UNIVERSITY OF ARKANSAS

DASC 4113 Machine Learning

Lecture 9

Ukash Nakarmi

Support Vector Machines

Learning Objectives

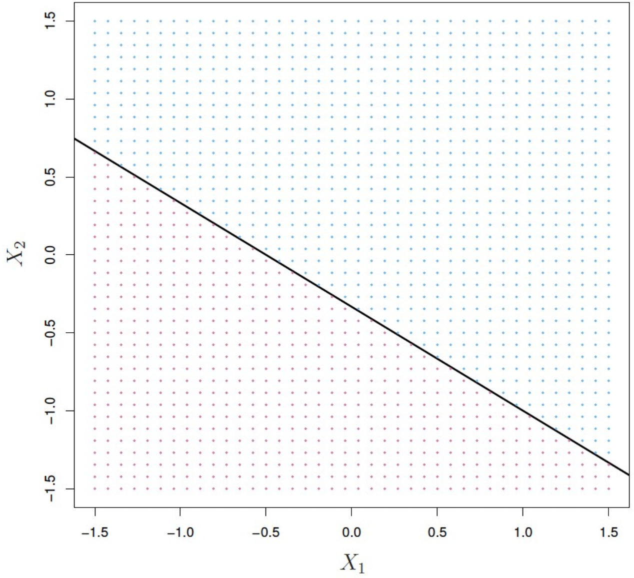
In this class, we will learn:

- A different approach to Classification termed " Maximal Marginal Classifier".
- Limitations of Maximal Marginal Classifier(MMC)
- An alternative: Support Vector Classifier that overcomes of MMC.
- A general approach termed "Support Vector Machines" that are able to handle non-linear Class Boundaries.
- Connection between Logistic Regression and Support Vector Machines.

Maximal Marginal Classifier

Hyperplane:

In a p-dimensional space, a hyperplane is a flat, affine subspace of a dimension p-1.



Example: A line in a 2 D space

2 Dimensions : X1 and X2

Hyperplane: A line, $1 + 2X_1 + 3X_2 = 0.$

Flat: Can use Euclidean Geometry/Linear Algebra. Affine: Does not have to contain an origin.

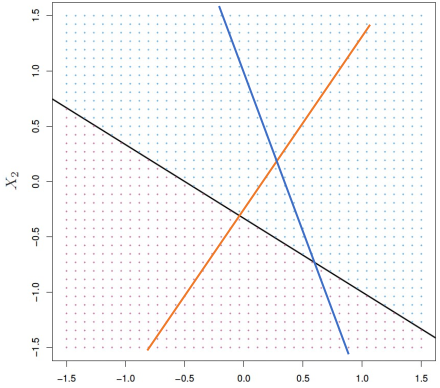
UNIVERSITY OF ARKANSAS

Hyperplane:

Expression for a Hyperplane in:

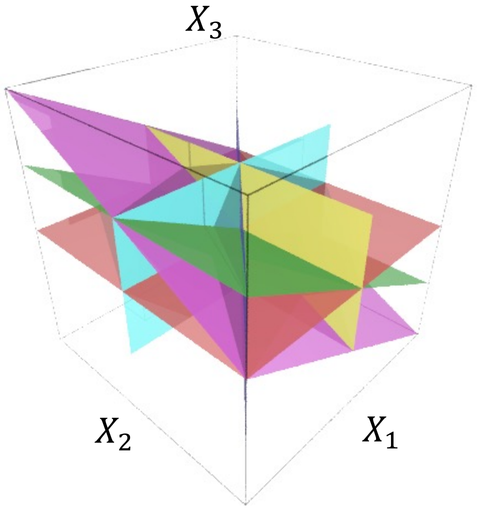
- A 2D space:

$\beta_o + \beta_1 X_1 + \beta_2 X_2 = 0$ Line, that does not pass-through origin



- A 3D space:

$\beta_o + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 = 0$ Plane, that does not pass-through origin



- A p-D space:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

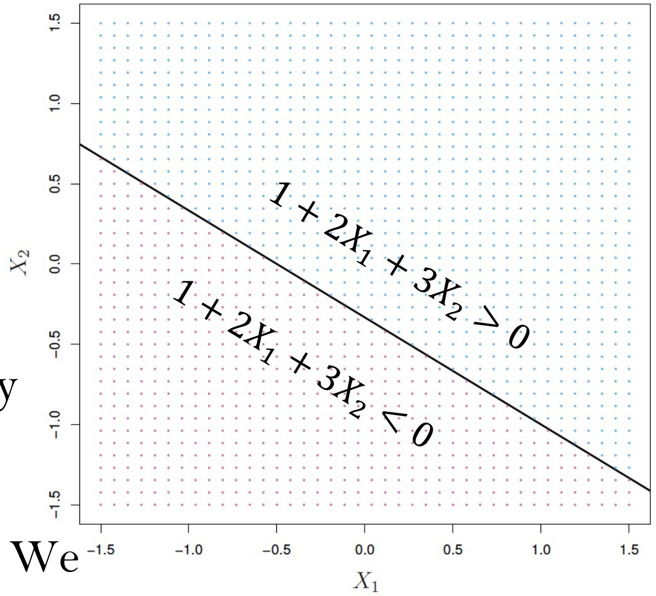
(p-1)dimension, flat, affine subspace.

Hyperplane and Points in Space

$$\beta_o + \beta_1 X_1 + \beta_2 X_2 = 0$$

There are set of points $X = (X_1, X_2)^T$ that satisfy the equation of the hyperplane. (The Points on the line)

And set of other points $X = (X_1, X_2)^T$ that do not satisfy the equation of the hyperplane.



For points that do not satisfy equation of a Hyperplane, We get:

$$\left.\begin{array}{c}\beta_o + \beta_1 X_1 + \beta_2 X_2 < 0\,, \\ or \\ \beta_o + \beta_1 X_1 + \beta_2 X_2 > 0\end{array}\right\}$$

Purple Region

(The Points on either side of the line)

Blue Region

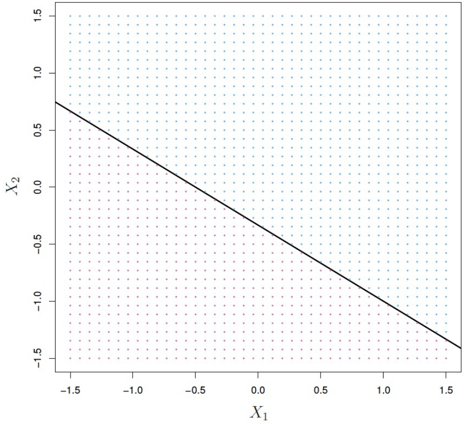
Hyperplane and Points in Space

For p - dimension space, a p dimension point(vector), $X = (X_1, X_2 . X_3 \dots X_p)^T$,

Lies on the hyperplane:

If:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$



Lies on the either side of a hyperplane:

If:

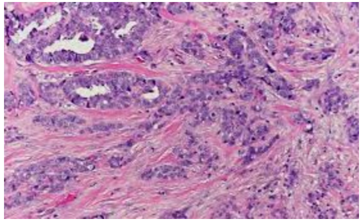
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0$$

OR

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0$$

Conclusion: A hyperplane divides/separates a p-dimensional space into two halves.

Classification using a Separating Hyperplane



Example:

- We measured p different tissue properties that plays role in whether a tissue is cancerous or not.
- We collected data from n different patients (samples).

