Alex Smola
Barnabas Poczos
Ina Fiterau
CMU - MLD

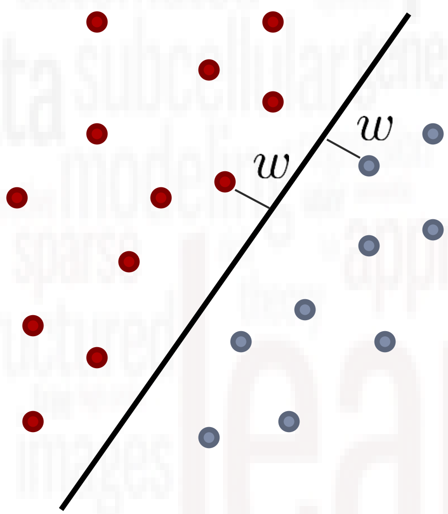
Recitation 6: Kernel SVM

SVM Revision. The Kernel Trick. Reproducing Kernels. Examples.

Main Source: F2009 10-701 course taught by Carlos Guestrin

SVM Primal

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



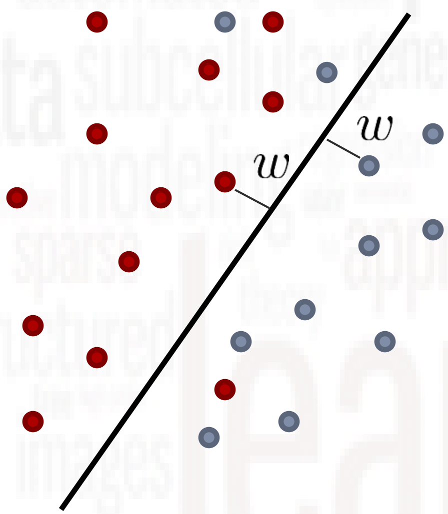
Hard Margin

$$\min_{w,b} ||w||^2$$

$$\text{subject to } (\langle w, x_i \rangle + b) y_i \geq 1$$

SVM Primal

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



Soft Margin

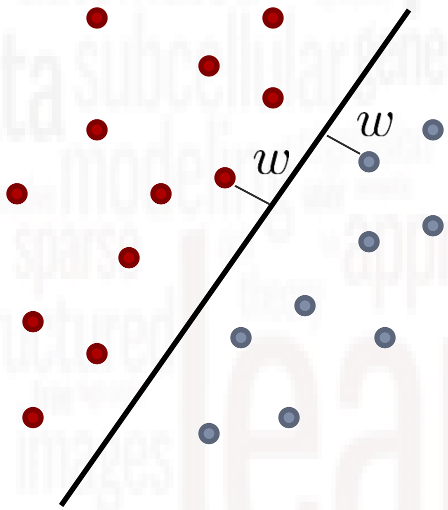
$$\min_{w,b} ||w||^2 + C \sum_i \xi_i$$

$$\text{subject to } (\langle w, x_i \rangle + b) y_i \geq 1$$

$$\xi_i \geq 0$$

SVM Dual

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



Dual for the hard margin SVM

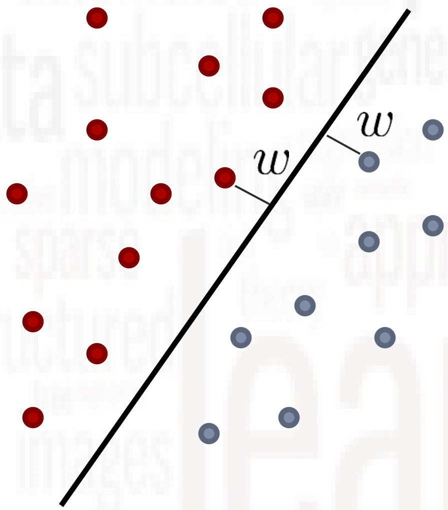
$$\mathcal{L}(w, \alpha) = \frac{1}{2}\langle w, w \rangle - \sum_i \alpha_i \Big[(\langle w, x_i \rangle + b) y_i - 1 \Big]$$

$$\alpha_i \geq 0$$

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \rightarrow \quad w = \sum_i \alpha_i y_i x_i$$

