Support Vector Machines

Dr. Ratna Babu Chinnam

Industrial & Systems Engineering

Wayne State University

Credits: Some Slides Derived from Lecture Slides of UoM - EECS445 (Winter 2018)

Generalization Error

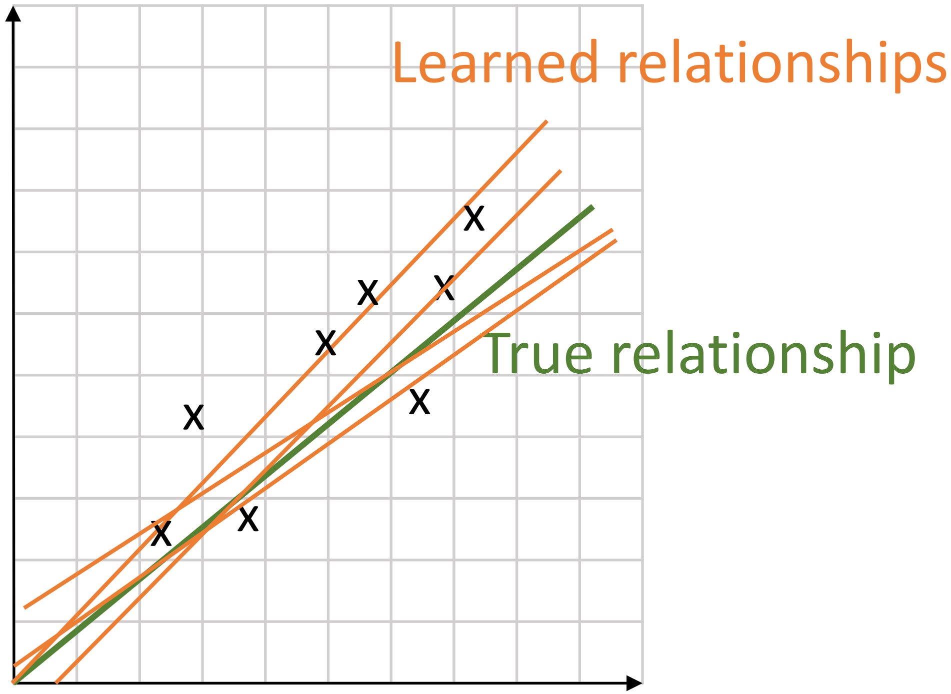
(Linear Regression Setting)

$$R_{Empirical}^{\text{Test}}(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(y_{(i)} - (\bar{\theta}\cdot\bar{x}_{(i)})\right)^2}{2}$$

n: testing data set

- Suppose we have multiple training datasets
- We have a notion of ‘average’ and ‘variance’ for $R_{Empirical}^{\text{Test}}(\bar{\theta})$

Variance

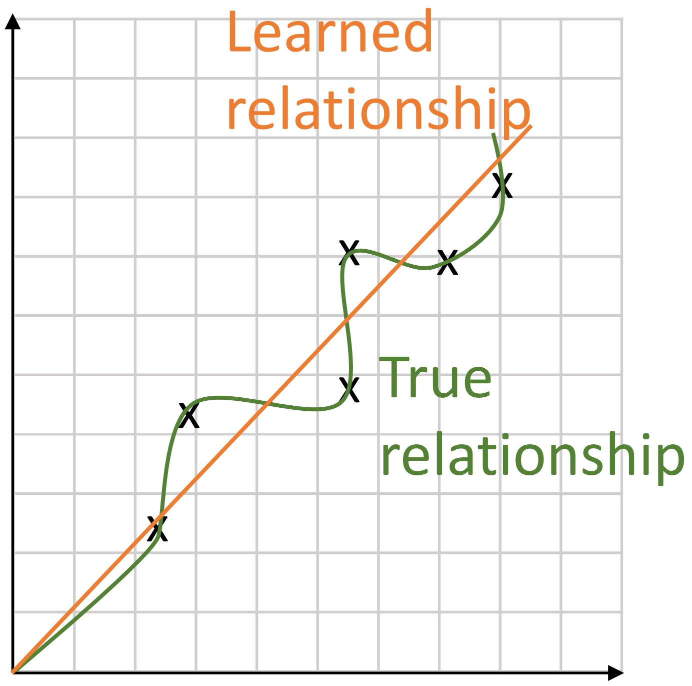


x: data points

As n increases,
variance decreases

Variance measures extent to which the solutions for individual datasets vary **around their average**

Bias



True relationship is non-linear but trying to fit a linear model:
- Increased n **does not help** bias

Increased n does allow more complex models:
- Increased n **does help** bias

Bias measures extent to which average prediction over all datasets differs from **desired function**

Bias-Variance Tradeoff

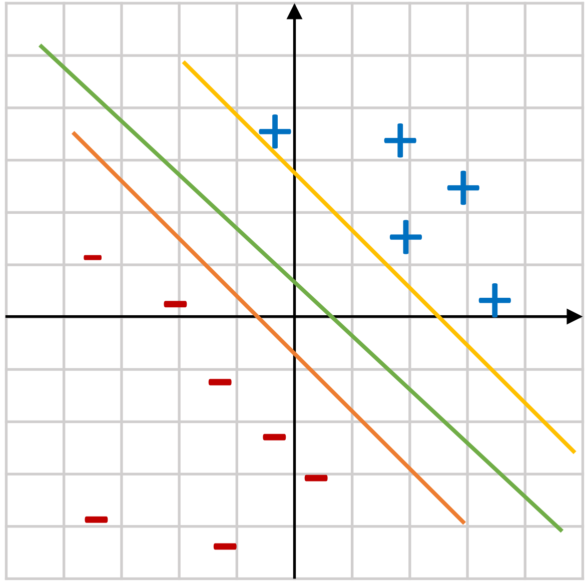
- To reduce bias
  - Need higher regression polynomials $(\mathcal{F})$

- However, if we have noisy/small dataset
  - This may increase variance

Picking a Decision Boundary

(Classification Setting)

Suppose data is linearly separable



Want:

- Boundary that classifies the training set correctly
- That is “maximally removed” from all training examples

SVM: Support Vector Machine

SVM: Structural Risk Minimization

$$R^{\text{Test}}_{Empirical}(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(y_{(i)} - (\bar{\theta}\cdot\bar{x}_{(i)})\right)^2}{2}$$

- Following bound holds with probability $1-\eta$ (Vapnik 1998):

$$\underbrace{R^{\text{Test}}_{Expected}(\bar{\theta})}_{\text{True Risk}} \leq R^{\text{Test}}_{Empirical}(\bar{\theta}) + \sqrt{\frac{h}{n}\left[\log\left(\frac{2n}{h}\right)\right] - \frac{1}{n}\log\left(\frac{\eta}{4}\right)}$$

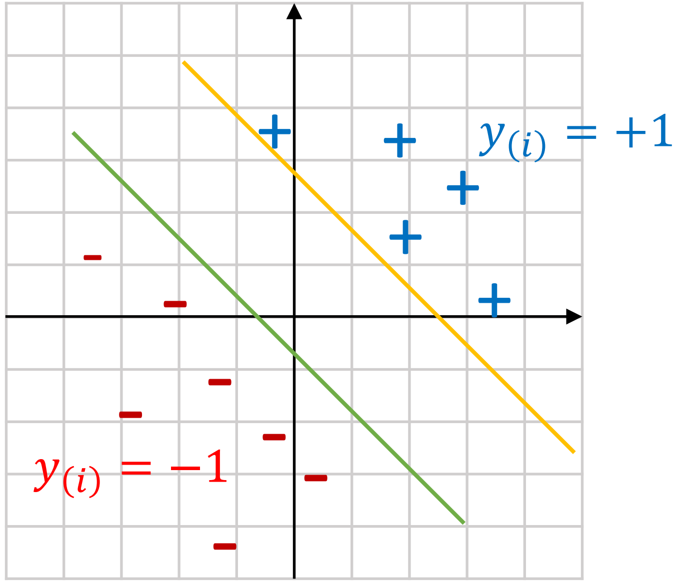
where h is the Vapnik-Chervonenkis (VC) dimension and measures the model capacity (complexity, flexibility)