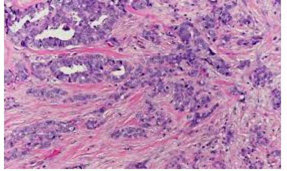
So, we have a data matrix X of size $n{\times}p$.
We can also say, we have n training observation in p dimensional space.

Task:

- We want to classify the observations into:

Cancerous (Y = +1) or
Non-cancerous (Y = -1)

Classification using a Separating Hyperplane



Task:

- We want to classify the observations into:

Cancerous $(Y = +1)$ or

Non-cancerous $(Y = -1)$

n Inputs/samples each of p dimension

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}$$

n responses, either sample is from class -1 or class 1

$$y_1, \ldots, y_n \in \{-1, 1\}$$

If we can find a hyperplane,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

Such that:

For All Cancerous Samples,
i.e., For Samples (x_i) whose response is $y_i = 1$,

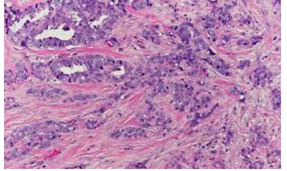
$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0$$

For All Non-cancerous Samples,
i.e., For Samples (x_i) whose response is $y_i = -1$,

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0$$

Job is Done!

Classification using a Separating Hyperplane



For All Cancerous Samples,
i.e., For Samples (x_i) whose response is $y_i = 1$,

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0$$

For All Non-cancerous Samples,
i.e., For Samples (x_i) whose response is $y_i = -1$,

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0$$

Equivalently, our separating hyperplane should be such that:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0 \quad , \forall\, i = 1, 2, 3, \ldots n$$

Property of Hyperplane

Then for any Test Samples: (x^*),
We can evaluate: $f(x^*)$, Sign of f(x) tells us the class, Magnitude tells us how far the sample is from the hyperplane

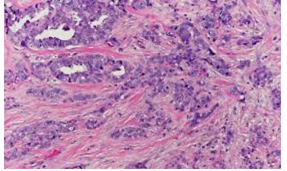
If:

$$f(x^*) > 0 : Test\ Sample\ belongs\ to\ class\ Y = 1, \text{(Cancerous Tissue)}$$

Else:

$$f(x^*) < 0 : Test\ Sample\ belongs\ to\ class\ Y = -1, \text{(Non} - \text{Cancerous)}$$

Classification using a Separating Hyperplane



Example in a 2D space:

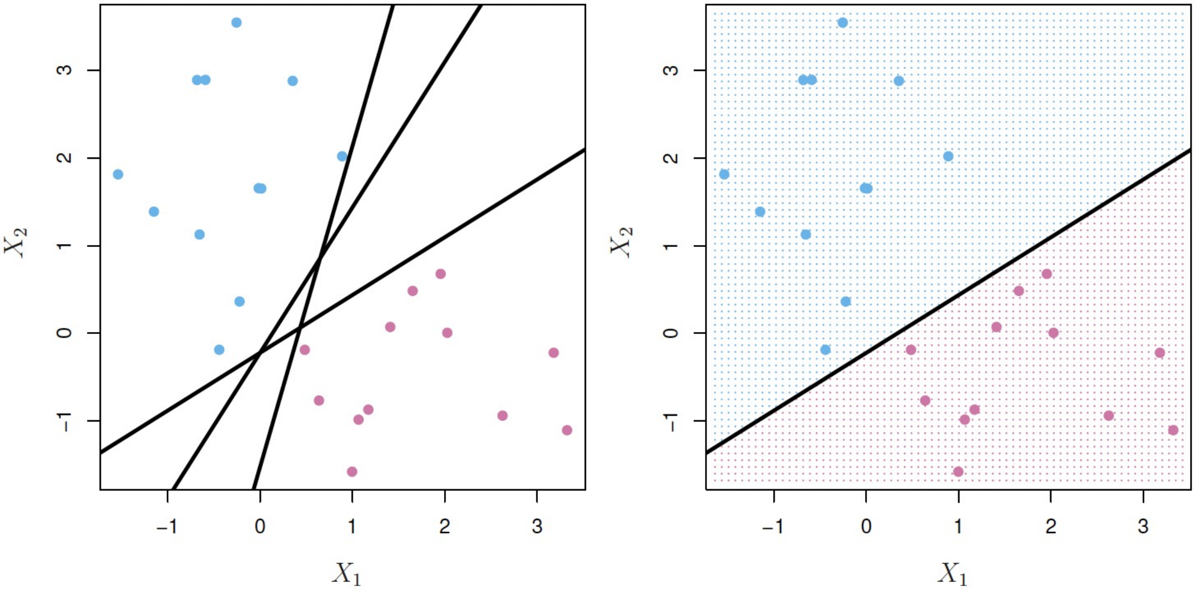
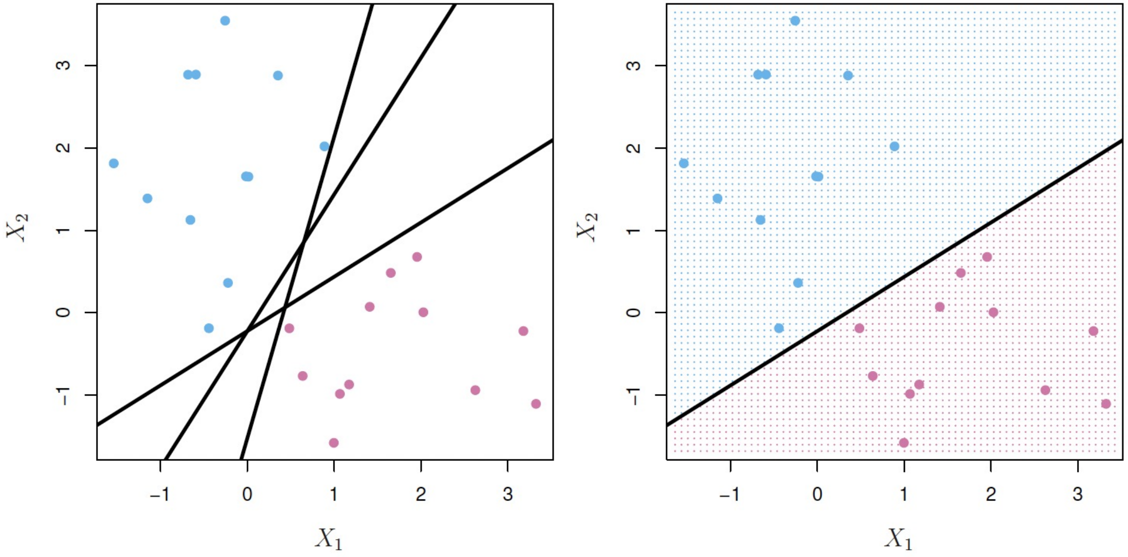


Fig: 9.2, X_1 *and* X_2 are 2 input features. Left: Some observation Samples (training) and MANY separating hyperplanes. Right: One of the separating hyperplanes and training and test samples in either side of hyperplane.

The Maximal Margin Classifier



We could get many separating hyperplanes.
Which one shall we choose?

