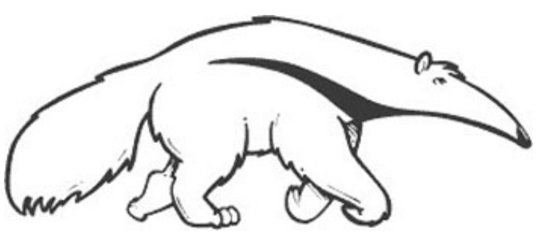
+

CS178: Machine Learning and Data Mining

Bayesian Classifiers & Naïve Bayes

Prof. Erik Sudderth



Some materials courtesy Alex Ihler & Sameer Singh

CS178 Zoom Lectures

CS178 zoom lectures are recorded by the instructor (the recording feature is disabled for students). Recordings are posted to YuJa, and only available to CS178 students and staff. To ask questions during lecture, you may:

- Use the Raise Hand feature. Prof. Sudderth will then call on you by name, unmute your microphone, and let you ask a question. *Your question will be recorded. Please be respectful of your instructor and classmates.*
- Use the Q&A Window to type a question. Prof. Sudderth will read your question to the class before answering it, but *will not personally identify you.*

Machine Learning

- **Bayesian Classification**
- Naïve Bayes
- Bayes Error Rates
- Gaussian Bayes Classifiers

A basic classifier

- Training data $D=\{x^{(i)},y^{(i)}\}$, Classifier $f(x ; D)$
  - Discrete feature vector x
  - $f(x ; D)$ is a contingency table
- Ex: credit rating prediction (bad/good)
  - X_1 = income (low/med/high)
  - How can we make the most # of correct predictions?

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

A basic classifier

- Training data D={$x^{(i)}$,$y^{(i)}$}, Classifier f(x ; D)
  - Discrete feature vector x
  - f(x ; D) is a contingency table
- Ex: credit rating prediction (bad/good)
  - X_1 = income (low/med/high)
  - How can we make the most # of correct predictions?
  - Predict more likely outcome for each possible observation

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

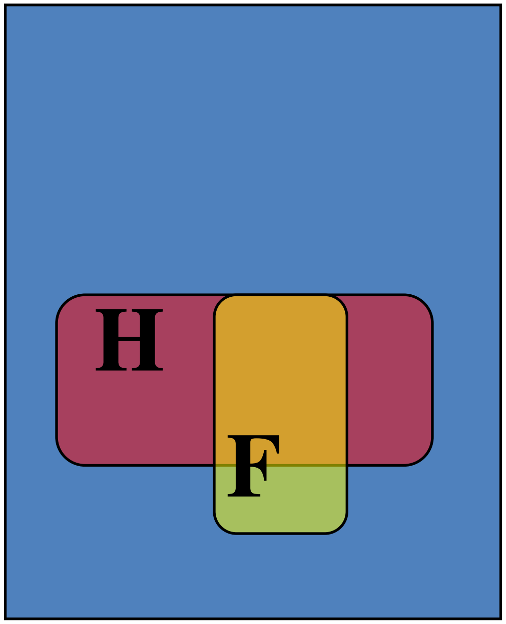
A basic classifier

- Training data D={$x^{(i)}$,$y^{(i)}$}, Classifier f(x ; D)
  - Discrete feature vector x
  - f(x ; D) is a contingency table
- Ex: credit rating prediction (bad/good)
  - X_1 = income (low/med/high)
  - How can we make the most # of correct predictions?
  - Predict more likely outcome for each possible observation
  - Can normalize into probability: p(y=good | X=c)
  - How to generalize?

Features	# bad	# good
X=0	.7368	.2632
X=1	.5408	.4592
X=2	.3750	.6250

Bayes Rule

- Two events: headache, flu
- $p(H) = 1/10$
- $p(F) = 1/40$
- $p(H|F) = 1/2$



You wake up with a headache.
What is the chance that you have the flu?

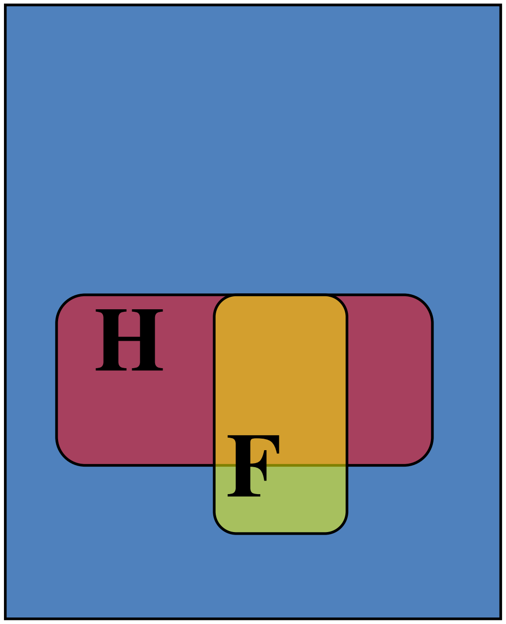
Example from Andrew Moore's slides

Bayes Rule

- Two events: headache, flu
- $p(H) = 1/10$
- $p(F) = 1/40$
- $p(H|F) = 1/2$

- $P(H \& F) = ?$

- $P(F|H) = ?$

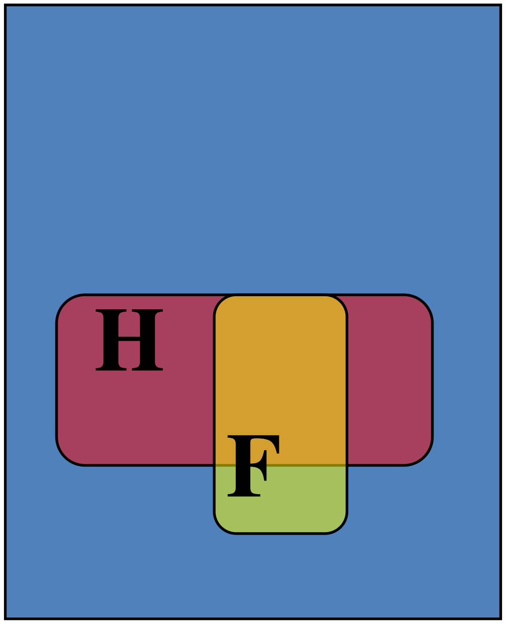


You wake up with a headache.
What is the chance that you have the flu?

Example from Andrew Moore's slides

Bayes rule

- Two events: headache, flu
- $p(H) = 1/10$
- $p(F) = 1/40$
- $p(H|F) = 1/2$



- $P(H \& F) = p(F)\, p(H|F)$
  $= (1/2) * (1/40) = 1/80$
- $P(F|H) = ?$

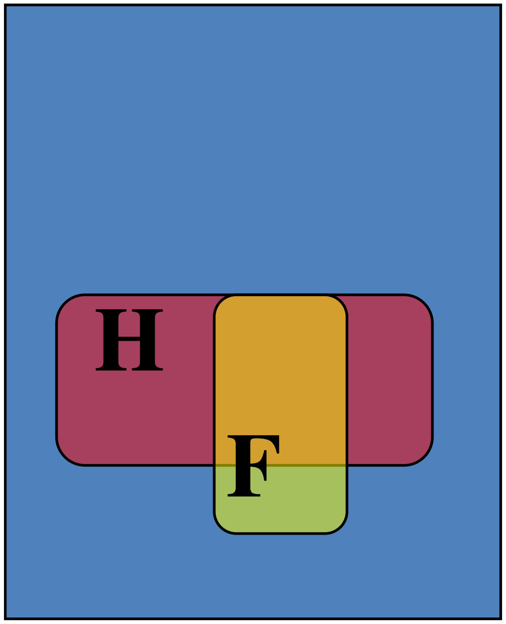
You wake up with a headache.
What is the chance that you have the flu?

Example from Andrew Moore's slides

Bayes rule

- Two events: headache, flu
- p(H) = 1/10
- p(F) = 1/40
- p(H|F) = 1/2

- P(H & F) = p(F) p(H|F)
  = (1/2) * (1/40) = 1/80
- P(F|H) = p(H & F) / p(H)
  = (1/80) / (1/10) = 1/8



You wake up with a headache.
What is the chance that you have the flu?

Example from Andrew Moore's slides

Classification and probability

- Suppose we want to model the data

- Prior probability of each class, p(y)
  - E.g., fraction of applicants that have good credit
- Distribution of features given the class, p(x | y=c)
  - How likely are we to see "x" in users with good credit?

- Joint distribution $p(y|x)p(x) = p(x,y) = p(x|y)p(y)$

- Bayes Rule: $\Rightarrow \quad p(y|x) = p(x|y)p(y)/p(x)$

$$= \frac{p(x|y)p(y)}{\sum_c p(x|y=c)p(y=c)}$$

(Use the rule of total probability to calculate the denominator!)

