10-701
Machine Learning

Naïve Bayes classifiers

Optional additional reading: Mitchell 6.1-6.10

Types of classifiers

- We can divide the large variety of classification approaches into three major types

1. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

2. Generative:
   - build a generative statistical model
   - e.g., Bayesian networks

3. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., decision tree

Bayes decision rule

- If we know the conditional probability P(X | y) we can determine the appropriate class by using Bayes rule:

$$P(y=i \mid X) = \frac{P(X \mid y=i)P(y=i)}{P(X)} \stackrel{def}{=} q_i(X)$$

But how do we determine p(X|y)?

Computing p(X|y)

Recall…

y – the class label

X – input attributes (features)

- Consider a dataset with 16 attributes (lets assume they are all binary). How many parameters to we need to estimate to fully determine p(X|y)?

age	employmer	education	edun	marital	…	job	relation	gender	hour	country	wealth
					…						
39	State_gov	Bachelors	13	Never_mar	…	Adm_cleric	Not_in_fam	Male	40	United_Sta	poor
51	Self_emp_i	Bachelors	13	Married	…	Exec_mana	Husband	Male	13	United_Sta	poor
39	Private	HS_grad	9	Divorced	…	Handlers_c	Not_in_fam	Male	40	United_Sta	poor
54	Private	11th	7	Married	…	Handlers_c	Husband	Male	40	United_Sta	poor
28	Private	Bachelors	13	Married	…	Prof_speci	Wife	Female	40	Cuba	poor
38	Private	Masters	14	Married	…	Exec_mana	Wife	Female	40	United_Sta	poor
50	Private	9th	5	Married_sp	…	Other_serv	Not_in_fam	Female	16	Jamaica	poor
52	Self_emp_i	HS_grad	9	Married	…	Exec_mana	Husband	Male	45	United_Sta	rich
31	Private	Masters	14	Never_mar	…	Prof_speci	Not_in_fam	Female	50	United_Sta	rich
42	Private	Bachelors	13	Married	…	Exec_mana	Husband	Male	40	United_Sta	rich
37	Private	Some_colle	10	Married	…	Exec_mana	Husband	Male	80	United_Sta	rich
30	State_gov	Bachelors	13	Married	…	Prof_speci	Husband	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	…	Adm_cleric	Own_child	Female	30	United_Sta	poor
33	Private	Assoc_acd	12	Never_mar	…	Sales	Not_in_fam	Male	50	United_Sta	poor
41	Private	Assoc_voc	11	Married	…	Craft_repai	Husband	Male	40	*MissingVa	rich
34	Private	7th_8th	4	Married	…	Transport_	Husband	Male	45	Mexico	poor
26	Self_emp_i	HS_grad	9	Never_mar	…	Farming_fis	Own_child	Male	35	United_Sta	poor
33	Private	HS_grad	9	Never_mar	…	Machine_o	Unmarried	Male	40	United_Sta	poor
38	Private	11th	7	Married	…	Sales	Husband	Male	50	United_Sta	poor
44	Self_emp_i	Masters	14	Divorced	…	Exec_mana	Unmarried	Female	45	United_Sta	rich
41	Private	Doctorate	16	Married	…	Prof_speci	Husband	Male	60	United_Sta	rich

Learning the values for the full conditional probability table would require enormous amounts of data

Naïve Bayes Classifier

- Naïve Bayes classifiers assume that given the class label (Y) the attributes are **conditionally independent** of each other:

$$X = \begin{bmatrix} x^1 \\ \vdots \\ x^n \end{bmatrix}$$

$$p(X \mid y) = \prod_j p_j(x^j \mid y)$$

Product of probability terms

Specific model for attribute *j*

- Using this idea the full classification rule becomes:

$$\begin{aligned} \hat{y} &= \arg\max_v \, p(y = v \mid X) \\ &= \arg\max_v \frac{p(X \mid y = v)\, p(y = v)}{p(X)} \\ &= \arg\max_v \prod_j p_j(x^j \mid y = v) p(y = v) \end{aligned}$$

v are the classes we have

Conditional likelihood: Full version

$$L(X_i \mid y_i = 1, \Theta) = \prod_j p(x_i^j \mid y_i = 1, \theta_1^j)$$

Vector of binary attributes for sample *i*

The set of all parameters in the NB model

The specific parameters for attribute *j* in class 1

Note the following:

1. We assume conditional independence between attributes **given** the class label
2. We learn a **different** set of parameters for the two classes (class 1 and class 2).

