CAP 5610: Machine Learning

Lecture 4:
Bayes Classifiers II

Instructor: Dr. Gita Sukthankar
Email: gitars@eecs.ucf.edu

Reference Materials From: GuoJun Qi, Chris Bishop

Assignment 1: kNN and naive Bayes (due Sept 24)

- [Iris dataset]
- Implement kNN:
  - Distance metrics: Euclidean and cosine
  - Different values of k
- Implement naive Bayes (MLE) assuming Gaussian distribution model for features
- Report:
  - Accuracy table
  - Confusion matrix
  - Discuss MLE vs. MAP estimation
- Do not use any external machine learning libraries or toolboxes for algorithms or evaluation
- Please cite any sources you refer to

Object Detection Example

Query Image

TP

FP

Retrieved Images

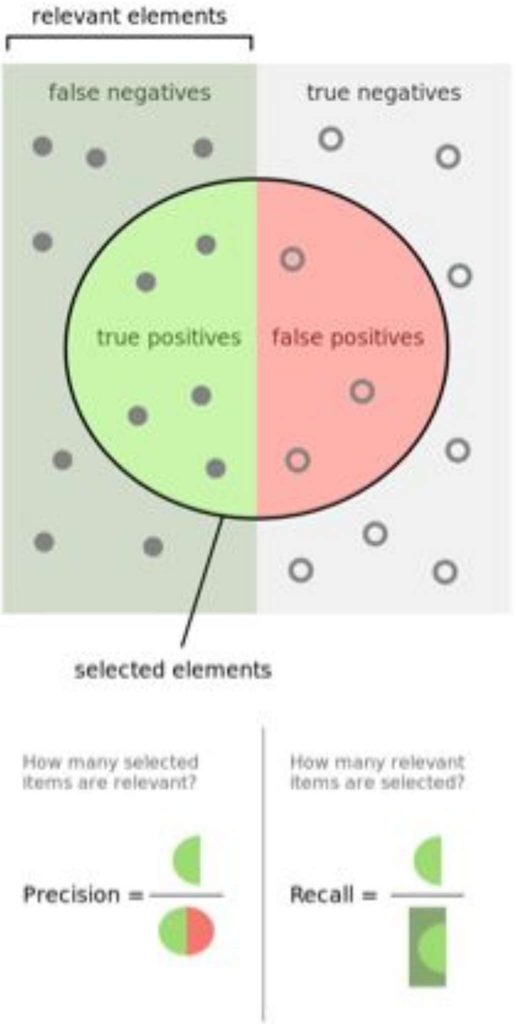
Unretrieved Images

FN

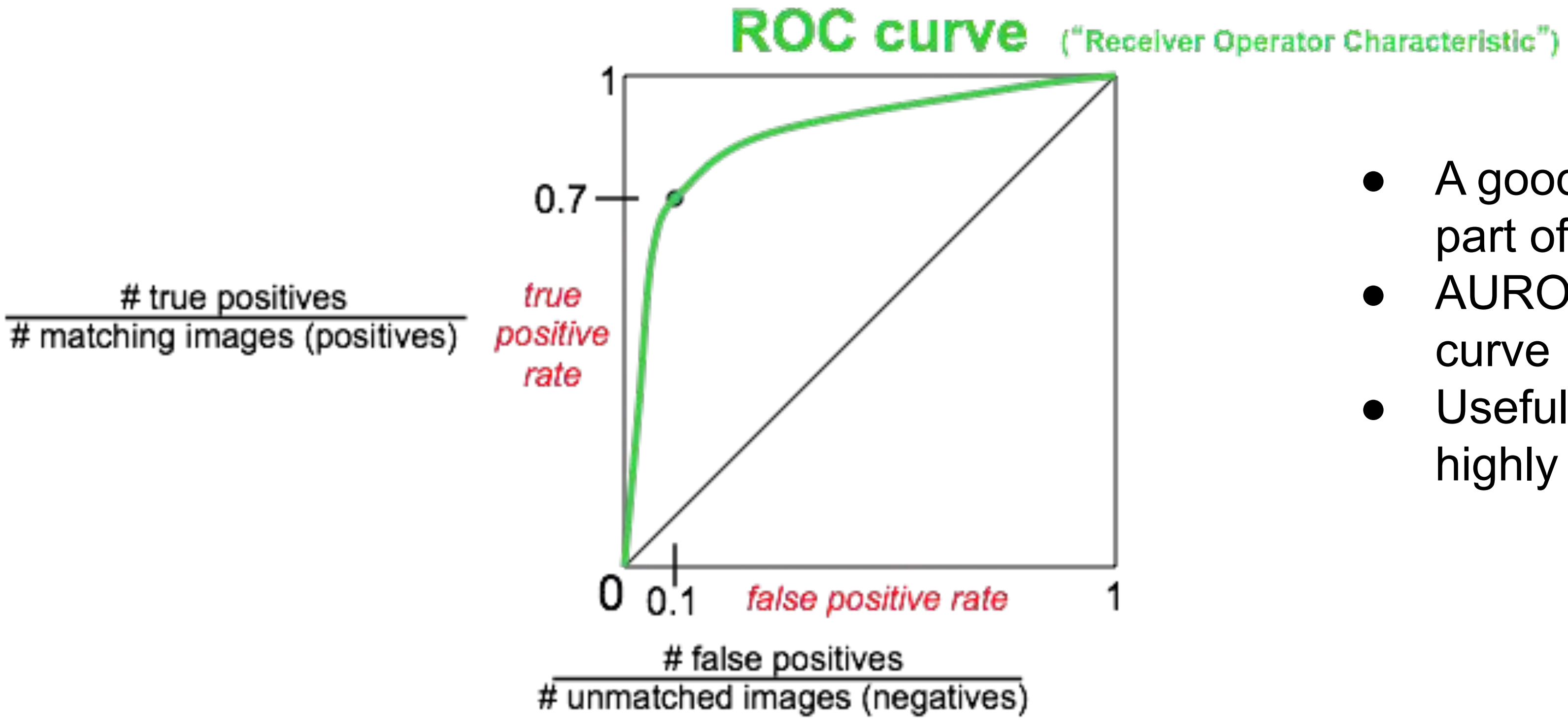
TN

Precision/Recall

- Precision
  - TP/(TP+FP)
- Recall (true positive rate)
  - TP/P or TP/(TP+FN)