- SVMs naturally limit the VC dimension and in turn attain strong bounds for the expected risk (i.e., generalization)

Picking a Decision Boundary

(Classification Setting)

Suppose data is linearly separable

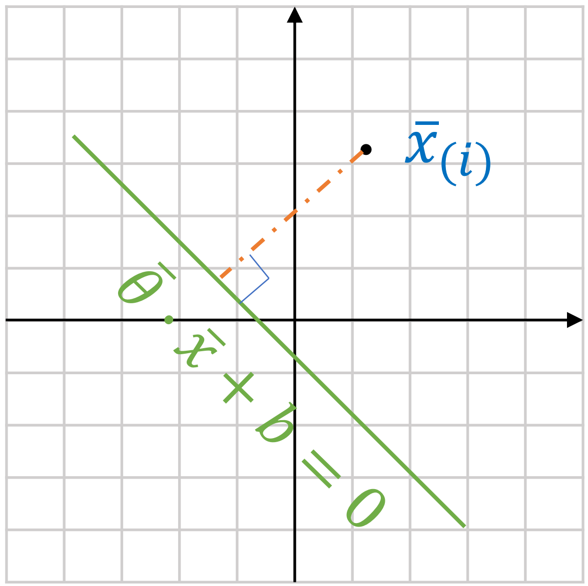


Want:

- Boundary that classifies the training set correctly
- That is “maximally removed” from all training examples

How to solve the problem? Suggest ideas!

Distance from a Point to Decision Boundary



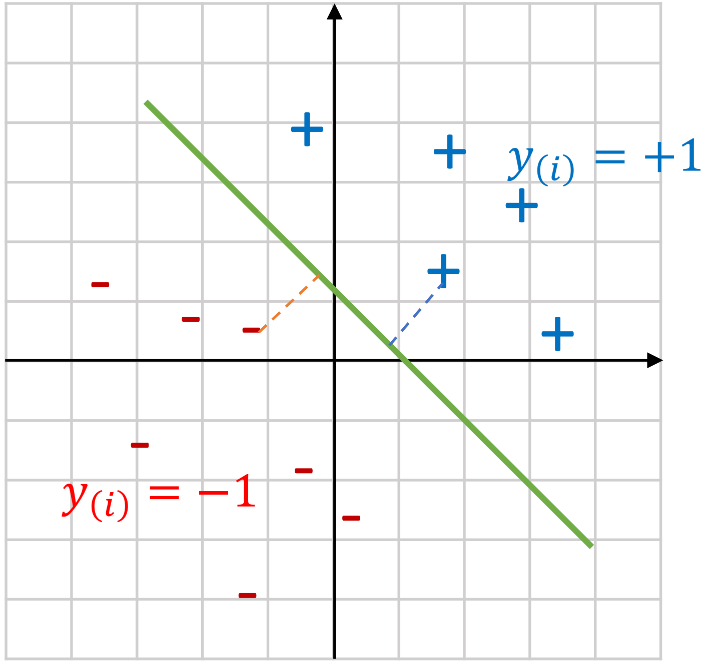
Margin for a Point: $\bar{x}_{(i)}$

$$r_{(i)}(\bar{\theta}, b) = \frac{\bar{\theta} \cdot \bar{x}_{(i)} + b}{\|\bar{\theta}\|} y_{(i)}$$

- Assumes that all points are correctly classified by
$$\bar{\theta} \cdot \bar{x} + b = 0$$
- Signed distance without $y_{(i)}$

Can we now solve the problem? Suggest ideas!

Maximum Margin Separator



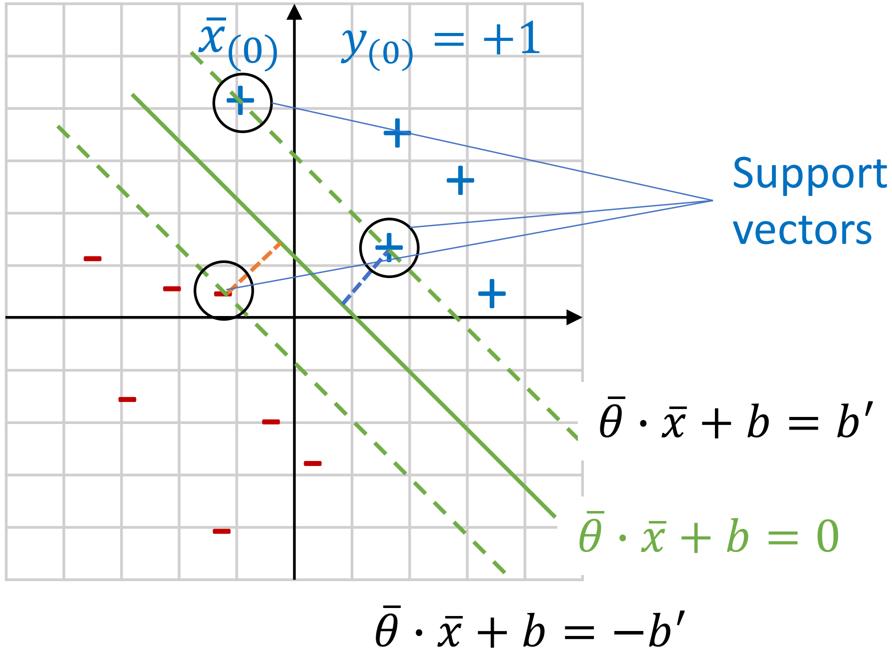
$$\max_{\bar{\theta}, b} \min_{i} r_{(i)}(\bar{\theta}, b)$$

where $\bar{\theta} \cdot \bar{x} + b = 0$ is a separating hyperplane

(Assumption: All points are correctly classified)

Distance to Decision Boundary

Suppose data is linearly separable



Margin for a point $\bar{x}_{(0)}$:

$$\frac{y_{(0)}\left(\bar{\theta} \cdot \bar{x}_{(0)} + b\right)}{\|\theta\|} = \frac{b'}{\|\theta\|}$$

Can rescale parameters (i.e., $\bar{\theta}$) so that $b' = 1$ for "support vectors"

Does adding or taking away a data point always change the solution?

Distance to Decision Boundary …

- Distance to decision boundary of a support vector

$$r_{(i)} = \frac{1}{\|\bar{\theta}\|}$$

- Can formulate problem of finding maximum margin separator as a "quadratic" program:

$$\min_{\bar{\theta},b} \frac{\|\bar{\theta}\|}{2} \;\; s.t.\; y_{(i)}\left(\bar{\theta}\bar{x}_{(i)} + b\right) \geq 1 \text{ for } i = 1, \ldots, n$$

Constrained Optimization!

How to make it Unconstrained Optimization?