Learning parameters

$$L(X_i \mid y_i = 1, \Theta) = \prod_j p(x_i^j \mid y_i = 1, \theta_1^j)$$

- Let $X_1 \ldots X_{k1}$ be the set of input samples with label 'y=1'
- Assume all attributes are **binary**
- To determine the MLE parameters for $p(x^j = 1 \mid y = 1)$ we simply count how many times the j'th entry of those samples in class 1 is 0 (termed n0) and how many times its 1 (n1). Then we set:

$$p(x^j = 1 \mid y = 1) = \frac{n1}{n0 + n1}$$

Final classification

- Once we computed all parameters for attributes in both classes we can easily decide on the label of a **new** sample X.

Can be easily extended to multi-class classification

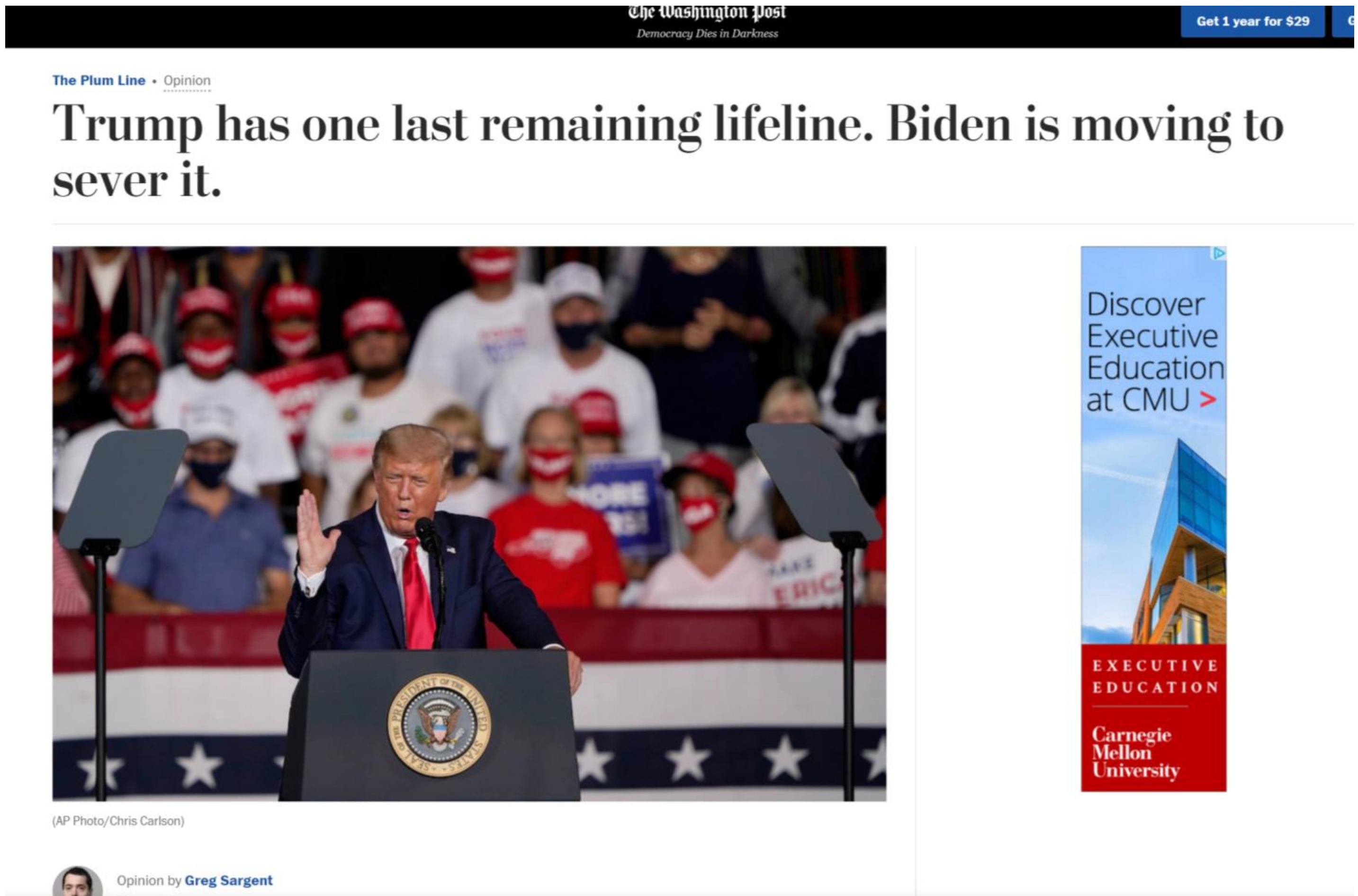
$$
\begin{aligned}
\hat{y} &= \arg\max_v \; p(y = v \mid X) \\
&= \arg\max_v \frac{p(X \mid y = v)\, p(y = v)}{p(X)} \\
&= \arg\max_v \prod_j p_j(x^j \mid y = v)\, p(y = v)
\end{aligned}
$$