- Specificity (true negative rate, 1-FPR)
  - TN/N or TN/(TN+FP)
- False positive rate
  - FP/N or FP/FP+TN
- F1 Score
  - Harmonic average of precision/recall
  - 2 (precision x recall)/precision+recall
  - Other F scores weight precision and recall differently



ROC Curve



- A good ROC curve should lie in the top left part of the graph.
- AUROC: measure of the area under the curve
- Useful metric even when the classes are highly unbalanced

ROC Curves

- Generated by counting # correct/incorrect matches, for different thresholds
- Want to maximize area under the curve
- Useful for comparing different feature matching methods
- For more info: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

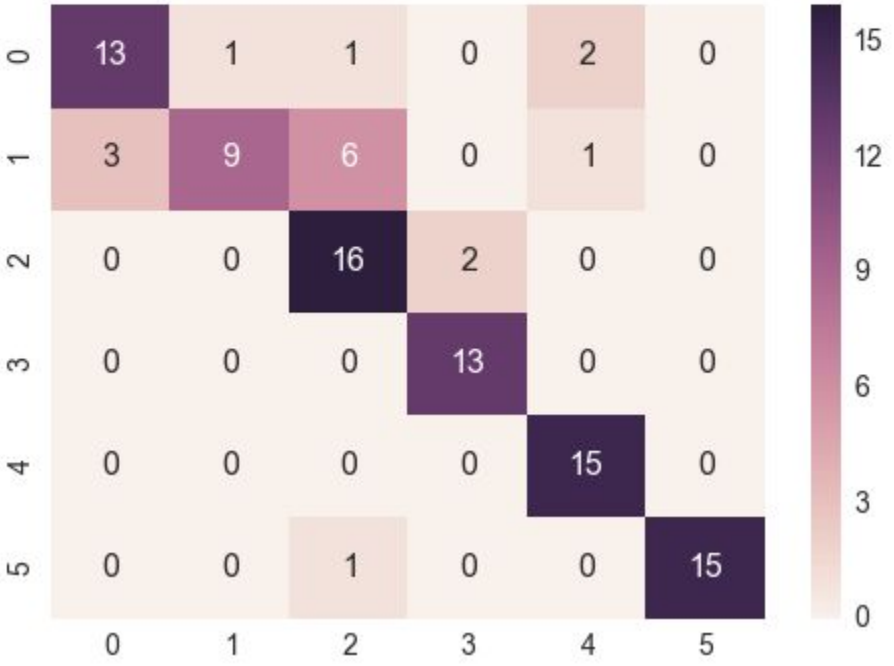
Confusion Matrix

Commonly to report classifier results in supervised learning

Rows are predicted class; columns are actual class.