The Maximal Margin Classifier

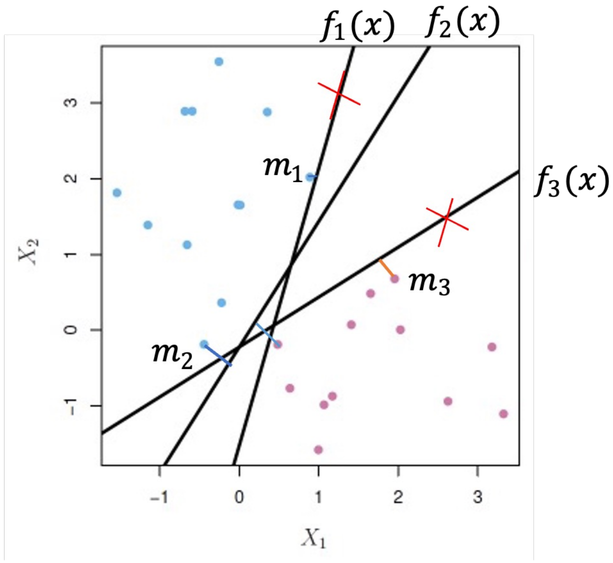
Which one shall we choose?

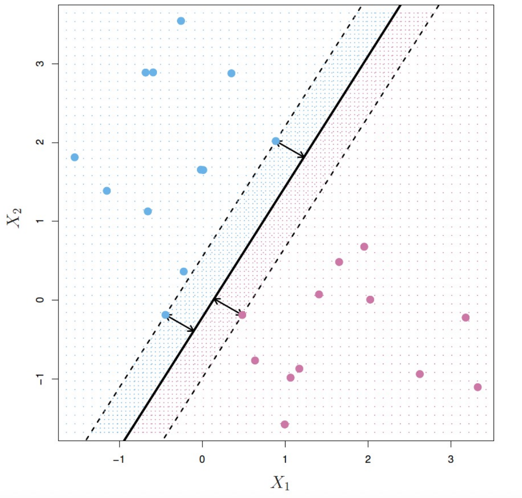
- Maximal Marginal Hyperplane/Optimal Separating Hyperplane
- A Hyperplane Farthest from the training observations.
- We can calculate perpendicular distance from each observation/training sample to the given hyperplane.
- The smallest of such distance is termed Margin.
- We want a hyperplane for which the Margin is Maximum.

  (Hence the term, Maximum Margin Classifier)

The Maximal Margin Classifier

- We want a hyperplane for which the Margin is Maximum.





- Three separating hyperplanes, $f_1(x)$, $f_1(x)$, $f_1(x)$ with corresponding minimum margins m_1, m_2 *and* m_3, respectively.
- $m_2 > m_3 > m_1$, m_2 is maximum among all minimum margins.

i.e., $f_2(x)$ is a maximal minimum margin Separating Hyperplane

The Maximal Margin Classifier

Construction:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{maximize}} \quad M$$

Two Constraints

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1,$$

Note: It does not apply to β_0

$$\underbrace{y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}_{\text{Recall: Property of Separating Hyperplane}} \geq M \quad \forall\, i = 1, \ldots, n.$$

What do these constraints enforce?

- Why $\geq$ M? Why not $\geq$ 0?, as given by property of Hyperplane?
- Why we need the first constraint?

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0$$

Property of Hyperplane

The Maximal Margin Classifier

What do these constraints enforce?

- Why $\geq$ M? Why not 0?
- Why we need the first constraint?

Two Constraints

$$\begin{cases} \text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \\ y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M \quad \forall\, i = 1, \ldots, n. \end{cases}$$

Let's first look at the second constraint:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0$$

Why $\geq$ M? Why not 0?

Property of Hyperplane: >0: Enforces each samples are on the correct side of the Hyperplane.

Adding Constraint , $\geq$ M (M is positive number)

Enforces: the minimum distance from the sample is at least M.

The Maximal Margin Classifier

What do these constraints enforce?

- Why $\geq$ M? Why not 0?
- Why we need the first constraint?

Two Constraints

$$\begin{cases} \text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \\ y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M \quad \forall\, i = 1, \ldots, n. \end{cases}$$

First constraint:

Its not a necessary condition, but allows us to calculate the perpendicular distance from any sample to the hyperplane as:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})$$

if $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} = 0$,Then, for any $k \neq 0$

$k(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) = 0$, as well.

The Maximal Margin Classifier

First constraint:

Its not a necessary condition, but allows us to calculate the perpendicular distance from any sample to the hyperplane as:

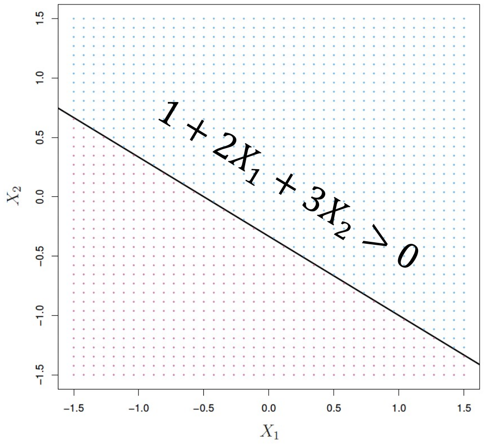
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})$$

Example:

$1 + 2X_1 + 3X_2 = 0$: *Hyperplane*

$\beta_0 = 1, \beta_1 = 2, \beta_2 = 3$

Point: (1,-1) lies on this hyperplane



k = 5, => $New\ \beta = 5 * old\ \beta, i.e., \beta_0, = 5,\ \beta_1 = 10, \beta_2 = 15$

$5 + 10X_1 + 15X_2 = 0$: *New Hyperplane*

Our Point: (1,-1) still lies on this New hyperplane.

No matter which hyperplane we had chosen, it would still correctly separate our data.
So, not a necessary condition.

