ECS 171: Introduction to Machine Learning

Lecture 8

Naïve Bayes Classifier

(Murphy book) Sections 3.2 and 3.5

Physical Address for Prof. Ilias Tagkopoulos
Computer Science:
Office: 3063 Kemper Hall
Phone: (530) 752-4821
Fax: (530) 752-4767

Genome and Biomedical Sciences Facility:
Office: 5313 GBSF
Phone: (530) 752-7707
Fax: (530) 754-9658

Instructor: Ilias Tagkopoulos
iliast@ucdavis.edu

A Probabilistic classification approach

- The simplest probabilistic classifier: Naïve Bayes
  - a **generative** method.
- The three main concepts here are:
  - **PRIOR probability**
  - **POSTERIOR probability**
  - **LIKELIHOOD**
- The main assumption is that given a class, any feature is **independent** of any other feature.
  - The reason the classifier is called “naïve”.
- In practice, it works quite well, even when features are not independent especially when
  - Features are not highly correlated, or their function is a complex one
  - The number of data points (training data) is modest, so more complex classifiers risk over-fitting.

Building a Bayesian classifier: Prior Probability

- Suppose that you have a total of n classes, i.e y ∈ $\{c_1, \dots, c_n\}$.
- Without having any other information, what is the *a priori* probability that any given object (j) belong to any given class?

$$p(y^{(j)} = c_i) = \frac{1}{n}$$

- This probability that a random object will belong to a certain class is called prior probability.
- What happens when classes have different frequencies in the dataset (population)?

$$p(y^{(j)} = c_i) = \frac{f_i}{\sum_1^n f_i}$$

- For instance, what is the prior probability for each class in the following example, given its member size?

	A	B	C	D	E
Members	100	300	5	500	95
Prior Probability	?	?	?	?	?

Building a Bayesian classifier: Prior Probability

- Suppose that you have a total of n classes, i.e y ∈ $\{c_1, \ldots, c_n\}$.
- Without having any other information, what is the *a priori* probability that any given object (j) belong to any given class?

$$p(y^{(j)} = c_i) = \frac{1}{n}$$

- This probability that a random object will belong to a certain class is called prior probability.
- What happens when classes have different frequencies in the dataset (population)?

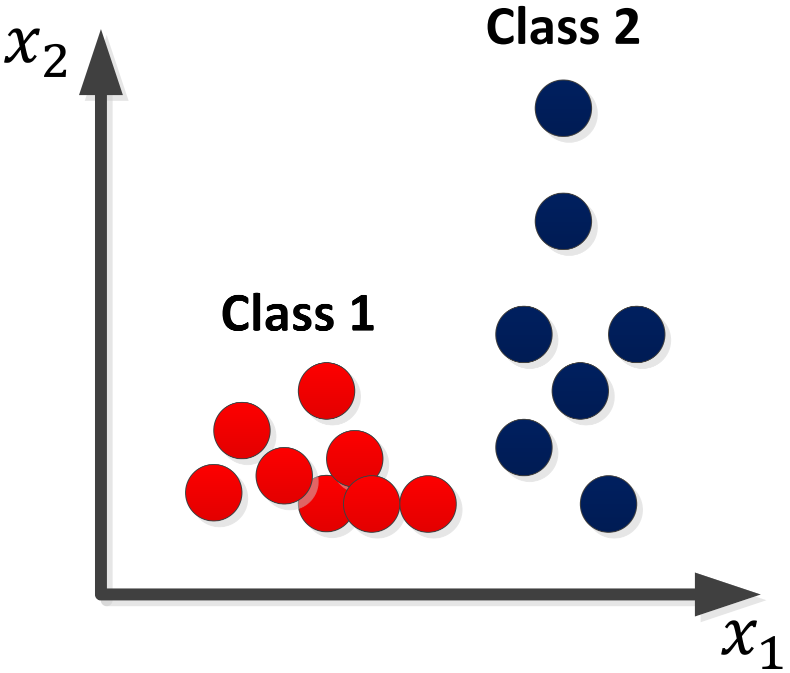
$$p\left(y^{(j)} = c_i\right) = \frac{f_i}{\sum_1^n f_i}$$

- For instance, what is the prior probability for each class in the following example, given its member size?