Perform this computation for both class 1 and class 2 and select the class that leads to a higher probability as your decision

Prior on the prevalence of samples from each class

Example: Text classification

- What is the major topic of this article?



The Washington Post
Democracy Dies in Darkness
Get 1 year for $29

The Plum Line • Opinion

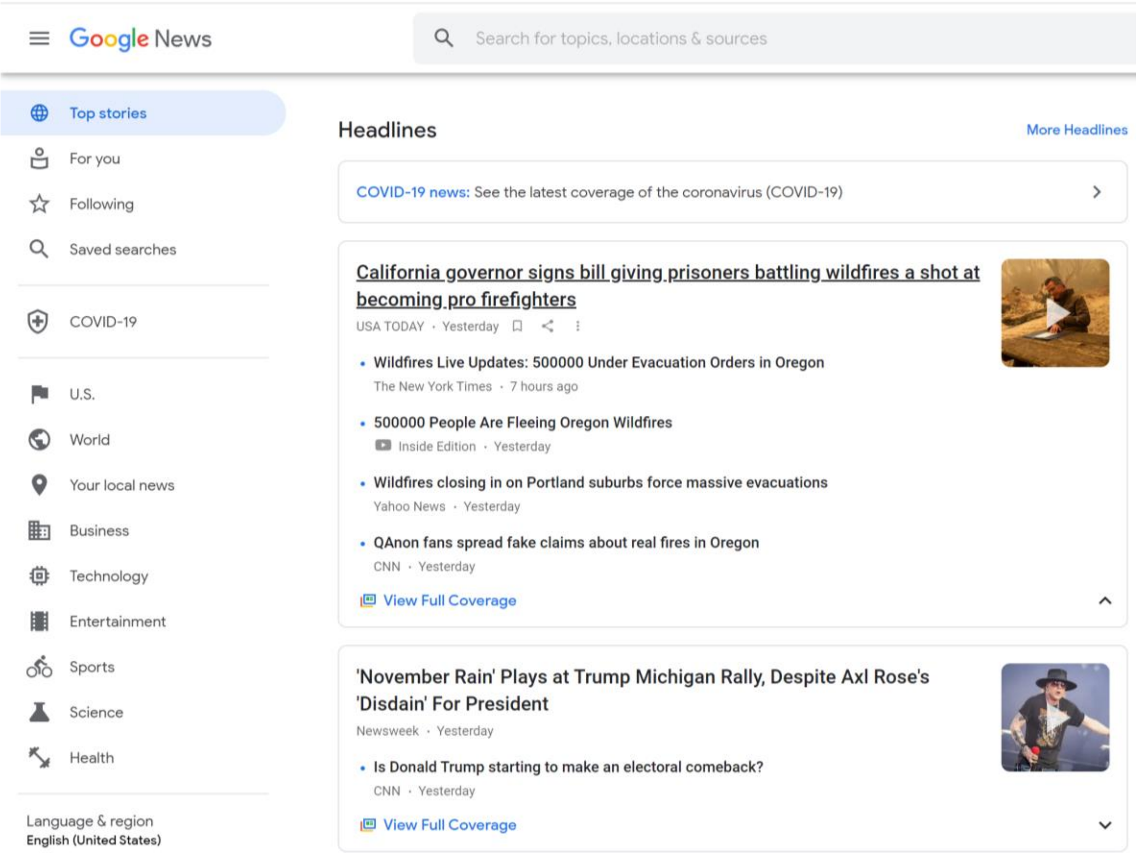
Trump has one last remaining lifeline. Biden is moving to sever it.

