Machine Learning

AIMLCZG565

Refresher – Question Paper Discussion

Raja vadhana P

Assistant Professor,

BITS - CSIS

BITS Pilani

Pilani Campus

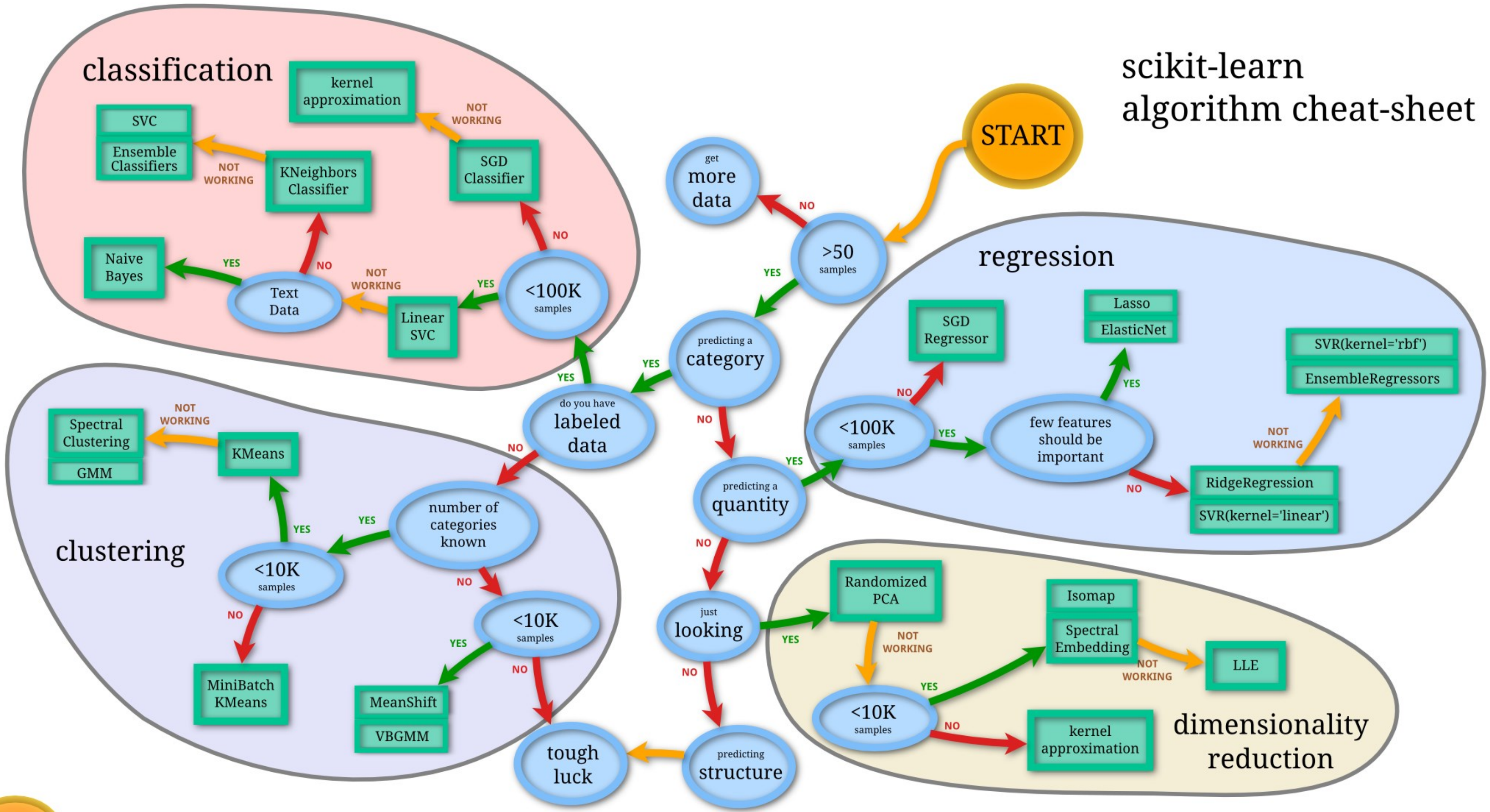
Disclaimer and Acknowledgement

- The content for these slides has been obtained from books and various other source on the Internet
- I here by acknowledge all the contributors for their material and inputs.
- I have provided source information wherever necessary
- I have added and modified the content to suit the requirements of the course

Source: Slides of Prof. Chetana, Prof.Vimal, Prof.Seetha, Prof.Sugata, Prof.Monali, Prof. Raja vadhana , Prof.Anita from BITS Pilani , CS109 and CS229 stanford lecture notes, Tom Mitchell, Andrew Ng and many others who made their course materials freely available online.

innovate achieve lead

Guide to Choose Estimator



Source Credit : https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Previous Semester Exam Answer Discussion

In addition to the previous QP examples discussed in class, check the live classes for below :

- Additional Problems discussed
- Practice Problems shared in the respective module's uploads
- Refresh different type of distributions from your ISM course and related them for MLE, MAP parameter estimation

Accommodate changes in the below, to study the effect of it on the hypothesis

- Know the design of Cost functions per unique ML techniques
- Refresh the notion of Bayes Theorem from ISM course and relate derivation of Naïve Bayes & Logistic Regression.
- ISM Course : Hypothesis Testing & Confidence Interval. Relate to the ML models

K-Means Algorithm

1. Proximity Matrix Calculation

2. Expectation Step

3. Maximization Step

4. Convergence

5. Interpretation

K-Means – Example 1

Consider the following dataset.

x1	-1	-1	-1	-1	0	4	4
x2	2	1	-1	-2	0	2	-2
Class label	1	1	1	1	1	2	2

The given class label exhibits two natural clusters formed in the given dataset and acts as a ground truth. Now remove class labels and use the K-means clustering algorithm to find the 2 clusters by initializing two cluster center's as follows:

A.C1(-1,2) and C2(0,0)

B.C1(-0.5,0) and C2(0,0)

- For both the above cases run the algorithm till centers do not change (convergence criteria) and give the final cluster assignment
- In each case, comment on the correctness of cluster assignment.
- Also, comment in no more than 20 words on the drawback of k-means which is depicted in above two cases

K-Means – Example 1

	c1	c2
x1	-1	2
x2	0	0

Case 1

x1	-1	-1	-1	-1	0	4	4
x2	2	1	-1	-2	0	2	-2
Class label	1	1	1	1	1	2	2
Dist(Xi, C1)	0	1	3	4	2.236068	5	6.403124
Dist(Xi, C2)	2.236068	1.414214	1.414214	2.236068	0	4.472136	4.472136
Cluster Assignment	1	1	2	2	2	2	2

	new c1	new c2
x1	-1	1.2
x2	1.5	-0.6

Dist(Xi, C1)	0.5	0.5	2.5	3.5	1.802776	5.024938	6.103278
Dist(Xi, C2)	3.405877	2.720294	2.236068	2.607681	1.341641	3.820995	3.130495
Cluster Assignment	1	1	2	2	2	2	2

Class label	1	1	1	1	1	2	2

	new c1	new c2
x1	-1	1.2
x2	1.5	-0.6

Algorithm has converged after 2 iterations but the cluster assignment does not depict the natural clusters in the datasets as given by the ground truth.