	A	B	C	D	E
Members	100	300	5	500	95
Prior Probability	0.1	0.3	0.005	0.5	0.095

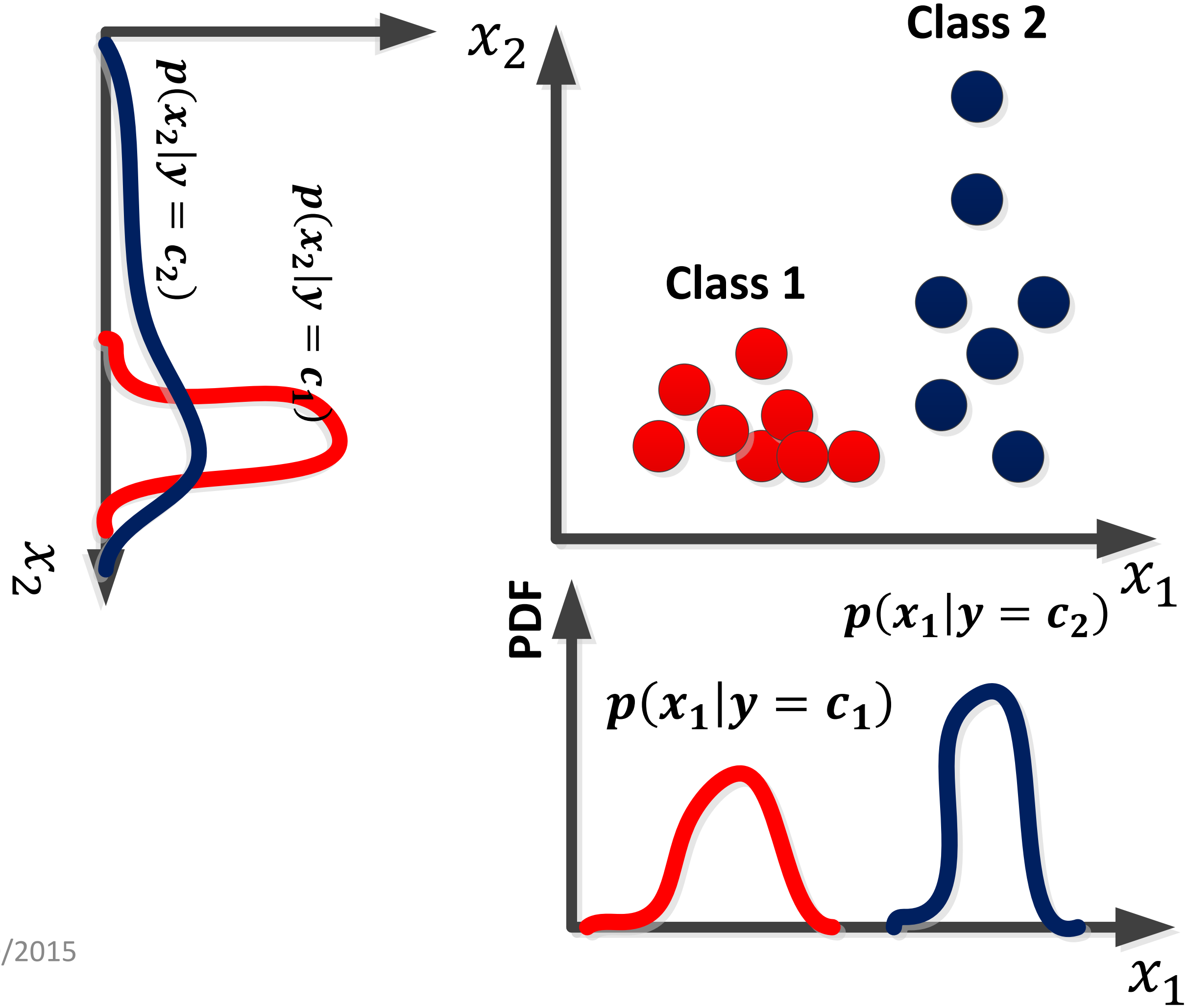
Building a Bayesian classifier: Likelihood

- The Likelihood $l(x|y) = p(x|y = c_i)$ can be thought as the class-conditional probability distribution of the feature (input) for a specific class c_i
- Example: assume two features and two classes, like in the picture. What will the $p(x|y = c_i)$ be?



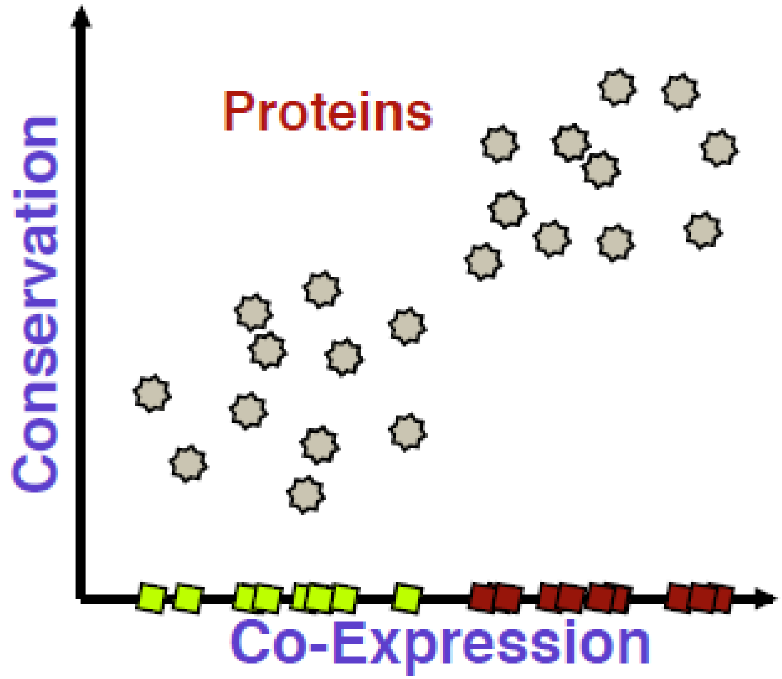
Building a Bayesian classifier: Likelihood

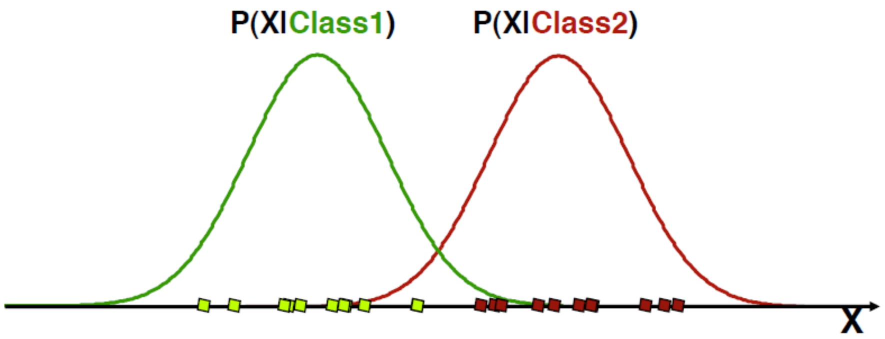
- The Likelihood $l(x|y) = p(x|y = c_i)$ can be thought as the class-conditional probability distribution of the feature (input) for a specific class c_i



