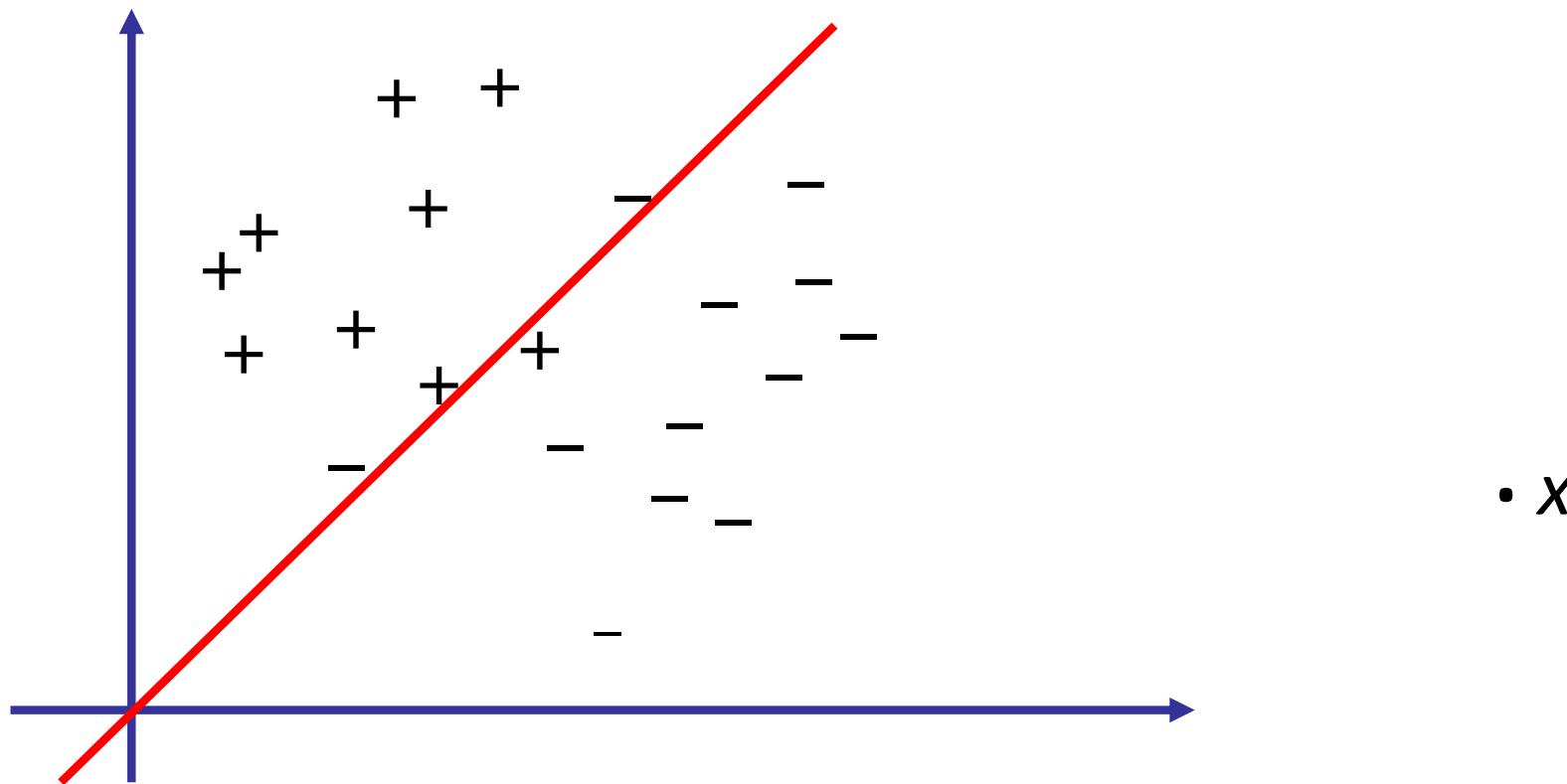


# Introduction to Machine Learning

## Discriminative vs. Generative Modeling

Prof. Andreas Krause  
Learning and Adaptive Systems ([las.ethz.ch](http://las.ethz.ch))

# Motivating example: Logistic regression



- What will logistic regression predict for data point  $x$ ?
- Logistic regression can be overconfident about labels for outliers

# Discriminative vs. Generative models

- Discriminative models aim to estimate

$$P(y \mid x)$$

$$\begin{aligned} P(x,y) &= P(x) P(y|x) \\ &\rightarrow P(y) P(x|y) \end{aligned}$$

- Generative models aim to estimate joint distribution

$$\underbrace{P(y, x)}$$

- Can derive conditional from joint distribution, but not vice versa!

$$\begin{aligned} \xrightarrow{\text{to}} P(y, x) \Rightarrow P(y|x) &= \frac{P(x, y)}{P(x)} \\ &\stackrel{L}{=} \sum_y P(x, y) \end{aligned}$$

