Lecture 9: Naive Bayes, SVM, Kernels

Instructor: Saravanan Thirumuruganathan

Outline

1. Probability basics
2. Probabilistic Interpretation of Classification
3. Bayesian Classifiers, Naive Bayes
4. Support Vector Machines

Probability Basics

Sample Space

- **Sample Space:** A space of events that we assign probabilities to
- Events can be binary, multi-valued, or continuous
- Events are mutually exclusive
- Examples:
  - Coin flip: {head, tail}
  - Die roll: {1,2,3,4,5,6}
  - English words: a dictionary
  - Temperature: $\mathbb{R}_+$ (Kelvin)

Random Variable

- A variable, X, whose domain is the sample space, and whose value is somewhat uncertain
- Examples:
  - X = coin flip outcome
  - X = first word in tomorrow's headline news
  - X = tomorrow's temperature

Probability for Discrete Events

- Probability $P(X = a)$ is the fraction of times x takes value a
- Often we write it as P(a)
- Examples:
  - Fair Coin: P(head)=P(tail)=0.5
  - Slightly Biased Coin: P(head)=0.51, P(tail)=0.49
  - Two Face's Coin: P(head)=1, P(tail)=0
  - Fair Dice: P(getting 1 in a die roll) = 1/6

Probability for Discrete Events

- P(A="head or tail in a fair coin")

$$0.5 + 0.5 = 1$$

- P(A="even number in a fair dice roll")

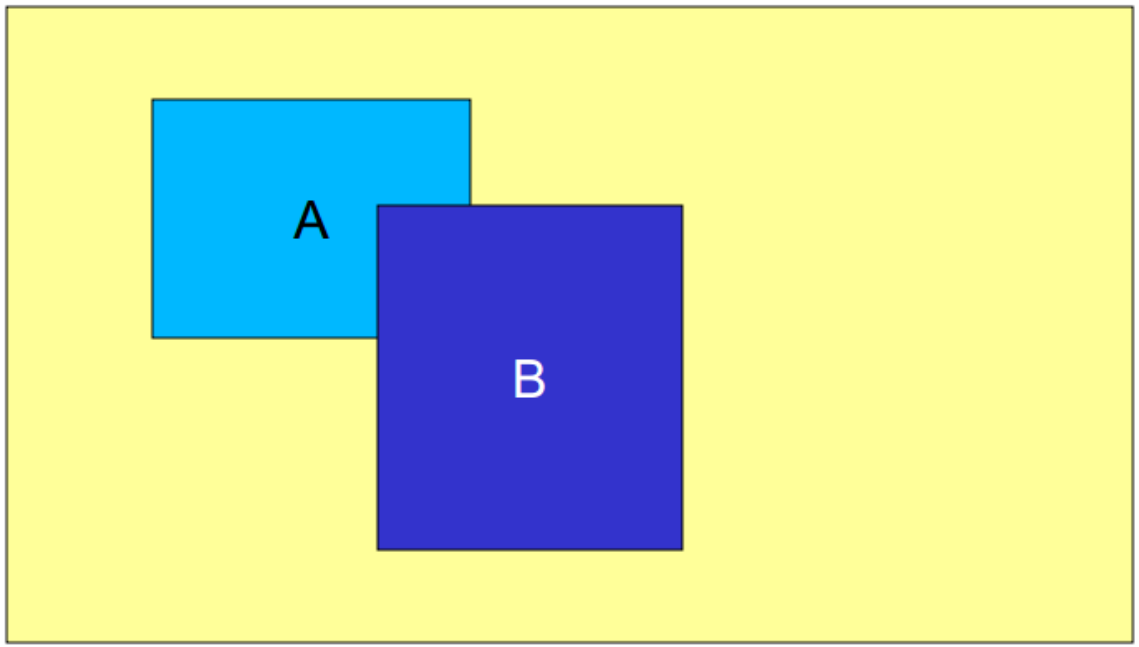
$$1/6 + 1/6 + 1/6 = 0.5$$

- P(A="two dice rolls sum to 2 in a fair dice")

$$1/6 * 1/6 = 1/36$$

Axioms of Probability

- $P(A) \in [0,1]$
- $P(true)=1, P(false)=0$
- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



Simple Corollaries

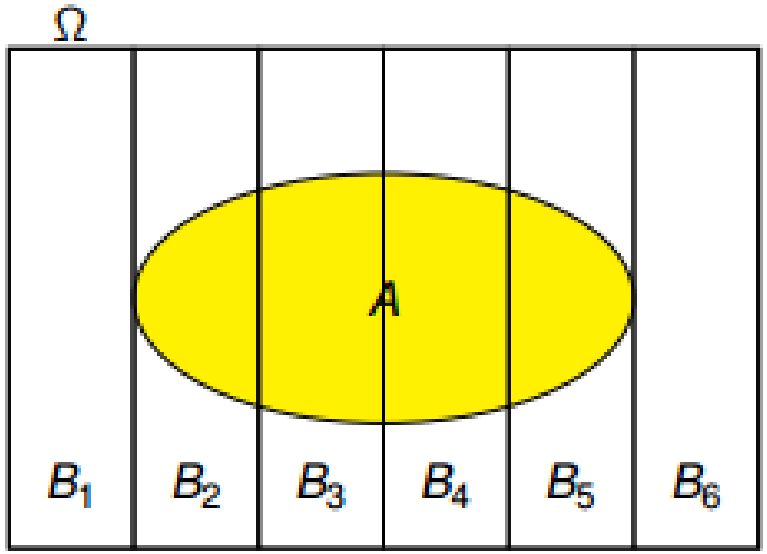
- $P(A') = 1 - P(A)$
- If A can take k different values $a_1, \ldots, a_k$,

$$P(A = a_1) + \ldots + P(A = a_k) = 1$$

- **Law of Total Probability:**
  - $P(A) = P(A \cap B) + P(A \cap B')$
  - $P(A) = \sum_{i=1}^{k} P(A \cap B = b_i)$ if B takes k values $b_1, \ldots, b_k$

Law of Total Probability[1]

Definition: A partition of the sample space Ω is a collection of disjoint events $B_1, B_2, \ldots B_k$ whose union is Ω. Such a partition divides any set A into disjoint pieces:



[1] http://people.reed.edu/~jones/Courses/P02.pdf

Probability Table

- Weather

Sunny	Cloudy	Rainy
200/365	100/365	65/365

- P(Weather = sunny) = P(sunny) = 200/365
- P(Weather) = {200/365, 100/365, 65/365}

Joint Probability Table

		weather		
		Sunny	Cloudy	Rainy
temp	hot	150/365	40/365	5/365
	cold	50/365	60/365	60/365

- P(temp=hot, weather=rainy) = P(hot, rainy) = 5/365
- The full joint probability table between N variables, each taking k values, has k^N entries (that's a lot!)