Building a Bayesian classifier: Likelihood

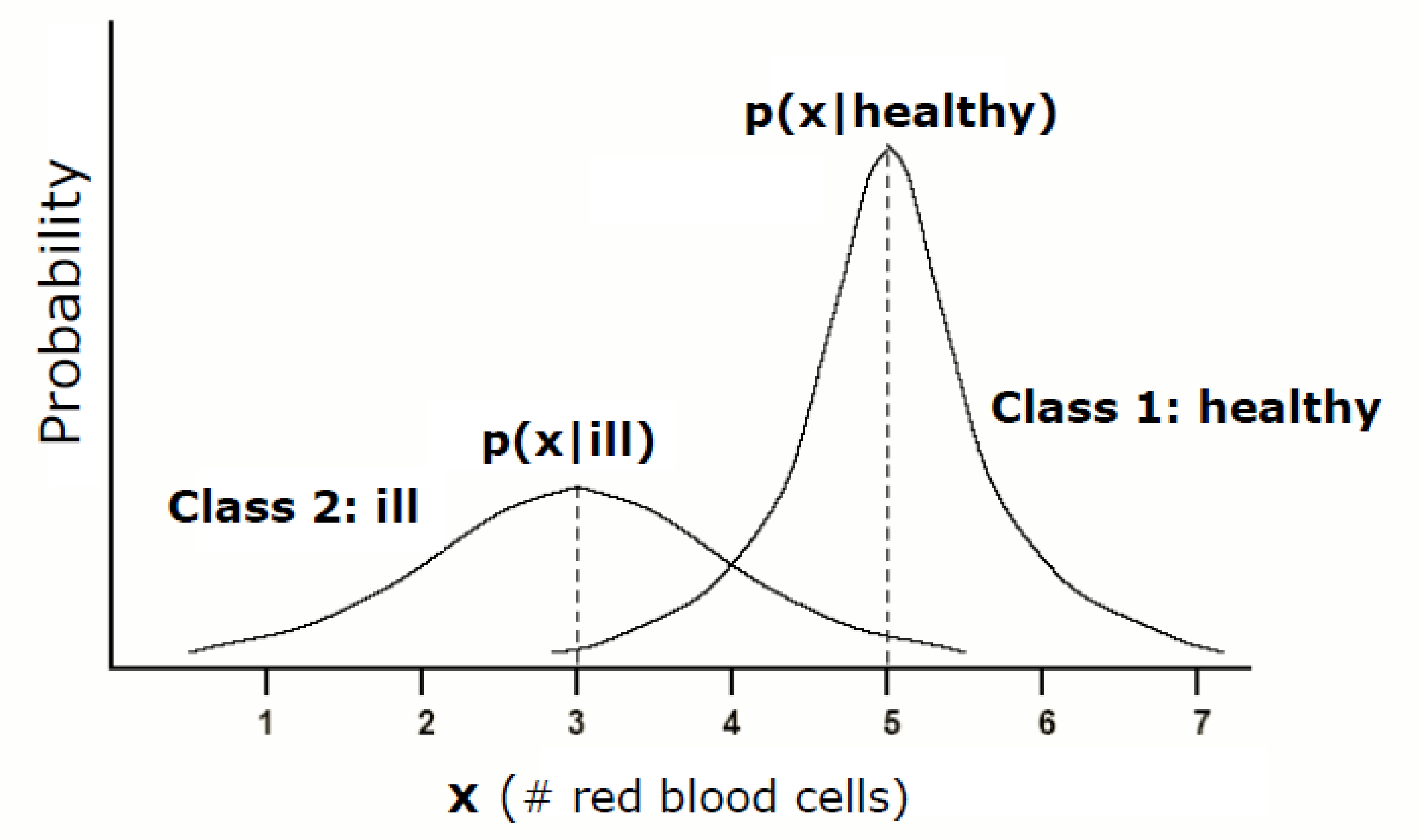
- Another example: proteins





Building a Bayesian classifier: Likelihood

- Another example: patients



Should we always model it as a Gaussian ? What if the features are binary?

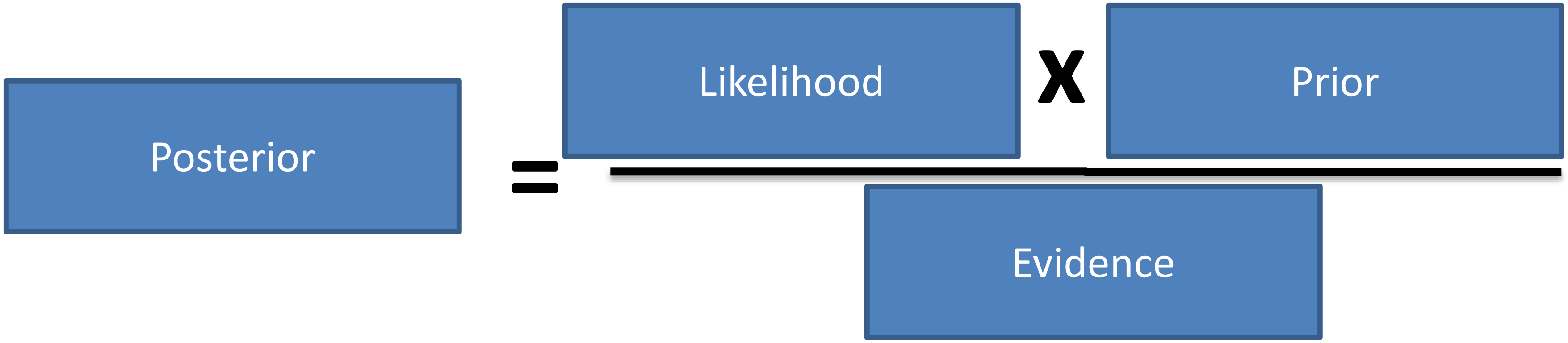
Building a Bayesian classifier

- So far, we have calculated:
  - **Prior Probability:** The "default", *a priori* probabilities for each class
  - **Likelihood:** The probability of a feature(s) having a specific value(s), given a certain class (also called class-conditional probabilities)
- What we want to calculate, however is the ***posterior probability:*** the probability of a sample being in a specific class given the feature values
- How can we link all these ?