A good confusion matrix has high values on the diagonal and low values everywhere else.

	image 0	image 1	image 2
class 0	5	2	1
class 1	2	7	1
class 2	3	1	8



Recap: Bayes Classifier

- MAP rule: given an input feature vector X=(X_1,X_2,…,X_N) with N attributes, the optimal binary prediction on its class Y is made by

$$Y^{*} = \mathrm{argmax}_{Y\in\{0,1\}} P(Y|X)$$

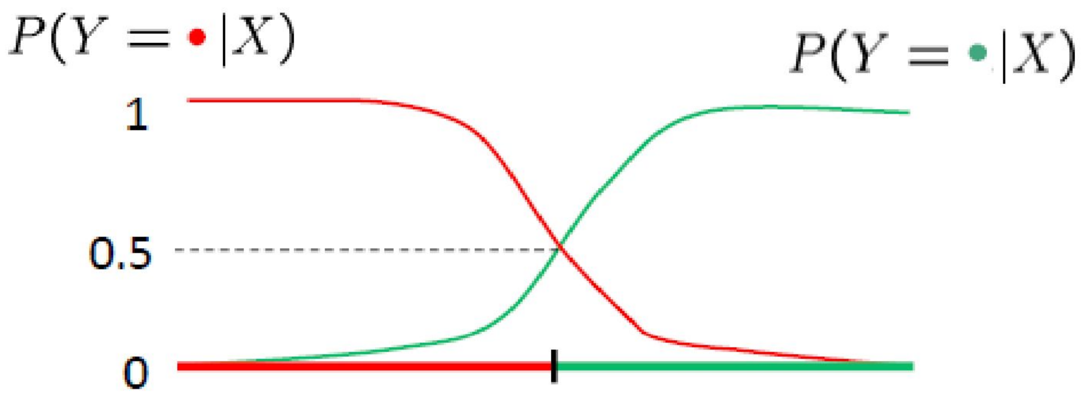
- Bayes rule: $P(Y|X) \propto P(X|Y)P(Y)$
  - where $P(X|Y)$ is the class-conditional distribution for class y, and
  - $P(Y)$ is the prior distribution.

Bayes Error

- Two types of prediction error

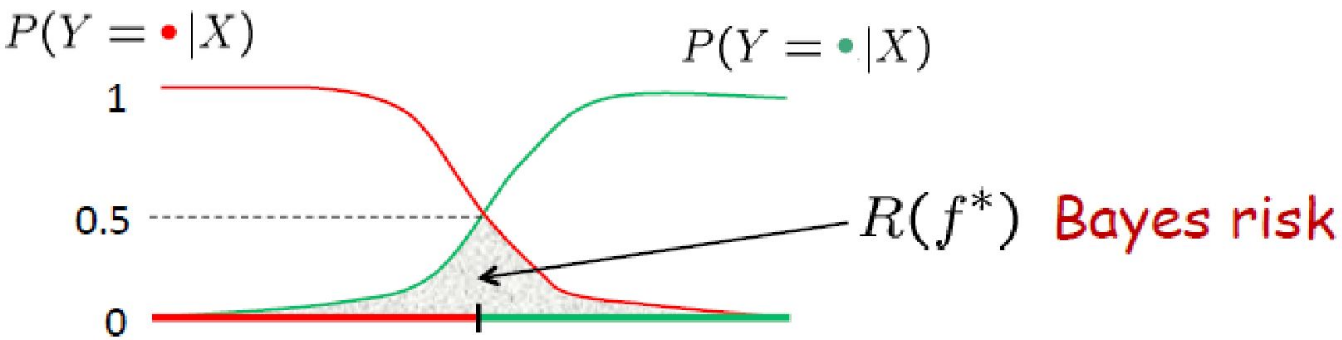
$$p(error|X) = \begin{cases} p(Y = C_1|X), if\ P(Y = C_2|X) > P(Y = C_1|X) \\ p(Y = C_2|X), if\ P(Y = C_1|X) > P(Y = C_2|X) \end{cases}$$
$$= \min\{p(Y = C_1|X), p(Y = C_2|X)\}$$