Marginal Probability Table

- Sum over other variables

		weather		
		Sunny	Cloudy	Rainy
temp	hot	150/365	40/365	5/365
	cold	50/365	60/365	60/365
	Σ	200/365	100/365	65/365

P(Weather)={200/365, 100/365, 65/365}

- The name comes from the old days when the sums are written on the margin of a page

Marginal Probability Table

- Sum over other variables

		weather			
		Sunny	Cloudy	Rainy	Σ
temp	hot	150/365	40/365	5/365	195/365
	cold	50/365	60/365	60/365	170/365

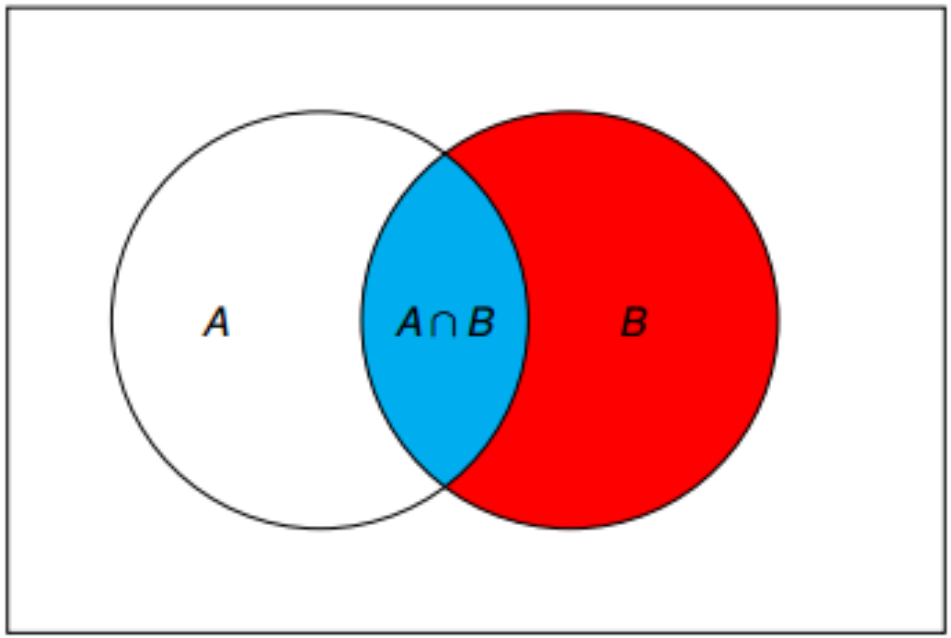
P(temp)={195/365, 170/365}

- This is nothing but $P(B) = \sum_{i=1...k} P(B \wedge A=a_i)$, if A can take *k* values

Conditional Probability

- $P(A = a|B = b)$ = fraction of times when random variable A took a value of a, within the region where random variable $B = b$

The definition $\mathbb{P}(A \mid B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$ restricts the sample space to B, and rescales to give $\mathbb{P}(B|B) = 1$:



Conditional Probability

- Consider a roll of a fair dice
- A: it rolled 1. $P(A) = 1/6$
- B: it rolled an odd number. $P(B) = 3/6 = 0.5$
- Suppose, I knew that B happened. What is the probability that A happened?

Conditional Probability

- Consider a roll of a fair dice
- A: it rolled 1. $P(A) = 1/6$
- B: it rolled an odd number. $P(B) = 3/6 = 0.5$
- Suppose, I knew that B happened. What is the probability that A happened?
- $$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1/6}{1/2} = \frac{1}{3}$$

Conditional Probability

- **Conditional Probability:** $P(A|B) = \frac{P(A \cap B)}{P(B)}$
- **Multiplication Rule:** $P(A \cap B) = P(A|B)P(B)$
- **Chain Rule:**
  - $P(A_1, A_2) = P(A_1)P(A_2|A_1)$
  - $$\begin{aligned} P(A_1, A_2, A_3) &= P(A_3|A_1, A_2)P(A_1, A_2) \\ &= P(A_3|A_1, A_2)P(A_2|A_1)P(A_1) \end{aligned}$$
  - $$\begin{aligned} P(A_1, A_2, A_3) &= P(A_1|A_2, A_3)P(A_2, A_3) \\ &= P(A_1|A_2, A_3)P(A_2|A_3)P(A_3) \end{aligned}$$
  - $P\left(\cap_{i=1}^{k} A_i\right) = \prod_{i=1}^{k} P\left(A_i | \cap_{j=1}^{i-1} A_j\right)$

Bayes Theorem

- $P(A|B) = \frac{P(A \cap B)}{P(B}$)
- $P(B|A) = \frac{P(A \cap B)}{P(A}$)
- Proof of Bayes Theorem:

$$
\begin{aligned}
P(A \cap B) &= P(A|B)P(B) \\
&= P(B|A)P(A) \\
P(A|B)P(B) &= P(B|A)P(A) \\
P(A|B) &= \frac{P(B|A)P(A)}{P(B)}
\end{aligned}
$$

Independence

- Two events A, B are independent, if (all 3 definitions are equivalent)
  - $P(A \cap B) = P(A)P(B)$
  - $P(A|B) = P(A)$
  - $P(B|A) = P(B)$

Independence Misused

A famous statistician would never travel by airplane, because he had studied air travel and estimated that the probability of there being a bomb on any given flight was one in a million, and he was not prepared to accept these odds.

One day, a colleague met him at a conference far from home. "How did you get here, by train?"

"No, I flew"

"What about the possibility of a bomb?"

"Well, I began thinking that if the odds of one bomb are 1:million, then the odds of two bombs are (1/1,000,000) x (1/1,000,000). This is a very, very small probability, which I can accept. So now I bring my own bomb along!"

An innocent old math joke

Independence

- Independence between random variables is typically obtained via domain knowledge
- Suppose A and B be two independent random variables that can take k different values $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$
- The joint probability table typically has k^2 parameters
- If random variables are independent, then only $2k - 2$ parameters
  - $k = 2$, 4 vs 2
  - $k = 10$, 100 vs 18
  - $k = 100$, $10,000$ vs 198
- This is something great for data mining!

Conditional Independence

- Random variables can be dependent, but **conditionally independent**
- Your house has an alarm
  - Neighbor John will call when he hears the alarm
  - Neighbor Mary will call when she hears the alarm
  - Assume John and Mary dont talk to each other
- JohnCall independent of MaryCall?

