INF8953CE - Fall 2020

Machine Learning

- Sarath Chandar

7. Probabilistic Generative Models.

7. Probabilistic Generative Models

Generalized linear models:-

In linear regression, the model prediction $y(x;w)$ was given by a linear function of the parameters 'w'.

In the simplest case, the model is also linear in the input variables.

$$y(x) = w^T x + w_0$$

↑ real number.

For classification, we wish to predict discrete class labels or posterior probabilities that lie in the range (0,1).

↳ We transform the linear function of w using a non-linear function $f(.)$ so that

$$y(x) = f(w^T x + w_0)$$

$f(.)$ - activation function.

Note: This model is no longer linear in the parameters due to the presence of non-linear $f(.)$.

The decision surface is given by

$$y(x) = \text{Constant}$$

proving that the decision surface is linear.

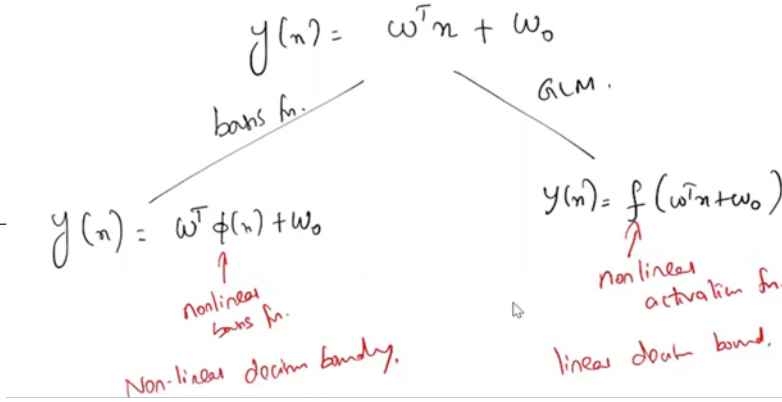
$$f(w^T x + w_0) = \text{Constant}$$

$$w^T x + w_0 = f^{-1}(\text{Constant})$$

$$w^T x + w_0 = \text{Constant}.$$

Decision surfaces are linear functions of x, even if $f(.)$ is non-linear! This type of models are called generalized linear models (GLM).

GLMs have complex analytical and computational properties than linear models. However, they are still relatively simple when compared to more general non-linear models



Probabilistic Generative Models :-

Model the class-conditional densities $p(x|C_k)$ and class priors $p(C_k)$ and then use Bayes' theorem to complete posterior probabilities $p(C_k|x)$.

2-class problem :-

$$p(C_1|x) = \frac{p(x|C_1)\,p(C_1)}{p(x)}$$

$$= \frac{p(x|C_1)\,p(C_1)}{p(x|C_1)\,p(C_1) + p(x|C_2)\,p(C_2)}$$

mul and div by numerator

$$= \frac{1}{1 + \frac{p(x|C_2)\,p(C_2)}{p(x|C_1)\,p(C_1)}}$$

reverse the fraction and take exp and ln which will cancel out with each other

$$= \frac{1}{1 + \exp\left(-\ln \frac{p(x|C_1)\,p(C_1)}{p(x|C_2)\,p(C_2)}\right)}$$

$$= \frac{1}{1+\exp(-a)} = \sigma(a)$$

where $a = \ln\left(\frac{p(x|C_1)\,p(C_1)}{p(x|C_2)\,p(C_2)}\right)$

$$\sigma(a) = \frac{1}{1+e^{-a}}$$ (logistic sigmoid fn).

Sigmoid function:-

→ Squashing function

→ range (0,1)

→ Symmetry property : $\sigma(-a) = 1 - \sigma(a)$

→ inverse : $a = \ln\left(\frac{\sigma}{1-\sigma}\right)$

logit function

logit function – log of the ratio of probabilities $\ln\left(\frac{P(C_1|x)}{P(C_2|x)}\right)$ for the two classes, also known as the log odds.