K-Means – Example 1

Correctness of the K- means the algorithm is sensitive to the initialization of cluster centres.

Case 2

	c1	c2
x1	-0.5	0
x2	0	0

x1	-1	-1	-1	-1	0	4	4
x2	2	1	-1	-2	0	2	-2
Class label	1	1	1	1	1	2	2
Dist(Xi, C1)	2.061553	1.118034	1.118034	2.061553	0.5	4.924429	4.924429
Dist(Xi, C2)	2.236068	1.414214	1.414214	2.236068	0	4.472136	4.472136
Cluster Assignment	1	1	1	1	2	2	2
Class label	1	1	1	1	1	2	2
Dist(Xi, C1)	2	1	1	2	1	5.385165	5.385165
Dist(Xi, C2)	4.176655	3.800585	3.800585	4.176655	2.666667	2.403701	2.403701
Cluster Assignment	1	1	1	1	1	2	2
Dist(Xi, C1)	2.009975	1.019804	1.019804	2.009975	0.8	5.2	5.2
Dist(Xi, C2)	5.385165	5.09902	5.09902	5.385165	4	2	2
Cluster Assignment	1	1	1	1	1	2	2

	new c1	new c2
x1	-1	2.6667
x2	0	0

	new c1	new c2
x1	-0.8	4
x2	0	0

	new c1	new c2
x1	-0.8	4
x2	0	0

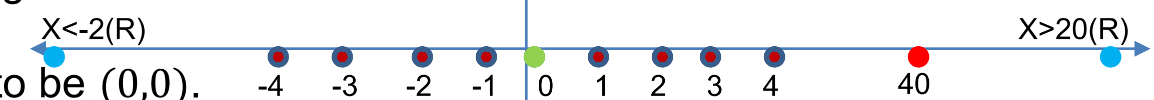
Algorithm has converged after 3 iterations and the cluster assignment shows the natural clusters in the datasets as given by the ground truth.

K-Means – Example 2

Assume that a number of points are distributed along the x-axis:

$(-R,0), (-R+1,0), \ldots (-1,0), (0,0), (1,0), \ldots$
$\ldots (R-1,0), (R,0)$ and an outlier point at $(10R,0)$.

We would like to use the K-Means algorithm to find two clusters for these points.
Initially, one cluster center is chosen to be $(0,0)$.
Where should the other cluster center be placed on the x-axis initially so that one of the clusters formed has all the given data points and the other cluster has none?
What will be the final locations of the cluster centers?
‘



In the second iteration of the algorithm the first cluster center will be updated from $(0,0)$ to $\left(\frac{\mathbf{10R}}{\mathbf{2R+2}}, \mathbf{0}\right)$ and the second cluster center will be unchanged

KNN Algorithm – Instance Based Classification/Regression

1. Proximity Matrix Calculation
2. Distance between Query and Training
3. Sorting based on KNN (Distance or Locally weighted using Kernels)
4. Classification : Majority Voting (or using weighted voting)
5. Regression : Weighted gradient descent to learn the weights

Instance Based Learning

K-Nearest Neighbor : Algorithm

Approximating a discrete-valued function $f : \Re^n \rightarrow V$.

Training algorithm:

- For each training example $\langle x, f(x) \rangle$, add the example to the list *training_examples*

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1 \ldots x_k$ denote the k instances from *training_examples* that are nearest to x_q
  - Return

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(v, f(x_i))$$

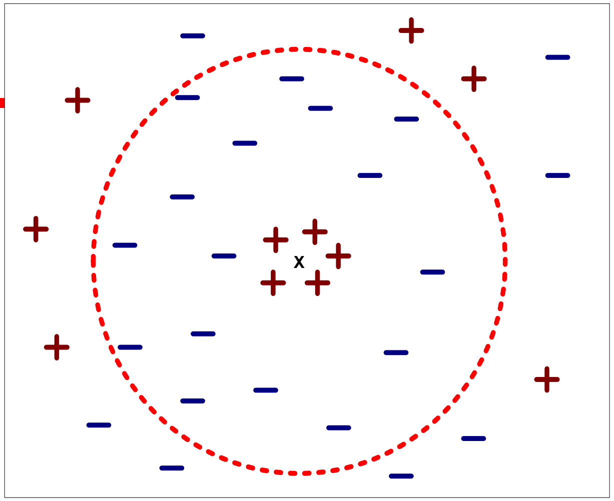
where $\delta(a, b) = 1$ if $a = b$ and where $\delta(a, b) = 0$ otherwise.

Approximate a real-valued target function $f : \Re^n \rightarrow \Re$

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

Instance Based Learning

K-Nearest Neighbor : Variation



- Locally Weighted K-NN algorithm or Distance Weighted K-NN algorithm

contribution of each of the k nearest neighbors is weighted accorded to their distance to x_q

- **discrete-valued target functions**

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{argmax} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$$

where $w_i \equiv \frac{1}{d(x_q, x_i)^2}$ and $\hat{f}(x_q) = f(x_i)$ if $x_q = x_i$

- **continuous-valued target function:**

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

<u>Kernel Functions</u>

$w_i = K(d(x_q, x_i))$

$K(d(x_q, x_i)) = 1/\ d(x_q, x_i)^2$

$K(d(x_q, x_i)) = 1/(d_0 + d(x_q, x_i))^2$

$K(d(x_q, x_i)) = \exp(-(d(x_q, x_i)/\sigma_0)^2)$

$d(x_q, x_i)$ is the distance between x_q and x_i

Locally weighted linear regression

- For a given query point $\mathbf{x}_q$ we solve the following weighted regression problem using gradient descent training rule:

$$\Delta w_j = \eta \sum_{x \in \, k \text{ nearest nbrs of } x_q} K(d(x_q, x)) \, (f(x) - \hat{f}(x)) \, a_j(x)$$

- Note that we need to solve a new regression problem for *every* query point—that's what it means to be *local*.
- In ordinary linear regression, we solved the regression problem once, globally, and then used the same $h_{\mathbf{w}}$ for any query point.

KNN – Example

Consider the following training set in 2-dimensional Euclidean space.

Point	Coordinate	Class
X1	(-1, 1)	Negative
X2	(0, 1)	Positive
X3	(0, 2)	Negative
X4	(1, -1)	Negative
X5	(1, 0)	Positive
X6	(1, 2)	Positive
X7	(2, 2)	Negative
X8	(2, 3)	Positive

- What is the class of the point (1, 1) if 7NN classifier is considered?
- If the value of K is reduced whether the class will change? (Consider K=3 and K=5).
- What should be the final class if the above 3 values of K are considered?