- An example with one dimensional X



Bayes Error

- The shaded area under the curve corresponds to the prediction errors incurred by the Bayes classifier



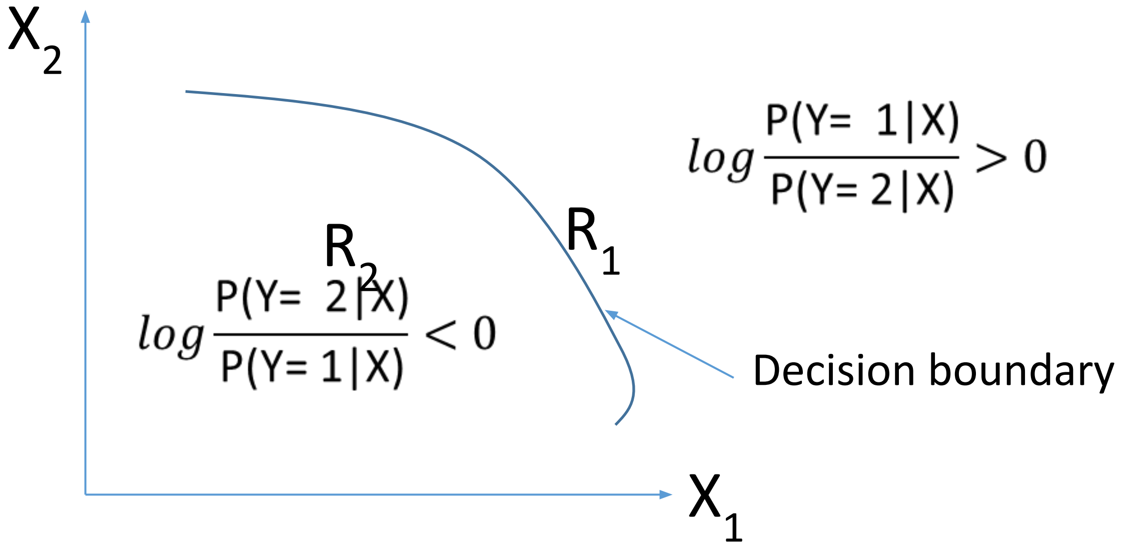
Optimality

- Bayes classifier is the optimal classifier we can obtained with the smallest prediction error.

- The prediction error of NN classifier is upper bounded by twice Bayes error asymptotically (i.e., when the size of training set is very large).

Decision region and boundary

- Log likelihood ratio divides the feature space into two regions by threshold 0.



The boundary

- is determined by the equation:

$$\log \frac{P(Y=1|X)}{P(Y=2|X)} = \log P(Y=1|X) - \log P(Y=2|X) = 0$$

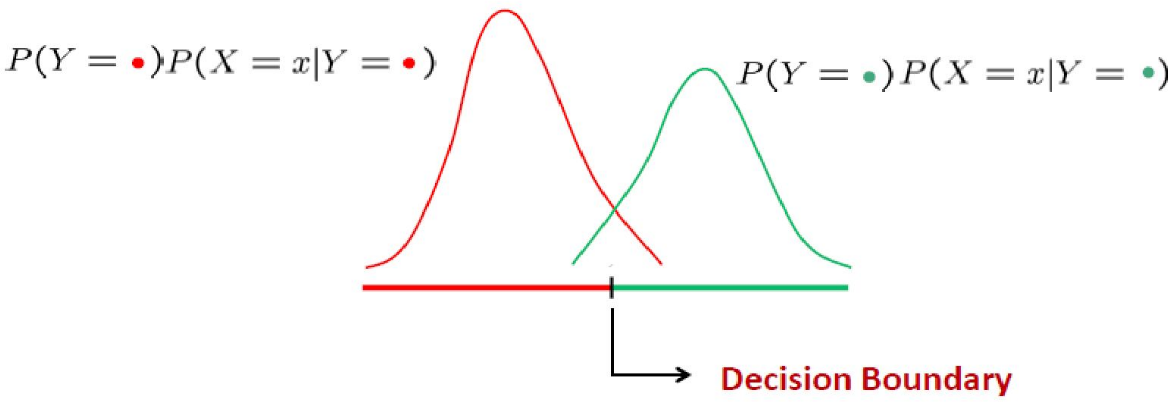
- Discriminant function $\mathrm{f(X)} = \log P(Y=1|X) - \log P(Y=2|X)$

Examples of binary decision boundary

- Gaussian class-conditional density (one dimensional X)

$$P(X = x|Y = y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x-\mu_y)^2}{2\sigma_y^2}\right)$$

with y = {1,2}.