SVM Dual

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



Dual for the hard margin SVM

$$\mathcal{L}(w, \alpha) = \frac{1}{2}\langle w, w \rangle - \sum_i \alpha_i \left[(\langle w, x_i \rangle + b) y_i - 1 \right]$$

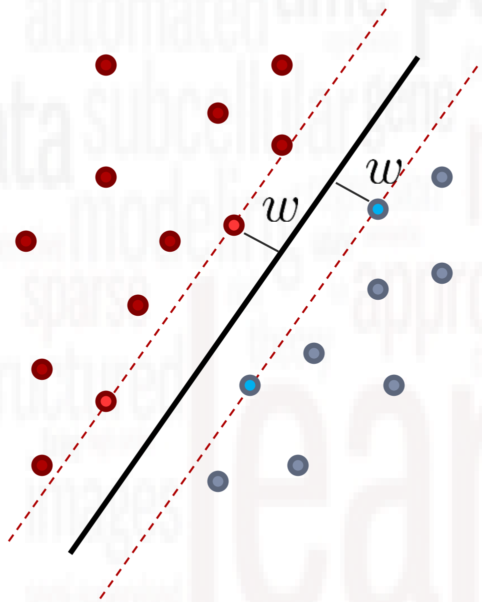
$$\alpha_j \geq 0$$

Substituting α for w

$$\begin{aligned} w &= \sum_i \alpha_i y_j x_j \\ \langle w, w \rangle &= \sum_{i,j} \langle \alpha_i y_i x_i, \alpha_j y_j x_j \rangle \\ &= \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \end{aligned}$$

SVM Dual

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



Dual for the hard margin SVM

$$\mathcal{L}(w, \alpha) = \frac{1}{2}\langle w, w \rangle - \sum_i \alpha_i \left[(\langle w, x_i \rangle + b) y_i - 1 \right]$$

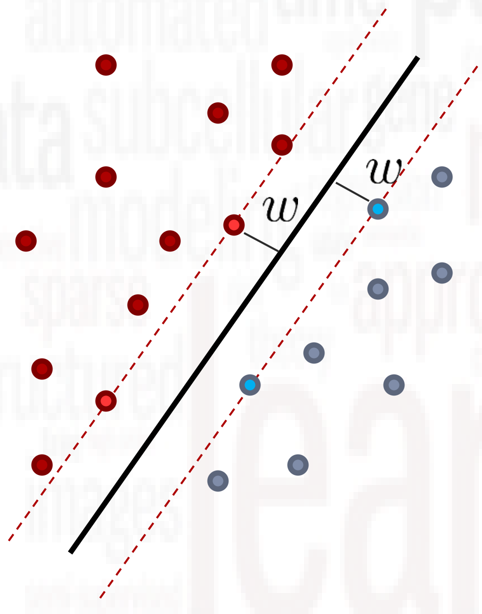
$$\alpha_j \geq 0$$

The constraints are active for the support vectors

$$\forall k \text{ s.t. } a_k > 0 \qquad b = y_k - \langle w, x_k \rangle$$

SVM Dual

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



Dual for the hard margin SVM

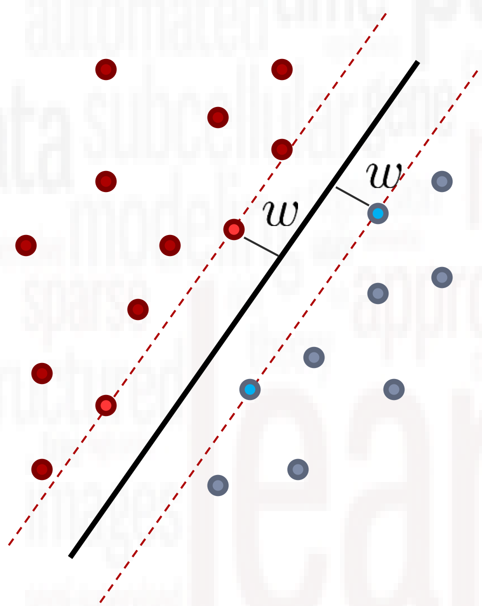
$$\max_{\alpha} \quad -\frac{1}{2} \sum_{i} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{i} \alpha_i$$