Conditional Independence

- Random variables can be dependent, but **conditionally independent**
- Your house has an alarm
  - Neighbor John will call when he hears the alarm
  - Neighbor Mary will call when she hears the alarm
  - Assume John and Mary dont talk to each other
- JohnCall independent of MaryCall?
  - No - If John called, likely the alarm went off, which increases the probability of Mary calling
  - $P(MaryCall|JohnCall) \neq P(MaryCall)$

Conditional Independence

- If we know the status of the alarm, JohnCall won't affect Mary at all
- $P(MaryCall|Alarm, JohnCall) = P(MaryCall|Alarm)$
- We say JohnCall and MaryCall are **conditionally independent**, given Alarm
- In general A, B are conditionally independent given C
  - $P(A|B, C) = P(A|C)$ or
  - $P(B|A, C) = P(B|C)$ or
  - $P(A, B|C) = P(A|C) * P(B|C)$

Probabilistic Interpretation of Classification

Probabilistic Classifiers

- Type of classifiers that, given an input, produces a *probability distribution* over a set of classes
  - Probability that this email is spam is X and not spam is Y
  - Probability that this person has tumour is X and no tumour is Y
  - Probability that this digit is 0 is X, 1 is Y, ...
- Most state of the art classifiers are probabilistic
- Even k-NN and Decision trees have probabilistic interpretations

Prior Probability

- $P(A)$: Prior or unconditional probability of A
- Your belief in A in the absence of additional information
- Uninformative priors
  - Principle of indifference: Assign equal probabilities to all possibilities
  - Coin toss, P(head)=P(tail)=1/2
- Often you get from domain knowledge or from data
- P(email is spam) = 0.8

Conditional Probability as Belief Update

- $P(A)$: Prior belief in A
- $P(A|B)$: Belief after obtaining information B
- $P(A|B, C)$: Belief after obtaining information B and C

Conditional Probability as Belief Update[2]

- Suppose you work as security guard in Airport
- Your job: look at people in security line and choose some for additional screening
- You want to pick passengers with high "risk"
- A: Passenger is high risk
- By experience, you know only 0.1% of passengers are high risk (Prior probability)

[2] http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work

Conditional Probability as Belief Update[3]

- Consider a random person:
  - The probability that this person is high risk is A is 0.1%
  - Suppose you notice that the person is male

[3] http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work

Conditional Probability as Belief Update[3]

- Consider a random person:
  - The probability that this person is high risk is A is 0.1%
  - Suppose you notice that the person is male
    - There are more male criminals than female ones
  - The passenger is nervous

[3] http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work

Conditional Probability as Belief Update[3]

- Consider a random person:
  - The probability that this person is high risk is A is 0.1%
  - Suppose you notice that the person is male
    - There are more male criminals than female ones
  - The passenger is nervous
    - Most criminals are nervous but most normal passengers are not
  - The passenger is a kid

[3] `http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work`

Conditional Probability as Belief Update[3]

- Consider a random person:
  - The probability that this person is high risk is A is 0.1%
  - Suppose you notice that the person is male
    - There are more male criminals than female ones
  - The passenger is nervous
    - Most criminals are nervous but most normal passengers are not
  - The passenger is a kid

[3] `http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work`

Conditional Probability as Belief Update

Then we observe candies drawn from some bag:

What kind of bag is it? What flavour will the next candy be?

Conditional Probability

- $X, A = \langle A_1, A_2, \ldots, A_d \rangle$: Input feature vector
- Y, C: Class value to predict
- $P(C|A)$ vs $P(A|C)$

Conditional Probability

- $X, A = \langle A_1, A_2, \ldots, A_d \rangle$: Input feature vector
- Y, C: Class value to predict
- $P(C|A)$ vs $P(A|C)$
- Key terms
  - $P(A), P(C)$: Prior probability
  - $P(A|C)$: Class conditional probability or likelihood (from training data)
  - $P(C|A)$: Posterior probability

Likelihood vs Posterior

- $P(A|C)$: Likelihood, $P(C|A)$: Posterior
- Examples:
  - P(Viagra|Spam) and P(Spam|Viagra):

Likelihood vs Posterior

- $P(A|C)$: Likelihood, $P(C|A)$: Posterior
- Examples:
  - P(Viagra|Spam) and P(Spam|Viagra): Likelihood, Posterior
  - P(High temperature|Flu) and P(Flu|High Temperature):

Likelihood vs Posterior

- $P(A|C)$: Likelihood, $P(C|A)$: Posterior
- Examples:
  - P(Viagra|Spam) and P(Spam|Viagra): Likelihood, Posterior
  - P(High temperature|Flu) and P(Flu|High Temperature): Likelihood, Posterior
  - P(Fever, Headache, Cough|Flu) and P(Flu|Fever, Headache, Cough)
  - P(Viagra, Nigeria, Lottery|Spam) and P(Spam|Viagra, Nigeria, Lottery)

Bayes' Theorem

- Training data gives us likelihood and prior
- Prediction requires Posterior
- Bayes rule allows us to do statistical inference
- $P(C|A) = \frac{P(A|C)P(C)}{P(A)}$
- $P(C|A) \propto P(A|C)P(C)$
- Posterior $\propto$ Likelihood $\times$ Prior

Bayes Decision Rule

- "When you hear hoofbeats, think of horses not zebras"
- When predicting, assign the class with highest posterior probability
- To categorize email:
  - Compute P(spam|email) and P(not spam|email)
  - If P(spam|email) $>$ P(not spam|email), decide email as spam. Else as not-spam

Bayesian Classifiers

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables

- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes P(C| A_1, A_2,…,A_n)

- Can we estimate P(C| A_1, A_2,…,A_n) directly from data?

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

  - Choose value of C that maximizes
    $P(C \mid A_1, A_2, \ldots, A_n)$
  - Equivalent to choosing value of C that maximizes
    $P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$

- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naive Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n | C) = P(A_1 | C_j)\, P(A_2 | C_j) \ldots P(A_n | C_j)$
  - Can estimate $P(A_i | C_j)$ for all A_i and C_j.
  - New point is classified to C_j if $P(C_j) \prod P(A_i | C_j)$ is maximal.

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(C) = N_c/N$
  - e.g., P(No) = 7/10, P(Yes) = 3/10
- For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}| / N_{ck}$$

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C
  - Examples:

P(Status=Married|No) = 4/7
P(Refund=Yes|Yes)=0

How to Estimate Probabilities from Data?

- For continuous attributes:
  - Discretize the range into bins
    - one ordinal attribute per bin
    - violates independence assumption
  - Two-way split: $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (A_i,c_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Example of Naive Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No)
  × P(Married| Class=No)
  × P(Income=120K| Class=No)
  = 4/7 × 4/7 × 0.0072 = 0.0024

- P(X|Class=Yes) = P(Refund=No| Class=Yes)
  × P(Married| Class=Yes)
  × P(Income=120K| Class=Yes)
  = 1 × 0 × 1.2 × 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)

=> Class = No

Naive Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m - estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability

m: parameter

Example of Naive Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: attributes

M: mammals

N: non-mammals

$$P(A \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

Naive Bayes Classifier Summary

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)