# Typical approach to generative modeling

1. Estimate prior on labels

$$\underbrace{P(y)}$$

$$P(y, x) = \underbrace{P(x|y)}_{\text{(chain rule)}} \underbrace{P(y)}$$

2. Estimate conditional distribution  $P(\mathbf{x} | y)$

for each class  $y$

3. Obtain predictive distribution using Bayes' rule:

$$P(y | \mathbf{x}) = \frac{1}{Z} P(y) P(\mathbf{x} | y)$$

$\downarrow P(x)$

# A note on generative modeling

- Generative modeling attempts to infer the process, according to which examples are generated
- First generate class label  $P(y)$
- Then, generate features given class  $P(\mathbf{x} | y)$

$y$ (label)	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
$\mathbf{x}$ (vector of pixel intensities)	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9

# Example: Naive Bayes Model

- Model **class label** as generated from **categorical** variable

$$P(Y = y) = p_y \quad y \in \mathcal{Y} = \{1, \dots, c\}$$

- Model **features** as **conditionally independent** given Y

$$P(X_1, \dots, X_d \mid Y) = \prod_{i=1}^d P(X_i \mid Y)$$

- I.e., given class label, each feature is „generated“ independently of the other features.
- Need to still specify feature distributions  $P(X_i \mid Y)$

# Gaussian Naive Bayes Classifiers

- Learning given data

$$D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$$

- MLE for class prior:

$$\hat{P}(Y = y) = \hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

- MLE for feature distribution:  $\hat{P}(x_i \mid y) = \mathcal{N}(x_i; \hat{\mu}_{y,i}, \sigma_{y,i}^2)$

$$\hat{\mu}_{y,i} = \frac{1}{\text{Count}(Y=y)} \sum_{j:y_j=y} x_{j,i}$$

$$\sigma_{y,i}^2 = \frac{1}{\text{Count}(Y=y)} \sum_{j:y_j=y} (x_{j,i} - \hat{\mu}_{y,i})^2$$

- Prediction given new point  $\mathbf{x}$ :

$$y = \arg \max_{y'} \hat{P}(y' \mid \mathbf{x}) = \arg \max_{y'} \hat{P}(y') \prod_{i=1}^d \hat{P}(x_i \mid y')$$

# Gaussian NB vs. Logistic regression

- Gaussian NB with shared variance uses discriminant

$$f(\mathbf{x}) = \log \frac{P(Y = 1 \mid \mathbf{x})}{P(Y = -1 \mid \mathbf{x})}$$

where  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$  and

$$\begin{aligned} w_0 &= \log \frac{\hat{p}_+}{1 - \hat{p}_+} + \sum_{i=1}^d \frac{\hat{\mu}_{-,i}^2 - \hat{\mu}_{+,i}^2}{2\hat{\sigma}_i^2} \\ w_i &= \frac{\mu_{+,i} - \mu_{-,1}}{\sigma_i^2} \end{aligned}$$

- The corresponding class distribution

$$P(Y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-f(\mathbf{x}))} = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

is of the same form as logistic regression!

- If model assumptions are met, GNB will make same predictions as Logistic Regression!

# Issue with Naive Bayes models

---

- Conditional independence assumption means that features are generated independently given class label
- If there is (conditional) correlation between class labels, then this assumption is violated
- Due to conditional independence assumption, predictions can become overconfident (very close to 1 or 0)
- This might be fine if we care about most likely class only, but not if we want to use probabilities for making decisions (e.g., asymmetric losses etc.)

# More general: Gaussian Bayes classifiers

- Model **class label** as generated from **categorical variable**

$$P(Y = y) = p_y \quad y \in \mathcal{Y} = \{1, \dots, c\}$$

- Model **features** as generated by **multivariate Gaussian**

$$P(\mathbf{x} \mid y) = \mathcal{N}(\mathbf{x}; \mu_y, \Sigma_y)$$

for GNB:

$$\Sigma_y = \begin{pmatrix} \sigma_{y,1}^2 & & & \\ & \ddots & & 0 \\ & & \ddots & \\ 0 & & & \sigma_{y,d}^2 \end{pmatrix}$$

- How do we estimate the parameters?

# More general: Gaussian Bayes classifiers

- Model **class label** as generated from **categorical** variable

$$P(Y = y) = p_y \quad y \in \mathcal{Y} = \{1, \dots, c\}$$

- Model **features** as generated by **multivariate Gaussian**

$$P(\mathbf{x} \mid y) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$$

- How do we estimate the parameters?