High dimensional case

- Gaussian class-conditional density in high dimensional space

$$P(X = x|Y = y) = \frac{1}{\sqrt{2\pi|\Sigma_y|}} \exp\left(-\frac{(x-\mu_y)\Sigma_y^{-1}(x-\mu_y)'}{2}\right)$$

- Decision boundary equation:

$$f(X) = \log\frac{P(Y=1)P(X|Y=1)}{P(Y=2)P(X|Y=2)}$$

$$= -\frac{(X-\mu_1)\Sigma_1^{-1}(X-\mu_1)' - (X-\mu_2)\Sigma_2^{-1}(X-\mu_2)'}{2} + \frac{1}{2}\log\frac{|\Sigma_2|\mathrm{P}(\mathrm{Y}=1)}{|\Sigma_1|\mathrm{P}(\mathrm{Y}=2)} = 0$$

  - A quadratic surface in high dimensional space

Special Case

- If $\Sigma_1 = \Sigma_2 = \Sigma$, discriminant function boils down to

$$f(X) = -X\Sigma^{-1}(\mu_1 - \mu_2) + \mu_1\Sigma^{-1}\mu_1 - \mu_2\Sigma^{-1}\mu_2 + \log\frac{P(Y=1)}{P(Y=2)}$$

$$f(X) = -X\underbrace{\Sigma^{-1}(\mu_1 - \mu_2)}_{\mathrm{W}} + \underbrace{\mu_1\Sigma^{-1}\mu_1 - \mu_2\Sigma^{-1}\mu_2 + \log\frac{P(Y=1)}{P(Y=2)}}_{\mathrm{b}}$$

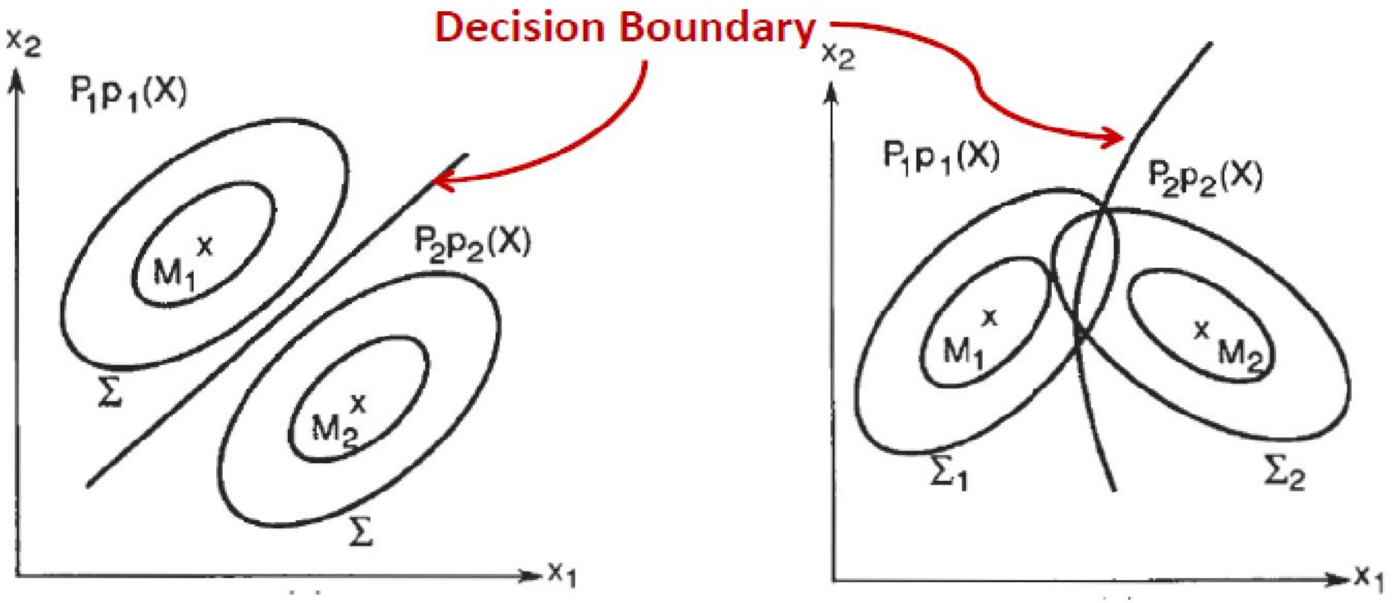
$$= XW + b$$

where W is a N dimensional vector, b is a real number.