Support Vector Machines

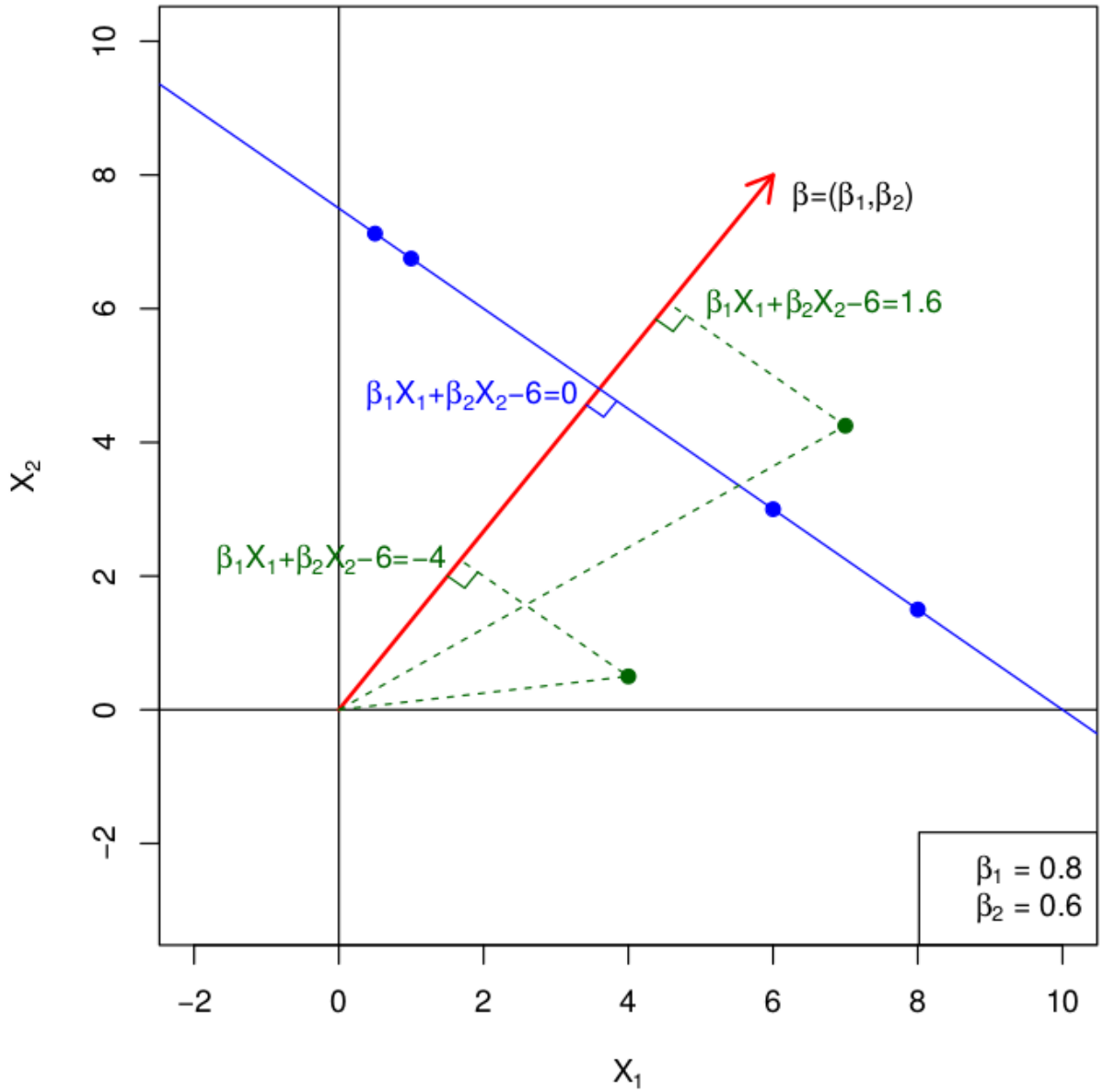
HyperPlane

- A hyperplane in p dimensions is a flat affine subspace of dimension $p - 1$.
- In general the equation for a hyperplane has the form

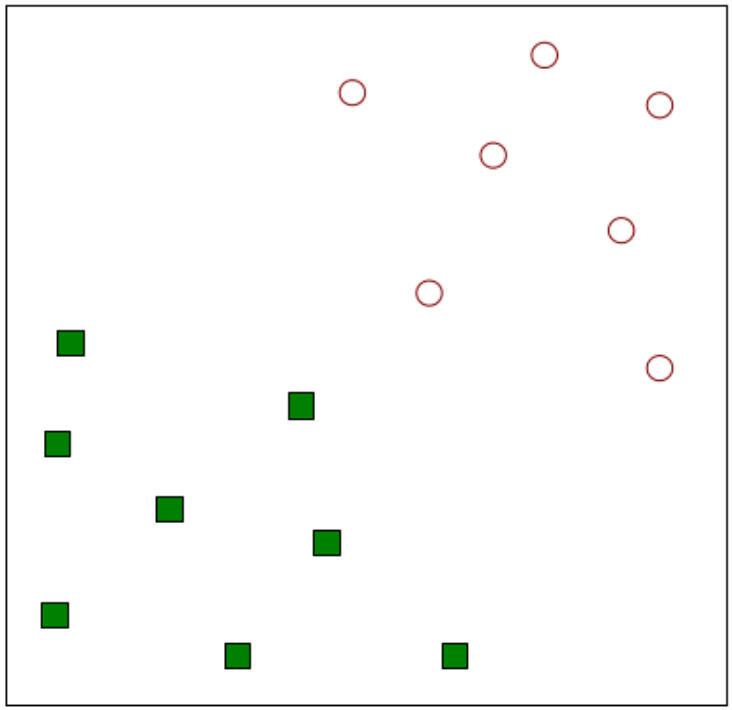
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

- In $p = 2$ dimensions a hyperplane is a line.
- If $\beta_0 = 0$, the hyperplane goes through the origin, otherwise not.
- The vector $\beta = (\beta_1, \beta_2, \cdots, \beta_p)$ is called the normal vector — it points in a direction orthogonal to the surface of a hyperplane.