# MLE for Gaussian Bayes Classifier

- Given data set  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$
- MLE for class label distribution  $\hat{P}(Y = y) = \hat{p}_y$

$$\hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

- MLE for feature distribution  $\hat{P}(\mathbf{x} \mid y) = \mathcal{N}(\mathbf{x}; \hat{\mu}_y, \hat{\Sigma}_y)$

$$\hat{\mu}_y = \frac{1}{\text{Count}(Y=y)} \sum_{i:y_i=y} \mathbf{x}_i$$

$$\hat{\Sigma}_y = \frac{1}{\text{Count}(Y=y)} \sum_{i:y_i=y} (\mathbf{x}_i - \hat{\mu}_y)(\mathbf{x}_i - \hat{\mu}_y)^T$$

# Discriminant functions for GBCs

- Given:  $P(Y = 1) = p$  and  $P(\mathbf{x} \mid y) = \mathcal{N}(\mathbf{x}; \mu_y, \Sigma_y)$

- Want:

$$f(\mathbf{x}) = \log \frac{P(Y = 1 \mid \mathbf{x})}{P(Y = -1 \mid \mathbf{x})}$$

- This discriminant function is given by

$$f(\mathbf{x}) = \log \frac{p}{1-p} + \frac{1}{2} \left[ \log \frac{|\hat{\Sigma}_-|}{|\hat{\Sigma}_+|} + \left( (\mathbf{x} - \hat{\mu}_-)^T \hat{\Sigma}_-^{-1} (\mathbf{x} - \hat{\mu}_-) \right) - \left( (\mathbf{x} - \hat{\mu}_+)^T \hat{\Sigma}_+^{-1} (\mathbf{x} - \hat{\mu}_+) \right) \right]$$

# Fisher's linear discriminant analysis LDA (c=2)

- Suppose we fix  $p=.5$
- Further, assume covariances are equal:  $\hat{\Sigma}_- = \hat{\Sigma}_+ = \hat{\Sigma}$
- Then the discriminant function

$$f(\mathbf{x}) = \log \frac{p}{1-p} + \frac{1}{2} \left[ \log \frac{|\hat{\Sigma}_-|}{|\hat{\Sigma}_+|} + \left( (\mathbf{x} - \hat{\mu}_-)^T \hat{\Sigma}_-^{-1} (\mathbf{x} - \hat{\mu}_-) \right) - \left( (\mathbf{x} - \hat{\mu}_+)^T \hat{\Sigma}_+^{-1} (\mathbf{x} - \hat{\mu}_+) \right) \right]$$

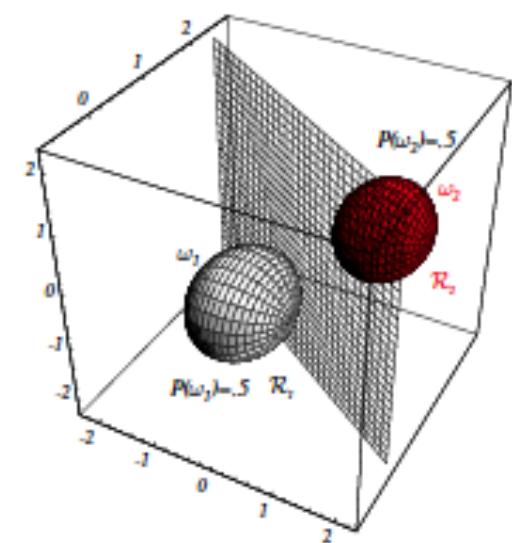
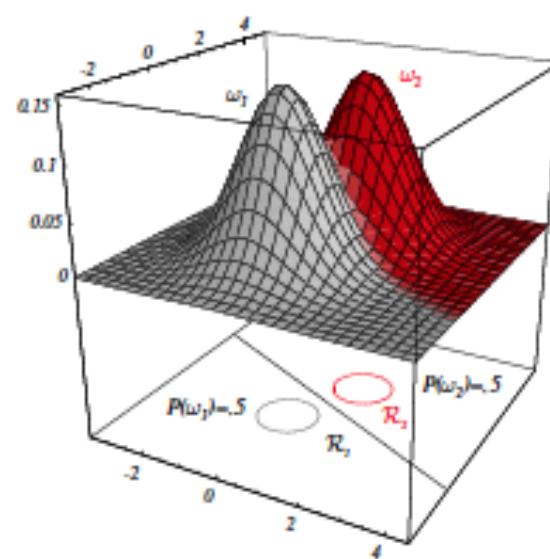
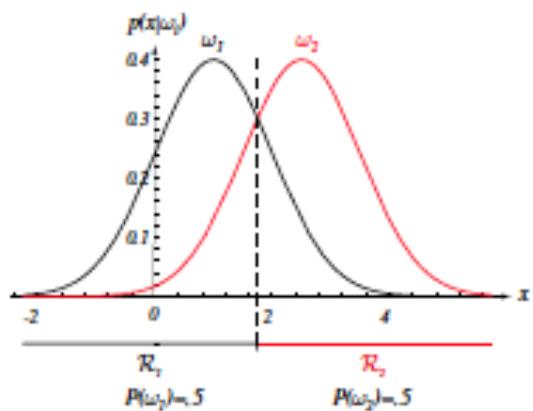
simplifies:  $f(\mathbf{x}) = \mathbf{x}^T \hat{\Sigma}^{-1} (\hat{\mu}_+ - \hat{\mu}_-) + \frac{1}{2} (\hat{\mu}_-^T \Sigma^{-1} \hat{\mu}_- - \hat{\mu}_+^T \Sigma^{-1} \hat{\mu}_+)$