The Maximal Margin Classifier

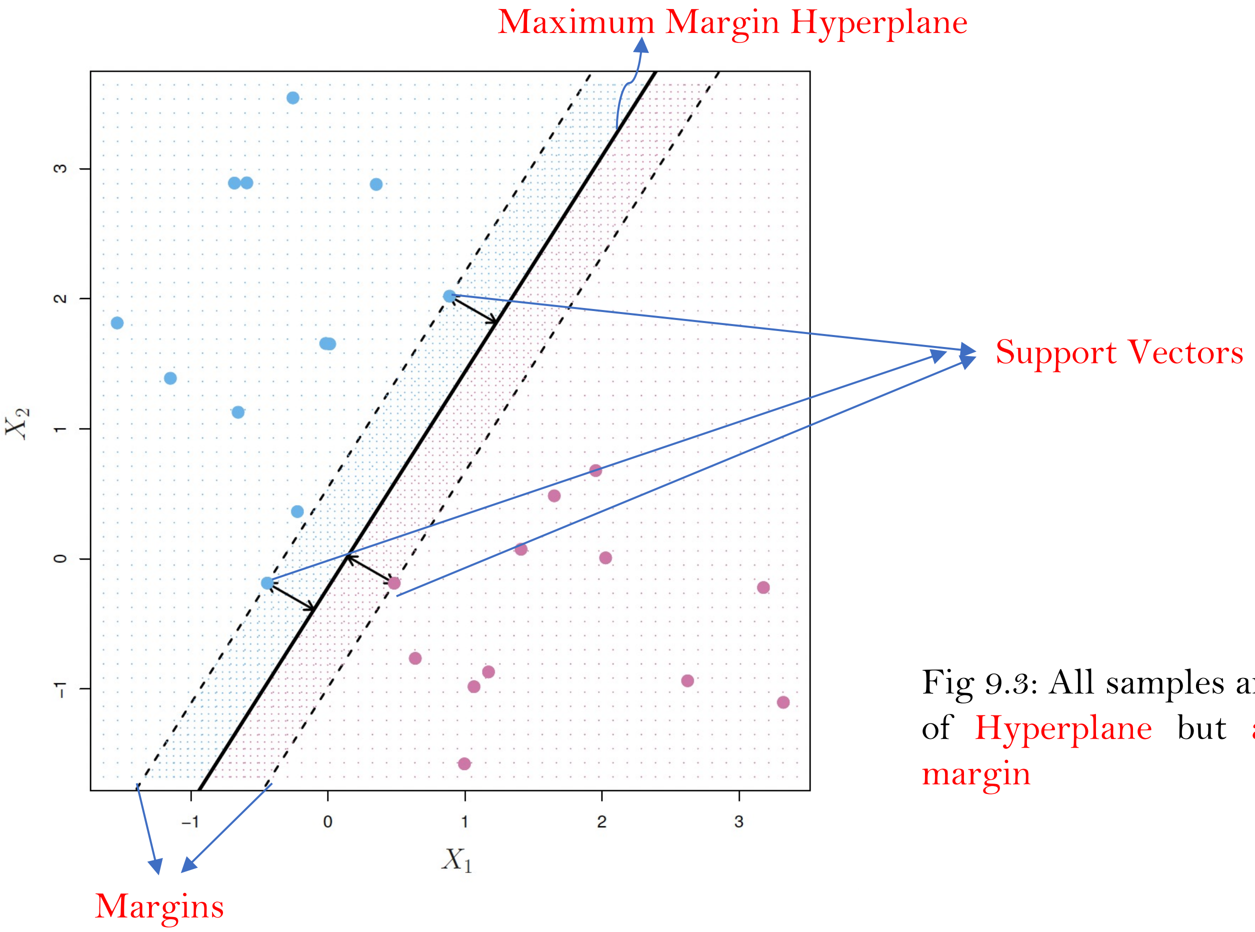
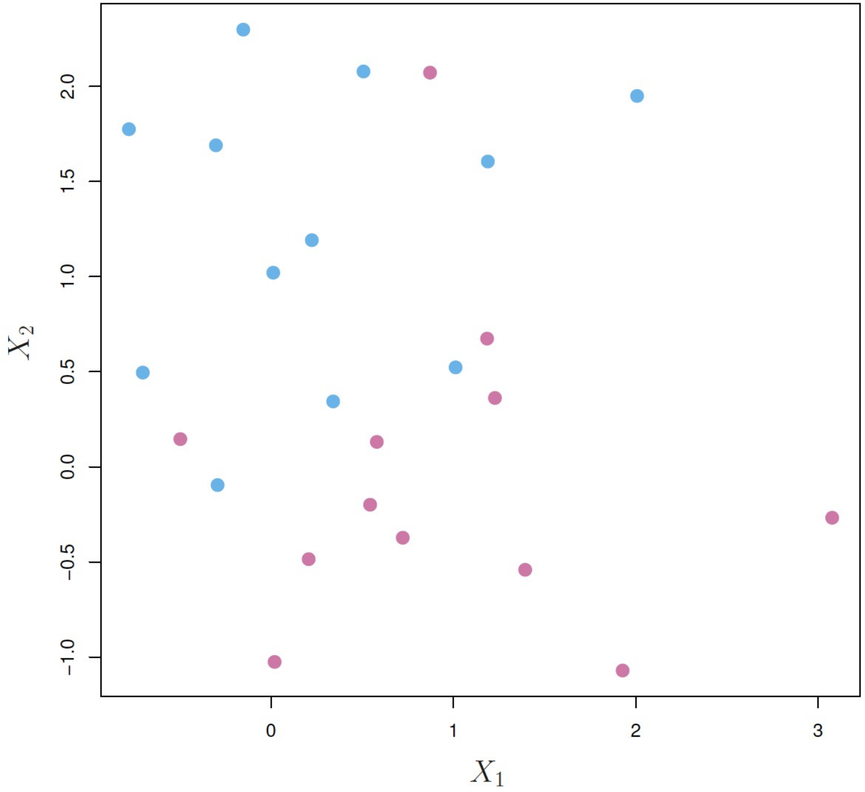


Fig 9.3: All samples are not only on the correct side of Hyperplane but also on the correct side of margin

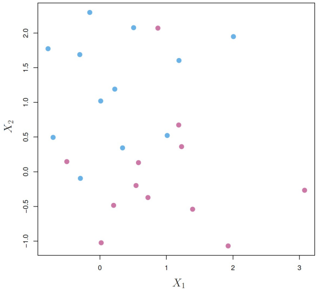
The Non-Separable Case

Data are not always linearly separable.



Example: Data not linearly separable. The separating Hyperplane does not exist.

Support Vector Classifier (SVC)



- Data are not always linearly separable. Separating hyperplane might not exist.
- Even when it exists, Perfectly separating Hyperplane may overfit the data, Not stable Learning Model.

SVCs

Introduce Some Tolerance.

i.e., Allow some misclassifications for some samples.

The process is called Support Vector Classifier (Soft margin)

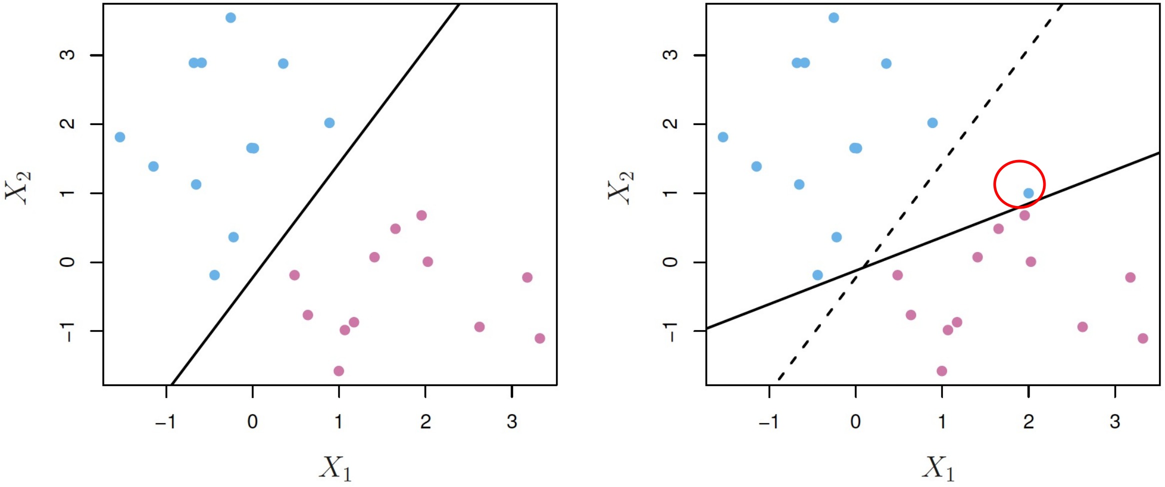
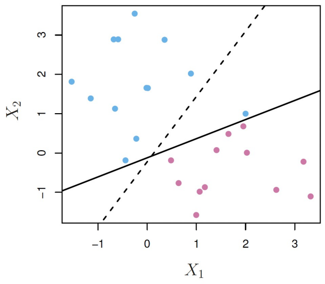


Fig: Left, A perfect separating Hyperplane. Right: Illustration: Just one extra sample point changed the separating hyperplane from a solid line on left to the solid line on right fig. Not stable learning.