KNN – Example

Point	Coordinate	Class	Distance from 1,1
X1	(-1, 1)	Negative	2
X2	(0, 1)	Positive	1
X3	(0, 2)	Negative	1.414
X4	(1, -1)	Negative	2
X5	(1, 0)	Positive	1
X6	(1, 2)	Positive	1
X7	(2, 2)	Negative	1.41
X8	(2, 3)	Positive	2.236

i. class of the point (1, 1) if 3NN classifier is considered x2, x5, x6 - Positive
ii. class of the point (1, 1) if 5NN classifier is considered? x2, x5, x6, x3, x7 - Positive
iii. class of the point (1, 1) if 7NN classifier is considered? x2, x5, x6, x3, x7, x1, x5- Negative

Final class value to be considered as Positive

GMM Algorithm

1. Expectation Step

2. Maximization Step

3. Convergence using Log Likelihood

4. Interpretation

GMM – Example

I Standardize the data if required:

II Fix the no.of.cluster expected

III Initialize the prototypes:
Mean, Covariance, Weights

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

V. Calculate the log likelihood of the points

VI Maximization Step: Fix the membership(responsibility matrix) and re-estimate the prototypes

VII Calculate the new log likelihood of the points. Repeat E & M Step till convergence is achieved:

Suppose we have the following one-dimensional data at -4.0, -.3.0, -2.0, -1.0, 0.0, 1.0, 2.0, 3.0, 4.0. Use the EM algorithm to find a Gaussian mixture model consisting of exactly one Gaussian that fits the data. Assume that the initial mean of the Gaussian is 10.0 and the initial variance is 1.0

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

$$P(z_{n1} = 1/x_n) = \gamma(z_{n1})$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \;\; = N$$

GMM – Example

I Standardize the data if required:

II Fix the no.of.cluster expected

III Initialize the prototypes:
Mean, Covariance, Weights

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

V. Calculate the log likelihood of the points

VI Maximization Step: Fix the membership(responsibility matrix) and re-estimate the prototypes

VII Calculate the new log likelihood of the points. Repeat E & M Step till convergence is achieved:

Suppose we have the following one-dimensional data at -4.0, -.3.0, -2.0, -1.0, 0.0, 1.0, 2.0, 3.0, 4.0. Use the EM algorithm to find a Gaussian mixture model consisting of exactly one Gaussian that fits the data. Assume that the initial mean of the Gaussian is 10.0 and the initial variance is 1.0

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

$$\mu_1^{new} = \frac{1}{N_1}\sum_{n=1}^{N}\gamma(z_{n1})x_n$$

$$= \frac{\sum_{n=1}^{n=N} x_n}{N}$$

$$= \frac{-4.0 \ + -3.0 \ + \ -2.0 + \ -1.0 \ + \ 0.0 + 1.0 + 2.0 + 3.0 + 4.0}{9}$$

$$= 0.0$$

GMM – Example

I Standardize the data if required:

II Fix the no.of.cluster expected

III Initialize the prototypes:
Mean, Covariance, Weights

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

V. Calculate the log likelihood of the points

VI Maximization Step: Fix the membership(responsibility matrix) and re-estimate the prototypes

VII Calculate the new log likelihood of the points. Repeat E & M Step till convergence is achieved:

Suppose we have the following one-dimensional data at -4.0, -.3.0, -2.0, -1.0, 0.0, 1.0, 2.0, 3.0, 4.0. Use the EM algorithm to find a Gaussian mixture model consisting of exactly one Gaussian that fits the data. Assume that the initial mean of the Gaussian is 10.0 and the initial variance is 1.0

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

$$\Sigma_k^{new} = \frac{1}{N_1}\sum_{n=1}^{n=N}(x_n - \mu_1^{new})(x_n - \mu_1^{new})^T$$

Here the x_n and μ_1^{new} are 1×1 matrices and the expression for Σ_k^{new} simplifies to $\frac{\sum_{n=1}^{N} x_n^2}{N}$ which is $\frac{2*(4.0^2+3.0^2+2.0^2+1.0^2)}{9}$=6.66.

GMM – Example

I Standardize the data if required:

II Fix the no.of.cluster expected

III Initialize the prototypes:
Mean, Covariance, Weights

IV Expectation-Step: Fix prototype & find the membership of each point weighted by the probability value

V. Calculate the log likelihood of the points

VI Maximization Step: Fix the membership(responsibility matrix) and re-estimate the prototypes

VII Calculate the new log likelihood of the points. Repeat E & M Step till convergence is achieved:

Suppose we have the following one-dimensional data at -4.0, -.3.0, -2.0, -1.0, 0.0, 1.0, 2.0, 3.0, 4.0. Use the EM algorithm to find a Gaussian mixture model consisting of exactly one Gaussian that fits the data. Assume that the initial mean of the Gaussian is 10.0 and the initial variance is 1.0

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

In the next iteration the E-step computes the posterior probabilities to be 1 and the M-step computes the same mean and covariance matrix as above, so the algorithm converges

Support Vector Machine Algorithm (SVC)

1. Select the support vectors

$$f(x) = \sum_i \alpha_i y_i (\mathbf{x}_i{}^\top \mathbf{x}) + b$$

2. Substitute in Lagrangian function & Find the Unconstrained Optimization Function:

$$L(w, b, \alpha_i) = \Sigma\, \alpha_i - \frac{1}{2}\left(\sum_i \sum_j \alpha_i\, \alpha_j\, y_i\, y_j\, xi \cdot xj \right)$$

3. Gradient of the Lagrangian
4. Solve the simultaneous linear equation and find the lagrange multiplier:

$$\alpha_i\,[-1\,(\mathbf{w} \cdot \mathbf{x}_i + b)] = -1$$
$$\alpha_i\,[+1\,(\mathbf{w} \cdot \mathbf{x}_i + b)] = 1$$

5. Substitute the Lagrange multiplier and obtain the weights

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$
$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i$$