Largrange Duality

- Suppose we want to minimize some function $f(\overline{w})$ subject to n constraints $h_i(\overline{w}) \leq 0$ for $i = 1, \ldots, n$

$$\min_{\overline{w}} f(\overline{w}) \quad s.t. \;\; h_i(\overline{w}) \leq 0 \text{ for } i = 1, \ldots, n$$

- Use Lagrange multiplier α_i to specify the Lagrangian

$$L(\overline{w}, \overline{\alpha}) = f(\overline{w}) + \sum_{i=1}^{n} \alpha_i h_i(\overline{w})$$

Relating the Lagrangian to $f(w)$

- Original problem:

$$\min_{\overline{w}} f(\overline{w}) \quad s.t. \;\; h_i(\overline{w}) \leq 0 \;\; \text{for} \;\; i = 1, \ldots, n$$

- Lagrangian formulation:

$$L(\overline{w}, \bar{\alpha}) = f(\overline{w}) + \sum_{i=1}^{n} \alpha_i h_i(\overline{w})$$

$$= f(\overline{w}) + \alpha_1 h_1(\overline{w}) + \cdots + \alpha_n h_n(\overline{w})$$

Requirements:

If constraints (i.e., $h_i(\overline{w}) \leq 0$) are violated, need $L(\overline{w}, \bar{\alpha})$ to be large

If all constraints are satisfied, $L(\overline{w}, \bar{\alpha}) = f(\overline{w})$; $\alpha_i h_i(\overline{w}) = 0$

Primal Formulation

- Original problem

$$\min_{\bar{w}} f(\bar{w}) \quad s.t. \;\; h_i(\bar{w}) \leq 0 \text{ for } i = 1, \dots, n$$

- Lagrangian formulation:

$$L(\bar{w}, \bar{\alpha}) = f(\bar{w}) + \sum_{i=1}^{n} \alpha_i h_i(\bar{w})$$

$$= f(\bar{w}) + \alpha_1 h_1(\bar{w}) + \cdots + \alpha_n h_n(\bar{w})$$

- **"Primal" formulation:**

$$g_p(\bar{w}, \alpha) = \min_{\bar{w}} \max_{\bar{\alpha}, \alpha_i \geq 0} L(\bar{w}, \bar{\alpha})$$

Dual Formulation

- Primal formulation:

$$\min_{\bar{w}} \max_{\bar{\alpha}, \alpha_i \geq 0} L(\bar{w}, \bar{\alpha})$$

- "Dual" formulation:

$$\max_{\bar{\alpha}, \alpha_i \geq 0} \min_{\bar{w}} L(\bar{w}, \bar{\alpha})$$

For some problems, dual formulation is easy to solve!
Particularly the case for SVMs (with kernels)

Duality Gap

1. The difference between solutions to the primal and dual formulations is called the duality gap
2. The dual (typically) gives a lower bound to solution of primal
3. Under certain conditions* duality gap is zero

* Objective function is convex + inequality constraints are continuously differentiable and convex

For our problem (SVMs) the duality gap is zero!

Deriving Dual Formulation for SVM

WAYNE STATE UNIVERSITY

Step 1:

$$\min_{\bar{\theta}} \frac{\|\bar{\theta}\|}{2} \quad s.t. \quad \underbrace{y_{(i)}\left(\bar{\theta}\cdot\bar{x}_{(i)} + b\right) \geq 1}_{1 - y_{(i)}\left(\bar{\theta}\cdot\bar{x}_{(i)} + b\right) \leq 0} \text{ for } i = 1, \ldots, n$$

$1 - y_{(i)}\left(\bar{\theta}\cdot\bar{x}_{(i)}\right) \leq 0$ if b (offset) is zero

Move the constraints into objective function

$$L(\bar{\theta}, \bar{\alpha}) = \frac{\|\bar{\theta}\|}{2} + \sum_{i=1}^{n} \alpha_i \left(1 - y_{(i)}\bar{\theta}\cdot\bar{x}_{(i)}\right)$$

How to handle b (offset)? What if we introduced a dummy feature $x_0 = 1$?

Dual Formulation for SVM …

Step 2:

$$\max_{\bar{\alpha}, \alpha_i \geq 0} \min_{\bar{\theta}} L(\bar{\theta}, \bar{\alpha})$$

$$= \max_{\bar{\alpha}, \alpha_i \geq 0} \min_{\bar{\theta}} \frac{\|\bar{\theta}\|}{2} + \sum_{i=1}^{n} \alpha_i \left(1 - y_{(i)} \bar{\theta} \cdot \bar{x}_{(i)}\right)$$

Dual Formulation for SVM …

Step 3:

$$L(\bar{\theta}, \bar{\alpha}) = \frac{||\bar{\theta}||}{2} + \sum_{i=1}^{n} \alpha_i \left(1 - y_{(i)}(\bar{\theta} \cdot \bar{x}_{(i)})\right)$$

Solve for $\bar{\theta}^*$ by setting gradients to zero

$$\nabla_{\bar{\theta}} L(\bar{\theta}, \bar{\alpha}) = \bar{\theta} + \sum_{i=1}^{n} -\alpha_i y_{(i)} \bar{x}_{(i)}$$

$$\nabla_{\bar{\theta}} L(\bar{\theta}, \bar{\alpha})\Big|_{\bar{\theta}=\bar{\theta}^*} = 0$$

$$\Rightarrow \bar{\theta}^* = \sum_{i=1}^{n} \alpha_i y_{(i)} \bar{x}_{(i)}$$

Dual Formulation for SVM …

Step 4:

$$L(\bar{\theta},\bar{\alpha}) = \frac{\|\bar{\theta}\|}{2} + \sum_{i=1}^{n} \alpha_i \left(1 - y_{(i)}\left(\bar{\theta} \cdot \bar{x}_{(i)}\right)\right)$$

substitute

$$\bar{\theta}^* = \sum_{i=1}^{n} \alpha_i y_{(i)} \bar{x}_{(i)}$$

$$L(\bar{\theta},\bar{\alpha}) =$$

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_{(i)} y_{(j)} \bar{x}_{(i)} \cdot \bar{x}_{(j)} + \sum_{i=1}^{n} \alpha_i \left(1 - \left(y^{(i)} \sum_{j=1}^{n} \alpha_j y_{(j)} \bar{x}_{(j)}\right) \bar{x}_{(i)}\right)$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \, y_{(i)} y_{(j)} \bar{x}_{(i)} \cdot \bar{x}_{(j)}$$

Dual Formulation for SVM …

$$\min_{\bar{\theta}} \frac{\|\bar{\theta}\|}{2} \quad s.t.\, y_{(i)}\left(\bar{\theta} \cdot \bar{x}_{(i)}\right) \geq 1 \text{ for } i = 1, \ldots, n$$

Dual formulation:

$$\max_{\bar{\alpha}, \alpha_i \geq 0} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \, y_{(i)} y_{(j)} \boxed{\bar{x}_{(i)} \cdot \bar{x}_{(j)}}$$

Inputs appear as "dot" products!

Done with Linear SVMs!

- Solve dual formulation (i.e., $\bar{\alpha}^*$)
- $\bar{\theta}^* = \sum_{i=1}^{n} \alpha_i^* y_{(i)} \bar{x}_{(i)}$
- $b^* = ?$ (Hint: Use any support vector)
- Why can't all $\alpha_i = 0$? Then there will be no $\bar{\theta}^*$!

Support vectors are points with $\alpha_i > 0$!

Support Vectors

$$\max_{\bar{\alpha},\alpha_i \geq 0} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \, y_{(i)} y_{(j)} \bar{x}_{(i)} \cdot \bar{x}_{(j)}$$

- Solution satisfies "complementary slackness constraints":

$$\hat{\alpha}_i > 0: y_{(i)}\left(\bar{\theta}^* \cdot \bar{x}_{(i)}\right) = 1 \quad (\text{support vector})$$

$$\hat{\alpha}_i = 0: y_{(i)}\left(\bar{\theta}^* \cdot \bar{x}_{(i)}\right) > 1 \quad (\text{non-support vector})$$

- Either the primal inequality is satisfied with equality, or the dual variable (Lagrangean coefficient α_i) is zero!