Bayes Theorem

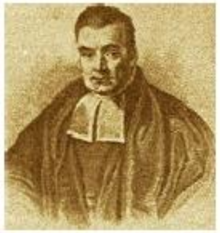
Building a Bayesian classifier: BAYES THEOREM

Bayes Theorem



Posterior = Likelihood x Prior / Evidence

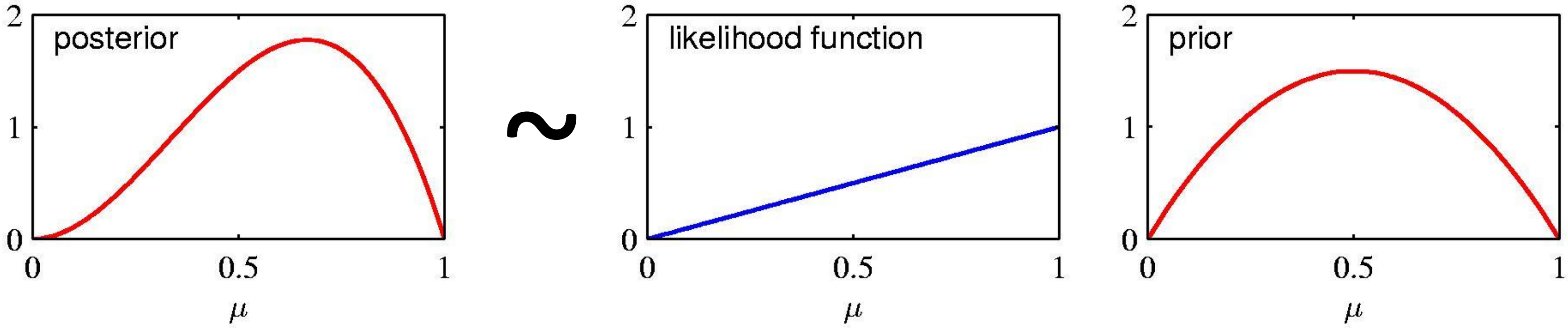
$$P(Class \mid Feature) = \frac{P(Feature \mid Class) \cdot P(Class)}{P(Feature)}$$



Building a classifier: Posterior – Likelihood – Prior relationship

From Bayes Theorem:
Posterior probability is proportional to
Likelihood times the Prior Probability

Posterior ~ Likelihood x Prior



Building a Bayesian classifier: How to assign classes

- We can now calculate the probability of a sample belonging to a class (posterior), for any given class, assuming that we know the prior, evidence and the likelihood

- One way to build a classifier: calculate all posterior probabilities for the sample and assign it to the class that is has the highest probability.

- Another way to build a classifier: Create a discriminant function. Assign to the first class when:

$$P(Feature\ X\ |Class1)\cdot P(Class1) > P(Feature\ X\ |Class2)\cdot P(Class2)$$

- Or equivalently if G(X)>0, where G(X) is given by

$$G(X) = log\ \frac{P(Feature\ X\ |Class1)\cdot P(Class1)}{P(Feature\ X|Class2)\cdot P(Class2)}$$

Building a classifier: Multiple features

- Which one is better? Having a discriminant function doesn’t need the calculation of the evidence
- Until now we focused on **a single feature X**. What happens when we have **multiple features**?
- In that case, our discriminant function becomes:

$$G(X) = \ log \frac{P(Feature\ X_1, Feature\ X_2, \ldots Feature\ X_n\ |Class1) \cdot P(Class1)}{P(Feature\ X_1, Feature\ X_2, \ldots Feature\ X_n\ |Class2) \cdot P(Class2)}$$

- We can build a table with the probability for each feature.
  - Good luck with that! Assume that each feature can take k values, what is the complexity of your problem?

Building a classifier: Naïve Bayes Classifier

- Naïve Bayes Classifier
  - Assumes that all features are **INDEPENDEND**

- With the assumption of independence, our likelihood is given by:

$$P(Feature\ X_1, Feature\ X_2, \dots Feature\ X_n\ | Class1) = P(Feature\ X_1 | Class1)\ P(Feature\ X_2 | Class1) \dots\ P(Feature\ X_n | Class1)$$