- *f*(*X*) is linear! Decision boundary is a hyperplane in high-dimensional space.

Decision boundary

- We got the first linear classifier from Bayes classifier model.



Linear classifier

- W and b is indirectly obtained from two class-conditional distribution

$$f(X) = -X\Sigma^{-1}(\mu_1 - \mu_2) + \mu_1\Sigma^{-1}\mu_1 - \mu_2\Sigma^{-1}\mu_2 + \log\frac{P(Y=1)}{P(Y=2)}$$

$$f(X) = -X\underbrace{\Sigma^{-1}(\mu_1 - \mu_2)}_{\mathrm{W}} + \underbrace{\mu_1\Sigma^{-1}\mu_1 - \mu_2\Sigma^{-1}\mu_2 + \log\frac{P(Y=1)}{P(Y=2)}}_{\mathrm{b}}$$

$$= XW + b$$

- We waste much effort on estimating Gaussian covariance matrix and mean vector, but what we really need is simply *W* and b, can we learn W and b directly? Yes!
- Directly estimating W and b reduces the number of parameters we have to estimate.

Bayes classifier for continuous feature vectors

- Input feature vector $X=(X_1,\ldots,X_N)$ with N attributes
- Step 1: design class-conditional density for $Y \in \{0,1,\ldots,9\}$

$$P(X = x|Y = y) = \frac{1}{\sqrt{2\pi|\Sigma_y|}} \exp\left(-\frac{(x-\mu_y)\Sigma_y^{-1}(x-\mu_y)'}{2}\right)$$

- Naïve Gaussian Bayes classifier

$$P(X|Y) = P(X_1|Y) \ldots P(X_N|Y)$$

Where each individual term is

$$P(X_n|Y) = \frac{1}{\sqrt{2\pi}\sigma_{y,n}} = \exp\left\{\frac{(X_n - \mu_{y,n})^2}{2\sigma_{y,n}^2}\right\}$$

Bayes classifier for continuous feature vectors

- Maximum Likelihood estimation of $\mu_{y,n}$ and $\sigma_{y,n}$ for

$$P(X_n|Y) = \frac{1}{\sqrt{2\pi}\sigma_{y,n}} = \exp\left\{\frac{(X_n - \mu_{y,n})^2}{2\sigma_{y,n}^2}\right\}$$

are

$$\mu_{y,n} = \frac{1}{m_y}\sum_{i=1}^{m} X_{i,n}\delta[Y_i = y]$$

$$\sigma_{y,n} = \frac{1}{m_y - 1}\sum_{i=1}^{m} \delta[Y_i = y](X_{i,m} - \mu_{y,n})^2$$

with a set of (X_i, Y_i) for ith training example, and $X_{i,n}$ is the nth feature for X_i, m_y is the number of training examples of class y, and $\delta[\![Y_i = y]\!]$ is indicator function.

Bayes classifier for continuous feature vectors

- Step 2: Estimate the prior distribution

$$P(y = d) = \theta_d \, with \, \sum_{d=0}^{9} \theta_d = 1$$

  - Solution 1: Maximum likelihood estimation

$$\theta_d = \frac{\#\ of\ digit\ d\ in\ traing\ set}{\#\ of\ total\ digits\ in\ training\ set}$$

Bayes classifier for continuous feature vectors

- Solution 2: Maximum A Posterior parameter estimation
  - Imposing a prior $\mathrm{P}(\theta)$ on the parameter of prior distribution $\mathrm{P}(\mathrm{y}|\theta)$: Prior on prior
  - Instead of only estimating a single point for θ, we estimate a posterior distribution $\mathrm{P}(\theta|D_Y)$ over all possible θ, where D_Y is the class labels for training examples
  - Dirichlet distribution $\mathrm{P}(\theta) \propto \theta_1^{\alpha_1-1} \dots \theta_9^{\alpha_9-1}$, conjugate to $\mathrm{P}(\mathrm{y}|\theta)$
  - By the property of conjugate distribution, we have

$$\underset{\theta_d}{\mathrm{argmax}}\, \mathrm{P}(\theta_d|D_Y) = \frac{\#\ of\ digit\ d\ in\ trainng\ set + \alpha_d}{\#\ of\ \ trainng\ examples + \sum_{d=0}^{9} \alpha_d}$$