Support Vector Classifier



$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,\,M}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

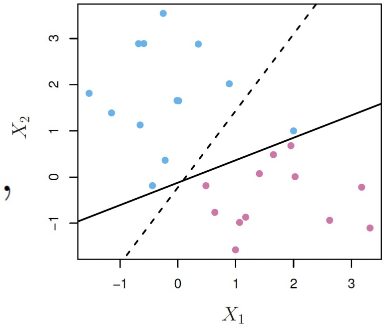
- $\epsilon_i, \epsilon_2 \cdots \epsilon_n$, are called slack variables that allows individual sample to be on the wrong side of margin or even wrong side of the separating hyperplane.
- C is a non-negative tuning parameter.

Once we know model parameters β, Then to classify Test Samples:

Compute: $$f(x^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^*$$ And Check Sign of $f(x^*)$

$f(x^*) > 0 \Rightarrow Class\ 1$ or $f(x^*) < 0 \Rightarrow Class\ -1,$

Support Vector Classifier



Consequence of Slack Variable ϵ:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

- When $\epsilon_i = 0$,

Sample x_i is on the correct side of the margin.

- When $\epsilon_i > 0$,

Sample x_i is on the wrong side of the margin, but on the correct side of separating hyperplane.

- When $\epsilon_i > 1$,

Sample x_i is not only the wrong side of the margin, but also on the wrong side of the separating hyperplane.

Support Vector Classifier

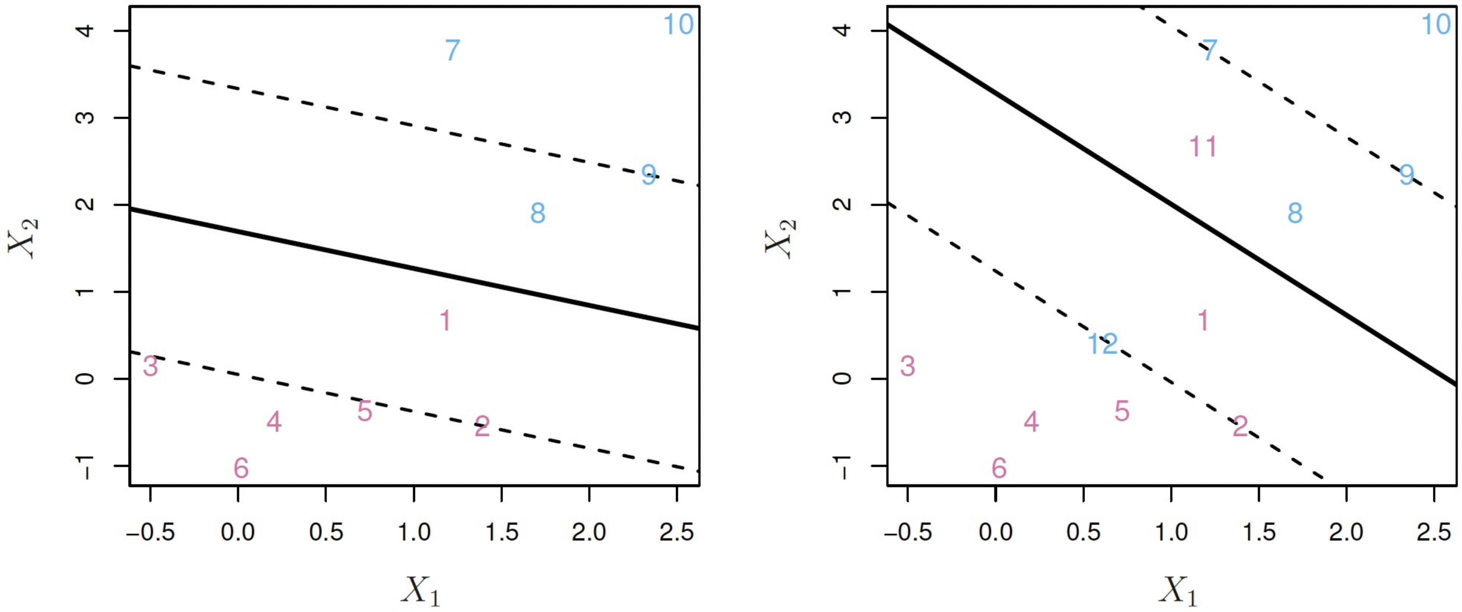


Figure 9.6, Support vector classifier. Left: some Samples only on the wrong side of margin (Still all correct classification), Right: some samples on wrong side of both margin and hyperplane (misclassification)

Support Vector Machines

Support Vector Machines

Support Vector Machines: Classification with Nonlinear Decision Boundaries

Recall : In Polynomial Regression/Regression with interaction variables:
We created transformation of input features/interaction variables as new variables.

Similarly,
Instead of fitting support vector classifiers using p features, $X_1, X_2. \dots X_p$,
We could fit a support vector classifier using $2p$ features, $X_1, X_1^2, X_2, X_2^2. \dots X_p X_p^2$,

Support Vector Machines: Classification with Nonlinear Decision Boundaries

Then, the Support Vector Classifier Model

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,\, M}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

Becomes

$$\underset{\beta_0,\beta_{11},\beta_{12},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n,\, M}{\text{maximize}} \quad M$$

$$\text{subject to } y_i\left(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2\right) \geq M(1 - \epsilon_i),$$

$$\sum_{i=1}^{n} \epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1.$$

Support Vector Machines: Classification with Nonlinear Decision Boundaries

$$\underset{\beta_0,\beta_{11},\beta_{12},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n,\, M}{\text{maximize}} \quad M$$

$$\text{subject to } y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right) \geq M(1-\epsilon_i),$$

$$\sum_{i=1}^{n}\epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1.$$

Interesting Points:

1. The decision Boundary is nonlinear in the original input feature space.
2. But the decision Boundary is linear in the modified, enlarged (high-dimensional) feature space.

Interesting Questions:

1. Why do we enlarge the feature space only with degree of 2?
2. Do we always know what is the best enlarging method?

Support Vector Machines: Classification with Nonlinear Decision Boundaries

Interesting Questions:

1. Why do we enlarge the feature space only with degree of 2?

We do not have to do only with degree of 2. We can use many ways to enlarge the feature space.

2. Do we always know what is the best feature space enlarging method?

No, we can use different methods to enlarge feature dimension, but we do not generally know which one is the best.

Support Vector Machines provide efficient computational tool to expand feature space (dimension of feature space) by use of *kernels*.

Support Vector Machines: Classification with Nonlinear Decision Boundaries

Support Vector Machines provide efficient computational tool to expand feature space (dimension of feature space) by use of *kernels* .

The Linear Support Vector Classifier is expressed as :

$$\boxed{f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle}$$

Inner Product

x_i : Training Samples
x : Test sample, new sample
α_i: Coeff of Support Vector Model

Support Vector Machines: Classification with Nonlinear Decision Boundaries

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle$$

What is interesting about this classifier?

1. To estimate parameters $\alpha_i, i = 1, 2, \dots n,$ we only need to compute $\binom{n}{2}$ inner products $<x_i, x_i{}'>$ between all training samples.