$$\sum_{i} \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

SVM – Computing w

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



Dual for the hard margin SVM

$$\max_{\alpha} \quad -\frac{1}{2}\sum_i \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

$$\sum_i \alpha_i y_i = 0$$

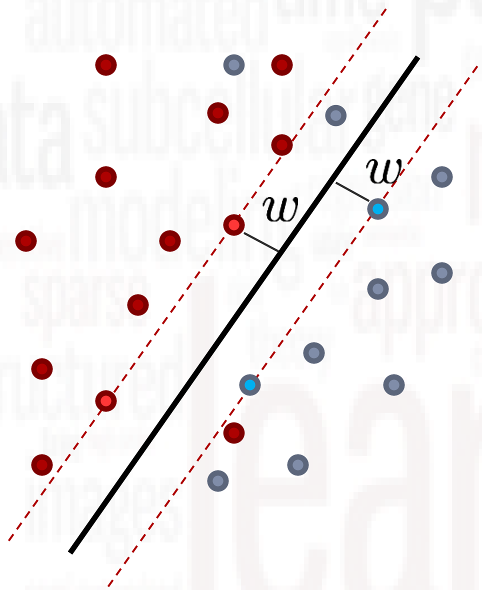
$$\alpha_i \geq 0$$

solve, get α_i

$$\boxed{\begin{aligned} w &= \sum_i \alpha_i y_i x_i \\ b &= y_k - \langle w, x_k \rangle \qquad \forall k \text{ for which } \alpha_k > 0 \end{aligned}}$$

SVM – Computing w

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



Dual for the soft margin SVM

$$\max_{\alpha} \quad -\frac{1}{2}\sum_i \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

$$\sum_i \alpha_i y_i = 0$$

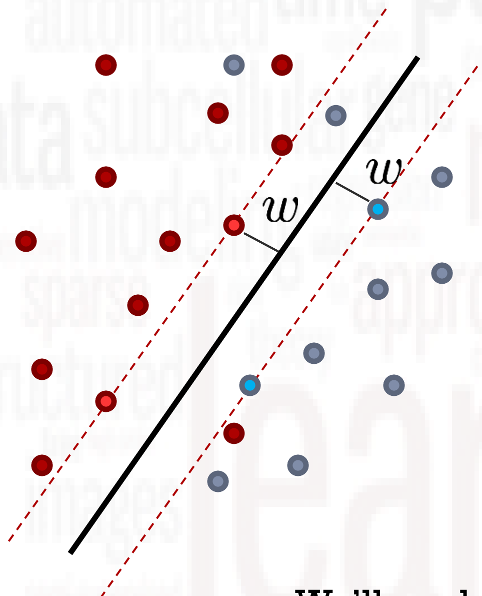
only difference from the separable case → $C \geq \alpha_i \geq 0$

solve, get α_i

$$\boxed{\begin{aligned} w &= \sum_i \alpha_i y_i x_i \\ b &= y_k - \langle w, x_k \rangle \quad \forall k \text{ for which } C > \alpha_k > 0 \end{aligned}}$$

SVM – the feature map

Find maximum margin hyper-plane $f(x) = \langle w, x \rangle + b = 0$



But data is not linearly separable ☹

$$\max_{\alpha} \quad -\frac{1}{2}\sum_i \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