HyperPlane in 2D

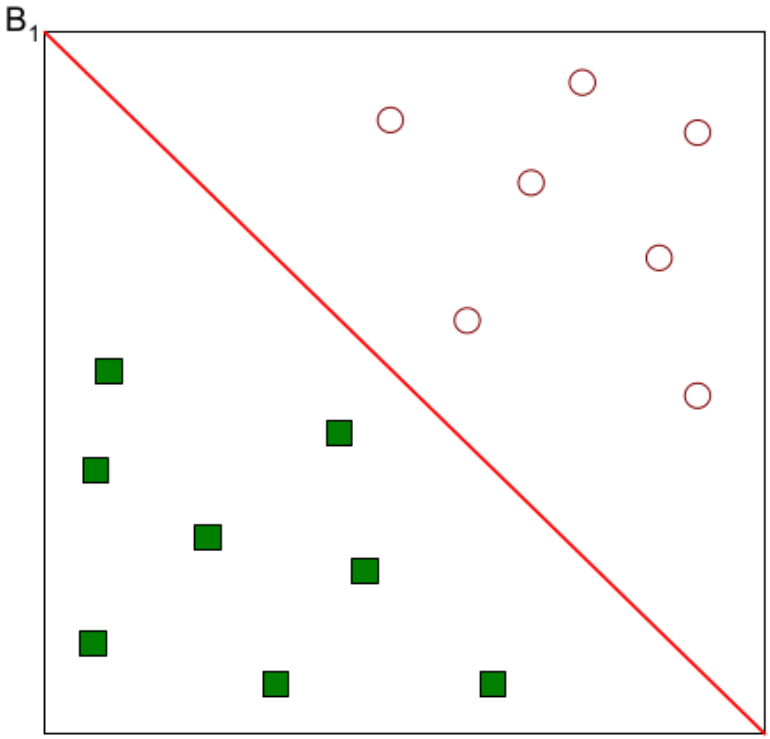


Support Vector Machines



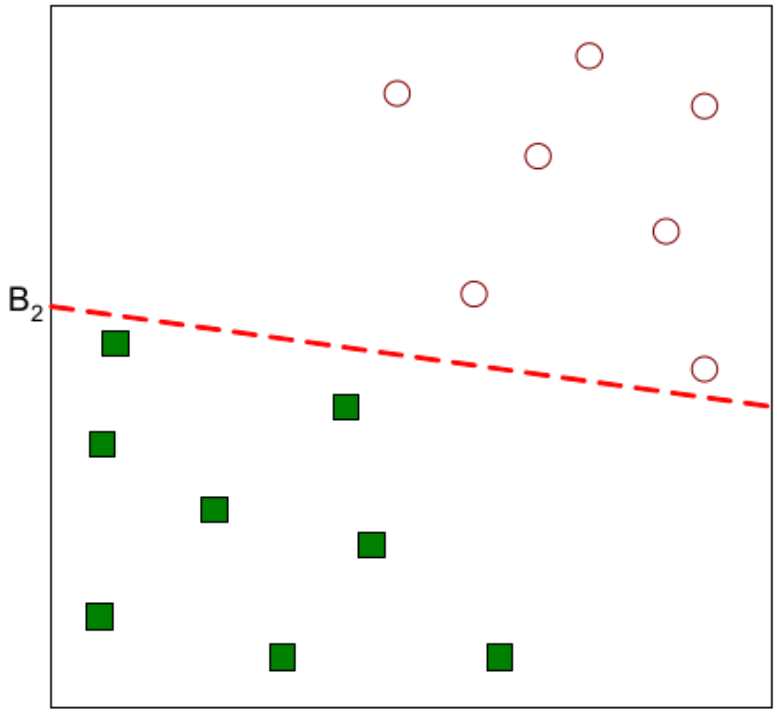
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