For $k > 2$ classes:

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{\sum_j P(x|C_j)P(C_j)}$$

$$p(C_k|x) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

where $a_k = \ln\left(p(x|C_k)\, p(C_k)\right)$

↓

Normalized exponential

↓

Multiclass generalization of the logistic sigmoid.

↓

also known as softmax function, as it represents a smoothed version of the 'max' function because if $a_k \gg a_j$ for all $j \neq k$, then $p(C_k|x) \simeq 1$ and $p(C_j|x) \simeq 0$.

Gaussian Discriminant Analysis:-

Assumptions:-

① class conditional densities are Gaussian.

② All classes share the same covariance matrix.

$$p(x|C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)\right\}$$

Mean : μ_k

Covariance : Σ

Note: $|\Sigma|$ is the determinant of Σ.

Consider 2-class problem:

$$p(C_1|x) = \sigma(a)$$

where $$a = \ln\left(\frac{p(x|C_1)\,p(C_1)}{p(x|C_2)\,p(C_2)}\right)$$

$$a = \ln\left(\frac{p(x|C_1)}{p(x|C_2)}\right) + \ln\left(\frac{p(C_1)}{p(C_2)}\right)$$

$$= \ln \exp\left\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\right\}$$

$$+ \ln \frac{p(C_1)}{p(C_2)}$$

$$= -\frac{1}{2}\Big(x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_1 - \mu_1^T\Sigma^{-1}x + \mu_1^T\Sigma^{-1}\mu_1$$

$$- x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_2 + \mu_2^T\Sigma^{-1}x - \mu_2^T\Sigma^{-1}\mu_2\Big)$$

$$+ \ln\left(\frac{p(C_1)}{p(C_2)}\right)$$

$$= x^T \Sigma^{-1} \mu_1 - x^T \Sigma^{-1} \mu_2 - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2$$

$$+ \ln \frac{P(C_1)}{P(C_2)}$$

$$= x^T \left(\Sigma^{-1} (\mu_1 - \mu_2) \right) - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2$$

$$+ \ln \frac{P(C_1)}{P(C_2)}$$

$$= w^T x + w_0$$

where $w = \Sigma^{-1} (\mu_1 - \mu_2)$

$$w_0 = -\frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 + \ln \frac{P(C_1)}{P(C_2)}$$

$$P(C_1 | x) = \sigma(w^T x + w_0)$$

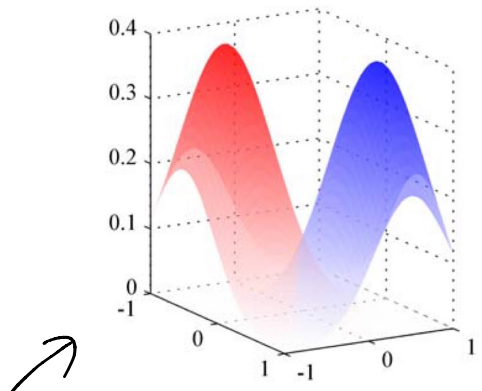
Note: Quadratic terms in x from the exponents of the Gaussian densities have cancelled due to the assumption of the common covariance matrices.

The model is a linear function of x.

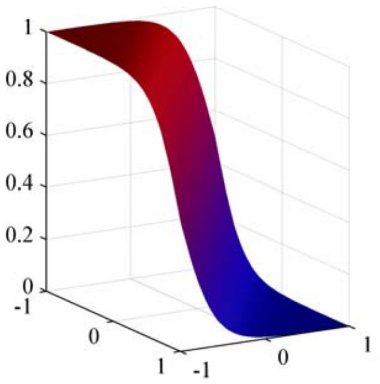
⇓

decision boundary is also a linear function of x.

Prior probabilities $p(C_k)$ enter only through the bias parameter w_0 so that the changes in the prior have the effect of making parallel shifts of the decision boundary.



Class Conditional density for 2 classes.