We'll apply a transformation to a high dimensional space where the data is linearly separable

feature map

inputs

features

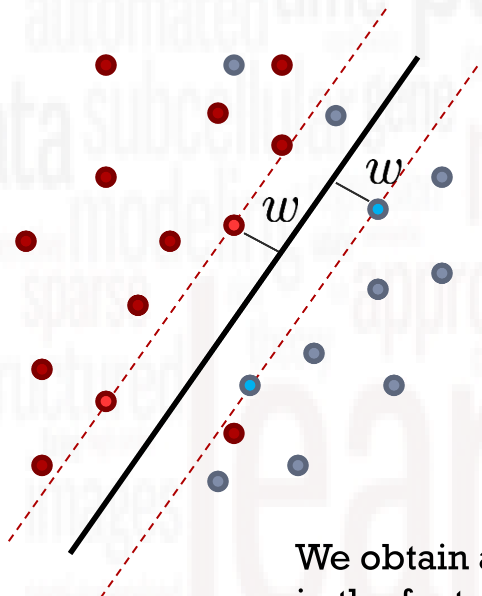
$$x \to \Phi(x)$$

$$\in R^m \qquad \in R^M$$

SVM – the feature map

Find maximum margin hyper-plane $f(x) = \langle w, \Phi(x) \rangle + b = 0$

But data is not linearly separable ☹



$$\max_{\alpha} \quad -\frac{1}{2}\sum_i \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle + \sum_i \alpha_i$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

We obtain a linear separator in the feature space.

!! $M >> m$

$\Phi(x)$ is expensive to compute!

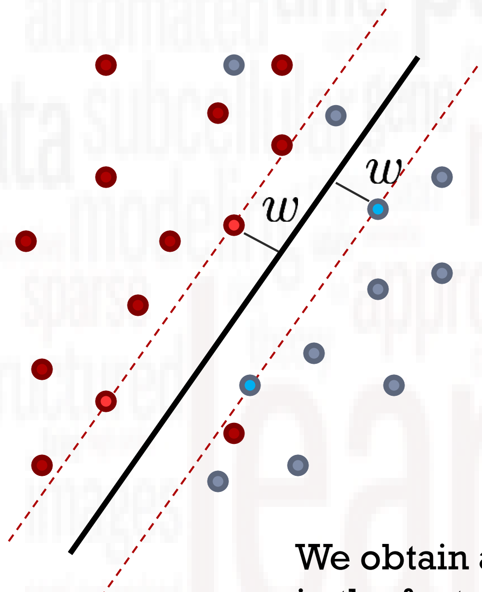
feature map

inputs $x \rightarrow \Phi(x)$ features

$\in R^m \qquad \in R^M$

Introducing the kernel

The dual formulation no longer depends on w, only on a dot product!



$$\max_{\alpha} \quad -\frac{1}{2}\sum_i \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle + \sum_i \alpha_i$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

We obtain a linear separator in the feature space.

!! $M >> m$

$\Phi(x)$ is expensive to compute!

But we don't have to!

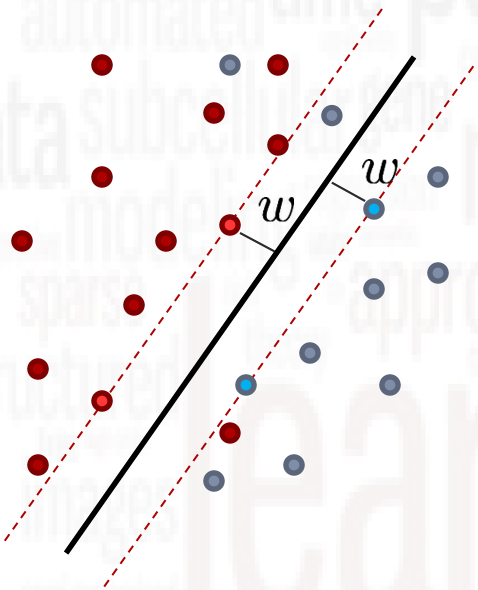
What we need is the dot product:

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

Let's call this a kernel
- 2-variable function
- can be written as a dot product

Kernel SVM

The dual formulation no longer depends on w, only on a dot product!



$$\max_{\alpha} \quad -\frac{1}{2}\sum_i \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_i \alpha_i$$

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

closed form

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

This is the famous 'kernel trick'.

- never compute the feature map
- learn using the closed form K
- constant time for HD dot products

Kernel SVM –Run time

What happens when we need to classify some x_0?