Non-Linear Separability

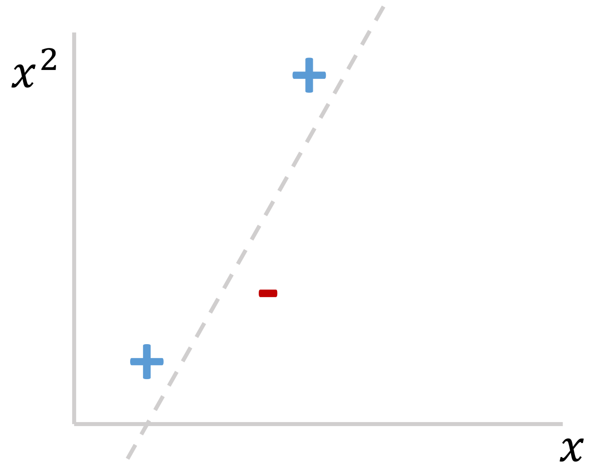
- Original 1D space

To classify these training examples, find $\theta, b \in \mathbb{R}$ $s.t.\ \ \theta x + b = 0$ (classifier)

Not possible

(Try this yourself for $x = 1$, $x = 2, x = 3$)

- Projected 2D space

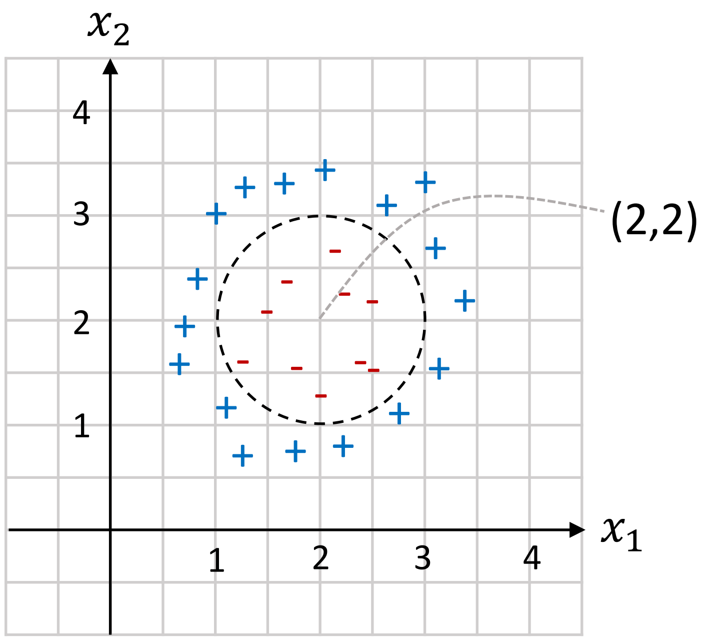


$\phi(x) = [x, x^2]$

Find $\theta \in \mathbb{R}^2, b \in \mathbb{R}$ s.t. $\phi(x) + b = 0$ correctly classifies examples.

Possible

Non-Linear Separability: Other Examples



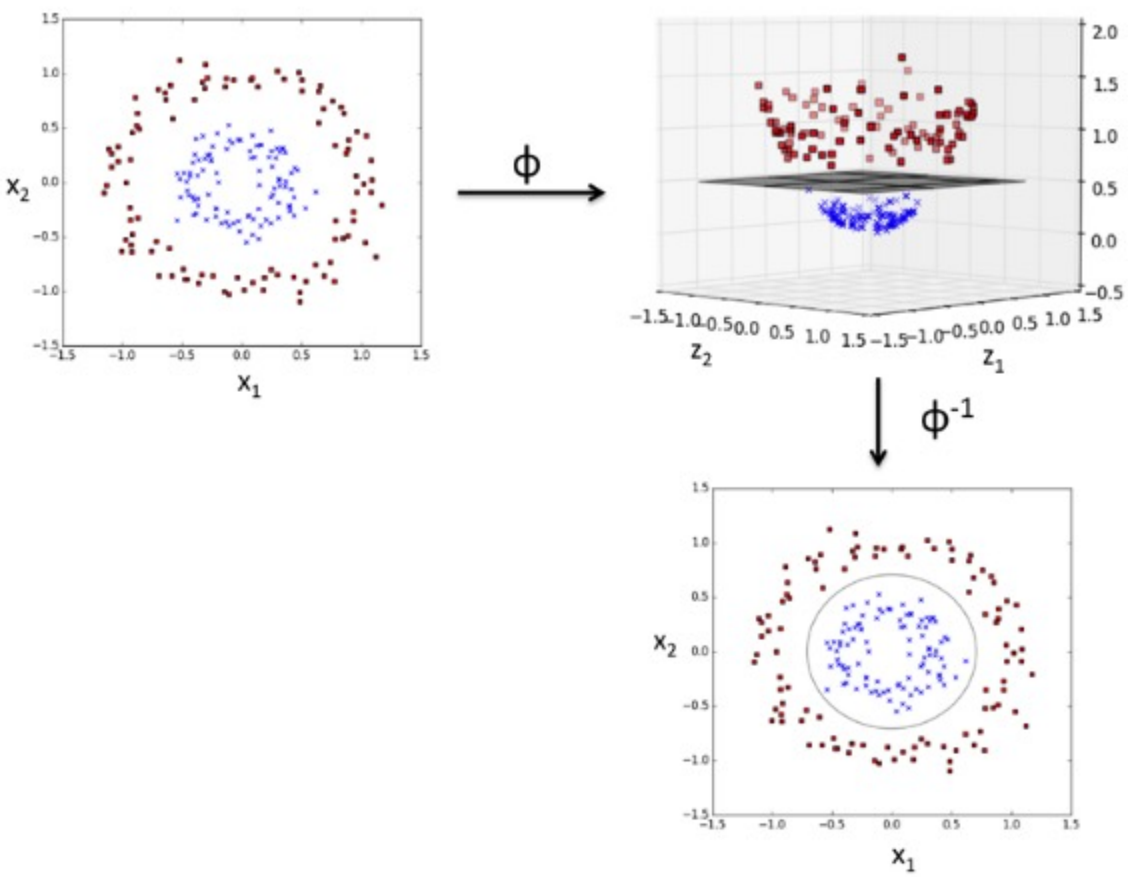
Mapping: $\phi(x_1, x_2) = (z_1, z_2, z_3, z_4) = (x_1, x_2, x_1^2, x_2^2)$

Circle centered at (h, k) with radius r

$$(x_1 - h)^2 + (x_2 - k)^2 = r^2$$
$$(x_1 - 2)^2 + (x_2 - 2)^2 = 1$$

$Boundary \Rightarrow x_1^2 + x_2^2 - 4x_1 - 4x_2 + 7 = 0$

Mapping: $\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$



“Linear” Classifiers in Higher Dimensional Spaces

- So, for $\bar{x} = [x_1, x_2]$
- Define $\phi(\bar{x}) = [x_1^2, x_2^2, x_1, x_2, 1]$
- Then we can find $\theta \in \mathbb{R}^5 \quad s.t. \quad \bar{\theta} \cdot \phi(\bar{x}) = 0$ defines a decision boundary:

$$\bar{\theta} = [1,1,-4,-4,7]$$

- This means we could use a “linear” classifier to find a decision boundary in this higher dimensional feature space.

Problem with Feature Mapping

- $\phi(\bar{x})$: feature mapping

- Suppose $\bar{x} \in \mathbb{R}^d$. If we want a feature mapping that considers all degree 2 terms (i.e., $x_i^2, x_i x_j$) what would the dimensionality of $\phi(\bar{x})$ be?

$$\phi(x) \in \mathbb{R}^p$$

$$p = \binom{d}{2} + d = \frac{d(d-1)}{2} + d = \frac{d(d+1)}{2}$$

- For example, when $d = 1{,}000 \implies p > 500{,}000$
- When $d = 50 \implies p = 1{,}275$ (for just degree 2)!