$p(C_1|x)$

Parameters of the model: $\mu_1, \mu_2, \Sigma, p(C_1), p(C_2)$

How to learn the parameters?

Consider a dataset $\mathcal{D} = \{x^{(n)}, t^{(n)}\}_{n=1}^{N}$ of N examples.

$t_n = 1$ class C_1

$t_n = 0$ class C_2

Let $P(C_1) = \pi$ prior probability for C_1

$$P(C_2) = 1 - \pi$$

Consider data point x,

$$P(x^{(n)}, C_1) = P(C_1)\, P(x^{(n)} | C_1) = \pi \cdot \mathcal{N}(x^{(n)} | \mu_1, \Sigma)$$

$$P(x^{(n)}, C_2) = P(C_2)\, P(x^{(n)} | C_2) = (1-\pi)\, \mathcal{N}(x^{(n)} | \mu_2, \Sigma)$$

Likelihood of the datapoint $x^{(n)}, t^{(n)}$ is given by

$$\left[\pi\, \mathcal{N}(x^{(n)} | \mu_1, \Sigma)\right]^{t^{(n)}} \left[(1-\pi)\, \mathcal{N}(x^{(n)} | \mu_2, \Sigma)\right]^{1-t^{(n)}}$$

Assuming that the examples are independent and identically distributed (i.i.d)

$$P(\mathcal{D} | \pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^{N} \left[\left[\pi\, \mathcal{N}(x^{(n)} | \mu_1, \Sigma)\right]^{t^{(n)}} \left[(1-\pi)\, \mathcal{N}(x^{(n)} | \mu_2, \Sigma)\right]^{1-t^{(n)}} \right]$$

Likelihood function.

We want to find the parameters that maximizes the likelihood function.

Maximum likelihood solution :-

We will often maximize log of the likelihood function instead of the likelihood fn. itself.

$$\ln p(D|\pi,\mu_1,\mu_2,\Sigma) = \sum_{n=1}^{N} \ln \left[\left[\pi \mathcal{N}(x^{(n)}|\mu_1,\Sigma) \right]^{t^{(n)}} \left[(1-\pi)\mathcal{N}(x^{(n)}|\mu_2,\Sigma) \right]^{1-t^{(n)}} \right]$$

$$= \sum_{n=1}^{N} \left[t^{(n)} \ln \pi \mathcal{N}(x^{(n)}|\mu_1,\Sigma) + (1-t^{(n)}) \ln (1-\pi)\mathcal{N}(x^{(n)}|\mu_2,\Sigma) \right]$$

Maximization w.r.t. π :

$$\sum_{n=1}^{N} \left[t^{(n)} \ln \pi + (1-t^{(n)}) \ln (1-\pi) \right] + \text{Const}$$

diff. w.r.t. π and set it to zero.

$$\sum_{n=1}^{N} \left(\frac{t^{(n)}}{\pi} + \frac{1-t^{(n)}}{1-\pi}(-1) \right) = 0$$

$$\sum_{n=1}^{N} \left(\frac{t^{(n)}(1-\pi) - (1-t^{(n)})\pi}{\pi(1-\pi)} \right) = 0$$

$$\sum_{n=1}^{N} \left(t^{(n)} - t^{(n)}\pi - \pi + t^{(n)}\pi \right) = 0$$

$$\left(\sum_{n=1}^{N} t^{(n)} \right) - N\pi = 0$$

fraction of points in C_1.

$$\pi = \frac{\sum_{n=1}^{N} t^{(n)}}{N} = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

where N_1 - number of data points in class C_1

N_2 - number of data points in class C_2.