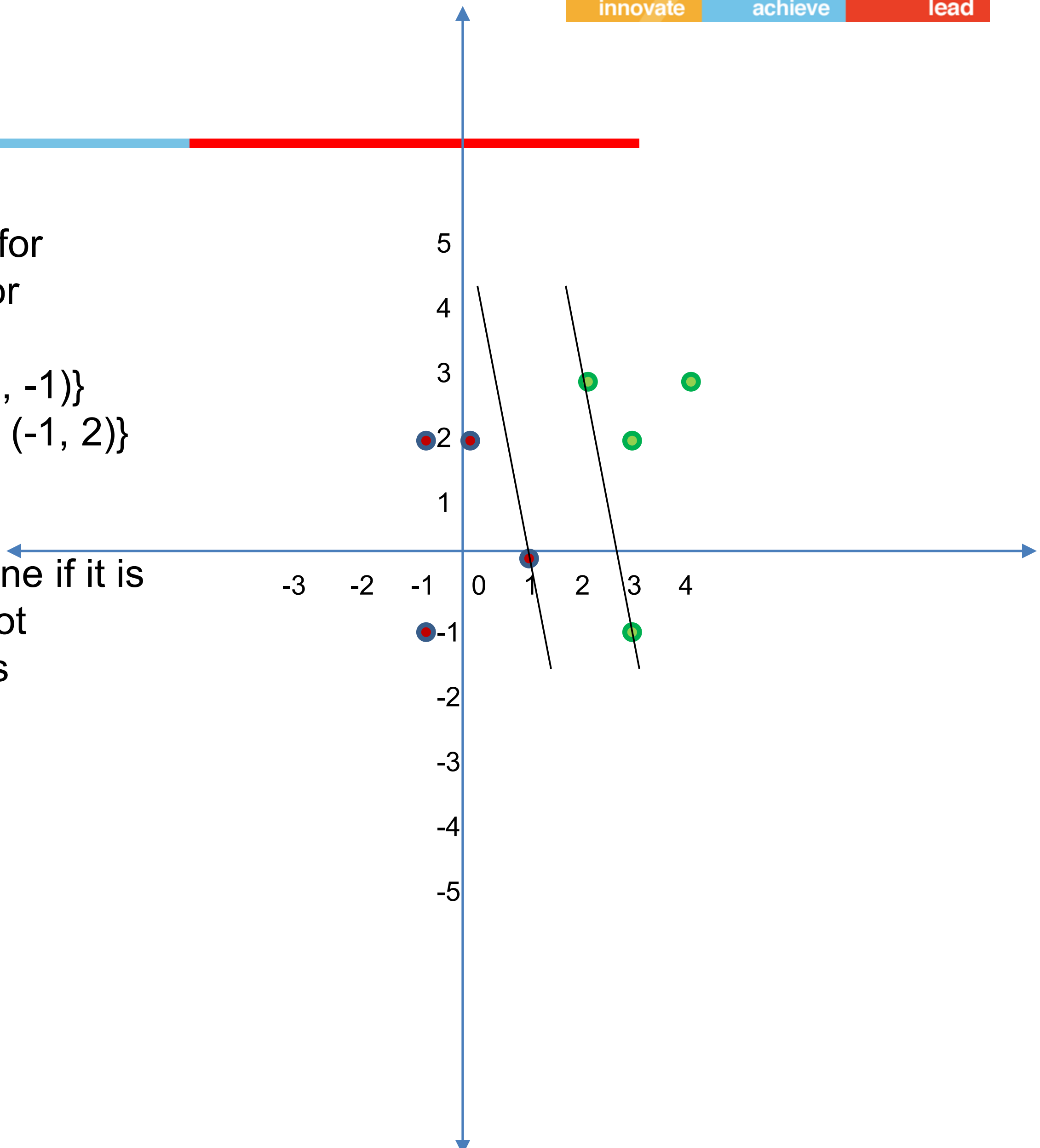
6. Construct the equation of the LSVM hyperplane
7. Estimate the width of the margin

$$\frac{2}{||W||}$$

SVC – Example 1

Solve the below and find the equation for hyper plane using linear Support Vector Machine method.
Positive Points: {(3, 2), (4, 3), (2, 3), (3, -1)}
Negative Points: {(1, 0), (-1, -1), (0, 2), (-1, 2)}

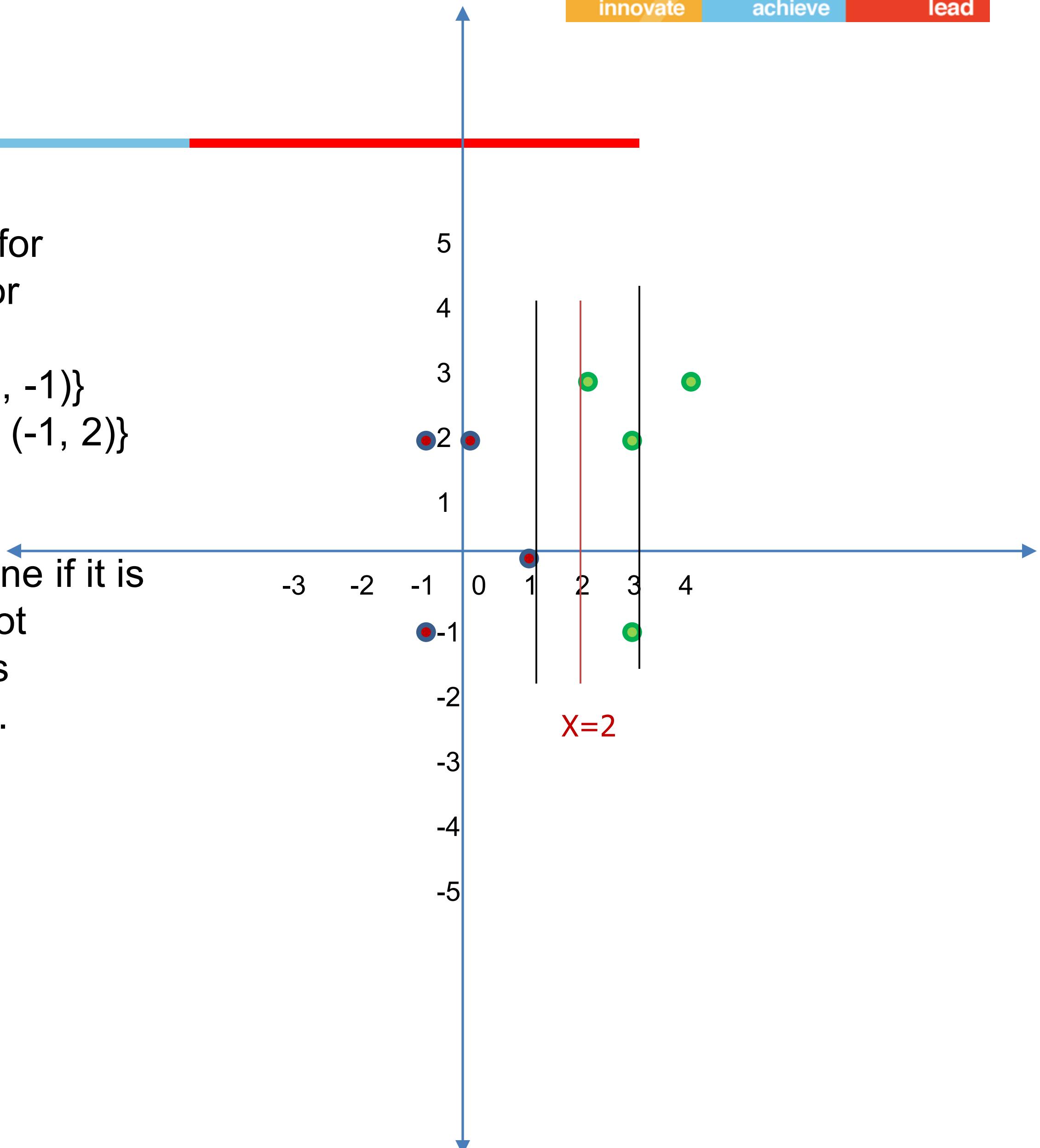
- Find the support vectors
- Determine the equation of hyperplane if it is changed and give a reason if it is not changed for the following two cases
  - ✓ If the point (2, 3) is removed.
  - ✓ If the point (5,4) is added
  - ✓ If the point (-2,-3) is added



SVC – Example 1

Solve the below and find the equation for hyper plane using linear Support Vector Machine method.
Positive Points: {(3, 2), (4, 3), (2, 3), (3, -1)}
Negative Points: {(1, 0), (-1, -1), (0, 2), (-1, 2)}

