 2 points, 1 point in each of the convex hulls. Then what does the direction of the classifier defined by this hyperplane look like?

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 $u - v$.

↳ $\perp$ to the line connecting 2 convex hulls.

has this as it's hyperplane.

If we have a pt v in class-1 v, and u in class+1, then vector that defines □ to the +ve region and □ to the -ve region, that the separating hyperplane has got a to be pointing in the direction from v to u.

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.

Therefore, we can construct the $\vec{w}$ by taking the point on the convex hull of +1 class and the point on the convex hull of -1 class and taking their difference.

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

pt. on the convex hull of +1 class.

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

the point on the convex hull of −1 class.

simply scaling it by some no. C. So that's simply streching it out / shrinking it.

corresponds to u

corresponds to v

 (difference- introduction of slack variable)

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

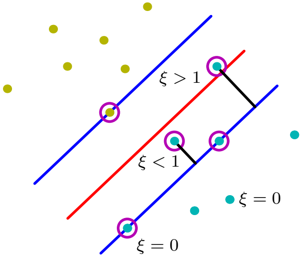
Replace the training rule $y_i(x_i^T w + w_0) \geq 1$ with

For some data points, allow $\xi_i \geq 1$, in which case we allow for that particular point x_i to be misclassified

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

with $\xi_i \geq 0$.

The ξ_i are called *slack variables*.



Soft-Margin SVM

Soft-margin objective function

(building it into the objective function)

Adding the slack variables gives a new objective to optimize

regularization parameter: says how strictly we're going to enforce linear separability

$$\min_{w, w_0, \xi_1, \ldots, \xi_n} \quad \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{n} \xi_i$$

in order to potentially allow the i^{th} data point to be misclassified.

$$\text{subject to} \quad y_i(x_i^T w + w_0) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, n$$

$$\xi_i \geq 0 \quad \text{for} \quad i = 1, \ldots, n$$

we want to minimize the sum of these slack variables, so we don't want to set too many greater than zero.

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

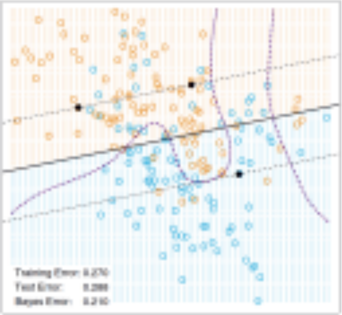
Role of λ

- Specifies the "cost" of allowing a point on the wrong side.
- If λ is very small, we're happy to misclassify. if λ is extremely small, say in the limit as it goes to 0, then we're essentially paying no penalty for setting $\epsilon_i > 0$ and so we're really allowing anything to be misclassified
- For $\lambda \to \infty$, we recover the original SVM because we want $\xi_i = 0$.
- We can use cross-validation to choose it.

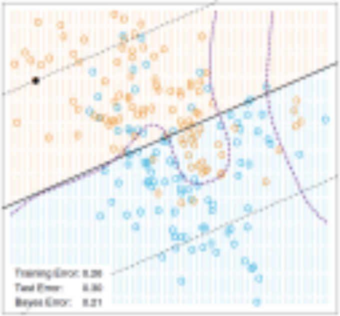
when λ gets bigger & bigger, for ex: $\lambda \to \infty$, then we're paying an infinitely large penalty by letting any of these $\epsilon_i > 0$. And we get back the original SVM.

INFLUENCE OF MARGIN PARAMETER

• → pts that have their corresponding $\alpha > 0$. Why?



$\lambda = 100000$



$\lambda = 0.01$

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

Hyperplane is going to adjust itself based on what we set λ to be.

Primal problem with slack variables

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

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

new for this constraint

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

additional constraint which comes from taking the derivative with respect to ξ_i

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

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

how?

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

simply replace the dot product with a kernel function btw 2 datapoints x_i and x_j

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

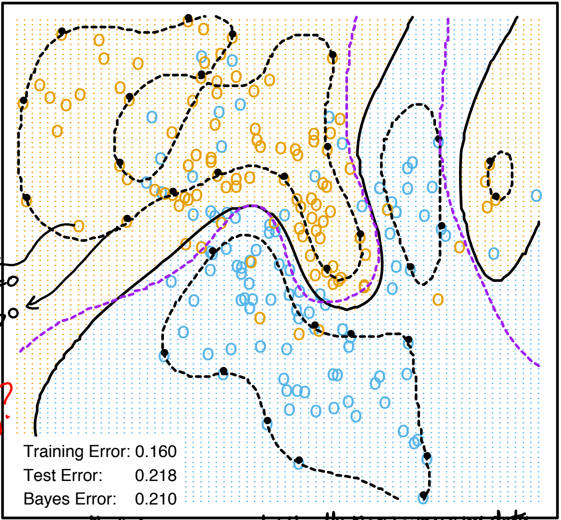
new thing

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

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

prediction

KERNELIZING THE SVM



$\alpha_i = 0$

$\alpha_i > 0$

why?

Black solid line

SVM decision boundary

Classification rule

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

many $\alpha_i = 0$. For any $\alpha_i = 0$, we can interpret this as saying that we don't consider the i^{th} data point in making our classification.

Dots

Support vectors ($\alpha_i > 0$)

Purple line

A Bayes classifier.

For any data point for α_i is + we is going to be a data point that we use to classify. Those are called support vectors.

After running the SVM, we could literally throw away any data that doesn't have a black dot over it. We only need to keep the data that has the black dot, and we only calculate the kernel btw the new point and points that have an $\alpha > 0$.

SUMMARY: SUPPORT VECTOR MACHINE

Basic SVM

- Linear classifier for linearly separable data.
- Position of affine hyperplane is determined to maximize the margin. btw the 2 classes
- The dual is a convex, so we can find exact solution with optimization. (so we have a convex optimisation problem)

Full-fledged SVM

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

↳ (when we solve for the dual, we see that we only need the kernels. we don't need the mapping at all.)

Use in practice

- Software packages (many options)
- Choose a kernel function (e.g., RBF) → means we have to choose a kernel function which if we choose the RBF means we have to choose the kernel width which gives a definition of proximity in the original space.
- Cross-validate λ parameter and RBF kernel width

↳ penalty on the slack.