(AP Photo/Chris Carlson)

Opinion by Greg Sargent
Columnist

Example: Text classification

- Text classification is all around us



Feature transformation

- How do we encode the set of features (words) in the document?
- What type of information do we wish to represent? What can we ignore?
- Most common encoding: ‘**Bag of Words**’
- Treat document as a collection of words and encode each document as a vector based on some dictionary
- The vector can either be binary (present / absent information for each word) or discrete (number of appearances)

- Google is a good example
- Other applications include job search adds, spam filtering and many more.

Feature transformation: Bag of Words

- In this example we will use a binary vector
- For document X_i we will use a vector of *m** indicator features $\{\phi^j(X_i)\}$ for whether a word appears in the document
  - $\phi^j(X_i) = 1$, if word *j* appears in document X_i;
    $\phi^j(X_i) = 0$ if it does not appear in the document
- $\Phi(X_i) = [\phi^1(X_i) \ldots \phi^m(X_i)]^T$ is the resulting feature vector for the entire dictionary for document X_i
- For notational simplicity we will replace each document X_i with a fixed length vector $\Phi_i = [\phi^1 \ldots \phi^m]^T$, where $\phi^j = \phi^j(X_i)$.

*The size of the vector for English is usually ~10000 words

Example

Assume we would like to classify documents as election related or not.