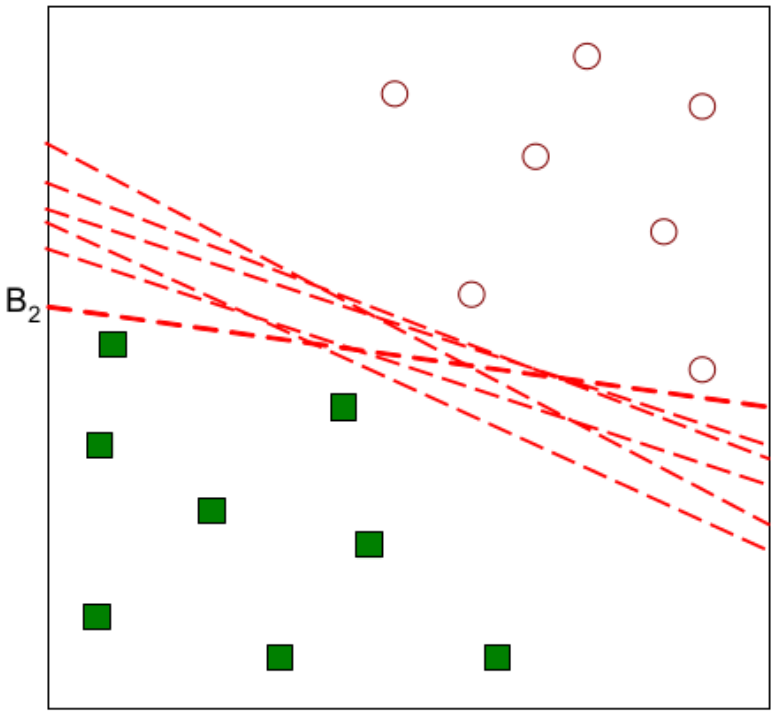
- One Possible Solution

Support Vector Machines

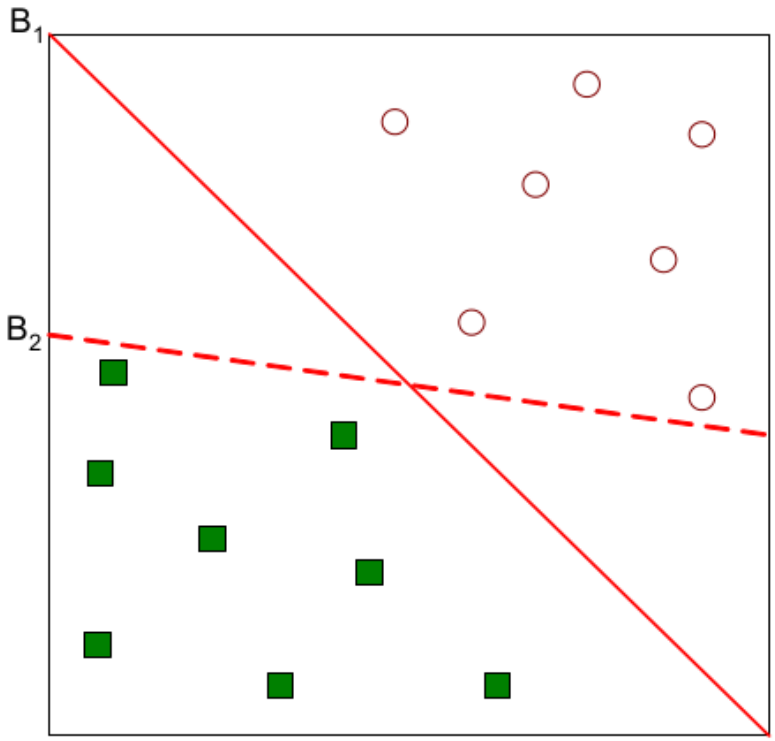


- Another possible solution

Support Vector Machines

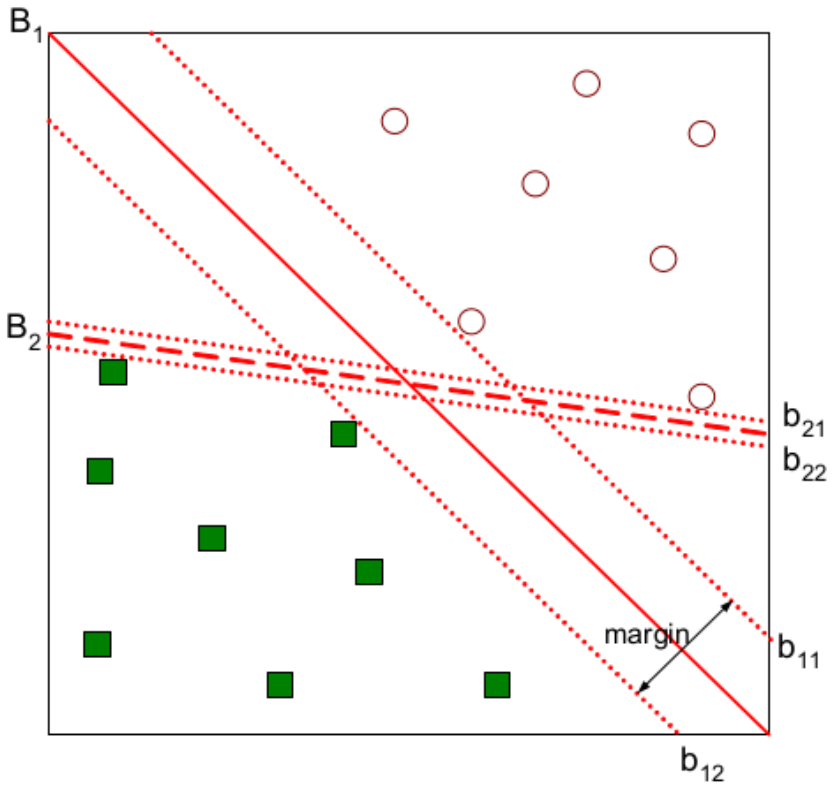


Support Vector Machines



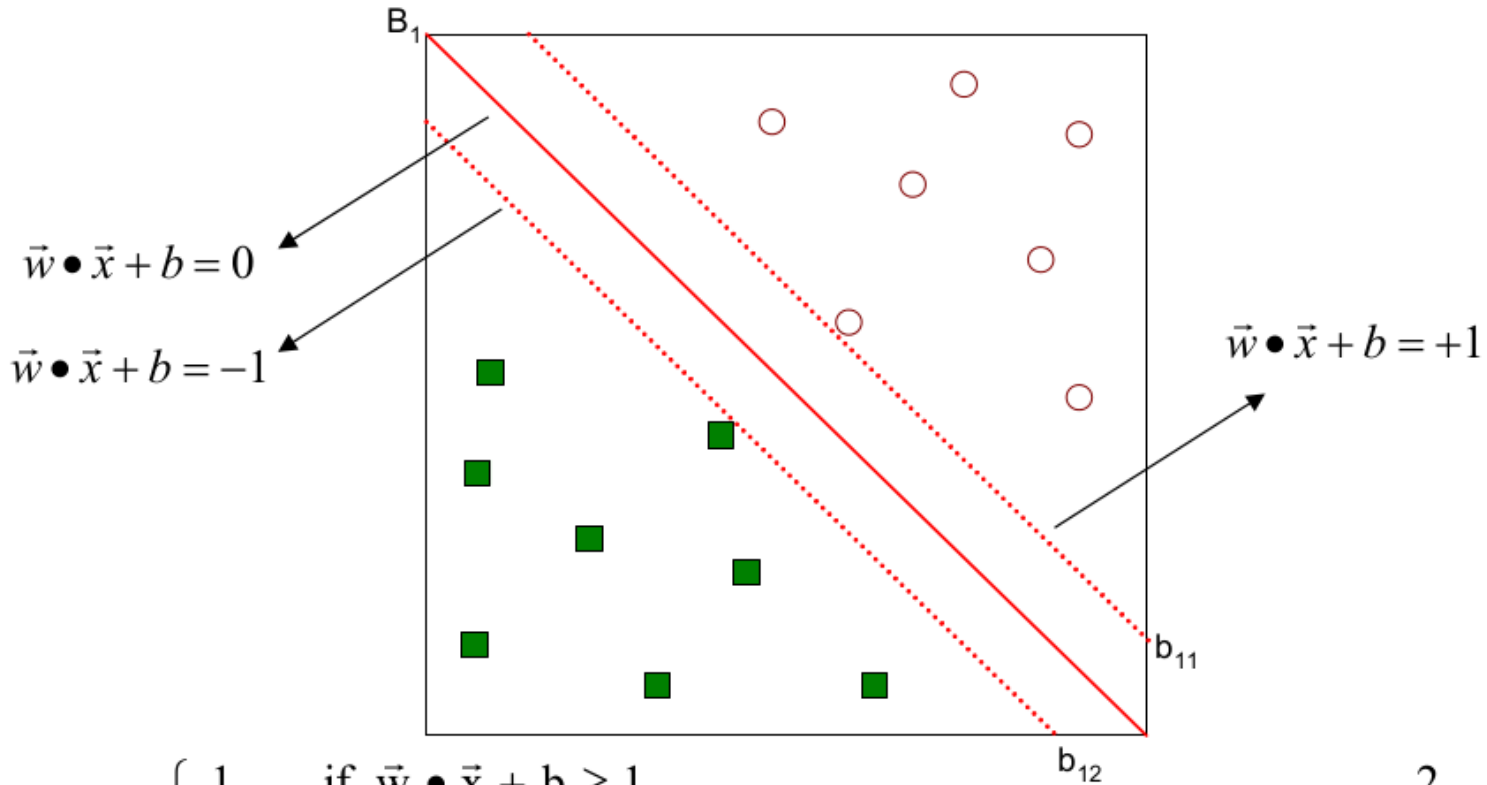
- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines



- Find hyperplane maximizes the margin => B1 is better than B2

Support Vector Machines



$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\| \vec{w} \|^2}$$

Support Vector Machines

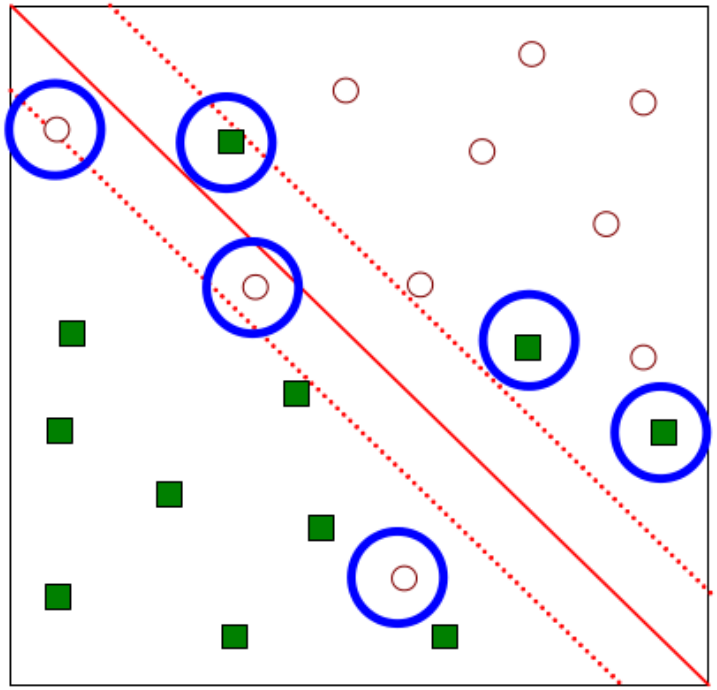
- We want to maximize: $\text{Margin} = \frac{2}{\| \vec{w} \|^2}$
  - Which is equivalent to minimizing: $L(w) = \frac{\| \vec{w} \|^2}{2}$
  - But subjected to the following constraints:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases}$$

    - This is a constrained optimization problem
      - Numerical approaches to solve it (e.g., quadratic programming)

Support Vector Machines

- What if the problem is not linearly separable?