Dirichlet Distribution

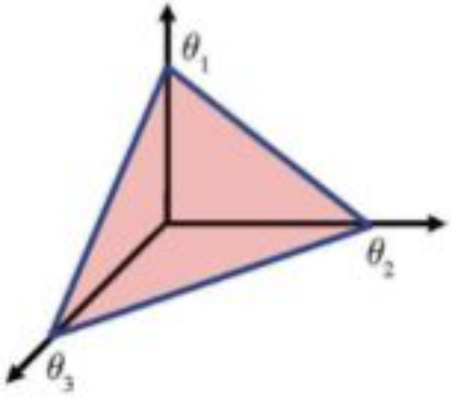
A multivariate generalization of the beta distribution

$$\mathrm{Dir}(\mathbf{x}|\boldsymbol{\alpha}) \triangleq \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} x_k^{\alpha_k - 1}$$

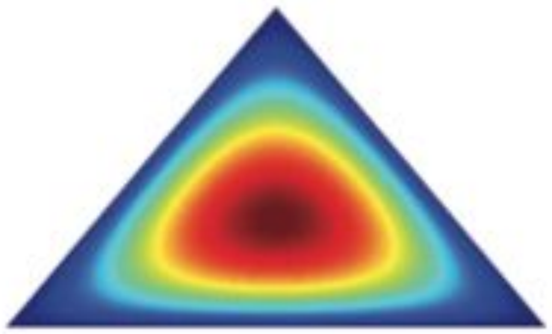
$$B(\boldsymbol{\alpha}) \triangleq \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\alpha_0)}$$

$$\alpha_0 \triangleq \sum_{k=1}^{K} \alpha_k$$

$$\mathbb{E}[x_k] = \frac{\alpha_k}{\alpha_0}, \ \mathrm{mode}[x_k] = \frac{\alpha_k - 1}{\alpha_0 - K}$$



$\boldsymbol{\alpha} = (2, 2, 2)$



Posterior
$$\begin{aligned} p(\boldsymbol{\theta}|\mathcal{D}) &\propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ &\propto \prod_{k=1}^{K} \theta_k^{N_k} \theta_k^{\alpha_k - 1} = \prod_{k=1}^{K} \theta_k^{\alpha_k + N_k - 1} \\ &= \mathrm{Dir}(\boldsymbol{\theta}|\alpha_1 + N_1, \ldots, \alpha_K + N_K) \end{aligned}$$

Bayes classifier for continuous feature vectors

- Test example X, its most possible digit is

$$Y^* = \mathrm{argmax}_{Y \propto \{0,1,\ldots,9\}} P(X_1|Y) \ldots P(X_N|Y)P(Y)$$
$$= \mathrm{argmax}_{Y \propto \{0,1,\ldots,9\}} \log P(X_1|Y) + \cdots + \log P(X_N|Y) + \log P(Y)$$

- A trick: working with log of probability
  - Convert multiplication to summation
  - Avoid arithmetic underflow: too large N will cause too small posterior that cannot be correctly operated by computer.

Why not use multivariate Gaussian?

- With full covariance matrix to model the class-conditional distribution?
- In MNIST, feature space dimension N=28X28, how many parameters are there in a full covariance matrix?
  - $\frac{N(N+1)}{2} = 307{,}720$, compared with 50000 training examples
  - Underdetermined: The parameters cannot be completely determined.

Text document: length-varying feature vectors

- Input
  - For each document m, a vector of length n represents all the words appearing in the document: X_n^m is the *n*-th word in document m.
    - The domain of X_n^m is a vocabulary of a dictionary (e.g., Webster dictionary)
  - The length of documents vary between each other, so feature vector $X^m = (X_1^m, X_2^m, \dots,)$ for a document does not have a fixed size.
- Output
  - Y_m defines the category of document *m* – {Ads, Non-Ads}

Naive Bayes: Spam Filtering

- Email spam filtering

	docID	X = words in Email	y = spam?
Training set	1	money click money	yes
	2	money money discount	yes
	3	money link	yes
	4	work lunch money	no
Test set	5	money money money work lunch	?