Dictionary

- Washington
- Congress

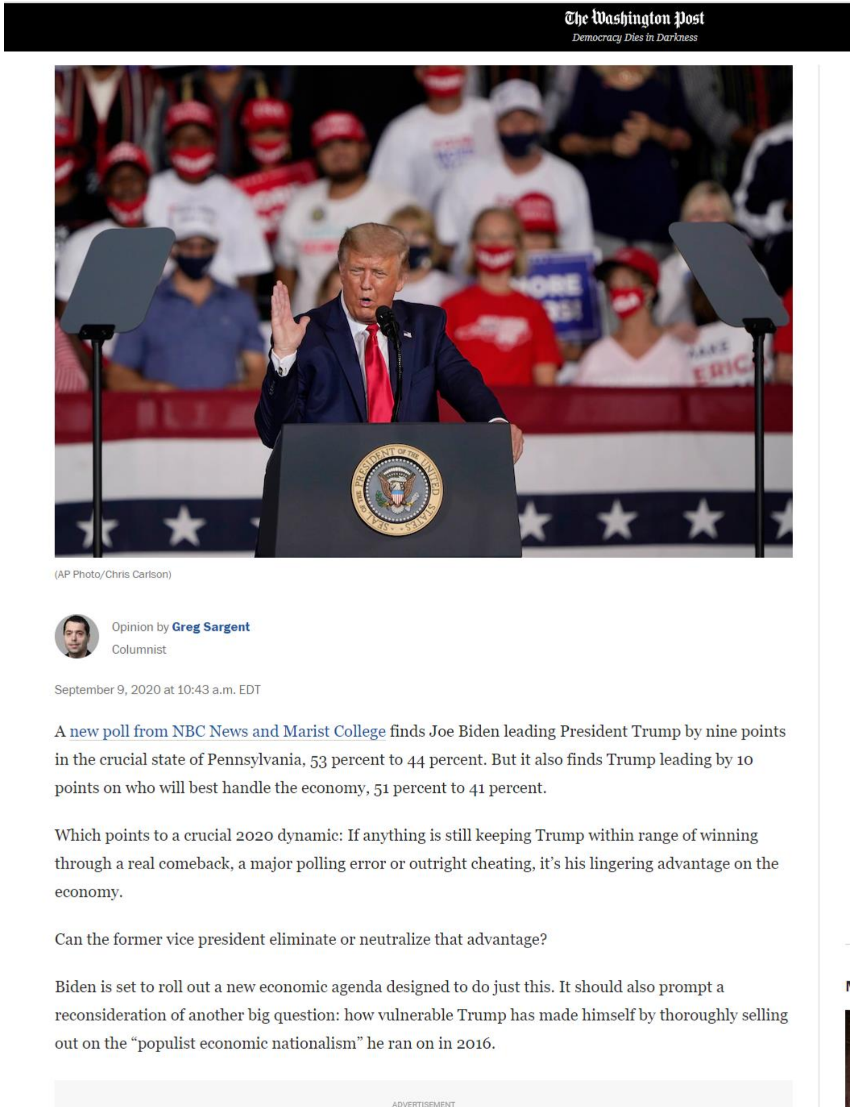
…

54. Trump

55. Biden

56. Russia

$$\phi^{54}=\phi^{54}(X_i) = 1$$
$$\phi^{55}=\phi^{55}(X_i) = 1$$
$$\phi^{56}=\phi^{56}(X_i) = 0$$


The Washington Post
Democracy Dies in Darkness

(AP Photo/Chris Carlson)

Opinion by Greg Sargent
Columnist

September 9, 2020 at 10:43 a.m. EDT

A new poll from NBC News and Marist College finds Joe Biden leading President Trump by nine points in the crucial state of Pennsylvania, 53 percent to 44 percent. But it also finds Trump leading by 10 points on who will best handle the economy, 51 percent to 41 percent.

Which points to a crucial 2020 dynamic: If anything is still keeping Trump within range of winning through a real comeback, a major polling error or outright cheating, it's his lingering advantage on the economy.

Can the former vice president eliminate or neutralize that advantage?

Biden is set to roll out a new economic agenda designed to do just this. It should also prompt a reconsideration of another big question: how vulnerable Trump has made himself by thoroughly selling out on the "populist economic nationalism" he ran on in 2016.

Example: cont.

We would like to classify documents as election related or not.

- Given a collection of documents with their labels ('training data') we learn the parameters for our model.
- For example, if we see the word 'Trump' in *n1* out of the *n* documents labeled as 'election' we set *p('Trump'|'election')=n1/n*
- Similarly we compute the priors (*p('election')*) based on the proportion of the documents from both classes.

The Washington Post
Democracy Dies in Darkness

(AP Photo/Chris Carlson)

Opinion by Greg Sargent
Columnist

September 9, 2020 at 10:43 a.m. EDT

A new poll from NBC News and Marist College finds Joe Biden leading President Trump by nine points in the crucial state of Pennsylvania, 53 percent to 44 percent. But it also finds Trump leading by 10 points on who will best handle the economy, 51 percent to 41 percent.

Which points to a crucial 2020 dynamic: If anything is still keeping Trump within range of winning through a real comeback, a major polling error or outright cheating, it's his lingering advantage on the economy.

Can the former vice president eliminate or neutralize that advantage?

Biden is set to roll out a new economic agenda designed to do just this. It should also prompt a reconsideration of another big question: how vulnerable Trump has made himself by thoroughly selling out on the "populist economic nationalism" he ran on in 2016.