Support Vector Machines

- What if the problem is not linearly separable?
  - Introduce slack variables
    - Need to minimize:

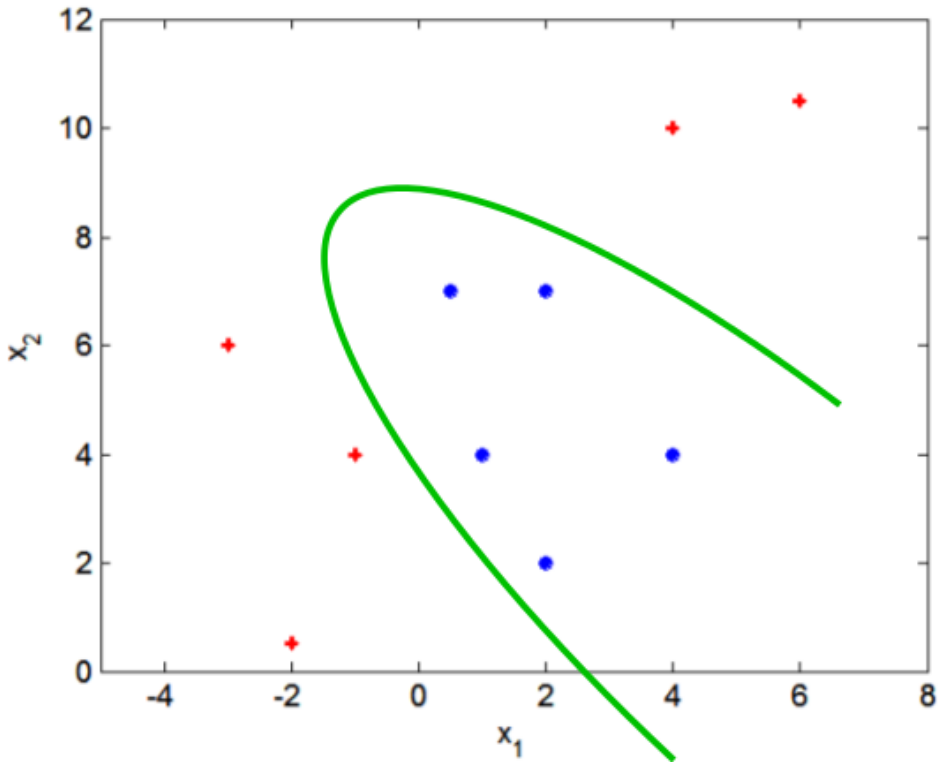
$$L(w) = \frac{\| \vec{w} \|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k \right)$$

    - Subject to:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

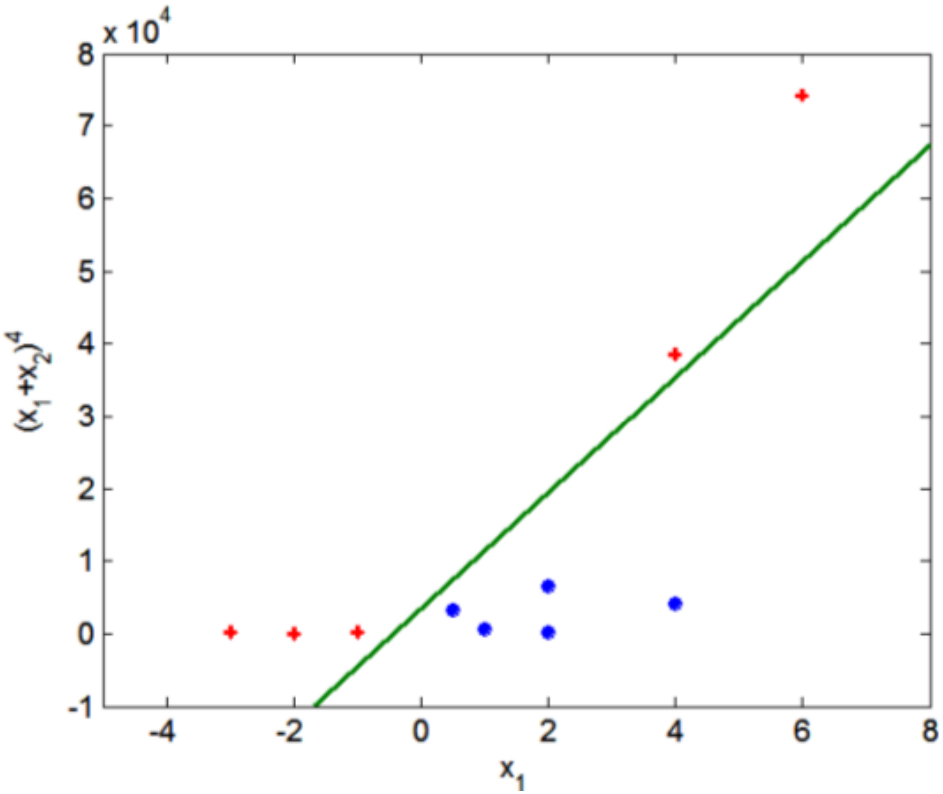
Nonlinear Support Vector Machines

- What if decision boundary is not linear?



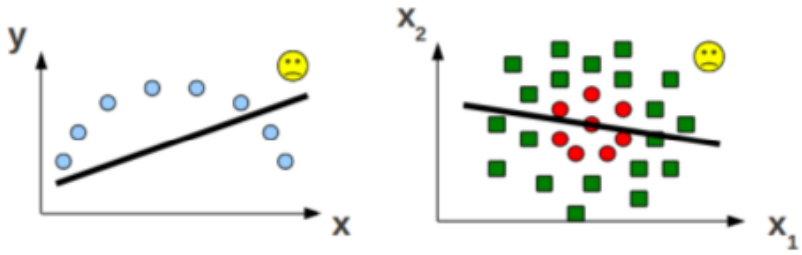
Nonlinear Support Vector Machines

- Transform data into higher dimensional space



Kernels

- Often we want to capture nonlinear patterns in the data
  - Nonlinear Regression: Input-output relationship may not be linear
  - Nonlinear Classification: Classes may not be separable by a linear boundary
- Linear models (e.g., linear regression, linear SVM) are not just rich enough