Maximization w.r.t. μ_1 :

$$\sum_{n=1}^{N} t^{(n)} \ln \mathcal{N}(x^{(n)} | \mu_1, \Sigma) + \text{Const.}$$

$$= -\frac{1}{2} \sum_{n=1}^{N} t^{(n)} (x^{(n)} - \mu_1)^T \Sigma^{-1} (x^{(n)} - \mu_1) + \text{Const}$$

diff. w.r.t. μ_1 and set it to zero.

$$\sum_{n=1}^{N} t^{(n)} (x^{(n)} - \mu_1) = 0$$

$$N_1 \mu_1 = \sum_{n=1}^{N} t^{(n)} x^{(n)}$$

$$\mu_1 = \frac{1}{N_1} \sum_{n=1}^{N} t^{(n)} x^{(n)}$$

↗ mean of all the input vectors $x^{(n)}$ assigned to class C_1.

Similarly, $\mu_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t^{(n)}) x^{(n)}$

Maximization w.r.t. Σ :

Maximization w.r.t Σ is bit involved.

$$\Sigma = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2$$

where $S_1 = \frac{1}{N_1} \sum_{n \in C_1} (x^{(n)} - \mu_1)(x^{(n)} - \mu_1)^T$

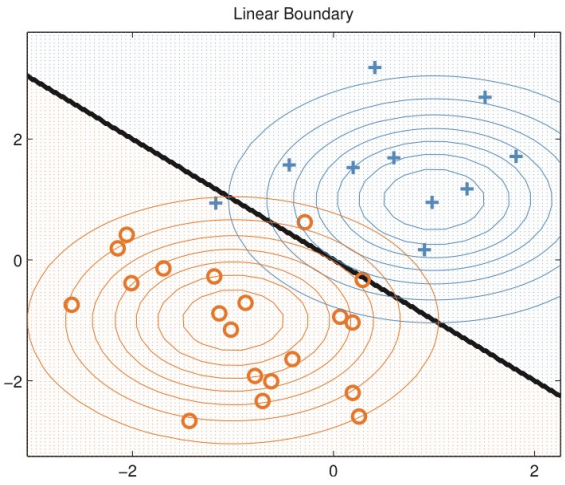
$$S_2 = \frac{1}{N_2} \sum_{n \in C_2} (x^{(n)} - \mu_2)(x^{(n)} - \mu_2)^T$$

Note : If we relax the assumption of a shared covariance matrix and allow each class conditional density $p(x|C_k)$ to have its own

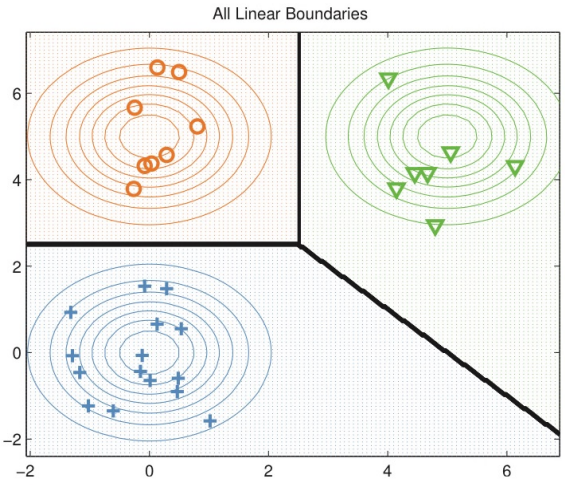
Covariance matrix Σ_k, then we will obtain quadratic functions of X.

↳ Quadratic Discriminant Analysis (QDA)