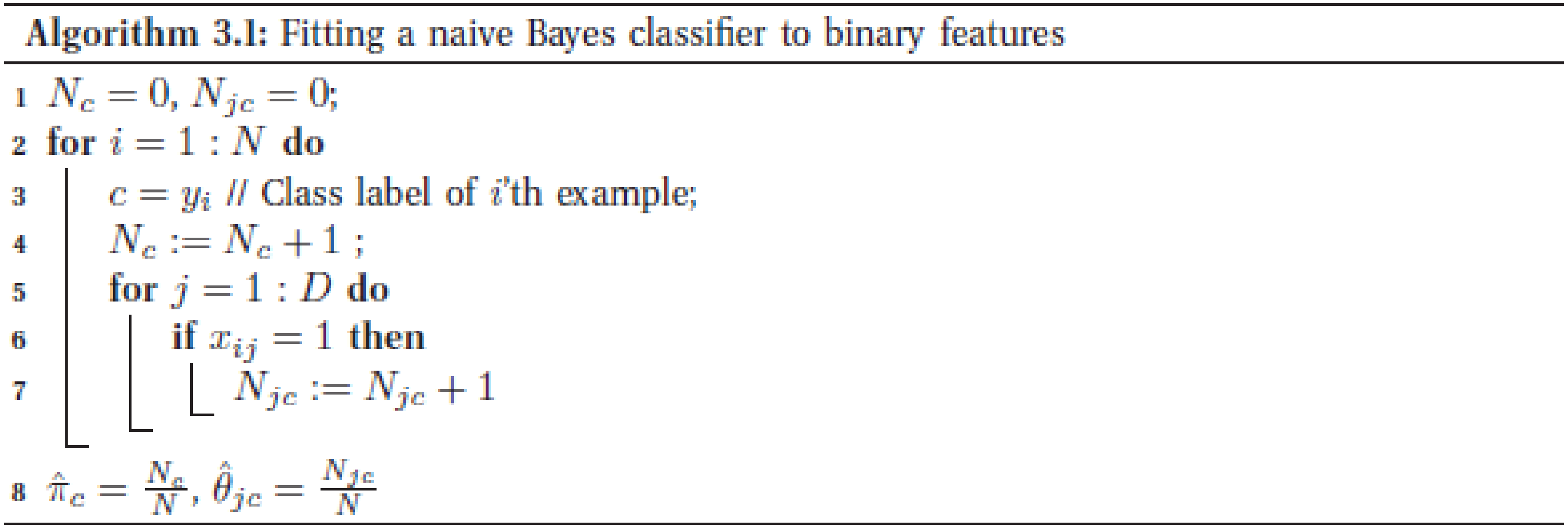
$$= \prod P(Feature\ X_i | Class1)$$

- And hence our discriminant function becomes:

$$G(X_1, \dots, X_n) = \ log\ \frac{\prod P(Feature\ X_i | Class1) \cdot P(Class1)}{\prod P(Feature\ X_i | Class2) \cdot P(Class2)}$$

Bayesian classification: an example

- How do we find the parameters for Naïve Bayes?
  - From your book:



Algorithm 3.1: Fitting a naive Bayes classifier to binary features

```
$N_c = 0, N_{jc} = 0;$
for $i = 1 : N$ do
    $c = y_i$ // Class label of $i$'th example;
    $N_c := N_c + 1$ ;
    for $j = 1 : D$ do
        if $x_{ij} = 1$ then
            $N_{jc} := N_{jc} + 1$
$\hat{\pi}_c = \frac{N_c}{N}, \hat{\theta}_{jc} = \frac{N_{jc}}{N}$
```

Bayesian classification: an example

- Some proteins are more expressed in cancerous vs. healthy tissues and thus they can be used as **features** to classify patients

- Assume two **classes/categories/labels:**
  - Class 1: Cancer
  - Class 2: Healthy

- Assume that we have measured the concentration of protein [X] in various people with and without cancer. We use 0 if the protein is expressed in low concentrations, 1 otherwise.

An example of Naïve Bayes classification

Class conditional probabilities for X

X (Protein)	Class A (Cancer)	Class B (Healthy)
0	0.9	0.2
1	0.1	0.8

- Additionally it is known that only 1% of the population is found to have cancer when a screening is performed.
- A new patient with high levels of protein X wants to know how probable it is for him to have cancer.

An example of Naïve Bayes classification

X (Protein)	Class A (Cancer)	Class B (Healthy)
0	0.9	0.2
1	0.1	0.8

- PRIOR Probability
  - P(Cancer) = 0.01
- LIKELIHOOD
  - P(X=1|Cancer) = 0.1
- EVIDENCE
  - P(X=1) = 0.1*0.01 + 0.8*0.99 = 0.793
- POSTERIOR Probability
  - **P(Cancer|X=1)** = 0.1*0.01/0.793 = **0.00126**
  - P(Healthy|X=1) = 0.8*0.99/0.793 = 0.99873

Example: Building the classifier

- Now we would like to build a classifier that can tell whether a patient is likely to have cancer or not based on a protein X concentration.
- Build a discriminative function:

$$G(X) = log \frac{P(Feature\ X\ |Class1) \cdot P(Class1)}{P(Feature\ X|Class2) \cdot P(Class2)}$$

X (Protein)	Class A (Cancer)	Class B (Healthy)
0	0.9	0.2
1	0.1	0.8

- When G(X) > 0 then patient is more likely to have cancer (Class 1) and vice versa.
- Example: G(X=1) = log(0.1*0.01/0.8*0.99)= -2.89
- Example: G(X=0) = log(0.9*0.01/0.2*0.99)= 2.64

Example: Adding more features

- Now assume that we measure protein Y too, which has the following pattern:

Y	Class A (Cancer)	Class B (Healthy)
0	0.9	0.5
1	0.1	0.5

- How can we add it as a feature?
- Assume a new patient with [X] high and [Y] low, what is his probability of cancer?

Feature independence

- Discriminative function for multiple features:

$$G(X) = \ log \frac{P(Feature\ X_1, Feature\ X_2, \dots Feature\ X_n\ |Class1) \cdot P(Class1)}{P(Feature\ X_1, Feature\ X_2, \dots Feature\ X_n\ |Class2) \cdot P(Class2)}$$

- But if we assume independence (naïve bayes)

$$G(X_1, \dots, X_n) = \ log \frac{\prod P(Feature\ X_i|Class1) \cdot P(Class1)}{\prod P(Feature\ X_i|Class2) \cdot P(Class2)}$$

- So here:

$$G(X, Y) = \ log \frac{P(X = 1|Cancer) \cdot P(Y = 0|Cancer) \cdot P(Cancer)}{P(X = 1|Healthy) \cdot P(Y = 0|Healthy) \cdot P(Healthy)}$$