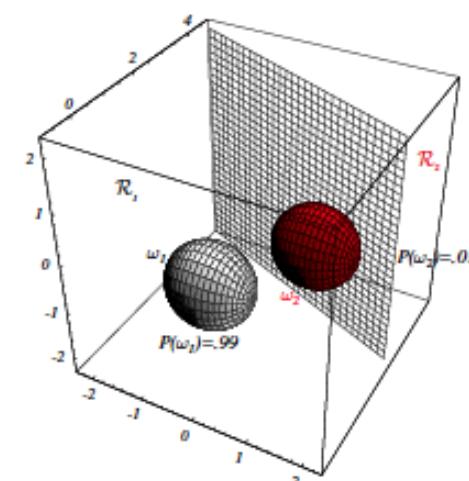
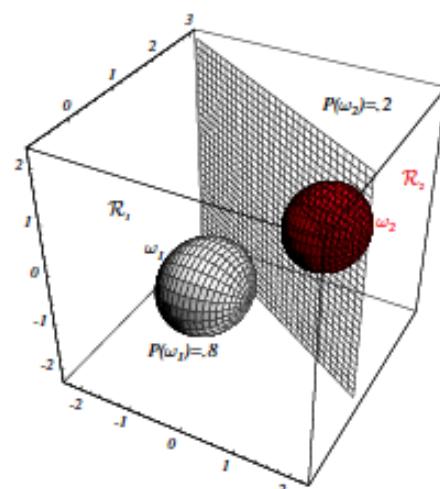
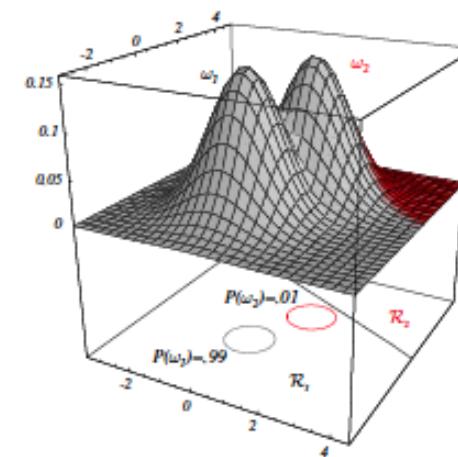
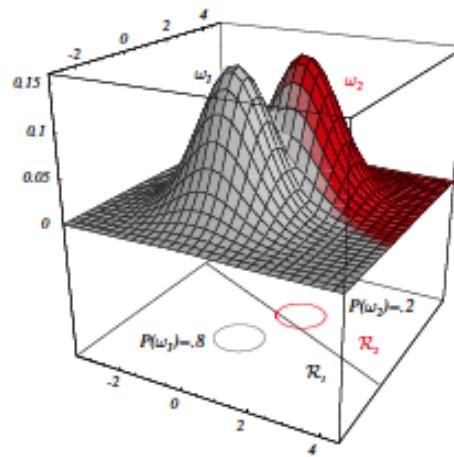
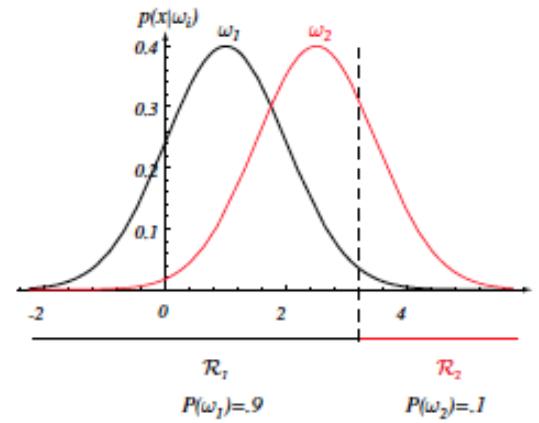
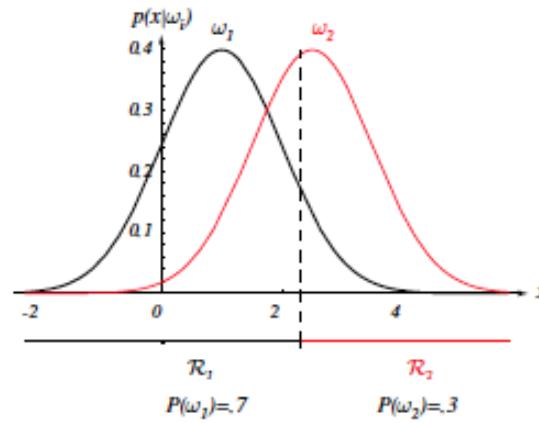
- Under these assumptions, we predict

$$y = \text{sign}(f(\mathbf{x})) = \text{sign}(\mathbf{w}^T \mathbf{x} + w_0) \quad \mathbf{w} = \hat{\Sigma}^{-1} (\hat{\mu}_+ - \hat{\mu}_-) \\ w_0 = \frac{1}{2} (\hat{\mu}_-^T \Sigma^{-1} \hat{\mu}_- - \hat{\mu}_+^T \Sigma^{-1} \hat{\mu}_+)$$