LDA

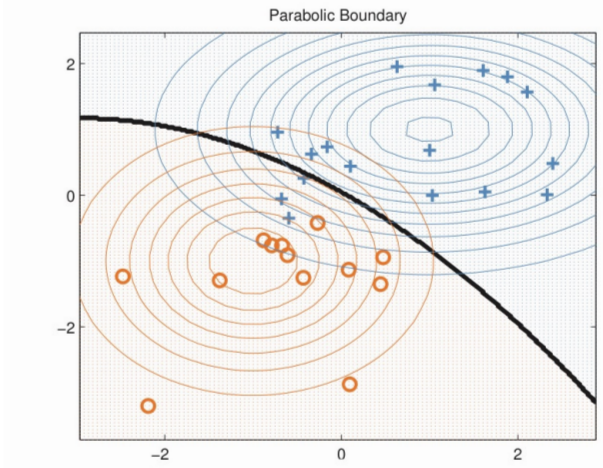


2-classes

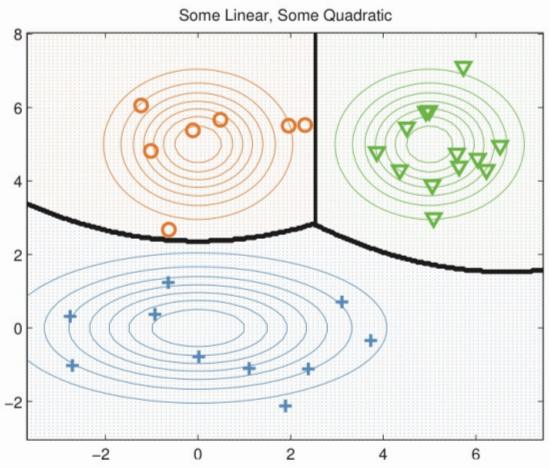


3-classes

QDA



2-classes



3-classes

Summary:

Generative models for classification :-

Compute $P(x|C_k)$ and $P(C_k)$ and then use Bayes' law to compute $P(C_k|x)$.

GDA: Class conditional densities are Gaussians with shared Covariance matrix.

QDA: class conditional densities are Gaussians with separate covariance matrix.

If there are M features, then

GDA: kM parameters for the means,

$\frac{M(M+1)}{2}$ Parameters for the shared Covariance matrix.

k-1 parameters for $P(C_k)$

QDA: kM parameters for means.

$k \cdot \frac{M(M+1)}{2}$ Parameters for Covariance matrices.

k-1 parameters for $P(C_k)$.

QDA has much more parameters. But the shared Covariance assumption is too restrictive.

Naive Bayes assumption:-

Features are conditionally independent given the class label.

$$P(x|C_k) = P(x_1 \ldots x_M | C_k)$$
$$= \prod_{i=1}^{M} P(x_i | C_k)$$

Now the Covariance matrix is diagonal.

This model is known as Gaussian Naive Bayes. (GNB)