Naïve Bayes example

X (Protein)	Class A (Cancer)	Class B (Healthy)
0	0.9	0.2
1	0.1	0.8

Y	Class A (Cancer)	Class B (Healthy)
0	0.9	0.5
1	0.1	0.5

$$\log[(0.1*0.9*0.01)/(0.8*0.5*0.99)] =$$
$$= -2.64$$

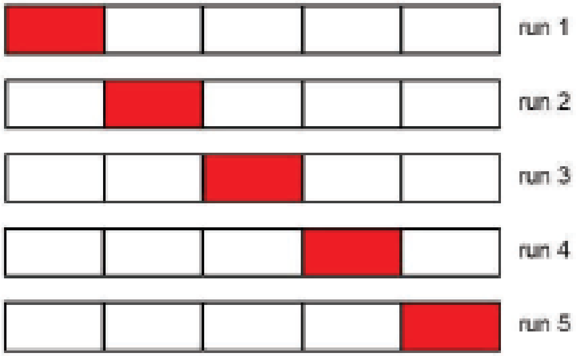
again he will be categorized as healthy, but we are now less certain (-2.64 vs -2.89) as low [Y] associates with more with cancer cases than healthy cases (0.9 vs. 0.5)

Measuring classification performance

Sections 5.7.2

Measuring classification performance

- Cross-validation



- Binary classification errors

	Really True	Really False
Predicted True	True Positive (TP)	False Positive (FP)
Predicted False	False Negative (FN)	True Negative (TN)

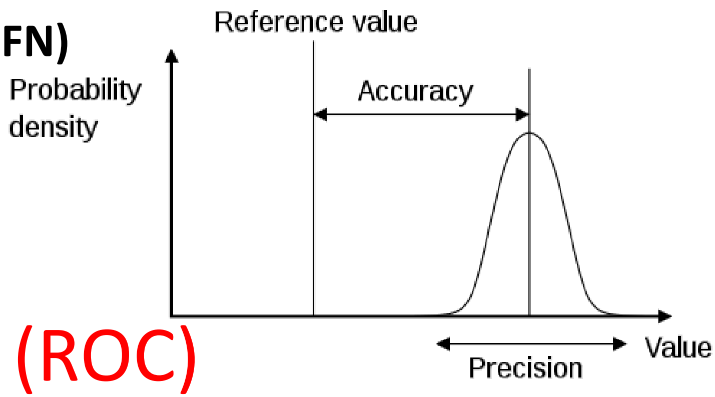
- Statistical measures :

True Positive Rate (TPR) = Recall = Sensitivity = TP/(TP + FN)

1- False Positive Rate (FPR) = Specificity = TN/(TN+FP)

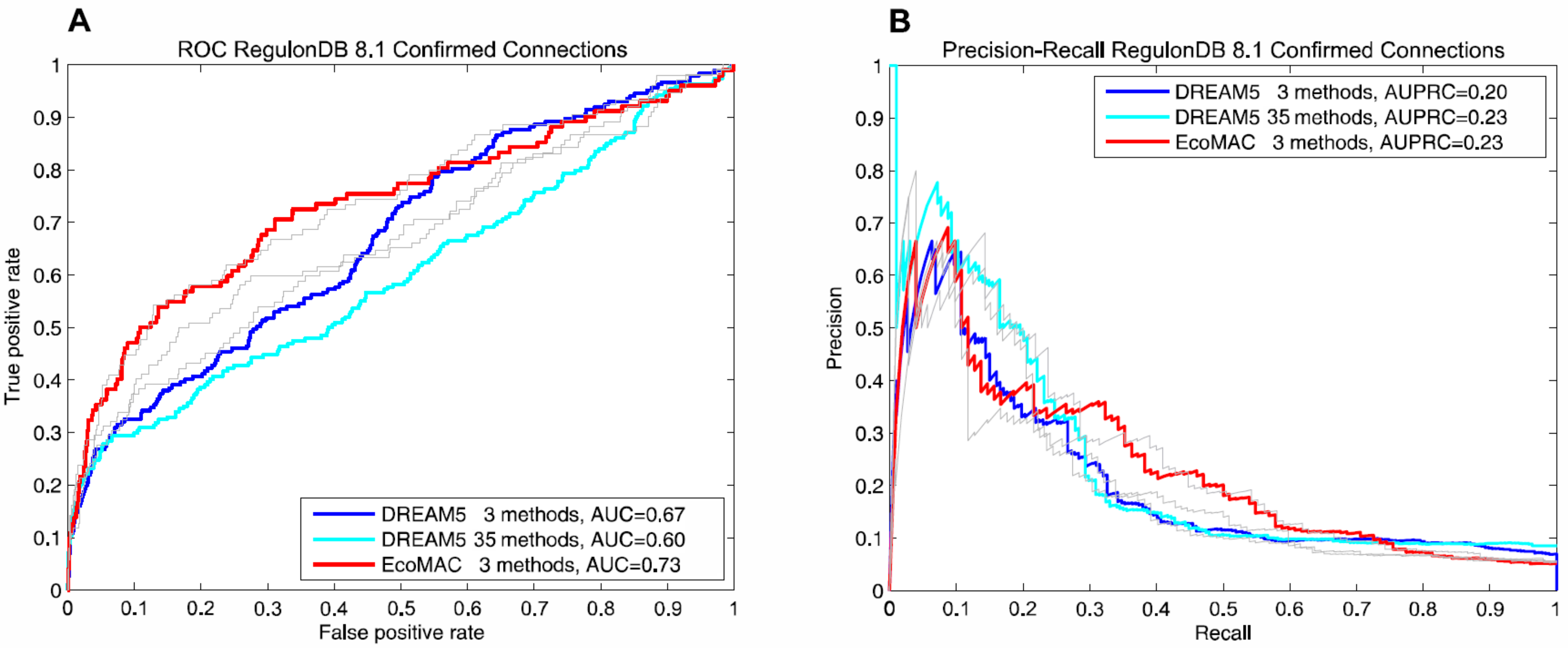
Accuracy = (TP + TN)/ (P + N)