- Find the support vectors
- Determine the equation of hyperplane if it is changed and give a reason if it is not changed for the following two cases
    - ✓ **If the point (2, 3) is removed.**
    - ✓ If the point (5,4) is added
    - ✓ If the point (-2,-3) is added

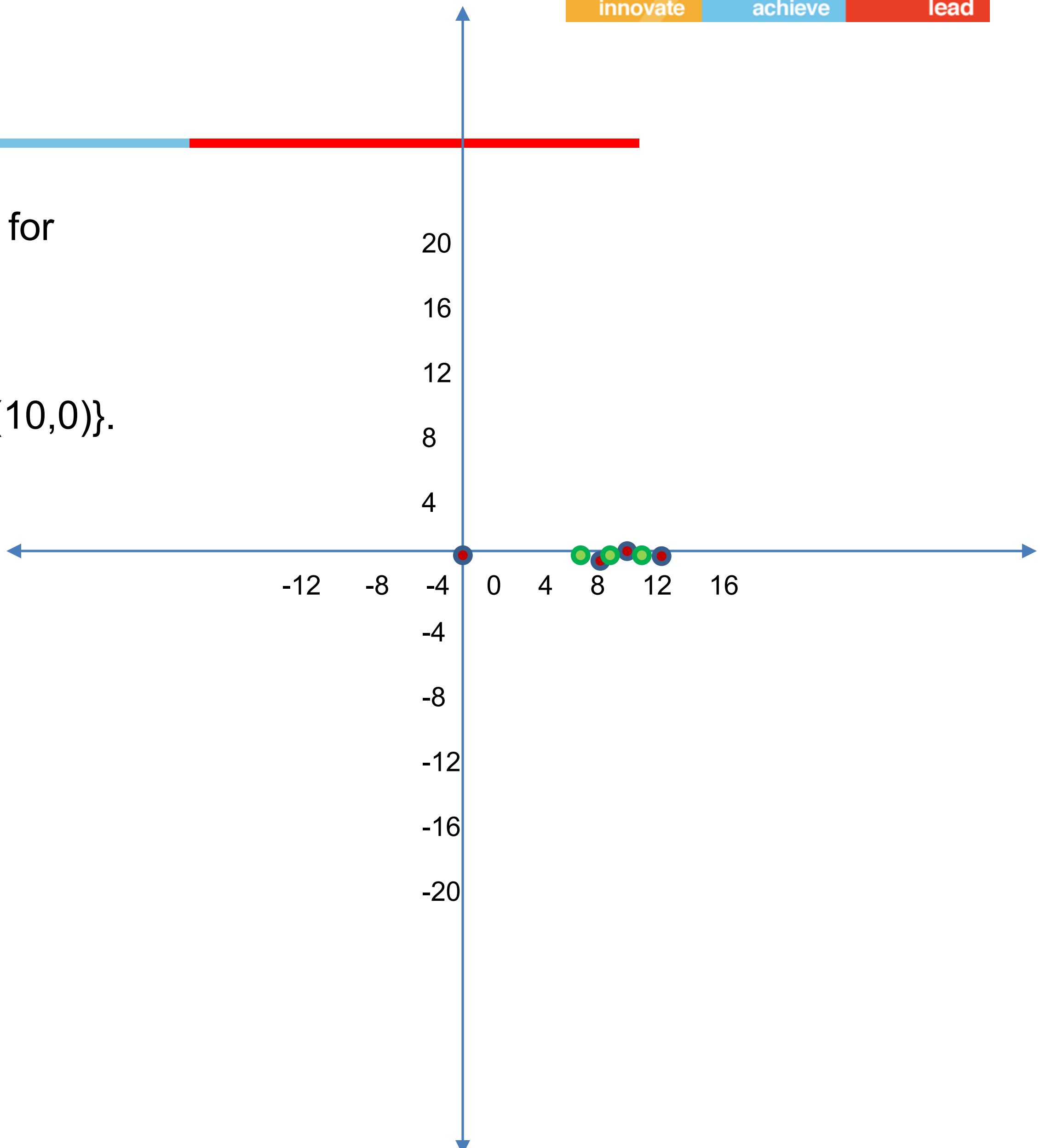


SVC – Example 2

Use kernel trick and find the equation for hyperplane using nonlinear SVM.

Positive Points: {(7,0), (9,0), (11,0)}
Negative Points: {(0,0), (8,0), (12,0), (10,0)}.

Plot the point before and after the transformation.