- This linear classifier is called  
**Fisher's linear discriminant analysis**

# Illustration





# Fisher's LDA vs. Logistic regression

- Fisher's LDA uses the discriminant function:

$$f(\mathbf{x}) = \log \frac{P(Y = 1 \mid \mathbf{x})}{P(Y = -1 \mid \mathbf{x})}$$

where  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$  and  $\mathbf{w} = \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-)$   
 $w_0 = \frac{1}{2}(\hat{\mu}_-^T \Sigma^{-1} \hat{\mu}_- - \hat{\mu}_+^T \Sigma^{-1} \hat{\mu}_+)$

- Can derive the class distribution

$$P(Y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-f(\mathbf{x}))} = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

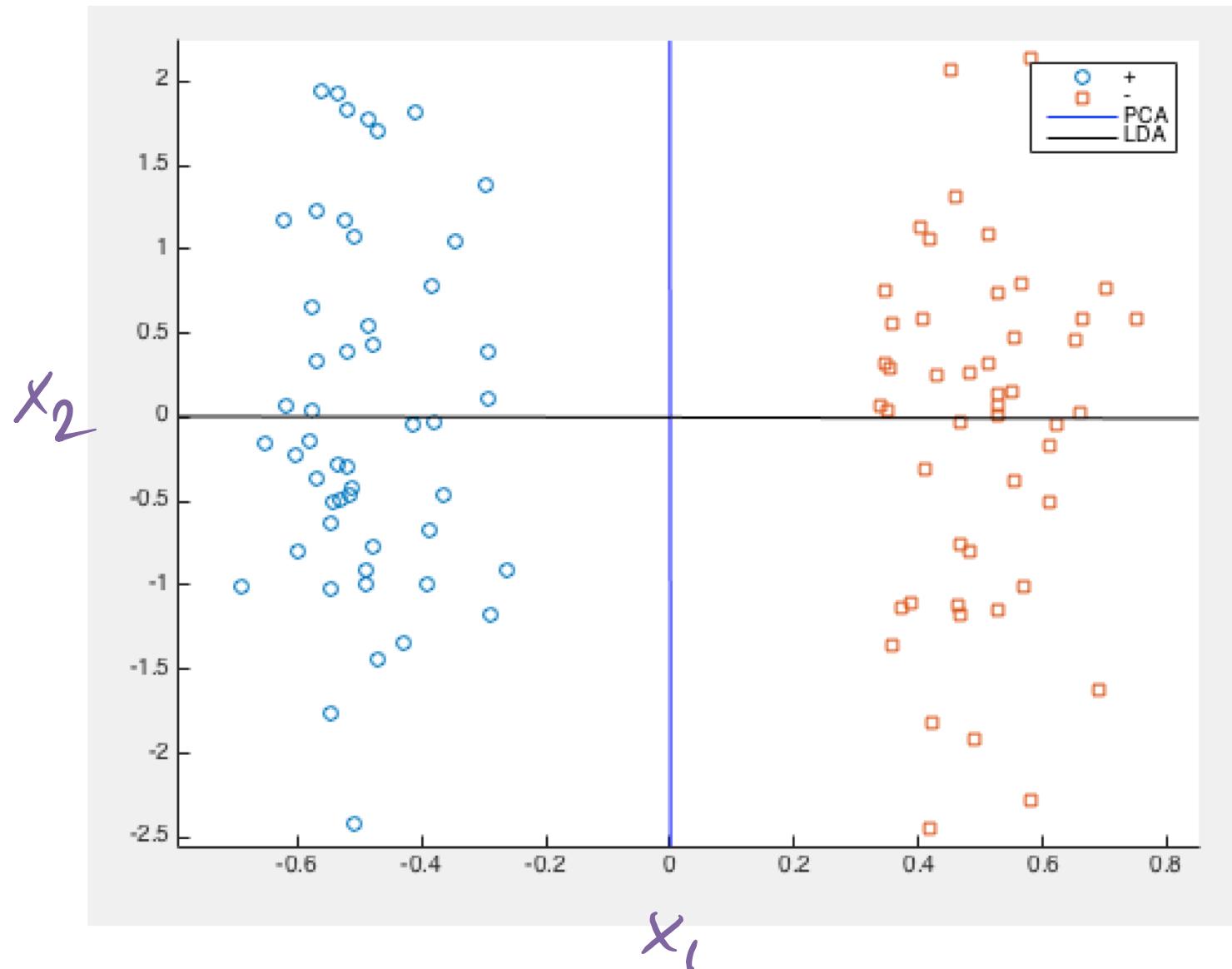
- This is the same form as logistic regression\*!
- If model assumptions are met, LDA will make same predictions as Logistic Regression!