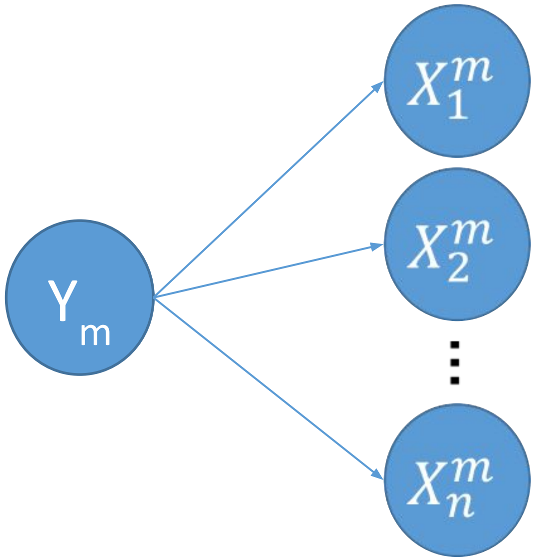
- Vocabulary
  - “money”, “click”, “discount”, “link”, “work”, “lunch”

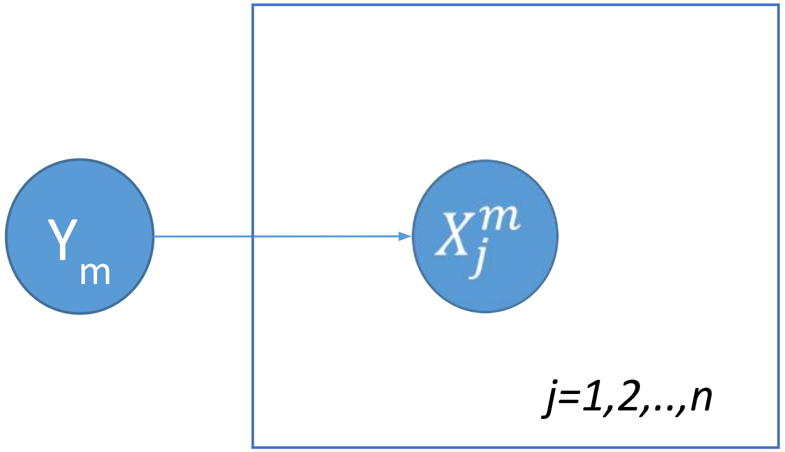
Class-conditional distribution

- Assume independence between words in a document

$$P(X_1^m, X_2^m, \dots, X_n^m | Y_m) = P(X_1^m | Y_m) P(X_2^m | Y_m) \dots P(X_n^m | Y_m)$$

- A graphical model representation for the independence decomposition of a distribution
  - Each circle node represent a random variable
  - Each arrow represents a conditional distribution, conditioned on the parent node





Class-conditional distribution

- Class-conditional distribution

$$P(X_1^m, X_2^m, \dots, X_n^m | Y_m) = P(X_1^m | Y_m) P(X_2^m | Y_m) \dots P(X_n^m | Y_m) = \prod_{i=1}^{V} P(w_i | Y_m)^{Count_i}$$

- MLE of $P(w_i | Y_m)$ for each word w_i in a vocabulary.

$$P(w_i | Y_m = y) = \frac{\# \ of \ word \ w_i \ in \ documents \ of \ class \ y}{\# \ of \ words \ in \ documents \ of \ class \ y}$$

Class-conditional distribution

- MAP estimate of $P(wi|Ym)$ for each word w_i in a vocabulary.

$$P(w_i|Y_m = y) = \frac{\#\ of\ word\ wi\ in\ documents\ of\ class\ y\ plus\ soft\ count\ \alpha_i}{\#\ of\ words\ plus\ all\ soft\ counts\ in\ documents\ of\ class\ y}$$

- Here a soft count α_i is added to each word w_i

Class prior

- Similar MLE/MAP methods can be applied to estimate P(Y)
- Given a test example X,

$$P(X_1 \ ,X_2 \ ,\dots,X_n \ |Y) = \prod_{i=1}^{V} P(w_i|Y)^{Count_i}$$

- Decide the optimal y

$$\mathrm{argmax}_y P(Y = y) \prod_{i=1}^{V} P(w_i|Y = y)^{Count_i}$$

Summary

- Recap Bayes classifier, Bayes error
- Decision region and boundary with Gaussian class-conditional distributions
  - Linear hyperplane and quadratic surface in high dimensional space
- Gaussian Naive Bayes
- Bayes classifier with length-varying feature vector for text document
- Basic concept of graphical model

References

Chap 4, Machine Learning: Probabilistic Perspective (Murphy)