Bayes classifiers

- Learn "class conditional" models
  - Estimate a probability model for each class
- Training data
  - Split by class
  - $D_c = \{ x^{(j)} : y^{(j)} = c \}$
- Estimate $p(x \mid y=c)$ using D_c

- For a discrete x, this recalculates the same table…

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

p(y)	383/690	307/690

⇒

p(x | y=0)	p(x | y=1)
42 / 383	15 / 307
338 / 383	287 / 307
3 / 383	5 / 307

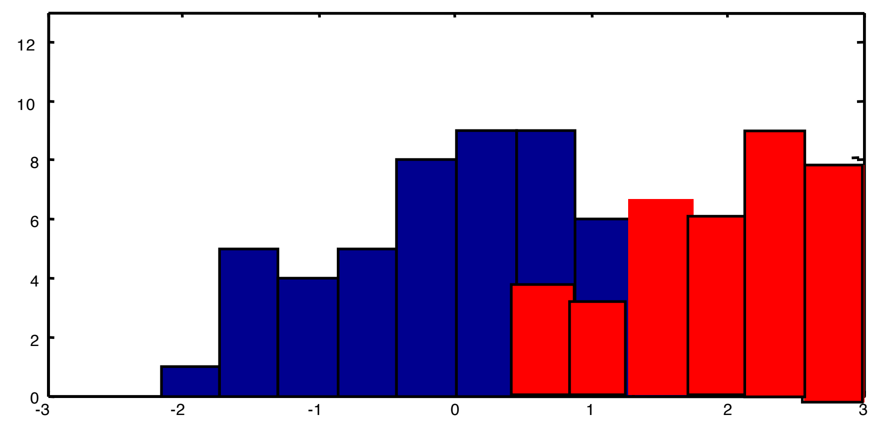
⇒

p(y=0|x)	p(y=1|x)
.7368	.2632
.5408	.4592
.3750	.6250

Bayes classifiers

- Learn "class conditional" models
  - Estimate a probability model for each class
- Training data
  - Split by class
  - $D_c = \{ x^{(j)} : y^{(j)} = c \}$
- Estimate $p(x \mid y=c)$ using D_c

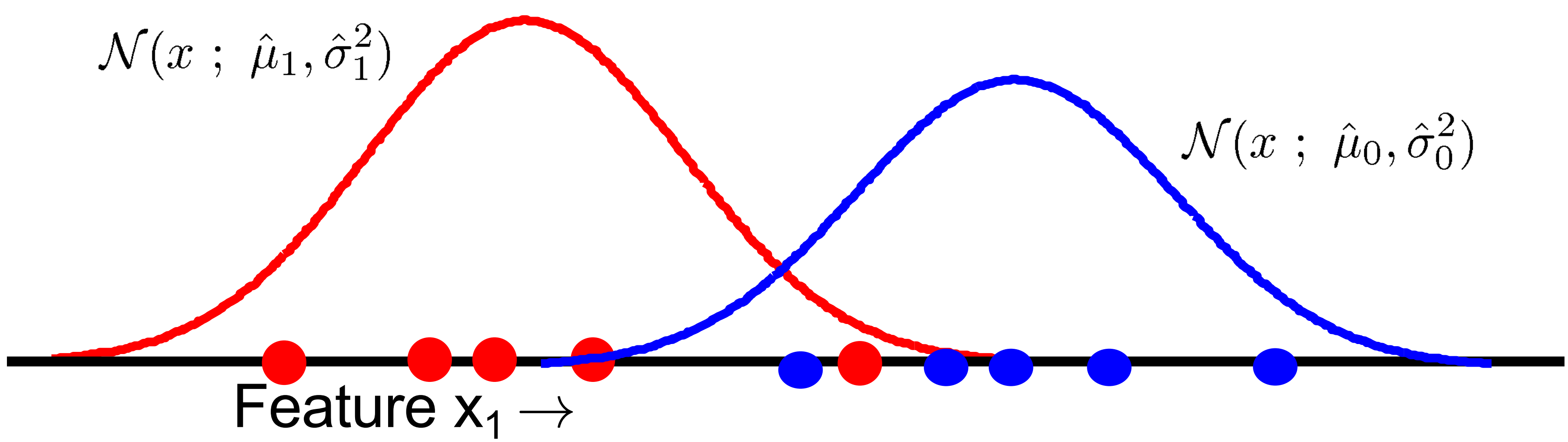
- For continuous x, can use any density estimate we like
  - Histogram
  - Gaussian
  - …



Gaussian models

- Estimate parameters of the Gaussians from the data

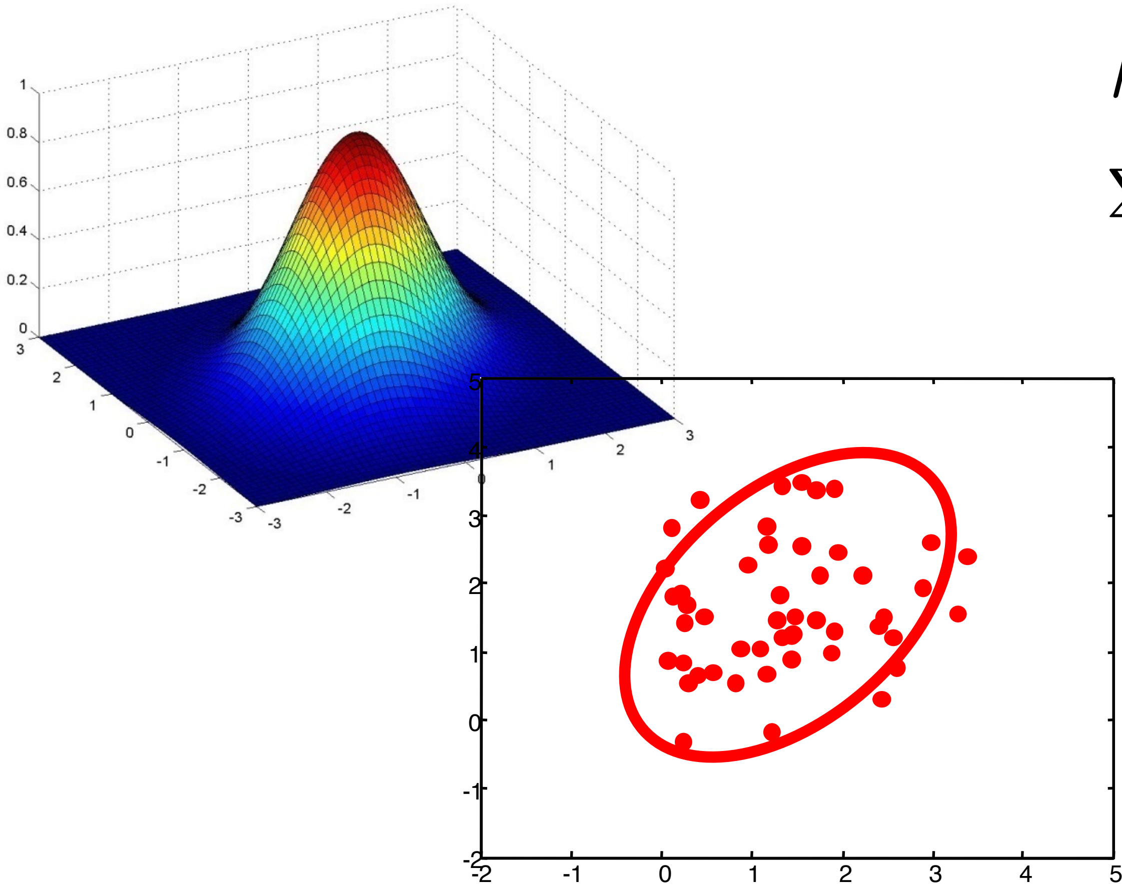
$$\alpha = \frac{m_1}{m} = \hat{p}(y = c_1) \qquad \hat{\mu} = \frac{1}{m}\sum_j x^{(j)} \qquad \hat{\sigma}^2 = \frac{1}{m}\sum_j (x^{(j)} - \mu)^2$$