Back to the Dual

$$\max_{\bar{\alpha}, \alpha_i \geq 0} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \, y_{(i)} y_{(j)} \bar{x}_{(i)} \cdot \bar{x}_{(j)}$$

- Notice that in the dual formulation where the feature vectors appear as $\bar{x}_{(i)} \cdot \bar{x}_{(j)}$, we can replace $\bar{x}_{(i)} \cdot \bar{x}_{(j)}$ with $\phi\left(\bar{x}_{(i)}\right) \cdot \phi\left(\bar{x}_{(j)}\right)$.
- Sometimes, even when $\phi\left(\bar{x}_{(i)}\right)$ is higher dimensional, $\phi\left(\bar{x}_{(i)}\right) \cdot \phi\left(\bar{x}_{(j)}\right)$ has an efficient representation!

Kernels

Suppose $\bar{x} = [x_1, x_2]$

Let $\phi(\bar{x}) = \left[x_1^2, x_2^2, \sqrt{2}x_1 x_2\right]$

Then $\bar{u}, \bar{v} \in \mathbb{R}^2$

$$\begin{aligned}\phi(\bar{u}) \cdot \phi(\bar{v}) &= \left[u_1^2, u_2^2, \sqrt{2}u_1 u_2\right] \cdot \left[v_1^2, v_2^2, \sqrt{2}v_1 v_2\right] \\ &= u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 u_2 v_1 v_2 \\ &= (u_1 v_1 + u_2 v_2)^2 = (\bar{u} \cdot \bar{v})^2\end{aligned}$$

Leads to "Kernel Trick":

$$K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \underbrace{\left(\bar{x}_{(i)} \cdot \bar{x}_{(j)}\right)^2 = \phi\left(\bar{x}_{(i)}\right) \cdot \phi\left(\bar{x}_{(j)}\right)}_{\text{Kernel}}$$

Kernels …

- Each kernel has an associated feature mapping

$$K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \phi\left(\bar{x}_{(i)}\right) \cdot \phi\left(\bar{x}_{(j)}\right)$$

- ϕ takes input $\bar{x} \in X$ (e.g., $\bar{x} \in \mathbb{R}^d$) and maps it to features space $\mathcal{F}$ (e.g., $\mathbb{R}^p$)

$$\phi: X \to \mathcal{F} \quad K: X \times X \to \mathbb{R}$$

- Kernel takes two inputs and gives their similarity in feature space
- Examples:

$$K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \left(\bar{x}_{(i)} \cdot \bar{x}_{(j)} + r\right)^p \quad r > 0$$

Polynomial Kernel

$$K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \exp\left(-\gamma \left\|\bar{x}_{(i)} - \bar{x}_{(j)}\right\|^2\right)$$

RBF Kernel
(∞ Dimensional)

Kernel Trick Example

Given a kernel: $K(\bar{x}, \bar{z}) = 1 + 2\bar{x} \cdot \bar{z} + 4(\bar{x} \cdot \bar{z})^2$

Find the corresponding feature mapping: $\phi(\bar{x})$ for $\bar{x}, \bar{z} \in \mathbb{R}^2$

$$
\begin{aligned}
K(\bar{x}, \bar{z}) &= 1 + 2x_1 z_1 + 2x_2 z_2 + 4(x_1 z_1 + x_2 z_2)^2 \\
&= 1 + 2x_1 z_1 + 2x_2 z_2 + 4x_1^2 z_1^2 + 4x_2^2 z_2^2 + 8x_1 z_1 x_2 z_2 \\
&= \phi(\bar{x}) \cdot \phi(\bar{z}) \\
&= \left[1, \sqrt{2}x_1, \sqrt{2}x_2, 2x_1^2, 2x_2^2, \sqrt{8}x_1 x_2\right] \cdot \left[1, \sqrt{2}z_1, \sqrt{2}z_2, 2z_1^2, 2z_2^2, \sqrt{8}z_1 z_2\right]
\end{aligned}
$$

So, **can all functions be kernels**?

No. There are some definitions of $k(\bar{x}, \bar{x}')$ for which there are no corresponding feature mapping!

Note: Sum of valid kernels is also a valid kernel!

Popular Kernels

Hyper Parameters:

- Linear Kernel: $K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \bar{x}_{(i)} \cdot \bar{x}_{(j)}$

- Polynomial Kernel: $\left(\bar{x}_{(i)} \cdot \bar{x}_{(j)} + r\right)^{p}\ \ r > 0$ — p, r

- RBF Kernel: $K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \exp(-\gamma \left\|\bar{x}_{(i)} - \bar{x}_{(j)}\right\|^{2})$ — γ

- Sigmoid Kernel: $K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \tanh\left(\gamma \bar{x}_{(i)} \cdot \bar{x}_{(j)} + c\right)$ — γ, c

Kernel SVM: Recap

$$\max_{\bar{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \, y_{(i)} y_{(j)} \phi\left(\bar{x}_{(i)}\right) \cdot \phi\left(\bar{x}_{(j)}\right)$$

subject to $\alpha_i \geq 0 \; \forall i = 1, \ldots, n$

$$\max_{\bar{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \, y_{(i)} y_{(j)} K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right)$$

subject to $\alpha_i \geq 0 \; \forall i = 1, \ldots, n$

Classifying a New Example

- Previously:

$$h(\bar{x}_{(new)}) = sign(\bar{\theta} \cdot \bar{x}_{(new)})$$

- Recall:

$$\bar{\theta}^* = \sum_{i=1}^{n} \alpha_i^* y_{(i)} \bar{x}_{(i)}$$

$\phi(\bar{x}_{(i)})$

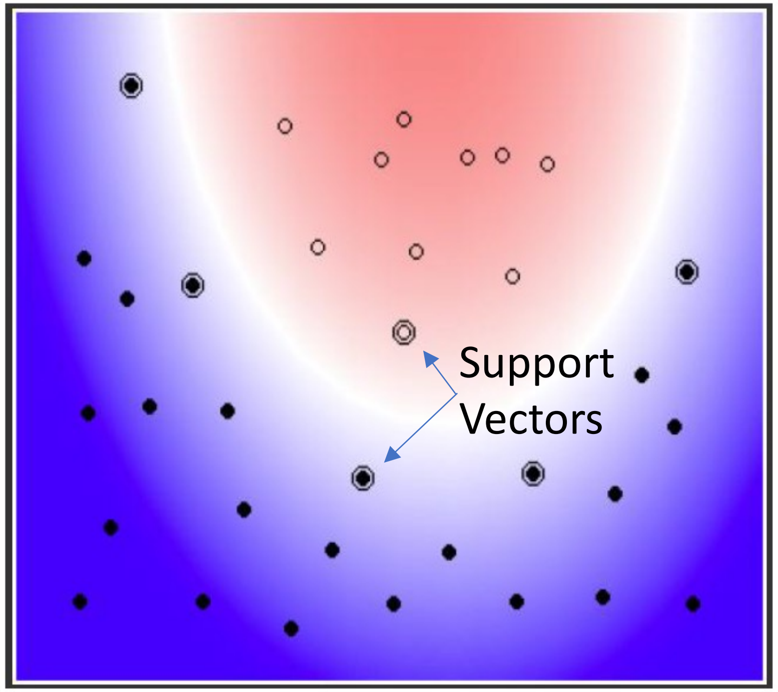
- So:

$$h(\bar{x}_{(new)}) = sign\left(\sum_{i=1}^{n} \alpha_i^* y_{(i)} \underbrace{\bar{x}_{(i)} \cdot \bar{x}_{(new)}}_{K(\bar{x}_{(i)}, \bar{x}_{(new)})}\right)$$