Recall that w depends on α

$$w = \sum_i \alpha_i y_i \Phi(x_i)$$

$$b = y_k - \langle w, \Phi(x_k) \rangle$$

$$\forall k \text{ s.t. } C > \alpha_k > 0$$

Our classifier for x_0 uses w

$$sign(\langle w, \Phi(x_0) \rangle + b)$$

Kernel SVM –Run time

What happens when we need to classify some x_0?

Recall that w depends on α

$$w = \sum_i \alpha_i y_i \Phi(x_i)$$

$$b = y_k - \langle w, \Phi(x_k) \rangle \quad \forall k \text{ s.t. } C > \alpha_k > 0$$

Our classifier for x_0 uses w

$$sign(\langle w, \Phi(x_0) \rangle + b)$$

Who needs w when we've got dot products?

$$\langle w, \Phi(x_0) \rangle = \sum_i \alpha_i y_i K(x_0, x_i)$$

$$b = y_k - \sum_i \alpha_i y_i K(x_k, x_i)$$

$$k \rightarrow \text{ support vectors}$$

Kernel SVM Recap

Pick kernel

Solve the optimization to get α

$$\max_{\alpha} \quad -\frac{1}{2}\sum_i \alpha_i\alpha_j y_i y_j K(x_i, x_j) + \sum_i \alpha_i$$

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

Compute b using the support vectors

$$b = y_k - \sum_i \alpha_i y_i K(x_k, x_i)$$

Classify as

$$sign\Big(\sum_i \alpha_i y_i K(x_0, x_i) + b\Big)$$

Other uses of Kernels in ML

- Logistic Regression
  - http://books.nips.cc/papers/files/nips14/AA13.pdf
- Multiple Kernel Boosting
  - http://siam.omnibooksonline.com/2011datamining/data/papers/146.pdf
- Trees and Kernels
  - http://users.cecs.anu.edu.au/~williams/papers/P175.pdf
- Conditional Mean Embeddings
  - http://arxiv.org/abs/1205.4656

More on Kernels

- Gram Matrix
  - of a set of vectors $x_1 \ldots x_n$ in the inner product space defined by the kernel K
  - $G_{ij} = K(x_i, x_j) \qquad \forall i, j \in 1 \ldots n$
- Reproducing Kernels
  - Point evaluation function for a Hilbert sp. of functions

    $$f(x) = \langle f, K_x \rangle \quad \forall f \in H$$

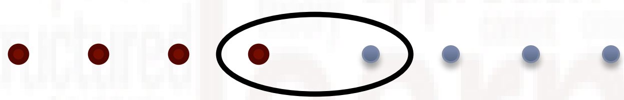
  - Reproducing property

    $$K(x,y) \stackrel{def}{=} \overline{K_x(y)} \quad \rightarrow \quad K(x,y) = \overline{K(y,x)} = \langle K_y, K_x \rangle$$

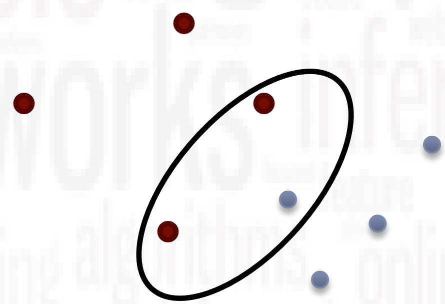
SVM Pop Quiz

- What's the maximum number of Support Vectors for a linear classification problem?
  - Hint: it's related to a concept you've recently studied

1-d case



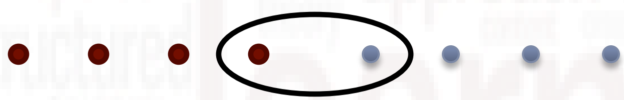
2-d case



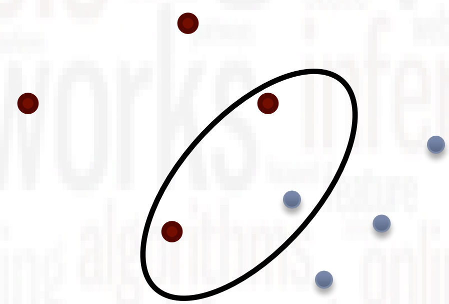
SVM Pop Quiz

- What's the worst case number of Support Vectors for a [linear] classification problem?
  - Hint: it's related to a concept you've recently studied