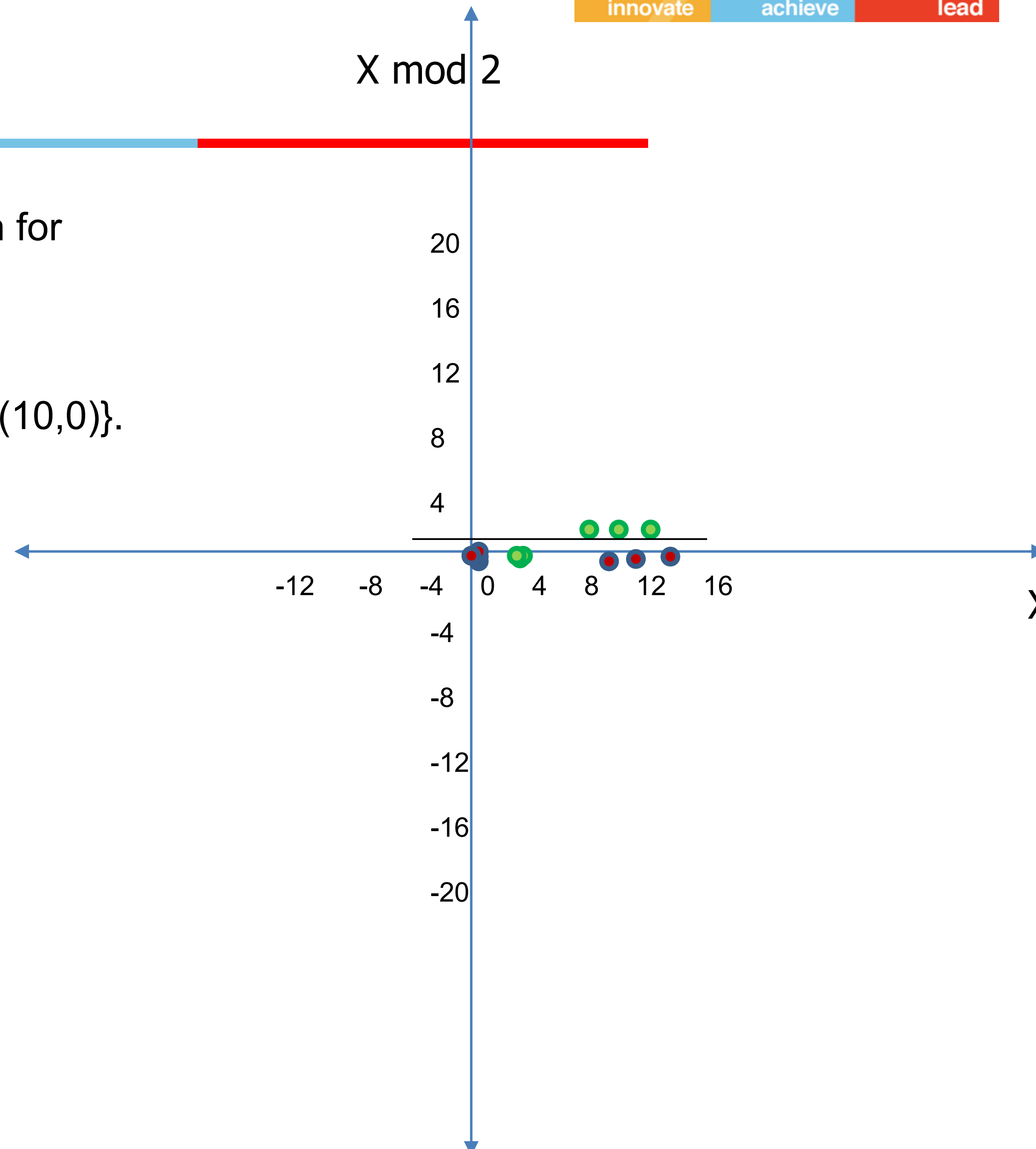


SVC – Example 2

Use kernel trick and find the equation for hyperplane using nonlinear SVM.

Positive Points: {(7,0), (9,0), (11,0)}
Negative Points: {(0,0), (8,0), (12,0), (10,0)}.

Plot the point before and after the transformation.



$\Phi(x)$ = x mod 2

Equation of hyperplane : y=0.5

Ensemble Learners

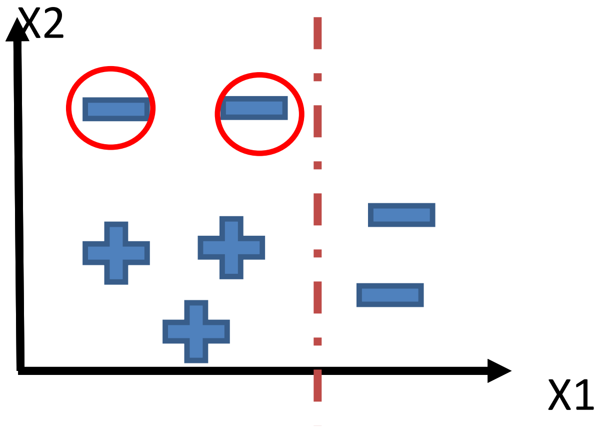
1. Sampling (with/without Boosting)
2. Decision Stumps/ Regressors /Other Classifiers
3. Aggregation of Predictor’s output
4. Voting / Gradient Boosted result

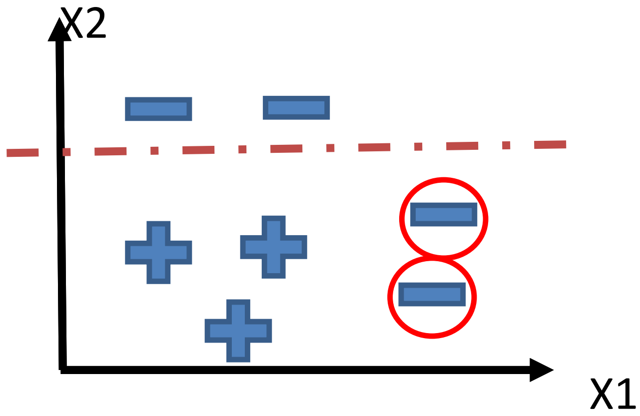
Ensemble – Example 1

Consider training a boosting classifier using decision stumps on the following data set.

Circle the examples which will have their weights increased at the end of each iteration.

Run the iteration till zero training error is achieved.



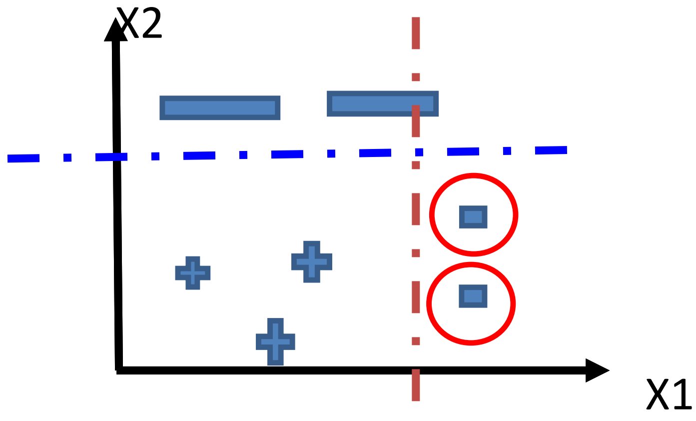


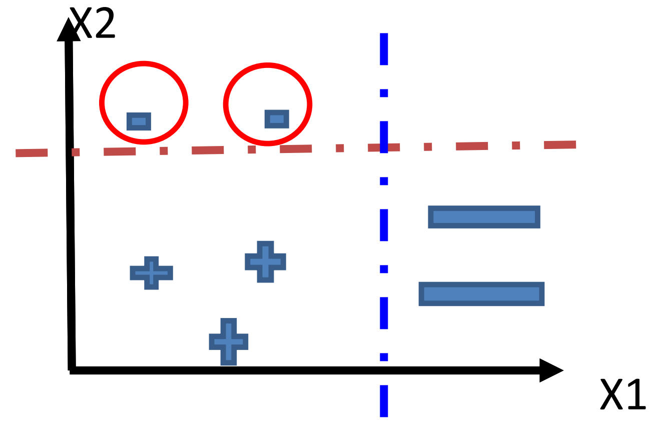
Ensemble – Example 1

Consider training a boosting classifier using decision stumps on the following data set.

Circle the examples which will have their weights increased at the end of each iteration.

Run the iteration till zero training error is achieved.



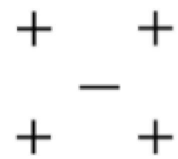


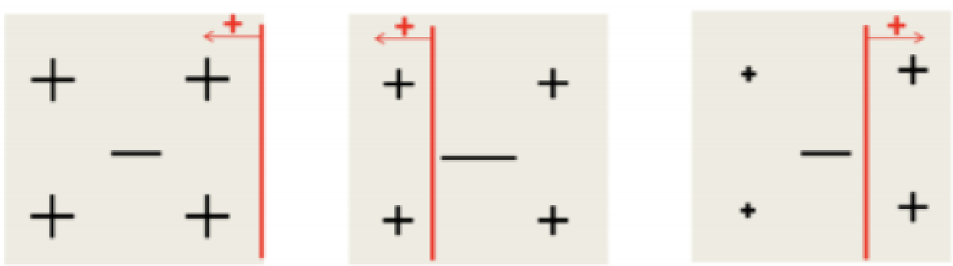
Ensemble – Example 2

Consider training a boosting classifier using decision stumps on the following data set.

Circle the examples which will have their weights increased at the end of each iteration.

Run the iteration till zero training error is achieved.