Multivariate Gaussian models

- Similar to univariate case

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$



$$\mu = n \times 1 \text{ mean vector}$$

$$\Sigma = n \times n \text{ covariance matrix}$$

Maximum likelihood estimate:

$$\hat{\mu} = \frac{1}{m} \sum_j \underline{x}^{(j)}$$

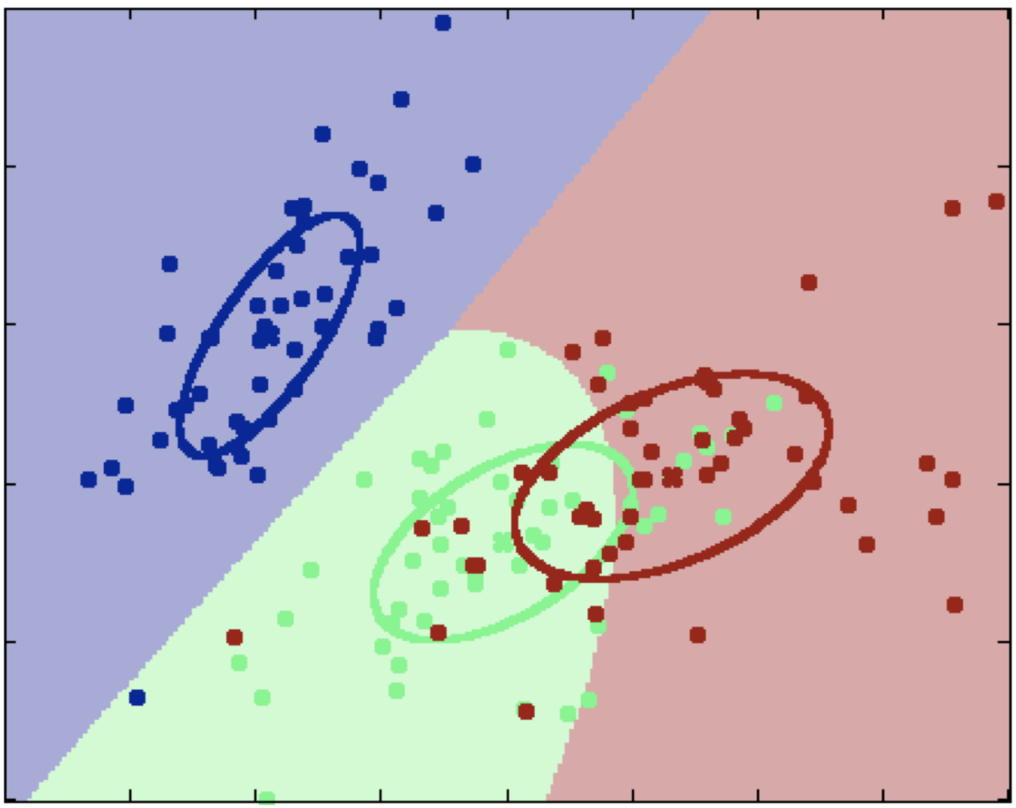
$$\hat{\Sigma} = \frac{1}{m} \sum_j (\underline{x}^{(j)} - \underline{\hat{\mu}})^T (\underline{x}^{(j)} - \underline{\hat{\mu}})$$

Example: Gaussian Bayes for Iris Data

- Fit Gaussian distribution to each class {0,1,2}

$$p(y) = \text{Discrete}(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3})$$

$$p(x_1, x_2 | y = 0) = \mathcal{N}(x\,;\, \mu_0, \Sigma_0)$$
$$p(x_1, x_2 | y = 1) = \mathcal{N}(x\,;\, \mu_1, \Sigma_1)$$
$$p(x_1, x_2 | y = 2) = \mathcal{N}(x\,;\, \mu_2, \Sigma_2)$$



Machine Learning

Bayesian Classification

Naïve Bayes

Bayes Error Rates

Gaussian Bayes Classifiers

Bayes classifiers

- Estimate p(y) = [p(y=0) , p(y=1) …]
- Estimate p(x | y=c) for each class c
- Calculate p(y=c | x) using Bayes rule
- Choose the most likely class c

- For a discrete x, can represent as a contingency table…
  - What about if we have more discrete features?

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

p(y)	383/690	307/690

$\Rightarrow$

p(x | y=0)	p(x | y=1)
42 / 383	15 / 307
338 / 383	287 / 307
3 / 383	5 / 307

$\Rightarrow$

p(y=0|x)	p(y=1|x)
.7368	.2632
.5408	.4592
.3750	.6250

Joint distributions

- Make a truth table of all combinations of values

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Joint distributions

- Make a truth table of all combinations of values
- For each combination of values, determine how probable it is
- Total probability must sum to one
- How many values did we specify?

A	B	C	p(A,B,C | y=1)
0	0	0	0.50
0	0	1	0.05
0	1	0	0.01
0	1	1	0.10
1	0	0	0.04
1	0	1	0.15
1	1	0	0.05
1	1	1	0.10

Overfitting & density estimation

- Estimate probabilities from the data
  - E.g., how many times (what fraction) did each outcome occur?

A	B	C	p(A,B,C | y=1)
0	0	0	4/10
0	0	1	1/10
0	1	0	0/10
0	1	1	0/10
1	0	0	1/10
1	0	1	2/10
1	1	0	1/10
1	1	1	1/10

- m data << 2^n parameters?
- What about the zeros?
  - We learn that certain combinations are impossible?
  - What if we see these later in test data?
- Overfitting!

Overfitting & density estimation

- Estimate probabilities from the data
  - E.g., how many times (what fraction) did each outcome occur?

A	B	C	p(A,B,C | y=1)
0	0	0	4/10
0	0	1	1/10
0	1	0	0/10
0	1	1	0/10
1	0	0	1/10
1	0	1	2/10
1	1	0	1/10
1	1	1	1/10

- m data << 2^n parameters?
- What about the zeros?
  - We learn that certain combinations are impossible?
  - What if we see these later in test data?
- One option: regularize $\hat{p}(a,b,c) \propto (M_{abc} + \alpha)$
- Normalize to make sure values sum to one…

Overfitting & density estimation

- Another option: reduce the model complexity
  by assuming the features are independent of one another

- Independence:
- p(a,b) = p(a) p(b)

- $p(x_1, x_2, \ldots x_N \mid y{=}1) = p(x_1 \mid y{=}1)\, p(x_2 \mid y{=}1) \ldots p(x_N \mid y{=}1)$
- Only need to estimate each individually

A	p(A|y=1)
0	.4
1	.6

B	p(B |y=1)
0	.7
1	.3

C	p(C |y=1)
0	.1
1	.9

$\Rightarrow$

A	B	C	p(A,B,C | y=1)
0	0	0	.4 * .7 * .1
0	0	1	.4 * .7 * .9
0	1	0	.4 * .3 * .1
0	1	1	…
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Example: Naïve Bayes

Observed Data:

x_1	x_2	y
1	1	0
1	0	0
1	0	1
0	0	0
0	1	1
1	1	0
0	0	1
1	0	1

$$\hat{p}(y=1) = \frac{4}{8} \qquad = (1 - \hat{p}(y=0))$$

$$\hat{p}(x_1, x_2 | y=0) = \hat{p}(x_1 | y=0)\, \hat{p}(x_2 | y=0)$$

$$\hat{p}(x_1 = 1 | y=0) = \frac{3}{4} \qquad \hat{p}(x_1 = 1 | y=1) = \frac{2}{4}$$

$$\hat{p}(x_2 = 1 | y=0) = \frac{2}{4} \qquad \hat{p}(x_2 = 1 | y=1) = \frac{1}{4}$$

Prediction given some observation x?

$$\hat{p}(y=1)\hat{p}(x=11|y=1) \quad \lessgtr \quad \hat{p}(y=0)\hat{p}(x=11|y=0)$$

$$\frac{4}{8} \times \frac{2}{4} \times \frac{1}{4} \qquad\qquad \frac{4}{8} \times \frac{3}{4} \times \frac{2}{4}$$

Decide class 0

Example: Naïve Bayes

Observed Data:

x_1	x_2	y
1	1	0
1	0	0
1	0	1
0	0	0
0	1	1
1	1	0
0	0	1
1	0	1

$$\hat{p}(y=1) = \frac{4}{8} \qquad = (1 - \hat{p}(y=0))$$

$$\hat{p}(x_1, x_2 | y=0) = \hat{p}(x_1 | y=0)\, \hat{p}(x_2 | y=0)$$

$$\hat{p}(x_1 = 1 | y=0) = \frac{3}{4} \qquad \hat{p}(x_1 = 1 | y=1) = \frac{2}{4}$$

$$\hat{p}(x_2 = 1 | y=0) = \frac{2}{4} \qquad \hat{p}(x_2 = 1 | y=1) = \frac{1}{4}$$

$$\hat{p}(y=1 | x_1 = 1, x_2 = 1) = \frac{\frac{4}{8} \times \frac{2}{4} \times \frac{1}{4}}{\frac{3}{4} \times \frac{2}{4} \times \frac{4}{8} + \frac{2}{4} \times \frac{1}{4} \times \frac{4}{8}}$$

$$= \frac{1}{4}$$

Example: Joint Bayes

Observed Data:

x_1	x_2	y
1	1	0
1	0	0
1	0	1
0	0	0
0	1	1
1	1	0
0	0	1
1	0	1

$$\hat{p}(y=1) = \frac{4}{8} \qquad = (1 - \hat{p}(y=0))$$

$$\hat{p}(x_1, x_2 | y=0) =$$

x_1	x_2	p(x | y=0)
0	0	1/4
0	1	0/4
1	0	1/4
1	1	2/4

$$\hat{p}(x_1, x_2 | y=1) =$$

x_1	x_2	p(x | y=1)
0	0	1/4
0	1	1/4
1	0	2/4
1	1	0/4

$$\hat{p}(y=1 | x_1=1, x_2=1) = \frac{\frac{4}{8} \times 0}{\frac{2}{4} \times \frac{4}{8} + 0 \times \frac{4}{8}}$$

$$= 0$$

Naïve Bayes Models

- Variable y to predict, e.g. "auto accident in next year?"
- We have *many* co-observed vars $\mathbf{x}=[x_1 \ldots x_n]$
  - Age, income, education, zip code, …
- Want to learn $p(y \mid x_1 \ldots x_n)$, to predict y
  - Arbitrary distribution: $O(d^n)$ values!
- Naïve Bayes:
  - $p(y|\mathbf{x}) = p(\mathbf{x}|y)\, p(y) / p(\mathbf{x})$; $p(\mathbf{x}|y) = \prod_i p(x_i|y)$
  - Covariates are independent given "cause"
- Note: may not be a good model of the data
  - Doesn't capture correlations in x's
  - Can't capture some dependencies
- But in practice it often does quite well!