$\binom{n}{2} = \frac{n(n-1)}{2}$

Note: We have not discussed how to calculate solve the optimization to compute α_i

2. In support vector classifier, only training observation that are support vectors (sample that lie on margin) are significant.
So, if S is a set of support vectors, Then:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

Support Vector Machines: Classification with Nonlinear Decision Boundaries

$$\boxed{f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle}$$

Conclusion:

To do classification using Support Vector, all we need is inner product.

$$\underset{\beta_0,\beta_{11},\beta_{12},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n,\ M}{\text{maximize}} \quad M$$

$$\text{subject to } y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right) \geq M(1-\epsilon_i),$$

$$\sum_{i=1}^{n}\epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1.$$

Support Vector Machines: Classification with Nonlinear Decision Boundaries

Inner Product, Dot Product and kernels

$$\boxed{f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle}$$

Let K be the kernel that generalize the inner product $<x_i, x_i'>$

i.e. $K(x_i, x_i') = <x_i, x_i'>$

One popular inner product we know is a dot product:

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

Support Vector Machines: Classification with Nonlinear Decision Boundaries

$$\boxed{f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle}$$

When we choose inner product to be the dot product

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

We get support vector classifier, and it is known as a linear kernel.
But, we could choose many different kernels.

Support Vector Machines: Classification with Nonlinear Decision Boundaries

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

We could choose many different kernels.

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d.$$

Polynomial Kernel

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2).$$

Radial kernel

Support Vector Machines: Classification with Nonlinear Decision Boundaries

$$\boxed{f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle}$$

Question: Why not expand the features itself using many polynomials and then solve regression/classification ?

Ans: Computational advantages. Using kernels:

1. We do not need to find explicit representation in new feature space.
2. Feature dimension can be very large (for example the implicit feature dimension of radial kernel is infinite).
3. All computations can be represented as inner product.

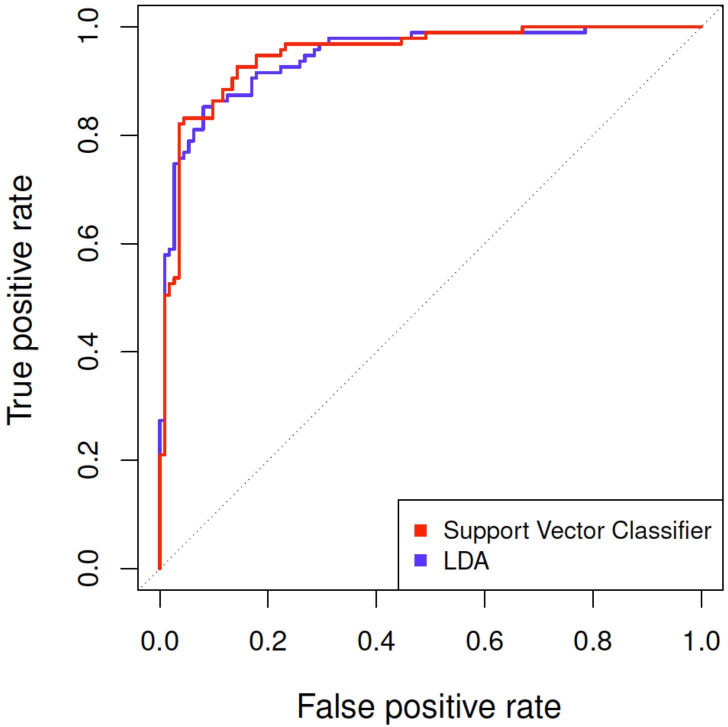
SVM and Relation to Regression

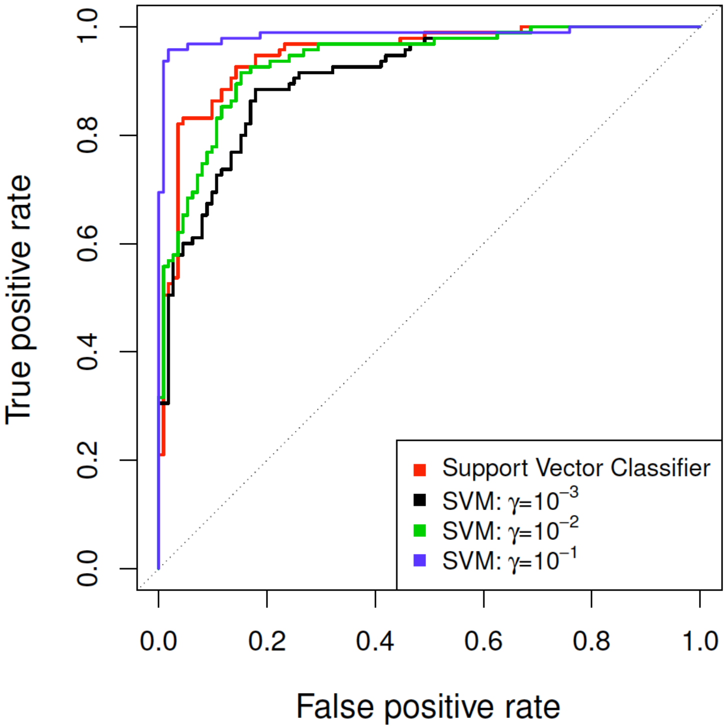
SVM has equivalent form of loss + penalty similar to Ridge Regression

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \left\{ \sum_{i=1}^{n} \max\left[0, 1 - y_i f(x_i)\right] + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