GNB requires kM parameters for means,

kM parameters for variance, and

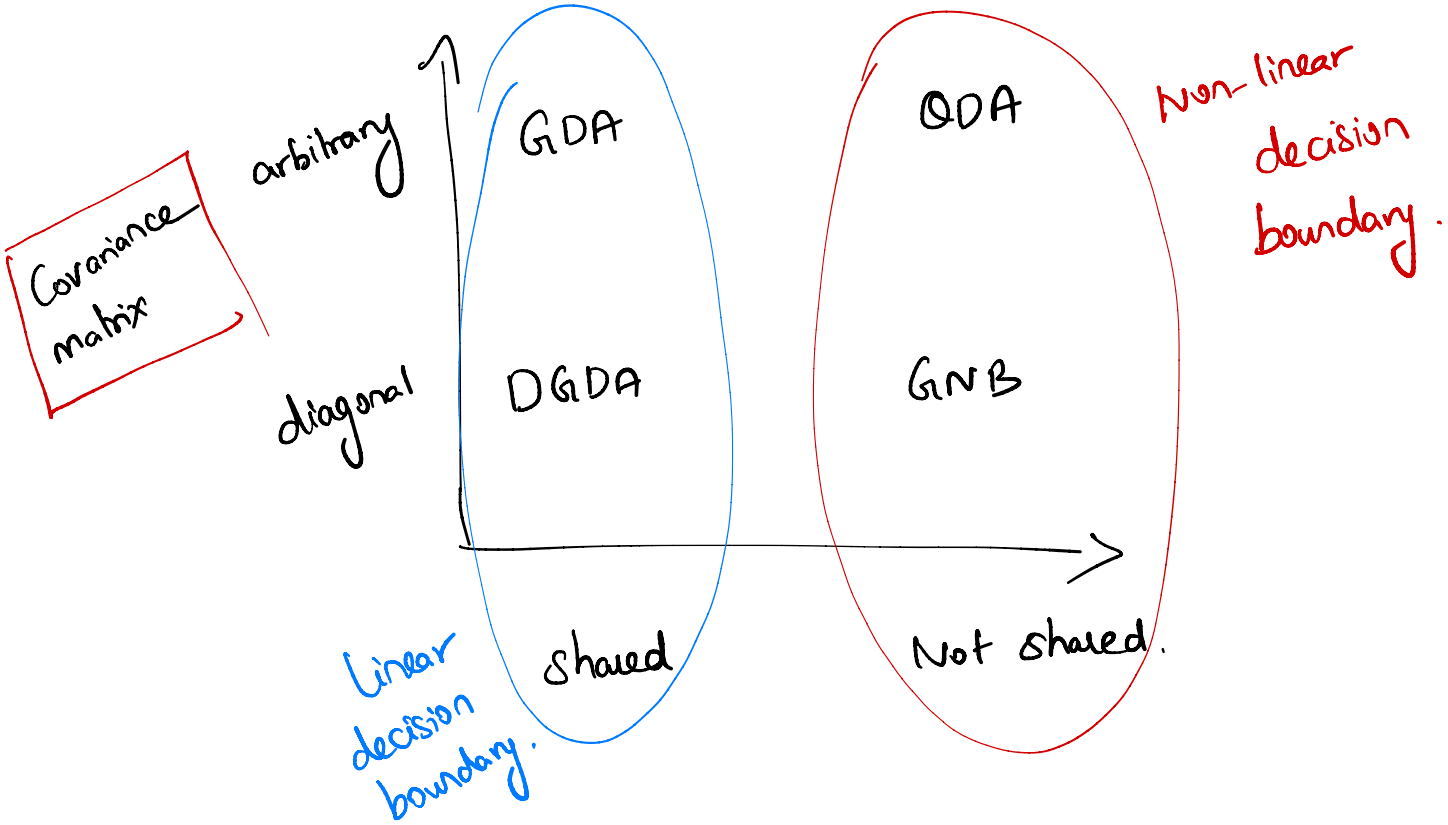
$k-1$ parameters for $P(C_k)$.

This requires lesser number of parameters and still has non-linear decision boundary like QDA.

Note 1: If the diagonal Covariance matrix is shared among the classes, then the decision boundary will be linear. GNB with shared diagonal Covariance matrices is known as DGDA (Diagonal GDA).

Note 2: We can learn the Covariance matrix in all these methods by using maximum likelihood.

Summary: Continuous i/p features



Discrete features:-

Consider binary features: $x_i \in \{0, 1\}$

If there are M inputs, then a general distribution would correspond to a table of 2^M numbers for each class, containing 2^{M-1} independent variables.

Naive Bayes' assumption :- $P(x|C_k) = \prod_{i=1}^{M} P(x_i|C_k)$

$$P(x|C_k) = P(x_1 \ldots x_M | C_k)$$

$$= P(x_1|C_k) P(x_2|C_k, x_1) P(x_3|C_k, x_1, x_2)$$

$$\ldots P(x_M|C_k, x_1 \ldots x_{M-1})$$

(chain rule)

$$= P(x_1|C_k) P(x_2|C_k) \ldots P(x_M|C_k)$$

(Naive Bayes assumption)

$$= \prod_{i=1}^{M} P(x_i|C_k)$$

Each $P(x_i|C_k)$ can be modeled as a Bernoulli with parameter μ_{k_i} (i.e. $P(x_i=1|C_k)$)

$$P(x|C_k) = \prod_{i=1}^{M} \mu_{ki}^{x_i} (1-\mu_{ki})^{1-x_i}$$

This requires only M parameters for each class.