Naïve Bayes Algorithm – discrete X_i

- Train Naïve Bayes (examples) for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \; \pi_k \prod_i \theta_{ijk}$$

The Gaussian Probability Distribution

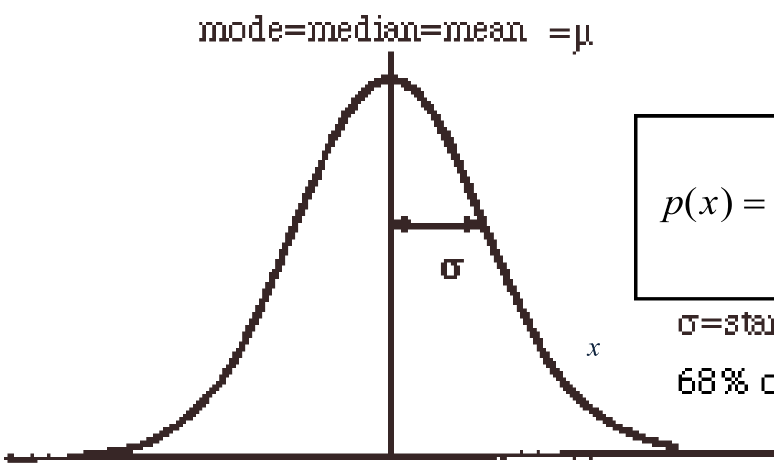
- It is a continuous distribution with pdf:

  μ = mean of distribution

  σ^2 = variance of distribution

  x is a continuous variable $(-\infty \leq x \leq)$

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



$$\boxed{p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad gaussian}$$

σ=standard deviation

68% of area within ±1σ

Continuous Features : learning and prediction

- For each target value Y_k (MLE estimate)

  $P(Y = y_k) \leftarrow$ No. of instances with Y_k class/No. of Total instances

- For each attribute value X_i estimate $P\,(X_i|Y = y_k)$
  - class conditional mean , variance

- Classify New Instance(x)

  Pick the most probable (MAP) Y

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}}\, P(Y = y_k)\Pi_i P(X_i|Y = y_k)$$

Naïve Bayes – Example 1

As a part of efforts to improve students' performance in the exams, you have been given the data showing number of study hours spent by students, their gender and their final results as pass or fail. Using this sample dataset, apply Naïve Bayes classification technique, to classify the test case:

{No of study hours = 3.5, Gender=”male”} either as “Pass”, or “Fail”.

$P(X|Y) \sim N(\mu, \sigma^2) \rightarrow$ GaussianNB (X_i – real valued)

No of study hours	Gender	Final result
4.5	Male	Pass
7	Female	Pass
2	Male	Fail
4	Female	Fail
2.5	Male	Fail
3	Female	Fail
8.3	Male	Fail
8	Female	Pass
9	Male	Pass

Naïve Bayes – Example 1

Look up tables

Maximum likelihood estimates (MLE's):

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

Gender	Pass	Fail
Male	2	3
Female	2	2

Final Result	Pass	Fail
Prior	4/9 = 0.44	5/9 = 0.56

No.of.Study	Pass = {4.5, 7, 8, 9}	Fail = {2, 4, 2.5, 3, 8.3}
Mean	7.2	3.9
Variance	2.95	4.64

No of study hours	Gender	Final result
4.5	Male	Pass
7	Female	Pass
2	Male	Fail
4	Female	Fail
2.5	Male	Fail
3	Female	Fail
8.3	Male	Fail
8	Female	Pass
9	Male	Pass

Naïve Bayes – Example 1

P(Pass | X) = P(X | Pass). P(Pass) / P(X)
= P(X | Pass). P(Pass)
= P(X | Pass). (0.44)
= P(Male | Pass). P(3.5 | Pass).(0.44)
= (2/4). 0.1056 . (0.44) = 0.0235

$$p(X_i = x | Y = y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_{ik}}{\sigma_{ik}}\right)^2}$$

P(Fail | X) = P(X | Fail). P(Fail) / P(X)
= P(X | Fail). P(Fail)
= P(X | Fail). (0.56)
= P(Male | Fail). P(3.5 | Fail).(0.56)
= (3/5). 0.1846 . (0.56) = 0.0615

Naïve Bayes – Example 2

•Consider a result prediction system where student's efforts are encoded as percent of
time a student has spent studying out of total available time.
•The input X is having just one feature representing the student's efforts having only four discrete values (25%, 50%, 75%, and 100%)
•The output Y is having 3 classes (First class, Second class, Fail)
•The priors for each class are: P(Y = First Class) = 0.5, P(Y = Second class) = 0.3, and P(Y = Fail) = 0.2.
•Based on the past data, the estimated the class-conditional probability P(X| Y) are shown in the following table.
Consider a following loss function

Note : The shared answer key has the Fail and First Class priors are Swapped.

Student's efforts	p(x|y=fail)	p(x|y=second class)	p(x|y=first class)
25	0.7	0.4	0.1
50	0.2	0.3	0.1
75	0.1	0.2	0.3
100	0	0.1	0.7

Naïve Bayes – Example 2

Consider modified Naïve Bayes hypothesis function: $l(\hat{y}, y)$ where $\hat{y}$ = predicted class label

Use this modified hypothesis function to classify each of the examples in the given table.

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}}\, l(y, \hat{y}) P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

$$l(\hat{y}, y) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} = Fail \text{ and } \hat{y} \neq y \\ 2 & \hat{y} = Second\ class \text{ and } \hat{y} \neq y \\ 4 & \hat{y} = First\ class \text{ and } \hat{y} \neq y \end{cases}$$

Student's efforts	p(x|y=fail)	p(x|y=second class)	p(x|y=first class)
25	0.7	0.4	0.1
50	0.2	0.3	0.1
75	0.1	0.2	0.3
100	0	0.1	0.7

Naïve Bayes – Example 2

Consider modified Naïve Bayes hypothesis function: $l(\hat{y}, y)$ where $\hat{y}$ = predicted class label

Use this modified hypothesis function to classify each of the examples in the given table.

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}}\, l(y, \hat{y}) P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

$$l(\hat{y}, y) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} = Fail \text{ and } \hat{y} \neq y \\ 2 & \hat{y} = Second\ class \text{ and } \hat{y} \neq y \\ 4 & \hat{y} = First\ class \text{ and } \hat{y} \neq y \end{cases}$$

Note : The shared answer key assumption that predicted and true values are not equal for below calculations

Student's efforts	p(x|y=fail) * P(y=fail) *loss	p(x|y=second class) * p(y=second class) * loss	p(x|y=first class) * p(y=first class)* loss	Highest value	New Y-Pred
25	0.35	0.24	0.08	0.35	fail
50	0.1	0.18	0.08	0.18	second class
75	0.05	0.12	0.24	0.24	first class
100	0	0.06	0.56	0.56	first class

Maximum Likelihood Estimation (MLE)

1. Determine formula for $LL(\theta)$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

3. Solve

MLE – Example 1

Let $T_1, T_2, \ldots. T_n$ be a random sample of a population describing the website loading time on a mobile browser with probability density function given as:

$$f(t/\theta) = \frac{1}{\theta} t^{\frac{(1-\theta)}{\theta}} \quad where \;\; 0 < t < 1 \; and \; 0 < \theta < \infty$$