Example: Classifying Election (E) or Sports (S)

Assume we learned the following model

$P(\phi^{trump} = 1 \ |E) = 0.8$, $P(\phi^{trumo} = 1 \mid S) = 0.1$ $P(S) = 0.5$

$P(\phi^{russia} = 1|E) = 0.9$, $P(\phi^{russia} = 1| S) = 0.05$ $P(E) = 0.5$

$P(\phi^{biden} = 1|E) = 0.9$, $P(\phi^{biden} = 1|S) = 0.05$

$P(\phi^{football} = 1|E) = 0.1$, $P(\phi^{football} = 1|S) = 0.7$

… and we have the following feature vector for an input document:

$$\phi^{trump} = 1,\ \phi^{russia} = 1,\ \phi^{biden} = 1,\ \phi^{football} = 0$$

$P(y = E \mid 1,1,1,0) \propto 0.8*0.9*0.9*0.9*0.5 = 0.5832$

$P(y = S \mid 1,1,1,0) \propto 0.1*0.05*0.05*0.3*0.5 = 0.000075$

So the document is classified as ‘Election’

Naïve Bayes classifiers for continuous values

- So far we assumed a binomial or discrete distribution for the data given the model ($p(X_i|y)$)
- However, in many cases the data contains continuous features:
  - Height, weight
  - Levels of genes in cells
  - Brain activity
- For these types of data we often use a Gaussian model
- In this model we assume that the observed input vector X is generated from the following distribution

$$X \sim N(\mu, \Sigma)$$

Gaussian Bayes Classifier Assumption

- The i'th record in the database is created using the following algorithm

1. Generate the output (the "class") by drawing $y_i \sim Multinomial(p_1, p_2, \ldots p_{Ny})$
2. Generate the inputs from a Gaussian PDF that depends on the value of y_i :

$$\boldsymbol{x}_j \sim N(\boldsymbol{\mu}_i, \Sigma_i).$$

Gaussian Bayes Classification

$$P(y=v \mid X) = \frac{p(X \mid y=v)P(y=v)}{p(X)}$$

- To determine the class when using the Gaussian assumption we need to compute p(X|y):

$$P(X \mid y) = \frac{1}{(2\pi)^{n/2} \mid \Sigma \mid^{1/2}} \exp\left[-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)\right]$$

Once again, we need lots of data to compute the values of the mean μ and the covariance matrix Σ

Gaussian Bayes Classification

- Here we can also use the Naïve Bayes assumption: Attributes are independent given the class label
- In the Gaussian model this means that the covariance matrix becomes a **diagonal matrix** with zeros everywhere except for the diagonal
- Thus, we only need to learn the values for the variance term for each attribute in each class: $x^j \sim N(\mu_v^j, \sigma_v^j)$

$$P(X|y=v) = \prod_j P(x^j|y=v) = \prod_j \frac{1}{(2\pi)^{1/2}\sigma_v^j} exp\left[-\frac{\left(x^j-\mu_v^j\right)^2}{2\left(\sigma_v^j\right)^2}\right]$$

Separate means and variance for each class

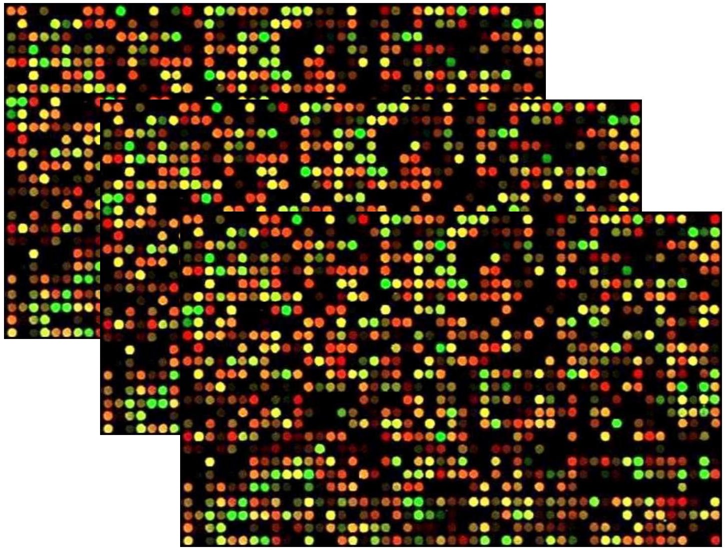
MLE for Gaussian Naïve Bayes Classifier

- For each class we need to estimate one global value (prior) and two values for each feature (mean and variance)
- The prior is computed in the same way we did before (counting) which is the MLE estimate
- Let the numbers of input samples in class 1 be k1. The MLE for mean and variance is computed by setting:

$$\mu_1^j = \sum_{i\ s.t. y_i=1} \frac{x_i^j}{k1} \qquad {\sigma_1^j}^2 = \sum_{i\ s.t. y_i=1} \frac{(x_i^j - \mu_1^j)^2}{k1}$$