Naïve Bayes Models for Spam

- $y \in$ {spam, not spam}
- X = observed words in email
  - Ex: [“the” … “probabilistic” … “lottery”…]
  - “1” if word appears; “0” if not

- 1000’s of possible words: 2^{1000s} parameters?
- # of atoms in the universe: $\approx 2^{270}$…

- Model words ***given*** email type as independent
- Some words more likely for spam (“lottery”)
- Some more likely for non-spam (“probabilistic”)
- Only 1000’s of parameters now…

Naïve Bayes Gaussian Models

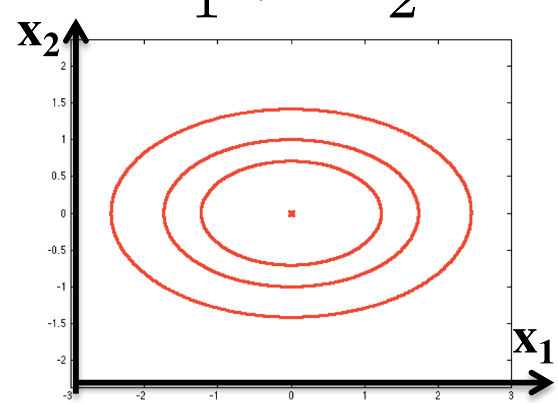
$$p(x_1) = \frac{1}{Z} \exp\left\{ -\frac{1}{2\sigma_1^2}(x_1 - \mu_1)^2 \right\} \qquad p(x_2) = \frac{1}{Z_2} \exp\left\{ -\frac{1}{2\sigma_2^2}(x_2 - \mu_2)^2 \right\}$$

$$p(x_1)p(x_2) = \frac{1}{Z_1 Z_2} \exp\left\{ -\frac{1}{2}(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$

$$\underline{\mu} = [\mu_1 \;\; \mu_2]$$

$$\Sigma = \mathrm{diag}(\sigma_1^2 \;,\; \sigma_2^2)$$

$$\sigma_1^2 > \sigma_2^2$$



Again, reduces the number of parameters of the model:

Bayes: $n^2/2$ Naïve Bayes: n

You should know…

- Bayes rule; $p(y | x)$
- Bayes classifiers
  - Learn $p(x | y=C)$, $p(y=C)$
- Naïve Bayes classifiers
  - Assume features are independent given class:

    $$p(x | y=C) = p(x_1 | y=C)\, p(x_2 | y=C) \ldots$$

- Maximum likelihood (empirical) estimators for
  - Discrete variables
  - Gaussian variables
  - Overfitting; simplifying assumptions or regularization

Machine Learning

Bayesian Classification

Naïve Bayes

Bayes Error Rates

Gaussian Bayes Classifiers

A Bayes Classifier

- Given training data, compute p(y=c| x) and choose largest
- What's the (training) error rate of this method?

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

A Bayes Classifier

- Given training data, compute p(y=c| x) and choose largest
- What's the (training) error rate of this method?

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

Gets these examples wrong:

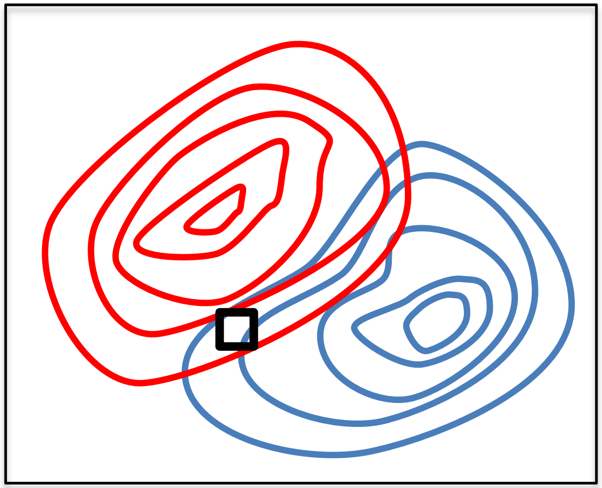
Pr[error] = (15 + 287 + 3) / (690)

(empirically on training data: better to use test data)

Bayes Error Rate

- Suppose that we knew the true probabilities:

$$p(x,y) \quad \Rightarrow p(y) \;, p(x|y=0) \;,\; p(x|y=1)$$



  - Observe any x: $\Rightarrow \; p(y=0|x)$ (at any x)

    $p(y=1|x)$

  - Optimal decision at that particular x is:

$$\hat{y} = f(x) = \arg\max_c \; p(y=c|x)$$

  - Error rate is:

$$\mathbb{E}_{xy}[\,y \neq \hat{y}\,] = \mathbb{E}_x[\,1 - \max_c p(y=c|x)\,]$$ = "Bayes error rate"

    - This is the best that **any** classifier can do!
    - Measures fundamental hardness of separating y-values given only features x

- Note: conceptual only!
  - Probabilities p(x,y) must be estimated from data
  - Form of p(x,y) is not known and may be very complex

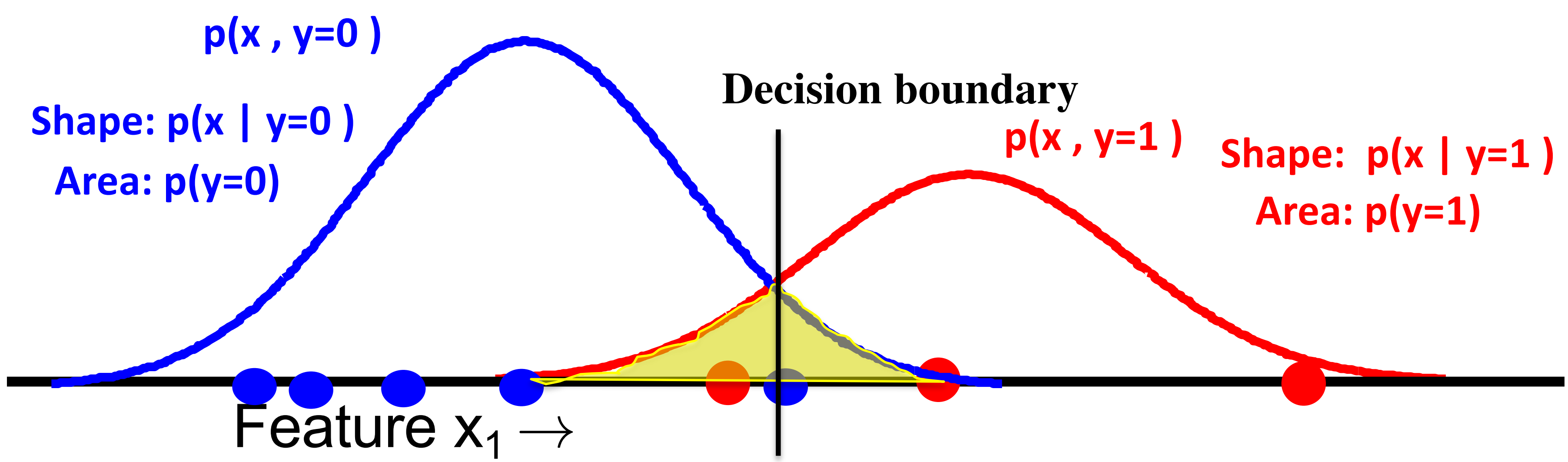
A Bayes classifier

- Bayes classification decision rule compares probabilities:

$$p(y=0|x) \underset{>}{<} p(y=1|x)$$

$$= \; p(y=0,x) \underset{>}{<} p(y=1,x)$$

- Can visualize this nicely if x is a scalar:

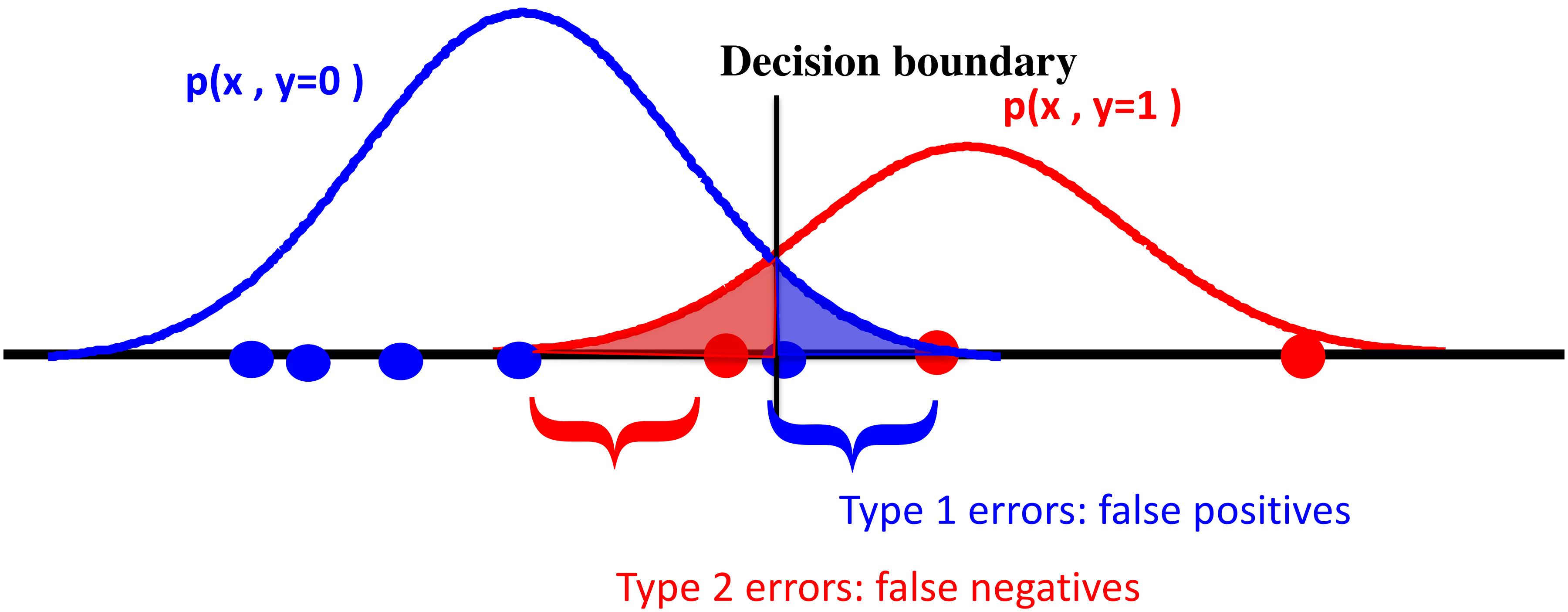


A Bayes classifier

- Not all errors are created equally…
- Risk associated with each outcome?

Add multiplier alpha:

$$\alpha \; p(y=0,x) \; \substack{<\\>} \; p(y=1,x)$$



False positive rate: (# y=0, ŷ=1) / (#y=0)

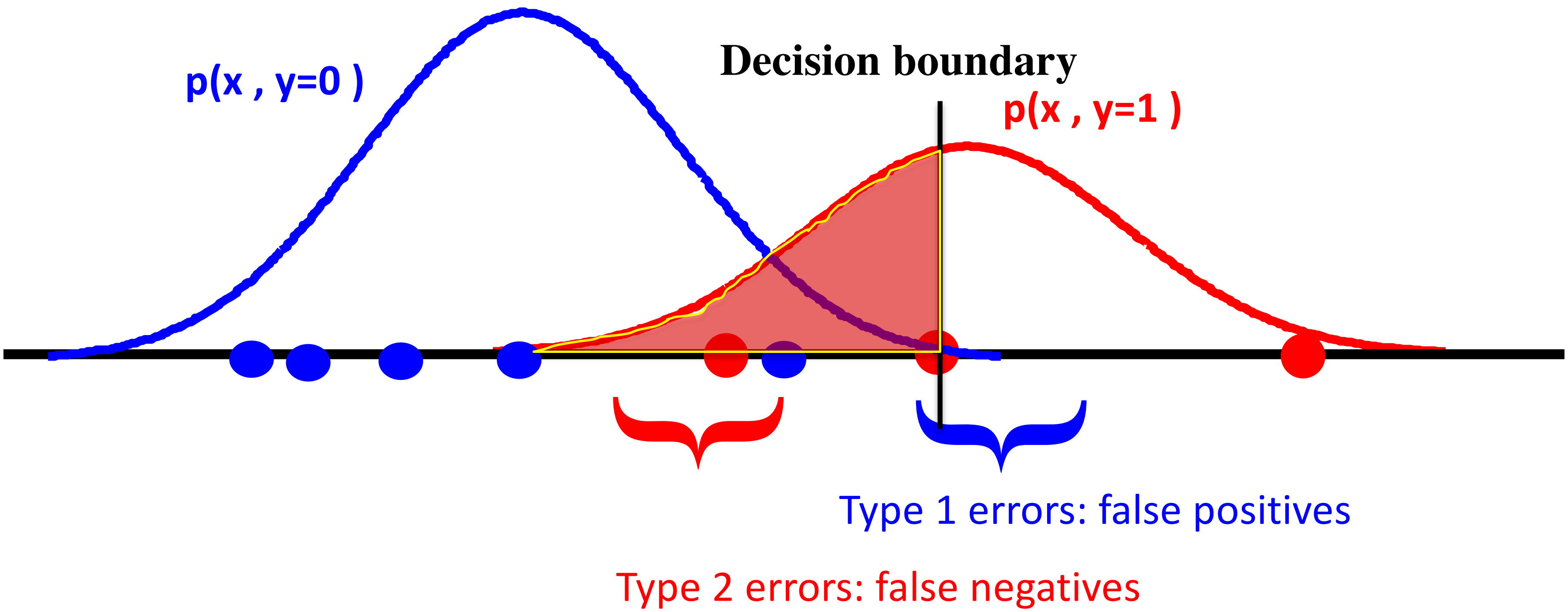
False negative rate: (# y=1, ŷ=0) / (#y=1)

A Bayes classifier

- Increase alpha: prefer class 0
- Spam detection

Add multiplier alpha:

$$\alpha\ p(y=0,x) \underset{>}{<} p(y=1,x)$$



False positive rate: (# y=0, ŷ=1) / (#y=0)

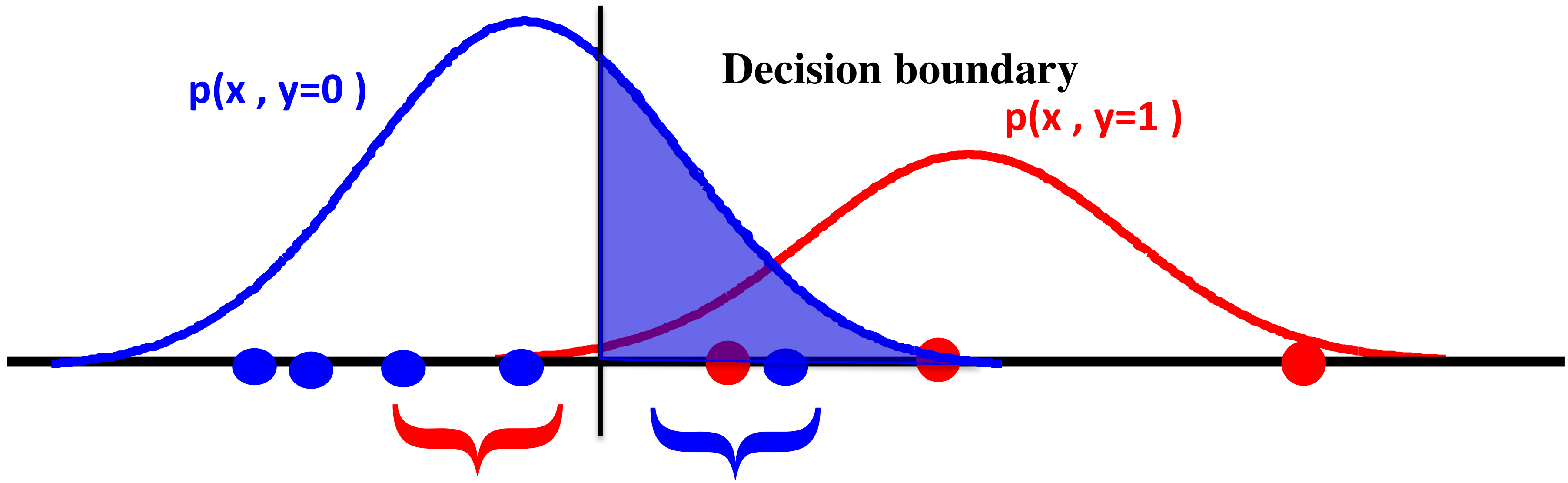
False negative rate: (# y=1, ŷ=0) / (#y=1)