SVM Example

Heart Disease Data

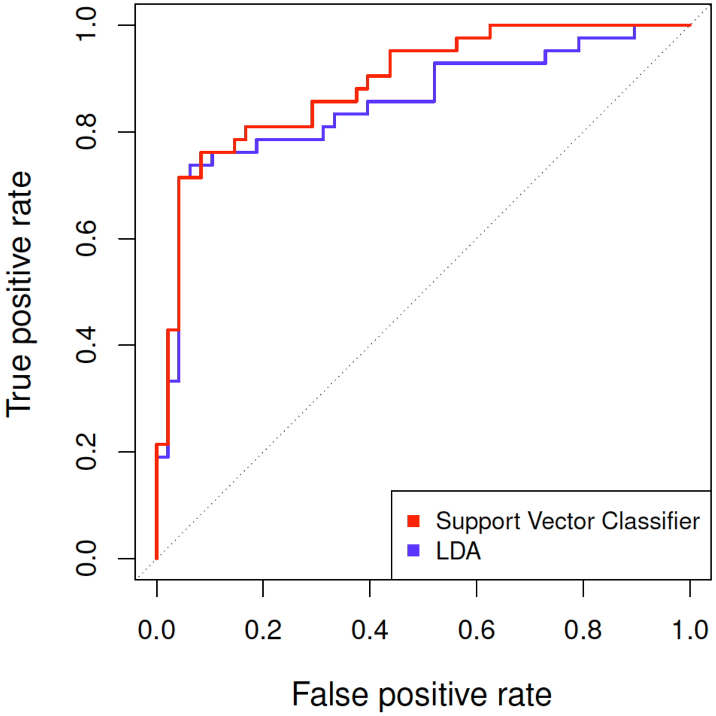


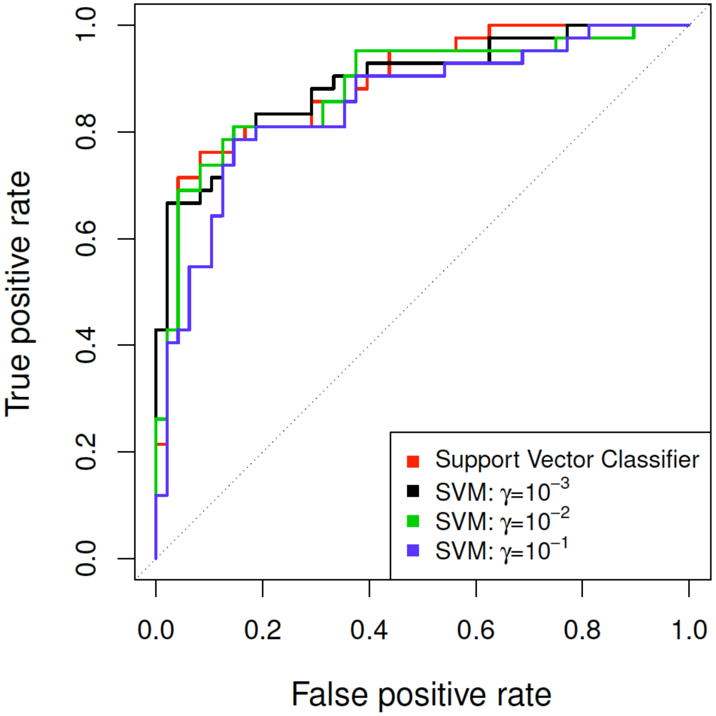


Performance for Training Data

SVM Example

Heart Disease Data





Performance for Test Data

VC Dimension

What does it tell us?
What is it?

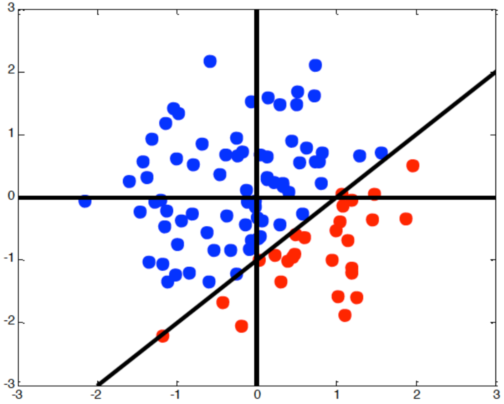
VC Dimension

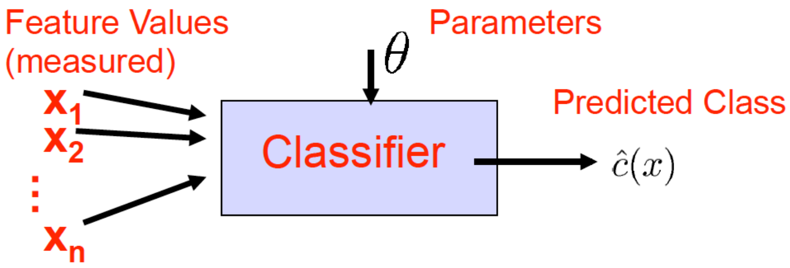
- Captures the complexity of learning model.
- How well the model can represent the data?

Rest of the Slides are based on Alexander Ihler and Andrew Moore's Slides

Learners and Complexity

- We've seen many versions of underfit/overfit trade-off
  - Complexity of the learner
  - "Representational Power"
- Different learners have different power



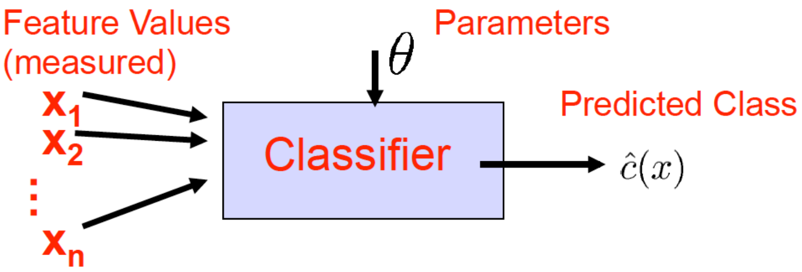


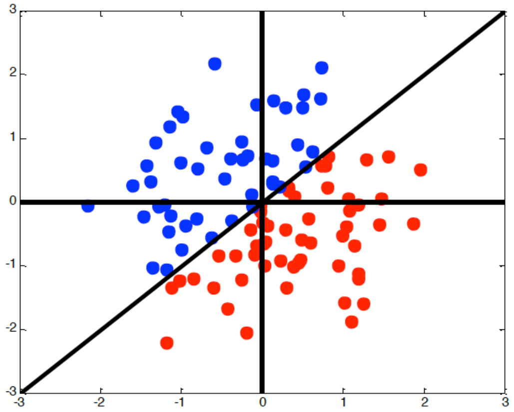
Example:

$$\hat{c}(x) = \mathrm{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$$

Learners and Complexity

- We've seen many versions of underfit/overfit trade-off
  - Complexity of the learner
  - "Representational Power"
- Different learners have different power



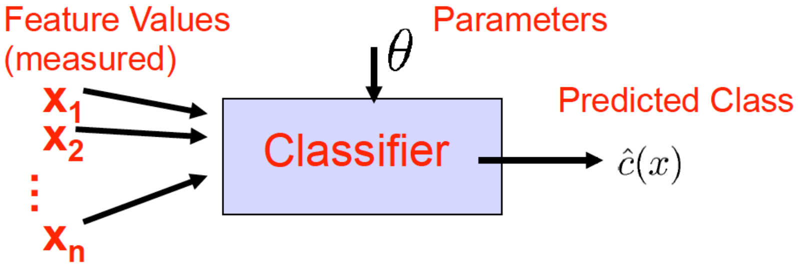


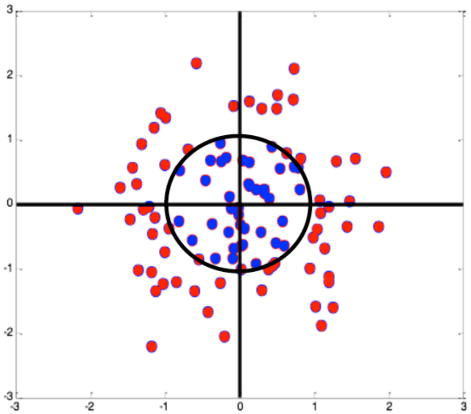
Example:

$$\hat{c}(x) = \text{sign}(\theta_1 x_1 + \theta_2 x_2)$$

Learners and Complexity

- We've seen many versions of underfit/overfit trade-off
  - Complexity of the learner
  - "Representational Power"
- Different learners have different power





Example:

$$\hat{c}(x) = \text{sign}(\,(x_1^2 + x_2^2) - \theta_0)$$

Learners and Complexity

- We've seen many versions of underfit/overfit trade-off
  – Complexity of the learner
  – "Representational Power"
- Different learners have different power

- Usual trade-off:
  – More power = represent more complex systems, might overfit
  – Less power = won't overfit, but may not find "best" learner

- How can we quantify representational power?
  – Not easily…
  – One solution is VC (Vapnik-Chervonenkis) dimension

Some notation

- Assume training data are iid from some distribution p(x,y)

- Define “risk” and “empirical risk”
  - These are just “long term” test and observed training error

$$R(\theta) = \text{TestError} = \mathbb{E}[\mathbb{1}[c \neq \hat{c}(x\,;\,\theta)]]$$

$$R^{\text{emp}}(\theta) = \text{TrainError} = \frac{1}{m}\sum_i \mathbb{1}[c^{(i)} \neq \hat{c}(x^{(i)}\,;\,\theta)]$$

- How are these related? Depends on overfitting…
  - Underfitting domain: pretty similar…
  - Overfitting domain: test error might be lots worse!