Visualization

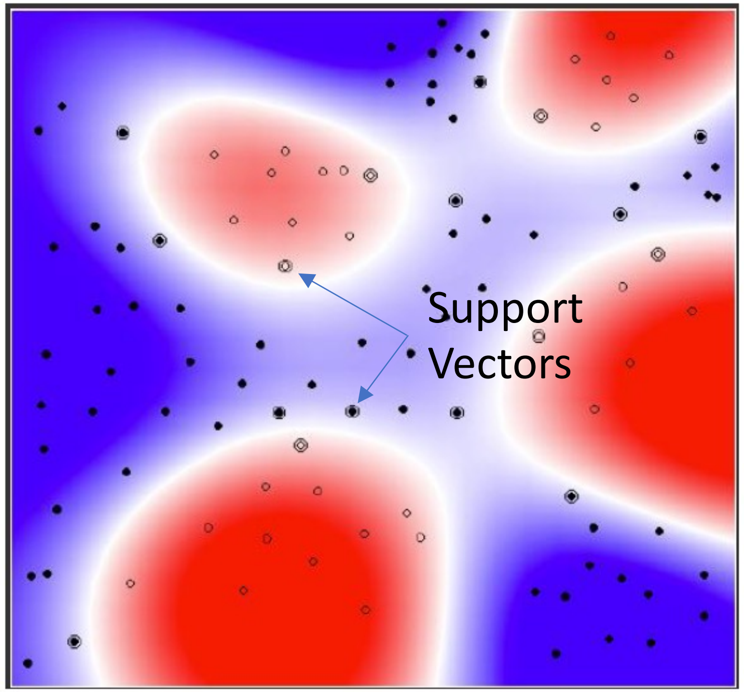
Quadratic Kernel

$$K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \left(\bar{x}_{(i)} \cdot \bar{x}_{(j)} + 1\right)^2$$



RBF Kernel

$$K\left(\bar{x}_{(i)}, \bar{x}_{(j)}\right) = \exp(-\gamma \|\bar{x}_{(i)} - \bar{x}_{(j)}\|^2)$$



“Soft-Margin” SVM: Non-Separable Case

- What to do when classes are not separable?
- Allow some misclassified points while still maximizing the margin using “slack variables”

$$\min_{\bar{\theta},b,\bar{\xi}} \frac{\|\bar{\theta}\|}{2} + C\sum_{i=1}^{n}\xi_i$$

$$s.t.\ \ y_{(i)}\left(\bar{\theta}\cdot\bar{x}^{(i)}+b\right) \geq 1-\xi_i \quad \forall\, i \in [1,\dots,n]$$

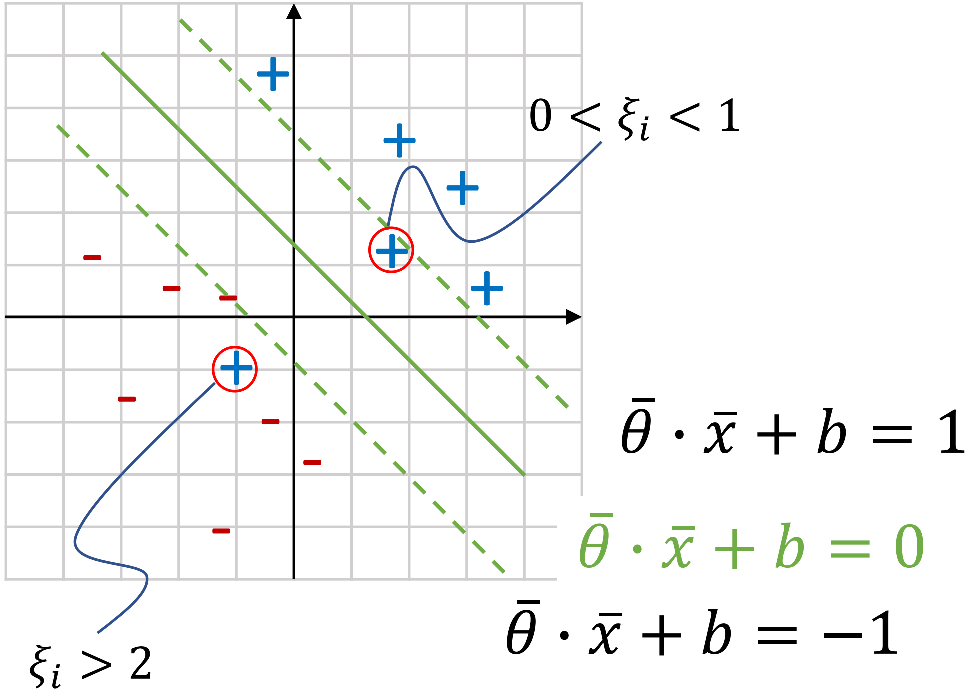
$$\xi_i \geq 0 \quad \forall\, i \in [1,\dots,n]$$

- As we increase C, we penalize errors more, and at $C = \infty$, it becomes a hard-margin SVM

Question: Is there a merit to working with soft-margin SVM when the classes are separable?

Maybe. Can reduce model complexity!

Soft-Margin SVM: Non-Separable …



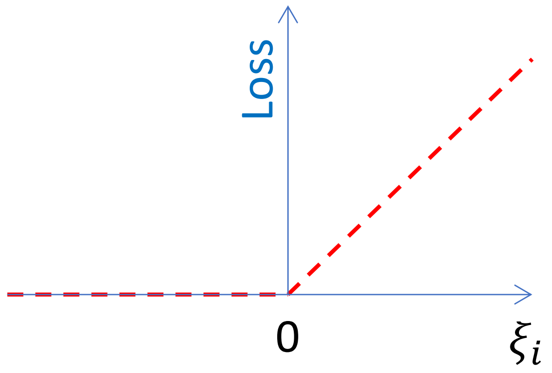
- It turns out that we can write this problem in terms of the hinge loss

Soft-Margin SVM: Non-Separable …

$$y_{(i)}\left(\bar{\theta} \cdot \bar{x}_{(i)} + b\right) \geq 1 - \xi_i$$

- **Hinge Loss**: Let

$$\xi_i = \max\left\{0, 1 - y_{(i)}\left(\bar{\theta} \cdot \bar{x}_{(i)} + b\right)\right\}$$