Precision = 1 – False Discovery Rate = TP / (TP + FP)



- Receiver Operating Characteristics (ROC)

Receiver Operating Characteristics (**ROC curves**) and Precision–Recall Curves



Carrera, R. Estrela, J. Luo, N. Rai, A. Tsoukalas, I. Tagkopoulos, "An integrative, multi-layer, genome-scale model reveals the phenotypic landscape of Escherichia coli", *Molecular Systems Biology*, 10(7):735, 2014

Receiver operating characteristic (ROC)

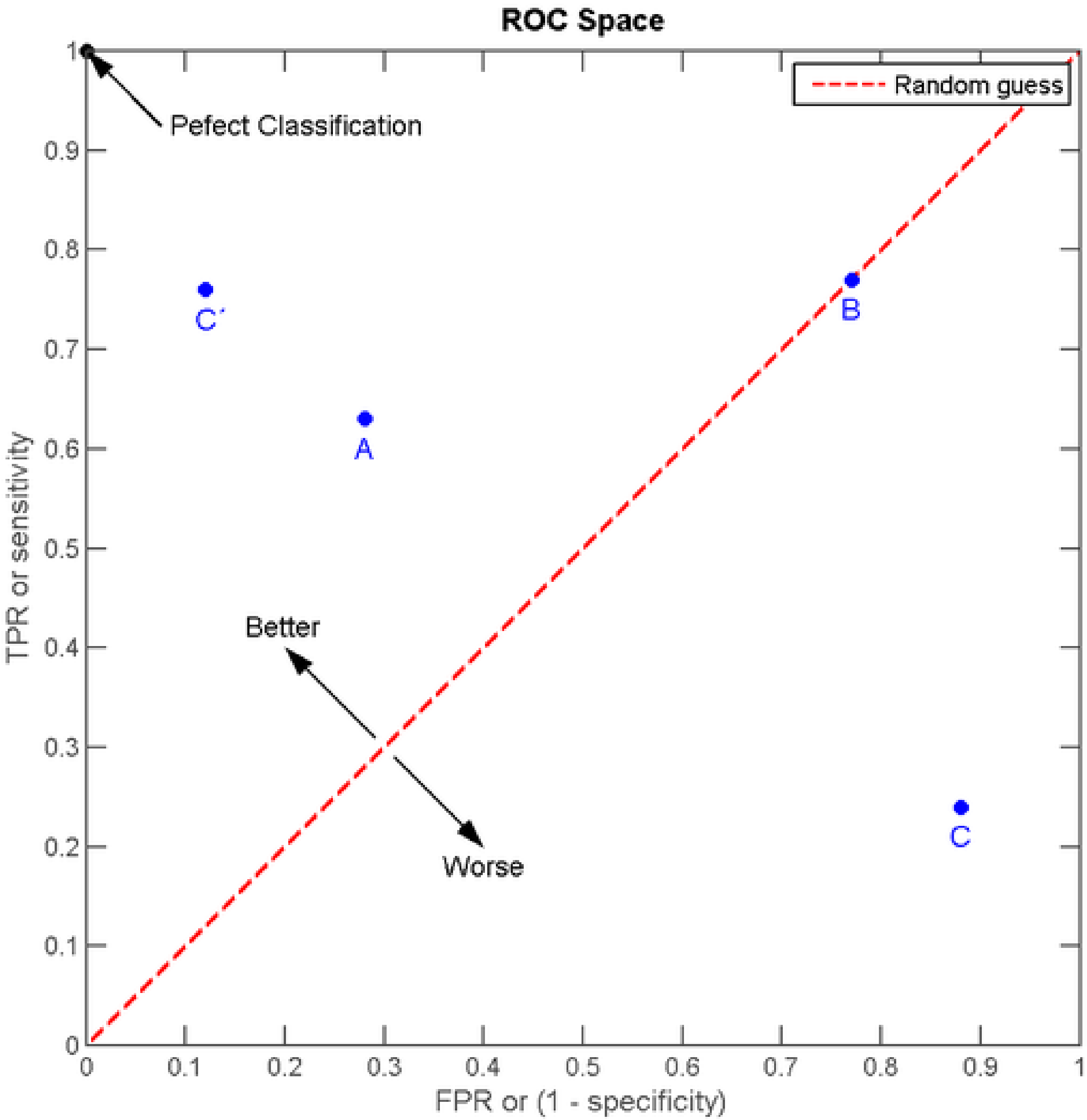
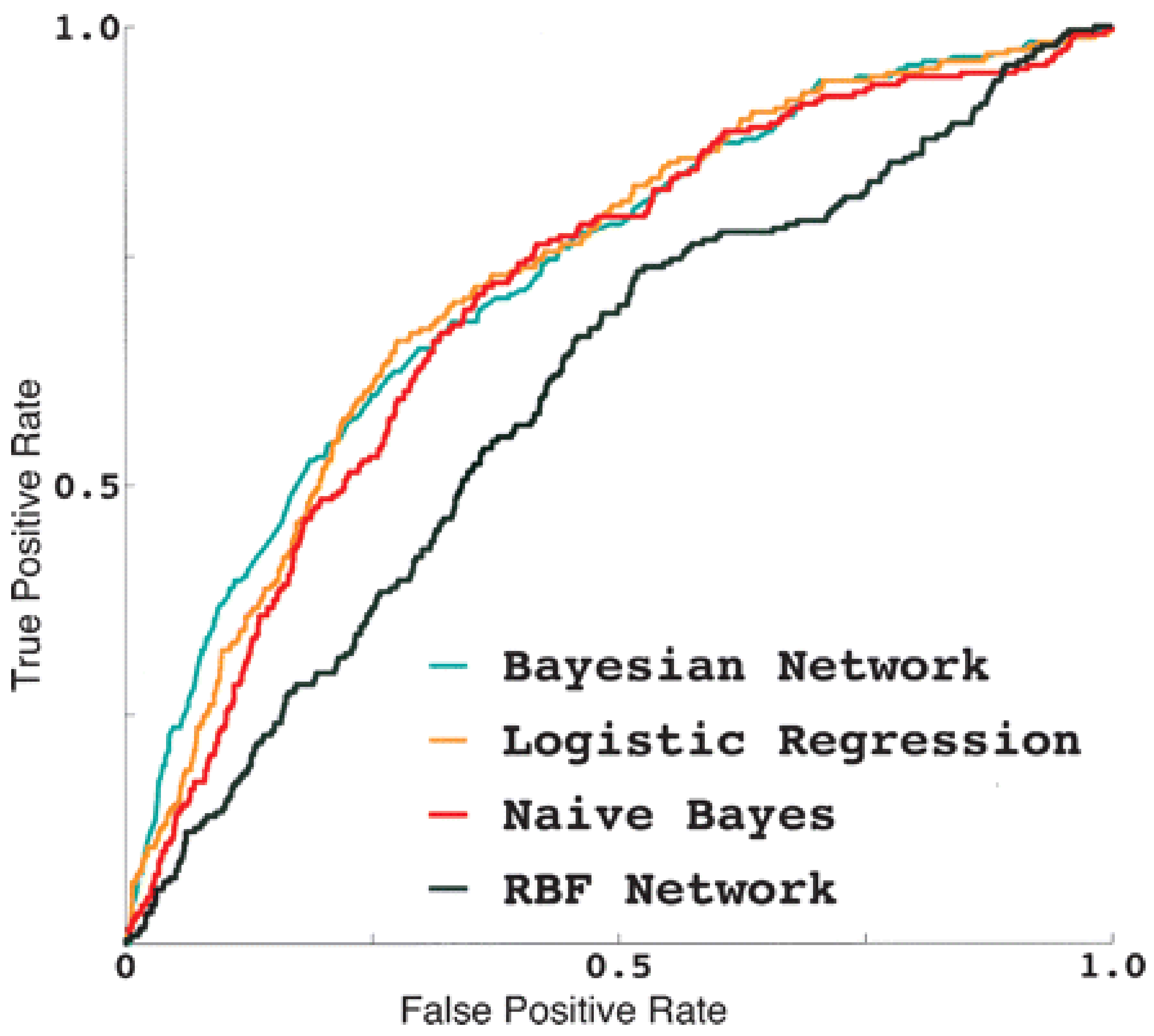