VC Dimension and Risk

- Given some classifier, let H be its VC dimension
  - Represents "representational power" of classifier

$$R(\theta) = \text{TestError} = \mathbb{E}[\mathbb{1}[c \neq \hat{c}(x\,;\,\theta)]]$$

$$R^{\text{emp}}(\theta) = \text{TrainError} = \frac{1}{m}\sum_i \mathbb{1}[c^{(i)} \neq \hat{c}(x^{(i)}\,;\,\theta)]$$

- With "high probability" (1-η), Vapnik showed

$$\text{TestError} \leq \text{TrainError} + \sqrt{\frac{H\log(2m/H) + H - \log(\eta/4)}{m}}$$

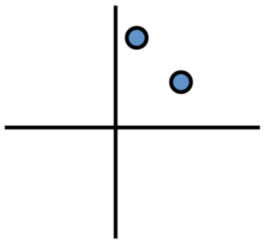
Shattering

- We say a classifier f(x) can shatter points $x^{(1)} \ldots x^{(h)}$ iff For *all* $y^{(1)} \ldots y^{(h)}$, f(x) can achieve zero error on training data $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, … $(x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)

- Can $\boxed{f(x;\theta) = \text{sign}(\theta_0 + \theta_1 x_1 + \theta_2 x_2)}$ shatter these points?

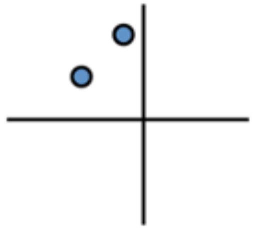
Some line that does not pass-through origin



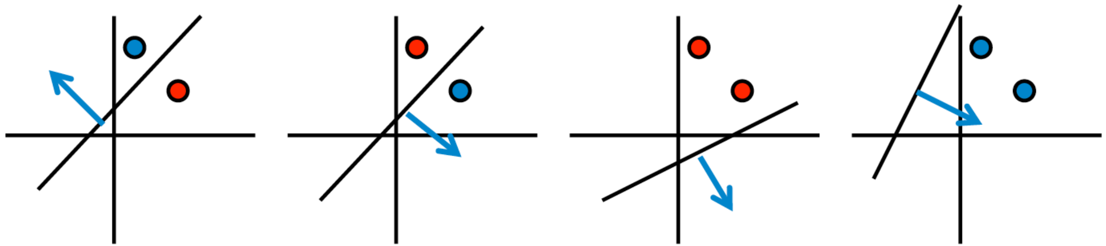
Shattering

- We say a classifier f(x) can shatter points $x^{(1)} \ldots x^{(h)}$ iff
  For *all* $y^{(1)} \ldots y^{(h)}$, f(x) can achieve zero error on
  training data $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, … $(x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)



- Can $f(x;\theta) = \text{sign}(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$ shatter these points?
- Yes: there are 4 possible training sets…



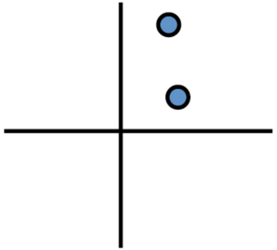
Shattering

- We say a classifier f(x) can shatter points $x^{(1)} \ldots x^{(h)}$ iff For *all* $y^{(1)} \ldots y^{(h)}$, f(x) can achieve zero error on training data $(x^{(1)},y^{(1)}), (x^{(2)},y^{(2)}), \ldots (x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)

- Can $\boxed{f(x;\theta) = \text{sign}(x_1^2 + x_2^2 - \theta)}$ shatter these points?

Our classifier now is a Circle with radius of θ

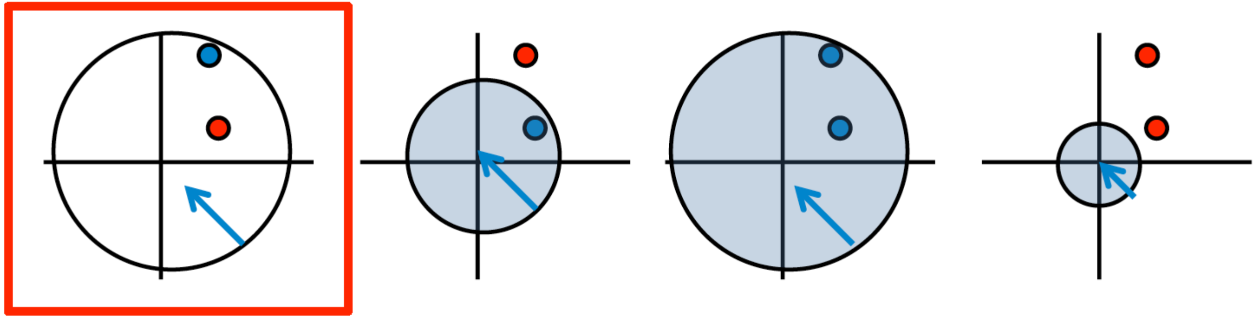


Shattering

- We say a classifier f(x) can shatter points $x^{(1)} \ldots x^{(h)}$ iff For *all* $y^{(1)} \ldots y^{(h)}$, f(x) can achieve zero error on training data $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, … $(x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)

- Can $f(x;\theta) = \text{sign}(x_1^2 + x_2^2 - \theta)$ shatter these points?
- Nope!

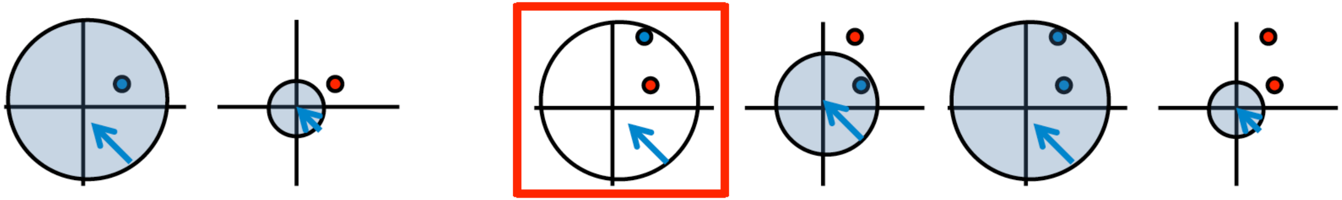


VC Dimension

- The VC dimension H is defined as

The maximum number of points h that *can be arranged* so that f(x) can shatter them

- Example: what's the VC dimension of the (zero-centered) circle, $f(x;\theta) = \text{sign}(x_1^2 + x_2^2 - \theta)$?
- VCdim = 1 : can arrange one point, cannot arrange two (previous example was general)

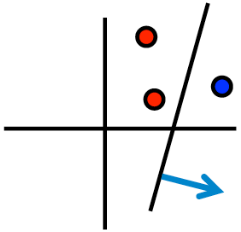


(c) Alexander Ihler

VC Dimension

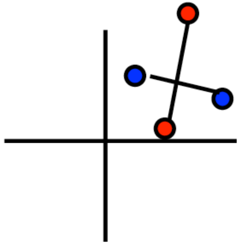
- Example: what's the VC dimension of the two-dimensional line, $f(x;\theta) = \text{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$?

- VC dim >= 3? Yes



- VC dim >= 4? No…

  Any line through these points must split one pair (by crossing one of the lines)



In General, for linear classifier in a d dimension with a constant term:

VC dimension = d+1.