A Bayes classifier

- Decrease alpha: prefer class 1
- Cancer detection

Add multiplier alpha:

$$\alpha \; p(y=0,x) \; \substack{<\\>} \; p(y=1,x)$$



Type 1 errors: false positives

Type 2 errors: false negatives

False positive rate: (# y=0, ŷ=1) / (#y=0)

False negative rate: (# y=1, ŷ=0) / (#y=1)

Measuring errors

- Confusion matrix
- Can extend to more classes

	Predict 0	Predict 1
Y=0	380	5
Y=1	338	3

- True positive rate: #(y=1 , ŷ=1) / #(y=1) -- “sensitivity”
- False negative rate: #(y=1 , ŷ=0) / #(y=1)
- False positive rate: #(y=0 , ŷ=1) / #(y=0)
- True negative rate: #(y=0 , ŷ=0) / #(y=0) -- “specificity”

Likelihood ratio tests

- Connection to classical, statistical decision theory:

$$p(y=0,x) \underset{>}{<} p(y=1,x) \quad = \quad \log \frac{p(y=0)}{p(y=1)} \underset{>}{<} \log \frac{p(x|y=1)}{p(x|y=0)}$$

"log likelihood ratio"

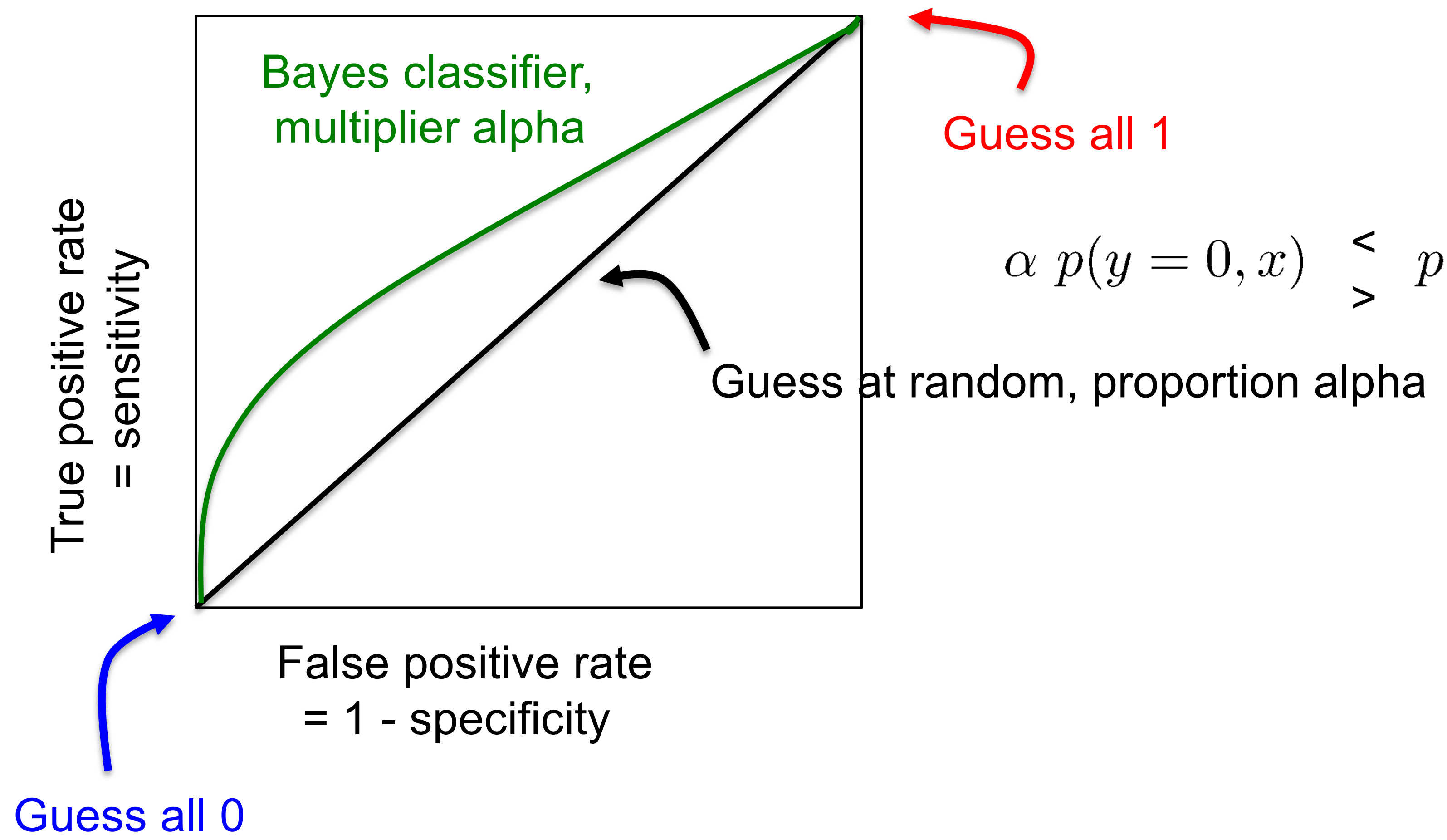
- Likelihood ratio: relative support for observation "x" under "alternative hypothesis" y=1, compared to "null hypothesis" y=0

- Can vary the decision threshold: $\gamma \underset{>}{<} \log \frac{p(x|y=1)}{p(x|y=0)}$

- Classical testing:
  - Choose gamma so that FPR is fixed ("p-value")
  - Given that y=0 is true, what's the probability we decide y=1?

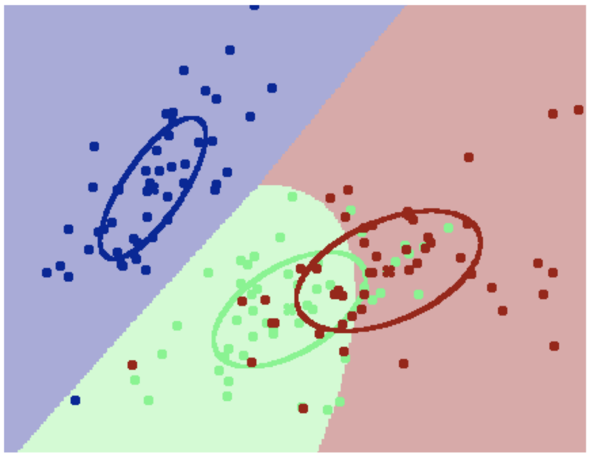
ROC Curves

- Characterize performance as we vary the decision threshold?

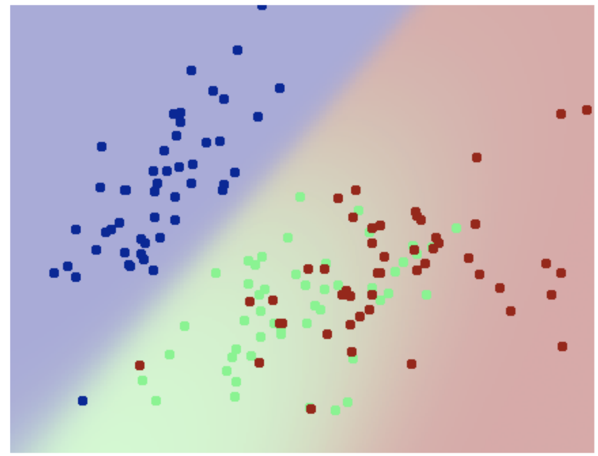


$$\alpha \; p(y=0,x) \;\underset{>}{<}\; p(y=1,x)$$

Probabilistic vs. Discriminative learning



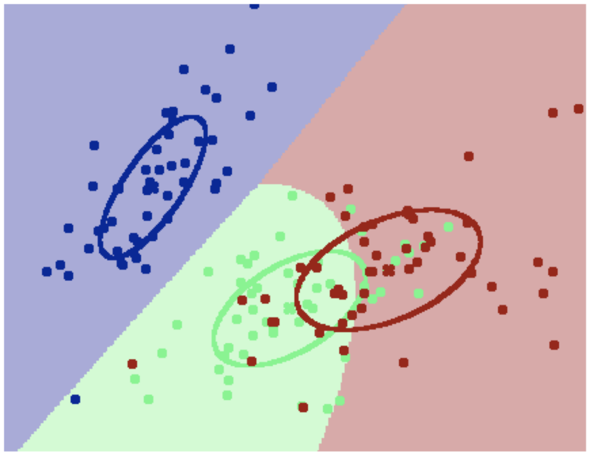
"Discriminative" learning:
Output prediction ŷ(x)