# LDA vs. PCA

---

- LDA can be viewed as a projection to a 1-dim. subspace that maximizes ratio of between-class and within-class variances
- In contrast, PCA ( $k=1$ ) maximizes the variance of the resulting 1-dim. projection

# LDA vs. PCA

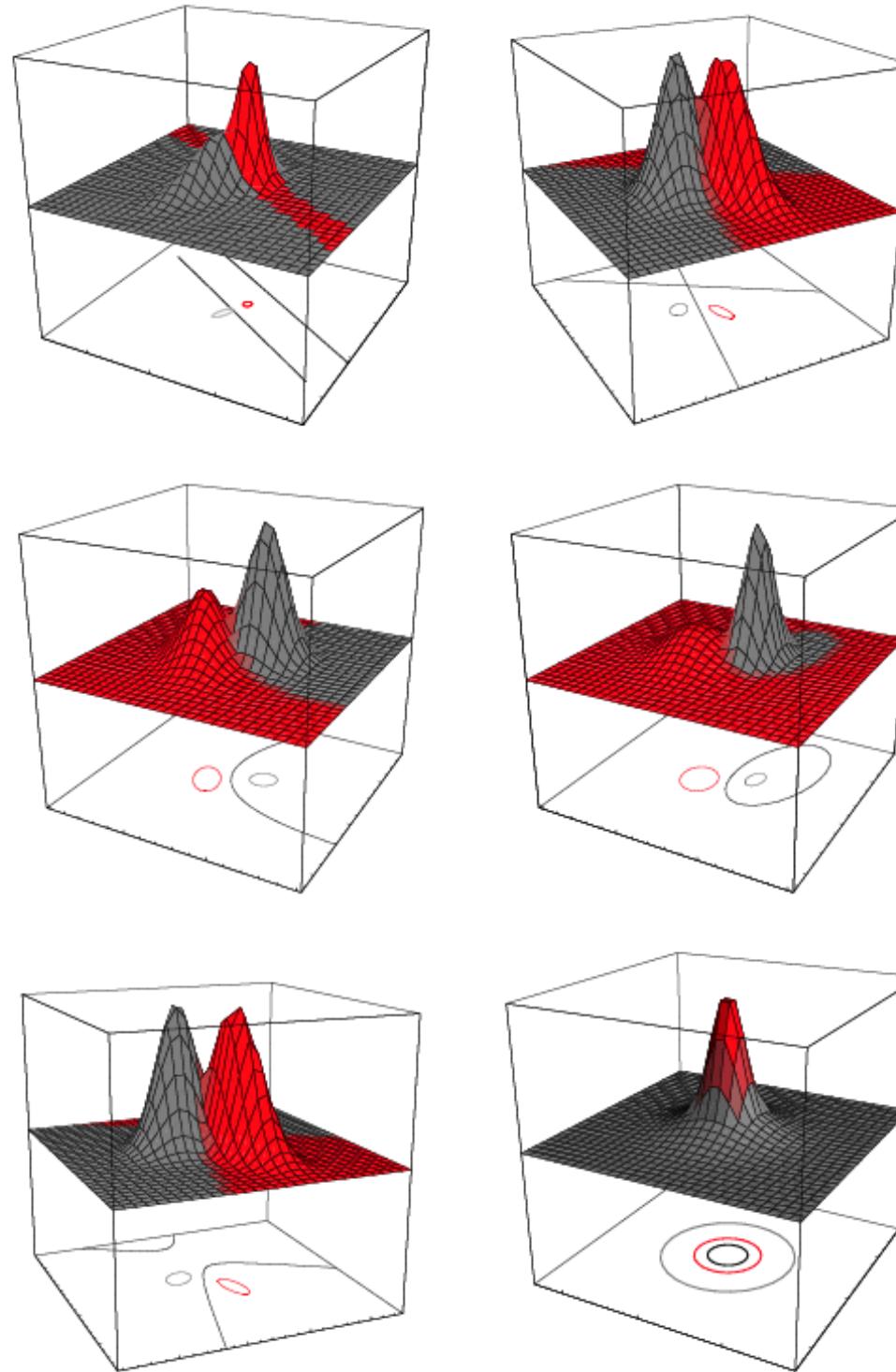


# Quadratic discriminant analysis

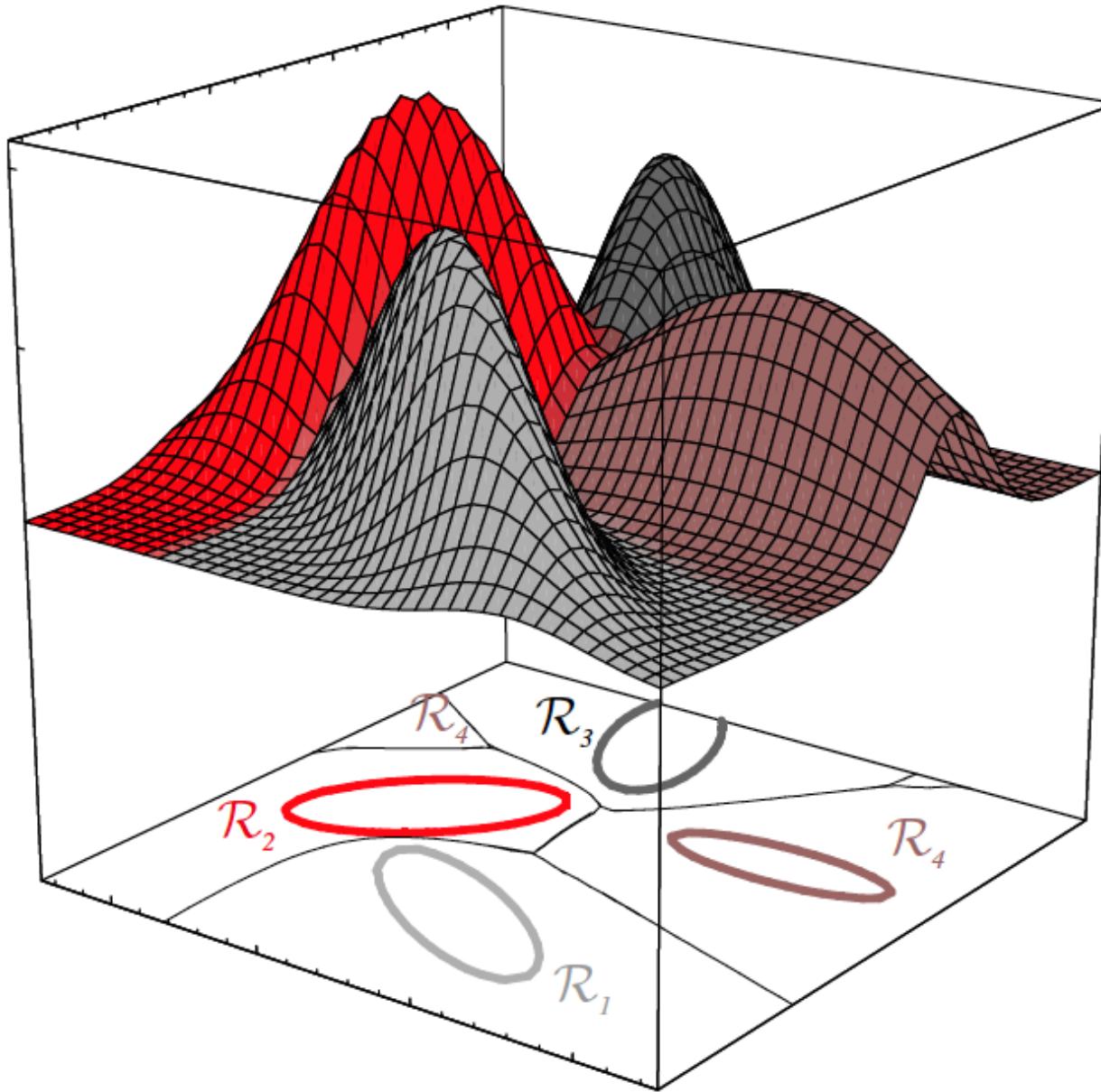
- In the general case,

$$f(\mathbf{x}) = \log \frac{p}{1-p} + \frac{1}{2} \left[ \log \frac{|\hat{\Sigma}_-|}{|\hat{\Sigma}_+|} + \left( (\mathbf{x} - \hat{\mu}_-)^T \hat{\Sigma}_-^{-1} (\mathbf{x} - \hat{\mu}_-) \right) - \left( (\mathbf{x} - \hat{\mu}_+)^T \hat{\Sigma}_+^{-1} (\mathbf{x} - \hat{\mu}_+) \right) \right]$$

- and we predict  $y = \text{sign}(f(\mathbf{x}))$
- This is called quadratic discriminant analysis



# Multiple classes



# Gaussian Bayes Classifiers Big Picture

Gaussian Bayes Classifiers

arbitrary  $\Sigma_y, c \geq 2$

Fisher's LDA

$$c = 2, p = 1/2 \\ \Sigma_+ = \Sigma_-$$

Gaussian Naive  
Bayes Classifiers

$$\Sigma_y = \text{diag}(\sigma_{y,1}^2, \dots, \sigma_{y,d}^2) \\ c \geq 2$$

Logistic regression

# Fisher's LDA vs. Logistic regression

---

- Fisher's LDA
  - Generative model, i.e., models  $P(\mathbf{X}, Y)$
  - Can be used to detect outliers:  $P(\mathbf{X}) < t$
  - Assumes normality of  $\mathbf{X}$
  - **not very robust** against violation of this assumption
- Logistic regression
  - Discriminative model, i.e., models  $P(Y|\mathbf{X})$  only
  - **Cannot detect outliers**
  - Makes no assumptions on  $\mathbf{X}$
  - **More robust**