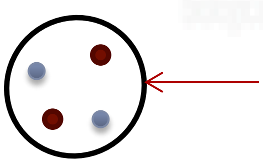
1-d case



2-d case



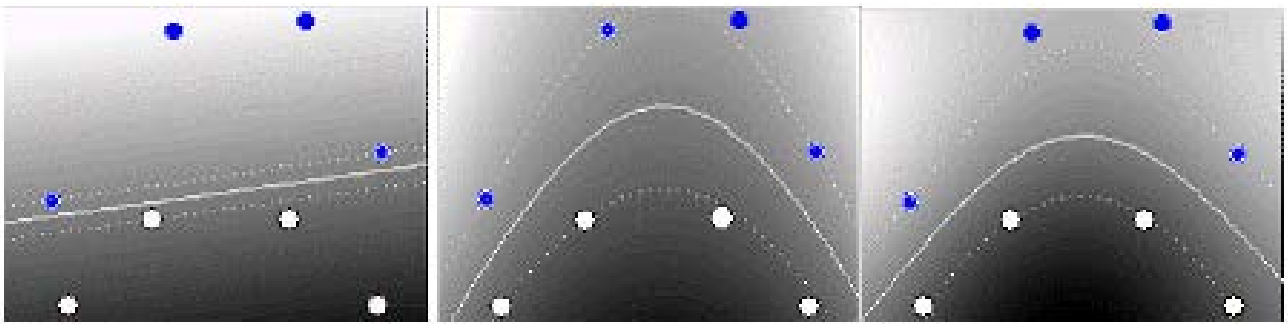
A: it's the same as the VC dimension of the classifier.



Because we can't have these as support vectors in 2D

K-SVM Pop Quiz

Here's the result of training different kernels on this dataset

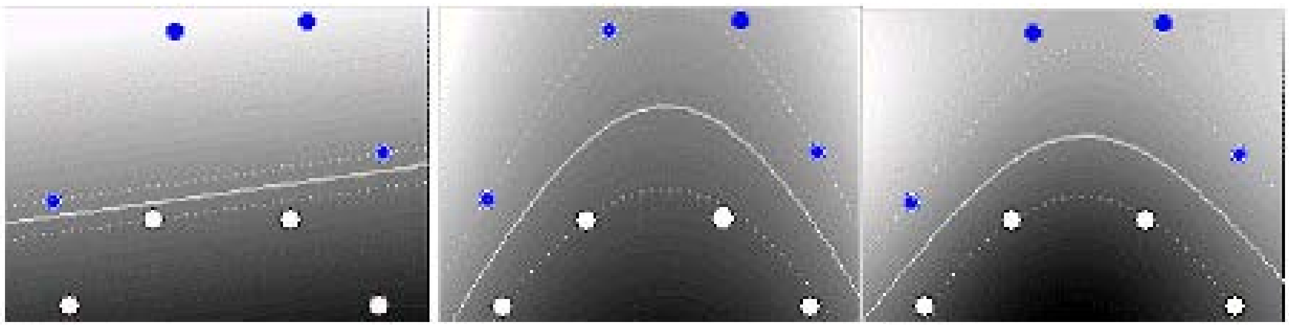


Linear Quadratic Polynomial RBF

What happens when we translate the points up by a large constant T on the vertical axis?

K-SVM Pop Quiz

Here's the result of training different kernels on this dataset



Linear Quadratic Polynomial RBF

What happens when we translate the points up by a large constant T on the vertical axis?

the bound retains relative position to points - it is shifted by 10 units

the bound depends more on the y value, therefore the bound becomes more arched

the value of the kernel is the same for each pair of points, so the bound retains position relative to points

Thanks!

Reminder: midterm is on Monday, feel free to ask questions about problems given in previous years