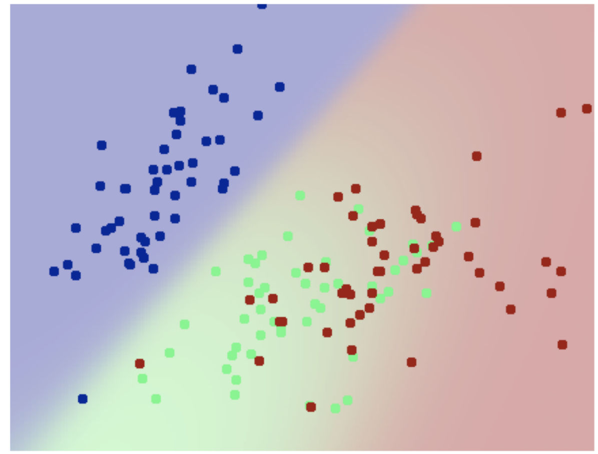
"Probabilistic" learning:
Output probability p(y|x)
(expresses confidence in outcomes)

- "Probabilistic" learning
  - Conditional models just explain y: p(y|x)
  - Generative models also explain x: p(x,y)
    - Often a component of unsupervised or semi-supervised learning
  - Bayes and Naïve Bayes classifiers are generative models

Probabilistic vs. Discriminative learning



"Discriminative" learning:
Output prediction $\hat{y}(x)$



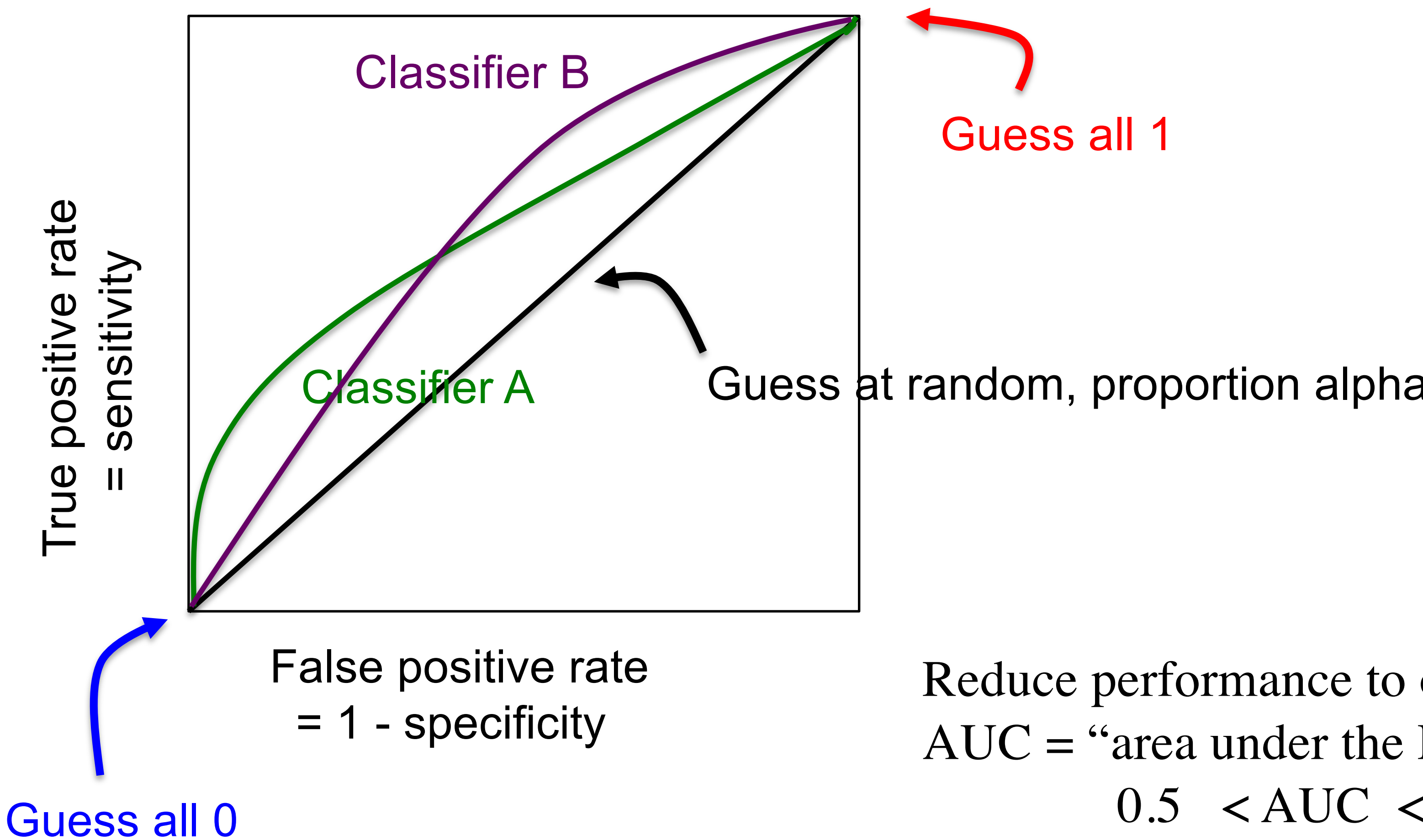
"Probabilistic" learning:
Output probability $p(y|x)$
(expresses confidence in outcomes)

- Can use ROC curves for discriminative models also:
  - Some notion of confidence, but doesn't correspond to a probability
  - In our code: "predictSoft" (vs. hard prediction, "predict")

```
learner = gaussianBayesClassify(X,Y)  # build a classifier
Ysoft = predictSoft(learner, X)        #  N x C matrix of confidences
plotSoftClassify2D(learner,X,Y)        #  shaded confidence plot
```

ROC Curves

- Characterize performance as we vary our confidence threshold?



Reduce performance to one number?
AUC = "area under the ROC curve"
$0.5 < \text{AUC} < 1$

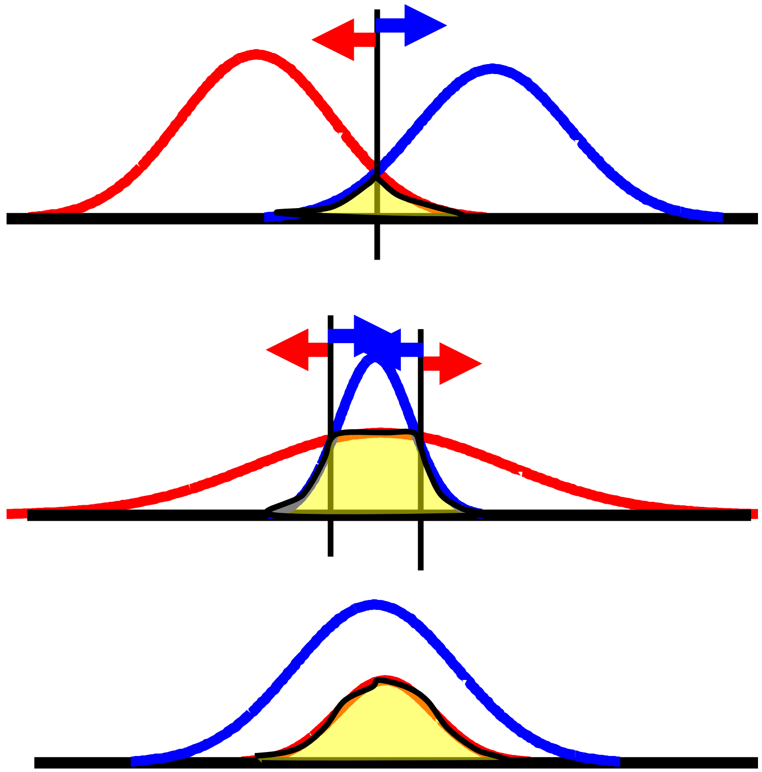
Machine Learning

- Bayesian Classification
- Naïve Bayes
- Bayes Error Rates
- **Gaussian Bayes Classifiers**

Gaussian models

- "Bayes optimal" decision
  - Choose most likely class
- Decision boundary
  - Places where probabilities equal

- What shape is the boundary?