Example: Classifying gene expression data

- Measures the levels (up or down) of genes in our cells
- Differs between healthy and sick people and between different disease types
- Given measurement of patients with two different types of cancer we would like to generate a classifier to distinguish between them



Classifying cancer types

- We select a subset of the genes (more in our 'feature selection' class later in the course).
- We compute the mean and variance for each of the genes in each of the classes
- Compute the class priors based on the input samples

Class 1 (ALL)

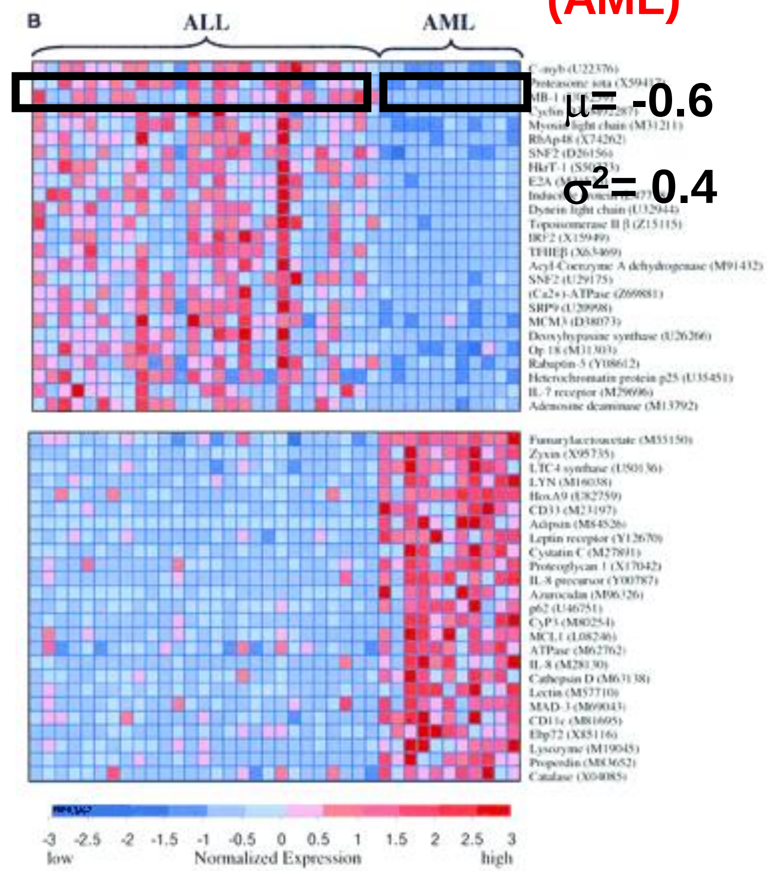
$\mu = 1.8$

$\sigma^2 = 1.1$

Class 2 (AML)