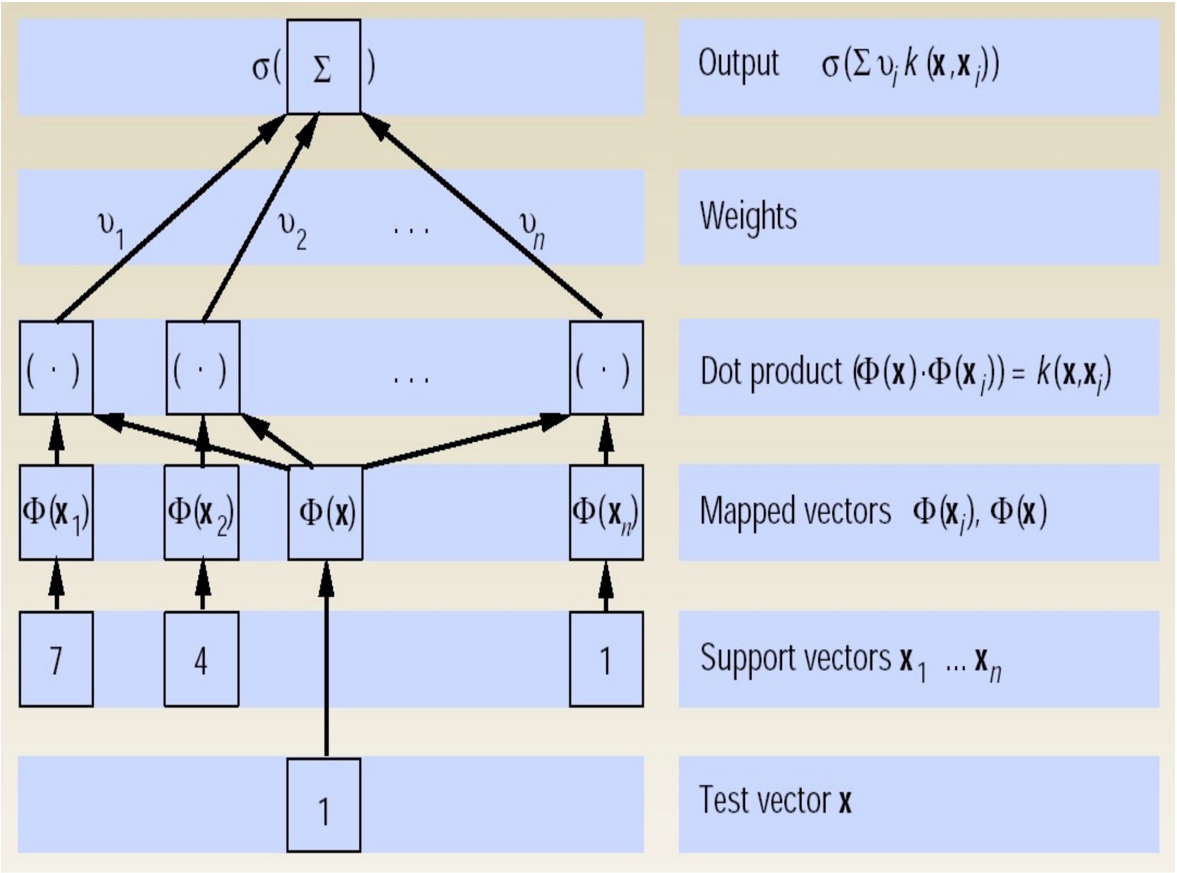
- Clearly

$$\xi_i \geq 0 \quad \text{and}$$
$$\xi_i \geq 1 - y_{(i)}\left(\bar{\theta} \cdot \bar{x}_{(i)} + b\right)$$

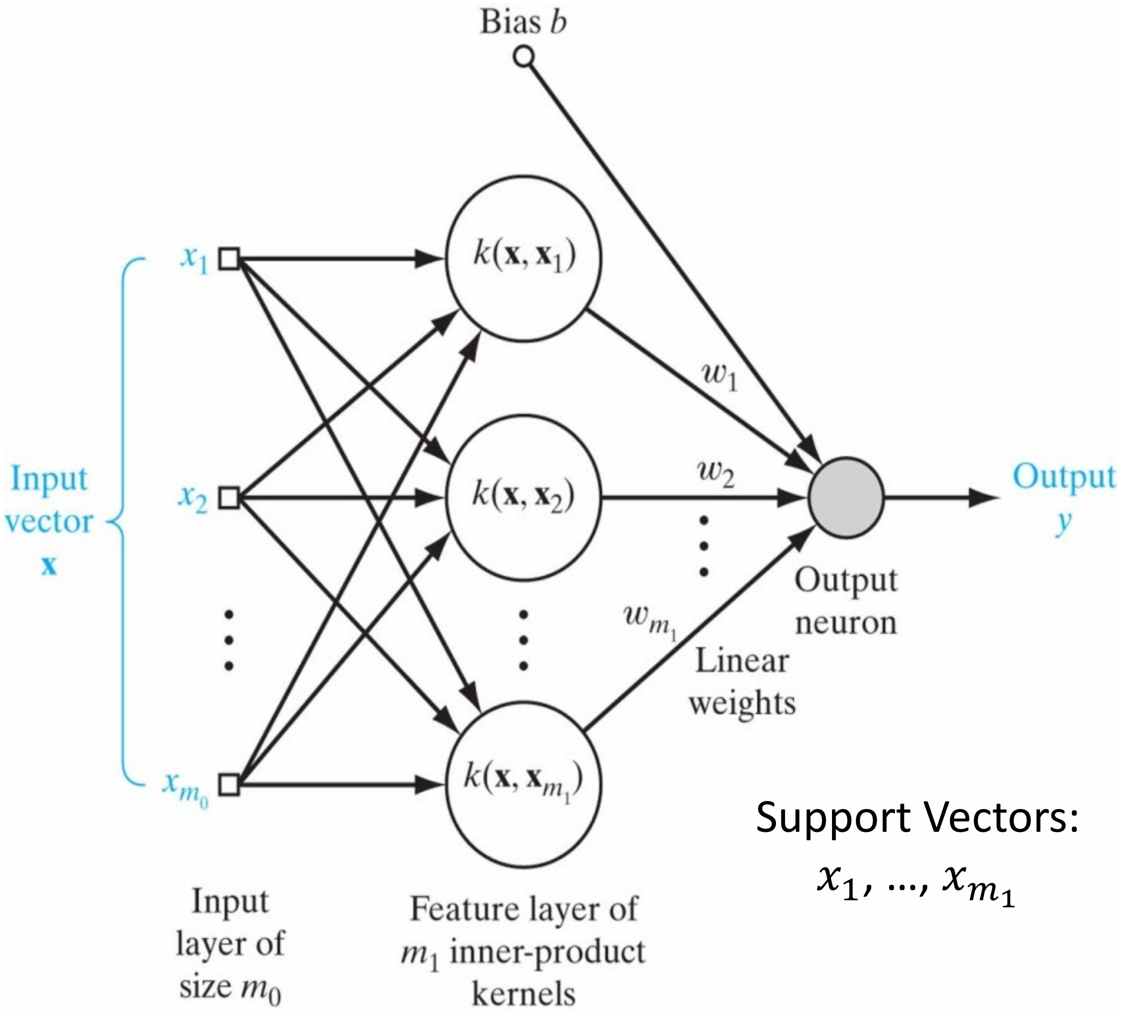
- So, we can rewrite the optimization problem as:

$$\min_{\bar{\theta}, b} \frac{\|\bar{\theta}\|}{2} + C \sum_{i=1}^{n} \underbrace{Loss_h\left(1 - y_{(i)}\left(\bar{\theta} \cdot \bar{x}_{(i)} + b\right)\right)}_{\text{Hinge Loss}}$$

SVM Architecture



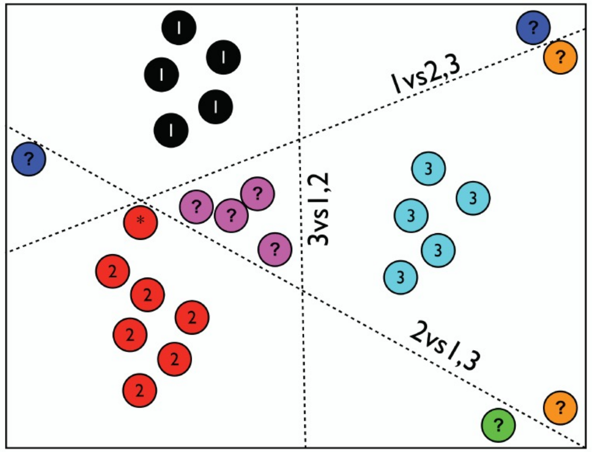
(Source: Schölkopf, 1998)