Consider 2 class problem :-

Likelihood : $P(D|\mu_1, \mu_2, \pi)$

$$= \prod_{n=1}^{N} \Big(\underbrace{\pi \prod_{i=1}^{M} \mu_{1i}^{x_i^{(n)}} (1-\mu_{1i})^{(1-x_i^{(n)})}}_{x} \Big)^{t^{(n)}} \Big(\underbrace{(1-\pi) \prod_{i=1}^{M} \mu_{2i}^{x_i^{(n)}} (1-\mu_{2i})^{(1-x_i^{(n)})}}_{y} \Big)^{1-t^{(n)}}$$

$$\ln P(D|\mu_1,\mu_2,\pi) = \sum_{i=1}^{N}\left[t^{(n)} \ln x + (1-t^{(n)}) \ln y \right]$$

$$\frac{\partial \ln P(D|.)}{\partial \mu_{1i}} = \frac{\partial}{\partial \mu_{1i}}\left[\sum_{n=1}^{N} \left(t^{(n)} \ln \mu_{1i}^{x_i^{(n)}} (1-\mu_{1i})^{1-x_i^{(n)}} \right) \right]$$

$$= \frac{\partial}{\partial \mu_{1i}}\left[\sum_{n=1}^{N} \left(t^{(n)} x_i^{(n)} \ln \mu_{1i} + t^{(n)}(1-x_i^{(n)}) \ln(1-\mu_{1i}) \right) \right]$$

diff. and set it to zero.

$$\sum_{n=1}^{N} \left(t^{(n)} x_i^{(n)} \frac{1}{\mu_{1i}} + t^{(n)}(1-x_i^{(n)}) \frac{1}{(1-\mu_{1i})}(-1) \right) = 0$$

$$\sum_{n=1}^{N} \left(t^{(n)} x^{(n)}(1-\mu_{1i}) - t^{(n)}(1-x_i^{(n)})\mu_{1i} \right) = 0$$

$$\sum_{n=1}^{N} \left(t^{(n)} x^{(n)} - t^{(n)} x^{(n)} \mu_{1i} - t^{(n)} \mu_{1i} + t^{(n)} x_i^{(n)} \mu_{1i} \right) = 0$$

$$\sum_{n=1}^{N} t^{(n)} \mu_{1i} = \sum_{n=1}^{N} t^{(n)} x_i^{(n)}$$

$$\boxed{\mu_{1i} = \frac{\sum_{n=1}^{N} t^{(n)} x_i^{(n)}}{\sum_{n=1}^{N} t^{(n)}}}$$

$$\mu_{1i} = \frac{\text{number of examples where } x_i = 1 \text{ and } t = 1}{\text{number of examples where } t = 1}$$

$$\mu_{2i} = \frac{\text{number of examples where } x_i = 1 \text{ and } t = 0}{\text{number of examples where } t = 0}$$

Note: If the features are Categorical instead of binary, then the maximum likelihood solution for the parameters of the Categorical distribution would be the frequency of occurrence of that Category in the given class.

Example application: Text classification.

Given a document, which is a collection of words, classify the document into one of the 'k' topics.

ex: Politics, sports, arts...

Assume there are M words in the vocabulary.

Each document can be represented as a

M dimensional feature vector

$$(x_1 \ldots x_M)$$

x_i = 1 if first word is present in the document
= 0 otherwise.

$$P(C_k|x) = P(x|C_k)\,P(C_k)$$

$$= \prod_{i=1}^{M} P(x_i|C_k)\,P(C_k)$$

Issue: Given limited number of examples per class, we might not see all words appearing atleast once in all documents of every class.

if $x_i = 0$ whenever $t = 1$, then

$$\mu_{1i} = 0$$

This would make $P(C_k|x)$ zero often.

How to avoid this?

Laplace Smoothing:-

$$\mu_{1i} = \frac{(\text{number of examples where } x_i = 1 \text{ and } t = 1) + 1}{(\text{number of examples where } t = 1) + 2}$$

→ If no example from a class, it reduces to a prior probability of $1/2$.