$\mu = -0.6$

$\sigma^2 = 0.4$



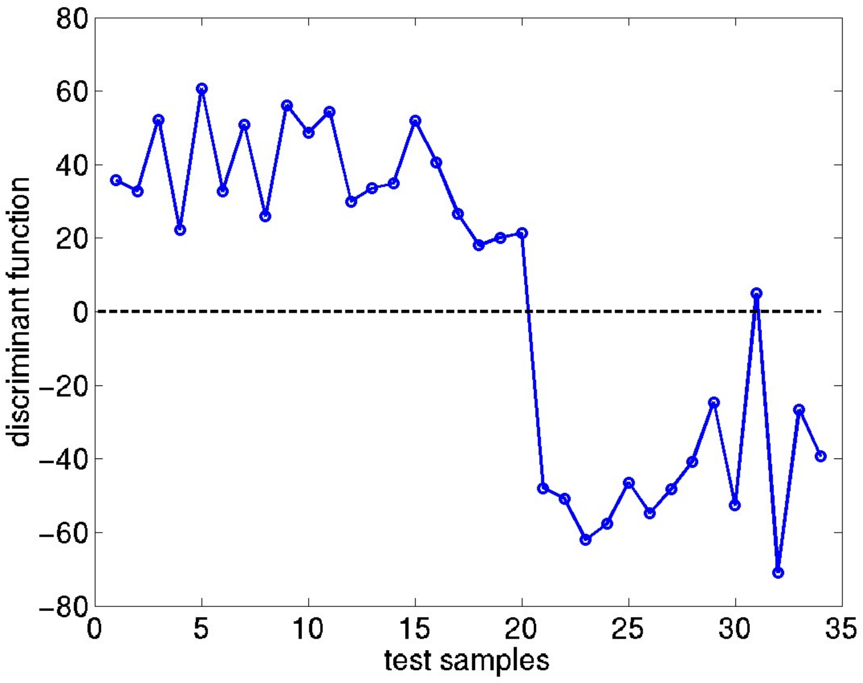
Classification accuracy

- The figure shows the value of the discriminate function

$$f(x) = \log \frac{p(y=1 \mid X)}{p(y=0 \mid X)}$$

across the test examples

- The only test error is also the decision with the lowest confidence



FDA Approves Gene-Based Cancer Test

“>400 DNA-sequenced genes and >250 RNA-sequenced genes

Combines DNA and RNA sequencing to detect all four main classes of genomic alterations, including sensitive identification of translocations and fusions.”

COMPARE OUR TESTS

Our Testing Portfolio

COMPARE	FOUNDATIONONE®CDX	FOUNDATIONONE®LIQUID CDX	FOUNDATIONONE®HEME
OVERVIEW	FDA-approved tissue-based companion diagnostic for all solid tumors, indicated for 20+ targeted therapies **View CDx Indications**	FDA-approved blood-based companion diagnostic for all solid tumors, indicated for 4 targeted therapies **View CDx Indications**	A laboratory developed test for hematologic malignancies, sarcomas or solid tumors where known or novel gene fusion detection is desired
CANCER TYPE	All Solid Tumors	All Solid Tumors	Hematologic Malignancies, Sarcomas, and Solid Tumors where known or novel gene fusion detection is desired
TYPICAL TURNAROUND TIME	**<2 weeks** from receipt of specimen	**<2 weeks** from receipt of specimen	**2 weeks** from receipt of specimen
NUMBER OF GENES ANALYZED	324 (DNA)	324 genes (DNA)*	406 genes (DNA), 265 genes (RNA)
SPECIMEN COLLECTION KIT			
SPECIMEN TYPE	FFPE Tissue **View Specimen Instructions**	Peripheral Whole Blood **View Specimen Instructions**	FFPE Tissue, Bone Marrow Aspirate, Peripheral Whole Blood

Foundation Medicine

Possible problems with Naïve Bayes classifiers: Assumptions

- In most cases, the assumption of conditional independence given the class label is violated

  - much more likely to find the word ‘Donald’ if we saw the word ‘Trump’ regardless of the class
- This is, unfortunately, a major shortcoming which makes these classifiers inferior in many real world applications (though not always)
- There are models that can improve upon this assumption without using the full conditional model (one such model are Bayesian networks which we will discuss later in this class).

Possible problems with Naïve Bayes classifiers: Parameter estimation

- Even though we need far less data than the full Bayes model, there may be cases when the data we have is not enough
- For example, what is $p(S=1,N=1|E=2)$?
- What if we have 20 variables, almost all pointing in the direction of the same class except for one for which we have no record for this class?
- Solutions?

Summer?	Num > 20	Evaluation
1	1	3
1	0	3
0	1	2
0	1	1
0	0	3
1	1	1

Important points

- Problems with estimating full joints
- Advantages of Naïve Bayes assumptions
- Applications to discrete and continuous cases
- Problems with Naïve Bayes classifiers
- Optional reading: Mitchell 6.1-6.10