Gaussian models

- Bayes optimal decision boundary
  - p(y=0 | x) = p(y=1 | x)
  - Transition point between p(y=0|x) >/< p(y=1|x)
- Assume Gaussian models with equal covariances

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$

$$0 \ \substack{< \\ >} \ \log \frac{p(x|y=0)}{p(x|y=1)} \frac{p(y=0)}{p(y=1)} = \log \frac{p(y=0)}{p(y=1)} +$$

$$-.5(x\Sigma^{-1}x - 2\mu_0^T \Sigma^{-1} x + \mu_0^T \Sigma^{-1} \mu_0)$$

$$+.5(x\Sigma^{-1}x - 2\mu_1^T \Sigma^{-1} x + \mu_1^T \Sigma^{-1} \mu_1)$$

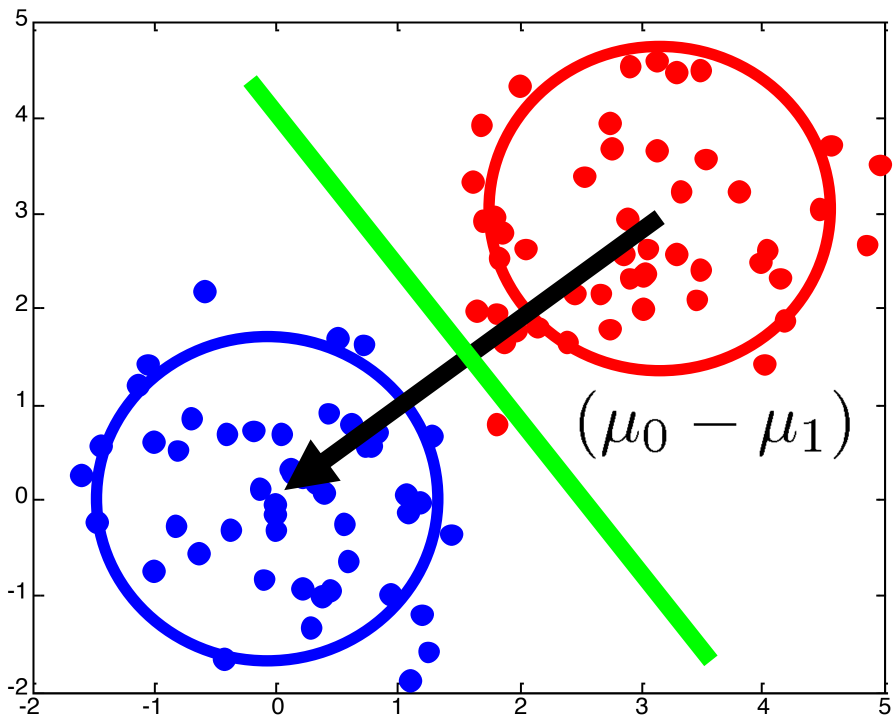
$$= (\mu_0 - \mu_1)^T \Sigma^{-1} x + constants$$

Gaussian example

- Spherical covariance: $\Sigma = \sigma^2 \mathrm{I}$
- Decision rule

$$= (\mu_0 - \mu_1)^T \Sigma^{-1} x + constants$$

$$(\mu_0 - \mu_1)^T x \begin{array}{c} < \\ > \end{array} C$$

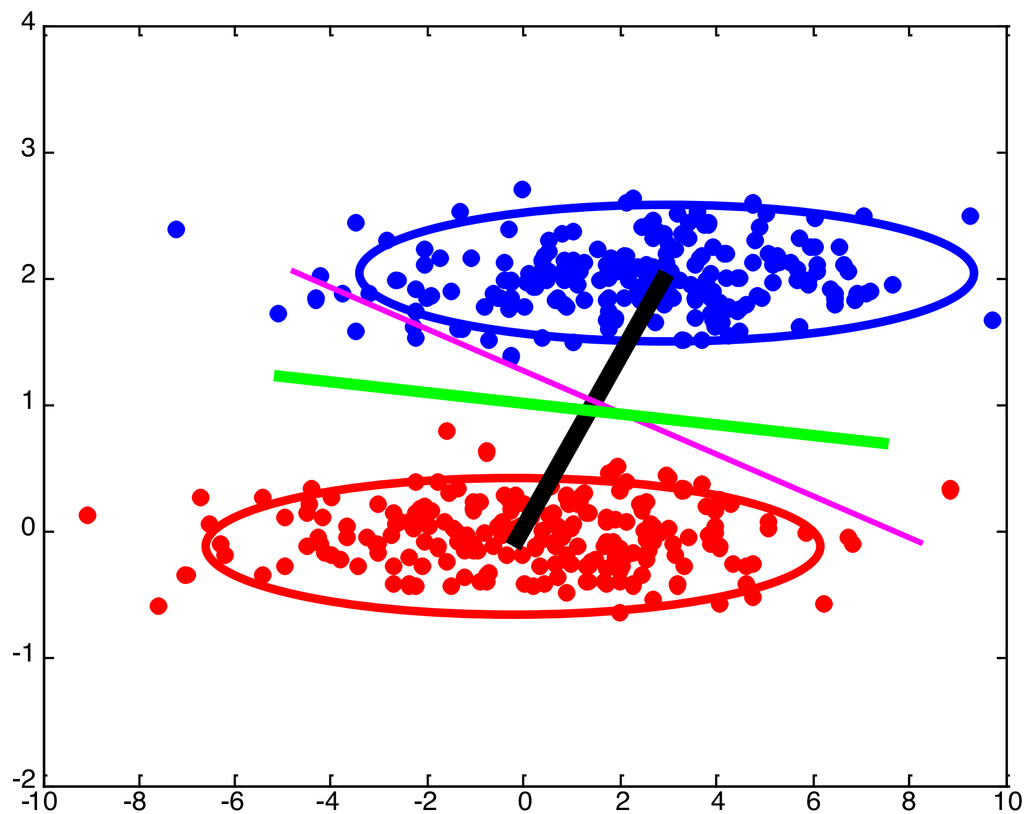


$$C = .5(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) - \log \frac{p(y=0)}{p(y=1)}$$

Non-spherical Gaussian distributions

- Equal covariances => still linear decision rule
  - May be “modulated” by variance direction
  - Scales; rotates (if correlated)

Ex:
Variance
[3   0   ]
[0 .25]



Class posterior probabilities

- Useful to also know class *probabilities*
- Some notation
  - p(y=0) , p(y=1) – class *prior* probabilities
    - How likely is each class in general?
  - p(x | y=c) – class conditional probabilities
    - How likely are observations "x" in that class?
  - p(y=c | x) – class posterior probability
    - How likely is class c *given* an observation x?

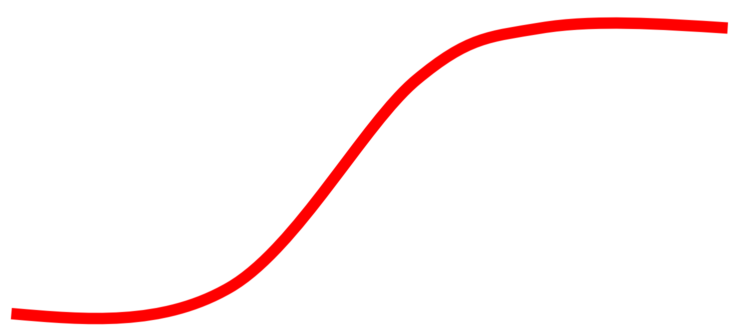
Class posterior probabilities

- Useful to also know class *probabilities*
- Some notation
  - $p(y=0)$, $p(y=1)$ – class *prior* probabilities
    - How likely is each class in general?
  - $p(x \mid y=c)$ – class conditional probabilities
    - How likely are observations "x" in that class?
  - $p(y=c \mid x)$ – class posterior probability
    - How likely is class c *given* an observation x?

- We can compute posterior using Bayes' rule
  - $p(y=c \mid x) = p(x|y=c)\, p(y=c) / p(x)$
- Compute p(x) using sum rule / law of total prob.
  - $p(x) = p(x|y=0)\, p(y=0) + p(x|y=1)p(y=1)$

Class posterior probabilities

- Consider comparing two classes
  - $p(x \mid y=0) * p(y=0) \quad \text{vs} \quad p(x \mid y=1) * p(y=1)$
  - Write probability of each class as
  - $p(y=0 \mid x) = p(y=0, x) / p(x)$
  - $\qquad = p(y=0, x) / (p(y=0,x) + p(y=1,x))$
  - $\quad = 1 / (1 + \exp(-a)) \qquad (**)$
  - $a = \log [p(x|y=0) p(y=0) / p(x|y=1) p(y=1)]$
  - (**) called the logistic function, or logistic sigmoid.



Gaussian models

- Return to Gaussian models with equal covariances

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2}(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$

$$0 \ \begin{matrix} < \\ > \end{matrix} \ \underbrace{\log \frac{p(x|y=0)}{p(x|y=1)} \frac{p(y=0)}{p(y=1)}}_{(**)} = (\mu_0 - \mu_1)^T \Sigma^{-1} x + constants$$

Now we also know that the probability of each class is given by:
p(y=0 | x) = Logistic(**) = Logistic(a^T x + b)

We'll see this form again soon…