↓

bias!

→ If all examples have $x_i = 1$, then

$$P(x_i = 0 \mid C_1) = \frac{1}{(\#\text{examples where } t=1) + 2}$$

→ the bias decreases, as we see the features set to 1 more often.

Evaluation metrics for classification problem:-

2-class problems:-

Positive / Negative

$$\text{Accuracy} = \frac{\text{number of correctly classified instances}}{\text{total number of instances.}}$$

Is this a good metric for all tasks?

Consider Cancer prediction. 9900 patients: Cancer = No

100 patients: cancer = yes

Classifier: default answer of "No"

↳ this classifier has 99% accuracy.

But this classifier is useless since it cant predict who has cancer.

Not every error has same cost!

Types of error this model can make:

① Classify patients without cancer with "yes"

② classify patients with cancer as "no".

Second error is more dangerous than the first error.

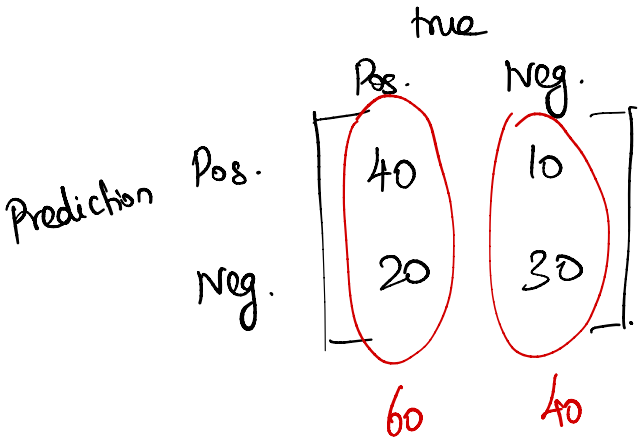
Confusion matrix:

Prediction \ True value	Pos.	Neg.
Pos.	True Positive	False Positive
Neg.	False Negative	True Negative

In previous example, ① is False positive and ② is false negative.

Example confusion matrix:

100 examples 60+ / 40 −



$\begin{pmatrix} 60 & 0 \\ 0 & 40 \end{pmatrix}$ ← ideal confusion matrix.

$\begin{pmatrix} 30 & 30 \\ 40 & 30 \end{pmatrix}$ ← bad confusion matrix.

$$\text{Precision} = \frac{\text{true Pos}}{\text{true Pos + false pos}} = \frac{tp}{tp+fp}$$

$$\text{recall} = \frac{\text{true Pos}}{\text{true pos + false neg}} = \frac{tp}{tp+fn}$$

Precision → Out of all positive predictions, how many of them are truely positive?

Recall → Out of all positive examples in the data, how many of them are correctly identified as positive by the classifier?

* Precision is important in face password systems.

* Recall is important in email Spam Classifier

↳ here good email
↓
Positive class.

It is easy to achieve high precision at the cost of low recall (and vice-versa).

① classifier which says default "yes"

↳ 100 % recall, bad precision.

② classifier which says default "no"

↳ 100% precision, bad recall.

What if both precision and recall are important?

$$F_1 \text{ measure} = \frac{2.P.r}{P+r}$$

P - Precision
r - recall

Harmonic mean of precision and recall.

You Should know!

1. Generalized linear models
2. logistic Sigmoid function
3. Softmax function
4. Gaussian discriminant analysis
5. independent and identically distributed (i.i.d)
6. Maximum likelihood approach
7. Quadratic discriminant analysis (QDA)
8. Naive Bayes assumption.
9. Gaussian Naive Bayes and Diagonal GDA.
10. Discrete Naive Bayes
11. Text classification
12. Laplace Smoothing.
13. Evaluation metrics for classification
    - Accuracy, Confusion matrix, false positive, false negative, precision, recall, F_1 measure.