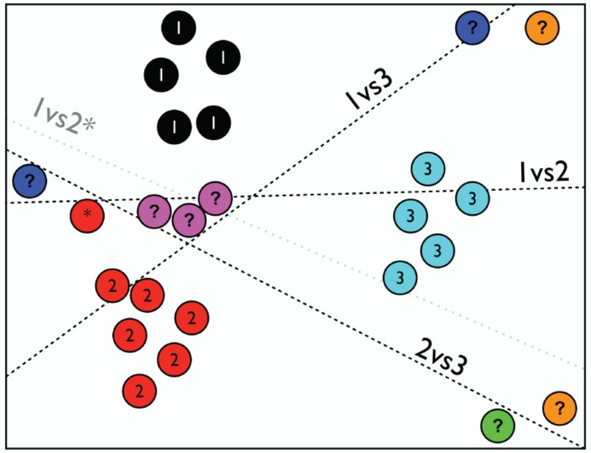
Support Vectors:
$x_1, \ldots, x_{m_1}$

Multi-class Classification

- SVMs are intrinsically binary classifiers!
- One vs All Approach
  - Computationally Efficient
  - M classes: M SVMs
  - Difficulty: Class boundaries overlap
- One vs One Approach
  - M classes: $\binom{M}{2}$ SVMs
    - $M = 26 \Rightarrow 325$ SVMs
  - Decision: Class with most votes' wins!
  - There can still be class ambiguity
  - Scoring Efficiency:
    - $C_i > C_j \;\; \Rightarrow$ Don’t bother checking $C_j > C_k \; \forall \; k$
- Hierarchical/Tree Methods



One vs All Approach

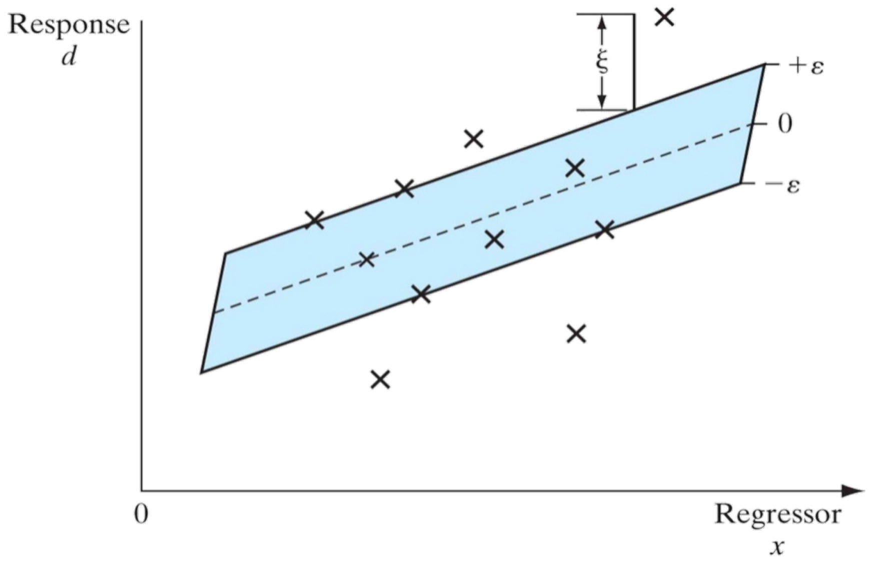


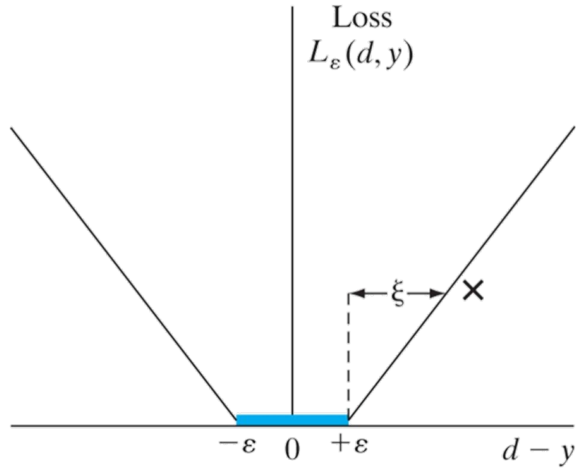
One vs One Approach

SVM Regression

- SVM formulation can be modified to handle regression
- Find a hyperplane that minimizes the distance to the farthest data points (enclose all points within the supporting hyperplanes)
- ε-insensitive loss function is for generalization:

$$L_\varepsilon(d,y) = \begin{cases} |d-y| - \varepsilon & \text{for } |d-y| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$





Implementing SVMs in Python

Comparing Basic Classifiers in Python using sklearn Package: Demonstated using Iris Dataset

Primary Source: Jason Brownlee | Source

In this notebook, we are going to build and compare six differnt machine learning models for classification using a popular and simple dataset (Iris flower dataset) for classifying flowers beased on features of flower petals.

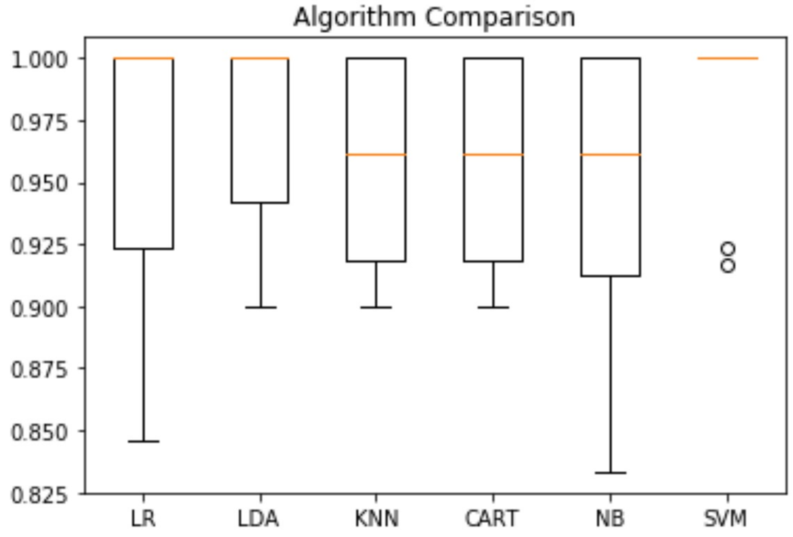
We shall explore the folloiwng machine learning models:

- Logistic Regression (LR)
- Linear Discriminant Analysis (LDA)
- K-Nearest Neighbors (KNN).
- Classification and Regression Trees (CART).
- Gaussian Naive Bayes (NB).
- Support Vector Machines (SVM).

We shall use the extremely popular scikit-learn (`sklearn`) package for implementing and testing the algorithms.

Select Best Model

```
# Compare Algorithms
pyplot.boxplot(results, labels=names)
pyplot.title('Algorithm Comparison')
pyplot.show()
```



Implementing SVMs in Python …

Comparing Classifiers in Python using sklearn Package: Synthetic Datasets

Source: [scikit-learn]() **|** Code: Gaël Varoquaux & Andreas Müller | Modified for documentation by Jaques Grobler

- A comparison of a several classifiers in scikit-learn on synthetic datasets. The point of this example is to illustrate the nature of decision boundaries of different classifiers. This should be taken with a grain of salt, as the intuition conveyed by these examples does not necessarily carry over to real datasets.
- Particularly in high-dimensional spaces, data can more easily be separated linearly and the simplicity of classifiers such as naive Bayes and linear SVMs might lead to better generalization than is achieved by other classifiers.
- We shall explore the following machine learning models: K-Nearest Neighbors (KNN), Gaussian Process (GP), Quadratic Discriminant Analysis (QDA), Decision Tree (DT), Random Forest (RF), AdaBoost (AB), Gaussian Naive Bayes (NB), Linear Support Vector Machines (SVM), RBF Support Vector Machines (SVM), Multi-Layer Perceptron (MLP)