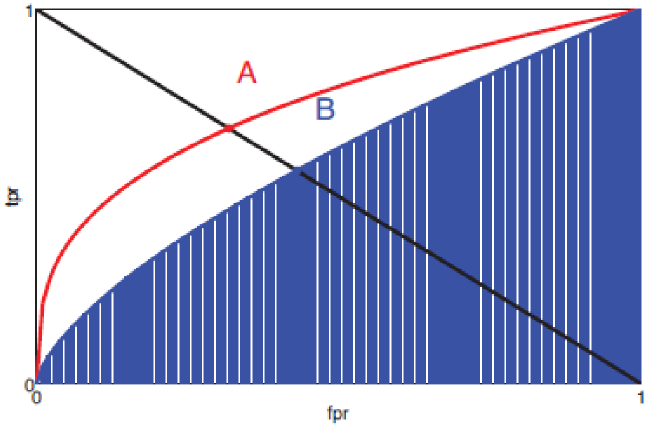
Image: Wikipedia

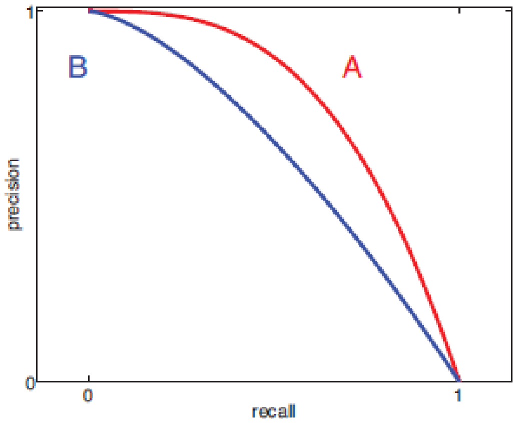
Receiver operating characteristic (ROC)



Quantitative systems-level determinants of human genes targeted by successful drugs
L. Yao and A. Rzhetsky
Genome Res. 18:206 (2008)

Receiver operating characteristic (ROC)





Which classifier is better? A or B?

Can we represent both precision and recall with one measure?

- **Area under the curve (AUC)**
- **F-score**

Putting everything together

- Now we have everything we need to design, build and evaluate any classifier. Assume that you are presented with a dataset with ***m* sample data, each with *n* features and one label.**
  - **Step 1: Select your model.** Neural Networks, Support Vector Machines, Logistic Regression, Bayesian classifiers. Which one will it be?
  - **Step 2: Select your features.** If n is small and m large, then your classifier can disregard irrelevant features without the risk of over-fitting. In most cases, however, you need to have a feature selection or extraction method to select the most informative features. Additionally, you might need to pre-process features (scaling, missing values, etc.).
  - **Step 3: Evaluate the classifier.** The classifier should always be trained in a training set and evaluated in a mutual exclusive testing set. You can either split the original dataset in any reasonable fraction (2/3-1/3, 70-30%, 80-20%) to form these two sets, or you can use K-fold cross-validation (preferred). Report generalization error, ROC/PR curves, AUC/AUPRC.
  - **Step 4: Train your classifier.** The final classifier should be trained with the complete dataset that was provided.

End of Lecture 9