Find the maximum likelihood estimator of θ. What is the estimate of θ, if the website loading time from four samples are $t_1 = 0.10$, $t_2 = 0.22$, $t_3 = 0.54$, $t_4 = 0.36$.

t_i	$f(t \mid \theta)$
0.10	$\frac{1}{\theta} 0.10^{\frac{(1-\theta)}{\theta}}$
0.22	$\frac{1}{\theta} 0.22^{\frac{(1-\theta)}{\theta}}$
0.54	$\frac{1}{\theta} 0.54^{\frac{(1-\theta)}{\theta}}$
0.36	$\frac{1}{\theta} 0.36^{\frac{(1-\theta)}{\theta}}$

1. Determine formula for $LL(\theta)$
2. Differentiate $LL(\theta)$ w.r.t. (each) θ
3. Solve

MLE – Example 1

1. Determine formula for $LL(\theta)$

$$L(\theta) = \frac{1}{\theta}\, 0.10^{\frac{(1-\theta)}{\theta}} * \frac{1}{\theta}\, 0.22^{\frac{(1-\theta)}{\theta}} * \frac{1}{\theta}\, 0.54^{\frac{(1-\theta)}{\theta}} * \ldots\ldots * \frac{1}{\theta}\, 0.36^{\frac{(1-\theta)}{\theta}}$$

$$= \theta^{-4} \left(\prod_{i=1}^{4} \mathbf{t}_{\mathbf{i}}\right)^{\frac{(1-\theta)}{\theta}}$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

3. Solve

t_i	$f(t\|\theta)$
0.10	$\frac{1}{\theta}\, 0.10^{\frac{(1-\theta)}{\theta}}$
0.22	$\frac{1}{\theta}\, 0.22^{\frac{(1-\theta)}{\theta}}$
0.54	$\frac{1}{\theta}\, 0.54^{\frac{(1-\theta)}{\theta}}$
0.36	$\frac{1}{\theta}\, 0.36^{\frac{(1-\theta)}{\theta}}$

MLE – Example 1

1. Determine formula for $LL(\theta)$

$$L(ɵ) = ɵ^{-4} \left(\prod_{i=1}^{4} \mathbf{t_i}\right)^{\frac{(1-ɵ)}{ɵ}}$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

$$LL(ɵ) = \log \left[ɵ^{-4} \left(\prod_{i=1}^{4} \mathbf{t_i}\right)^{\frac{(1-ɵ)}{ɵ}}\right]$$

$$= \log (ɵ^{-4}) + \log \left(\prod_{i=1}^{4} \mathbf{t_i}\right)^{\frac{(1-ɵ)}{ɵ}}$$

$$= -4 \log (ɵ) + \frac{(1-ɵ)}{ɵ} (\log (0.10*0.22*0.54*0.36))$$

$$= -4 \log (ɵ) + \frac{1}{ɵ} (\log (0.10*0.22*0.54*0.36)) - (\log (0.10*0.22*0.54*0.36))$$

3. Solve

$$\text{Gradient} (LL(ɵ)) = \frac{-4}{(ɵ)} - \frac{(\log (0.004276))}{(ɵ^2)} = 0$$

$$\frac{-4}{(ɵ)} = \frac{(\log (0.004276))}{(ɵ^2)}$$

$$ɵ = \frac{-(\log (0.004276))}{(4)}$$

ɵ = 1.3636 (base e) , ɵ = 1.9673 (base 2)

MLE – Example 2

Consider inputs x_i which are real valued attributes and the outputs y_i which are real valued of the form $y_i = f(x_i) + e_i$, where $f(x_i)$ is the true function and e_i is a random variable representing laplacian noise with PDF given by

$$\boxed{f\left(y_i/\theta\right) = \frac{1}{2\theta} * e^{\frac{-|y_i-\mu|}{\theta}}}$$

Implementing a linear regression model of the form , $h(x_i) = \sum_{i=0}^{n} \theta_i x_i$ and $\mu = h_i(x_i)$ find the maximum likelihood estimator of è. Comment on the loss function.

1. Determine formula for $LL(\theta)$
2. Differentiate $LL(\theta)$ w.r.t. (each) θ
3. Solve

MLE

1. Determine formula for $LL(\theta)$

$$L(\theta) = \frac{1}{2\theta}\, e^{\frac{-|y_1 - \mu|}{\theta}} * \frac{1}{2\theta}\, e^{\frac{-|y_2 - \mu|}{\theta}} * \frac{1}{2\theta}\, e^{\frac{-|y_3 - \mu|}{\theta}} * \ldots\ldots\ldots * \frac{1}{2\theta}\, e^{\frac{-|y_n - \mu|}{\theta}}$$

$$= \left(\prod_{i=1}^{n} \frac{1}{2\theta}\, e^{\frac{-|y_i - \mu|}{\theta}}\right)$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

3. Solve

MLE

1. Determine formula for $LL(\theta)$

   $L(\theta) = \left(\prod_{i=1}^{n} \frac{1}{2\theta}\, e^{\frac{-|y_i - \mu|}{\theta}}\right)$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ – natural log (ln)

   $LL(\theta) = \ln\left(\prod_{i=1}^{n} \frac{1}{2\theta}\, e^{\frac{-|y_i - \mu|}{\theta}}\right)$

   $= \sum_{i=1}^{n} \ln\left(\frac{1}{2\theta}\, e^{\frac{-|y_i - \mu|}{\theta}}\right)$

   $= -\ln(2\theta) \sum_{i=1}^{n} 1 + \sum_{i=1}^{n} \ln\left(e^{\frac{-|y_i - \mu|}{\theta}}\right)$

3. Solve

   $= \underset{\theta}{argmax}\; -n \ln(2\theta) - \sum_{i=1}^{n} \frac{|y_i - \mu|}{\theta}$

   $= \underset{\theta}{argmin}\; n \ln(2\theta) + \sum_{i=1}^{n} \frac{|y_i - \mu|}{\theta}$

MLE

1. Determine formula for $LL(\theta)$

   $L(\theta) = \left(\prod_{i=1}^{n} \frac{1}{2\theta}\, e^{\frac{-|\mathbf{y}_i - \mu|}{\theta}}\right)$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

   $LL(\theta) = \ln\left(\prod_{i=1}^{n} \frac{1}{2\theta}\, e^{\frac{-|\mathbf{y}_i - \mu|}{\theta}}\right)$

   $= \underset{\theta}{argmin}\ n \ln(2\theta) + \sum_{i=1}^{n} \frac{|\mathbf{y}_i - \mu|}{\theta}$

3. Solve

   Gradient ($LL(\theta)$) $= \frac{n}{(\theta)} - \frac{\sum_{i=1}^{n} |\mathbf{y}_i - \mu|}{(\theta^2)} = 0$

$$\frac{n}{(\theta)} = \frac{\sum_{i=1}^{n} |\mathbf{y}_i - \mu|}{(\theta^2)}$$

$$\theta = \frac{\sum_{i=1}^{n} |\mathbf{y}_i - \mu|}{\mathrm{n}}$$

Instead of MSE, MAE is the maximum likelihood hypothesis. So MAE is appropriate for the loss function

Happy Learning