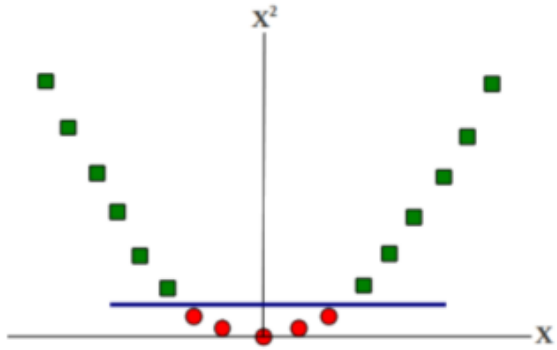


- Kernels: Make linear models work in nonlinear settings
  - By mapping data to higher dimensions where it exhibits linear patterns
  - Apply the linear model in the new input space
  - Mapping $\equiv$ changing the feature representation

Classifying non-linearly separable data

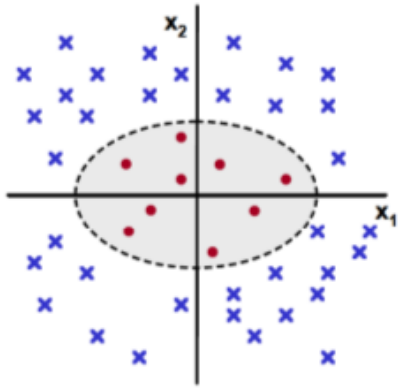
- Consider this binary classification problem

  - Each example represented by a single feature x
  - No linear separator exists for this data
- Now map each example as $x \rightarrow \{x, x^2\}$
  - Each example now has two features ("derived" from the old representation)
- Data now becomes linearly separable in the new representation

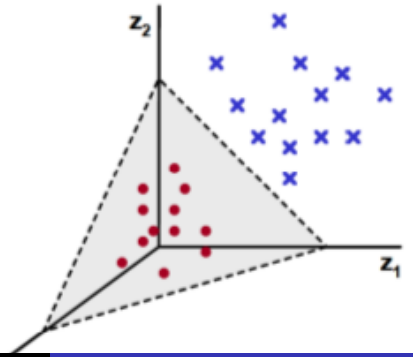


Classifying non-linearly separable data

- Let's look at another example:



  - Each example defined by a two features $\mathbf{x} = \{x_1, x_2\}$
  - No linear separator exists for this data
- Now map each example as $\mathbf{x} = \{x_1, x_2\} \rightarrow \mathbf{z} = \{x_1^2, \sqrt{2}x_1x_2, x_2^2\}$
  - Each example now has three features ("derived" from the old representation)
- Data now becomes linearly separable in the new representation



Feature Mapping

- Consider the following mapping ϕ for an example $\mathbf{x} = \{x_1, \ldots, x_D\}$

$$\phi : \mathbf{x} \rightarrow \{x_1^2, x_2^2, \ldots, x_D^2, , x_1x_2, x_1x_2, \ldots, x_1x_D, \ldots\ldots, x_{D-1}x_D\}$$

- It's an example of a quadratic mapping
  - Each new feature uses a pair of the original features
- Problem: Mapping usually leads to the number of features blow up!
  - Computing the mapping itself can be inefficient in such cases
  - Moreover, *using* the mapped representation could be inefficient too
    - e.g., imagine computing the similarity between two examples: $\phi(\mathbf{x})^\top \phi(\mathbf{z})$
- Thankfully, Kernels help us avoid both these issues!
  - The mapping doesn't have to be explicitly computed
  - Computations with the mapped features remain efficient

Kernels as High Dimensional Feature Mapping

- Consider two examples $\mathbf{x} = \{x_1, x_2\}$ and $\mathbf{z} = \{z_1, z_2\}$
- Let's assume we are given a function k (kernel) that takes as inputs $\mathbf{x}$ and $\mathbf{z}$

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^\top \mathbf{z})^2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2 \\ &= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^\top (z_1^2, \sqrt{2} z_1 z_2, z_2^2) \\ &= \phi(\mathbf{x})^\top \phi(\mathbf{z}) \end{aligned}$$

- The above k implicitly defines a mapping ϕ to a higher dimensional space

$$\phi(\mathbf{x}) = \{x_1^2, \sqrt{2} x_1 x_2, x_2^2\}$$

- Note that we didn't have to define/compute this mapping
- Simply defining the kernel a certain way gives a higher dim. mapping ϕ
- Moreover the kernel $k(\mathbf{x}, \mathbf{z})$ also computes the dot product $\phi(\mathbf{x})^\top \phi(\mathbf{z})$
  - $\phi(\mathbf{x})^\top \phi(\mathbf{z})$ would otherwise be much more expensive to compute explicitly
- All kernel functions have these properties

Kernels: Formally Defined

- Recall: Each kernel k has an associated feature mapping ϕ
- ϕ takes input $\mathbf{x} \in \mathcal{X}$ (input space) and maps it to $\mathcal{F}$ ("feature space")
- Kernel $k(\mathbf{x}, \mathbf{z})$ takes two inputs and gives their similarity in $\mathcal{F}$ space

$$\begin{aligned} \phi &: \mathcal{X} \to \mathcal{F} \\ k &: \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \quad k(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^\top \phi(\mathbf{z}) \end{aligned}$$

- $\mathcal{F}$ needs to be a *vector space* with a *dot product* defined on it
  - Also called a *Hilbert Space*
- Can just *any* function be used as a kernel function?
  - No. It must satisfy **Mercer's Condition**

Popular Kernels

The following are the most popular kernels for real-valued vector inputs

- Linear (trivial) Kernel:

$$k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z} \text{ (mapping function } \phi \text{ is identity - no mapping)}$$

- Quadratic Kernel:

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^2 \quad \text{or} \quad (1 + \mathbf{x}^\top \mathbf{z})^2$$

- Polynomial Kernel (of degree d):

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^d \quad \text{or} \quad (1 + \mathbf{x}^\top \mathbf{z})^d$$

- Radial Basis Function (RBF) Kernel:

$$k(\mathbf{x}, \mathbf{z}) = \exp[-\gamma ||\mathbf{x} - \mathbf{z}||^2]$$

  - γ is a hyperparameter (also called the kernel bandwidth)
  - The RBF kernel corresponds to an infinite dimensional feature space $\mathcal{F}$ (i.e., you can't actually write down the vector $\phi(\mathbf{x})$)

Using Kernels

- Kernels can turn a linear model into a nonlinear one
- Recall: Kernel $k(\mathbf{x}, \mathbf{z})$ represents a dot product in some high dimensional feature space $\mathcal{F}$
- Any learning algorithm in which examples only appear as dot products $(\mathbf{x}_i^\top \mathbf{x}_j)$ can be kernelized (i.e., non-linearlized)
  - .. by replacing the $\mathbf{x}_i^\top \mathbf{x}_j$ terms by $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$
- Most learning algorithms are like that
  - Perceptron, SVM, linear regression, etc.
  - Many of the unsupervised learning algorithms too can be kernelized (e.g., K-means clustering, Principal Component Analysis, etc.)

Summary

Major Concepts:

- Probabilistic interpretation of Classification
- Bayesian Classifiers
- Naive Bayes Classifier
- Support Vector Machines (SVM)
- Kernels

Slide Material References

- Slides from TSK Book, Chapter 5
- Slides from Piyush Rai
- See also the footnotes