# Gaussian Naive Bayes vs. General GBCs

---

- Gaussian Naive Bayes models
  - Conditional independence assumption may lead to overconfidence
  - Predictions might still be useful
  - $\# \text{parameters} = O(c d)$
  - Complexity (memory + inference) linear in  $d$
- General Gaussian Bayes models
  - Captures correlations among features
  - Avoids overconfidence
  - $\# \text{parameters} = O(c d^2)$
  - Complexity quadratic in  $d$

# What if we have discrete features?

---

- So far we assumed that  $\mathbf{x} \in \mathbb{R}^d$
- Suppose some  $X_i$  take discrete values
  - Gender
  - Nationality
  - ...
- Might not make sense to model  $X_i$  as a Gaussian
- Generative models allow to easily swap different distributions. E.g., model  $P(X_i | Y)$  as
  - Bernoulli
  - Categorical
  - Multinomial
  - ...

## Example: Categorical Naive Bayes classifiers

- Model class label as generated from categorical variable

$$P(Y = y) = p_y \quad y \in \mathcal{Y} = \{1, \dots, c\}$$

- Model features by (conditionally) independent categorical random variables

$$P(X_i = \underline{x} | Y = y) = \underline{\theta_{x|y}^{(i)}}$$

$$\forall i, x, y : \theta_{x|y}^{(i)} \geq 0$$

$$\forall i, y : \sum_{x=1}^c \theta_{x|y}^{(i)} = 1$$

- How do we estimate the parameters?

# MLE for Categorical Naive Bayes Classifier

- Given data set  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$
- MLE for class label distribution  $\hat{P}(Y = y) = \hat{p}_y$

$$\hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

- MLE for distribution of feature  $i$   $\hat{P}(X_i = c \mid y) = \theta_{c|y}^{(i)}$

$$\theta_{c|y}^{(i)} = \frac{\text{Count}(X_i = c, Y = y)}{\text{Count}(Y = y)}$$

# Categorical Naive Bayes Classifiers

- Learning given data  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ 
  - MLE for class prior:  $\hat{P}(Y = y) = \underline{\hat{p}_y} = \frac{\text{Count}(Y = y)}{n}$
  - MLE for distribution of feature  $i$ :  $\hat{P}(X_i = c \mid y) = \underline{\theta_{c|y}^{(i)}}$   
$$\theta_{c|y}^{(i)} = \frac{\text{Count}(X_i = c, Y = y)}{\text{Count}(Y = y)}$$

- Prediction given new point  $\mathbf{x}$ :

$$y = \arg \max_{y'} \hat{P}(y' \mid \mathbf{x}) = \arg \max_{y'} \hat{P}(y') \prod_{i=1}^d \hat{P}(x_i \mid y')$$

Categorical

$= \text{argmax}_{y'} \log \hat{p}_{y'} + \sum_{i=1}^d \log \theta_{x_i|y'}^{(i)}$

# Beyond categorical Naive Bayes

- Could in principle lift the Naive Bayes assumption by modeling joint (conditional) distribution of the features:  $P(X_1, \dots, X_d | Y)$
- What's the issue?

Need to  
Specify prob of  
each assignment:

$y$	$x_1$	$x_2$	$\dots$	$x_d$	$P(x_1 \dots x_d   y)$
0	0	0	...	0	0
0	0	1	...	0	1
0	1	0	...	1	6
1	0	0	...	0	1

- Requires exponentially (in  $d$ ) many parameters
  - Computationally intractable
  - Fantastic way to overfit!
- Remedy: Graphical models / Bayesian networks  
(see Probabilistic AI course)

## What if I have both discrete and continuous features?

- The (Naive) Bayes classifier does not require each feature to follow the same type of conditional distribution
- For example, can model some features as Gaussian, and some others as categorical
- Training (MLE) and prediction remains the same!

E.g.  $X_{1:10}$  discrete: for  $1 \leq i \leq 10$ ,  $P(X_i | y) = \text{Categorical}(x_i | y, \theta)$

$X_{11:20}$  Gaussian:  $P(x_i | y) = N(x_i; \mu_{iy}, \sigma_{iy}^2)$

$$P(X_{1:20} | y) = \prod_{i=1}^{10} \text{Categorical}(x_i | y, \theta) \prod_{i=11}^{20} N(x_i; \mu_{iy}, \sigma_{iy}^2)$$

# Avoiding overfitting

---

- So far we always used Maximum Likelihood Estimation -- didn't we say that's prone to overfitting?
- Can avoid overfitting by
  - Restricting model class (e.g., assumptions on covariance structure, e.g., Gaussian Naive Bayes) → fewer parameters
  - Using priors → „smaller“ parameters

## Prior over parameters (case c=2)

- As prior for our class probabilities, have assumed

$$P(Y = 1) = \theta$$

- Maximum likelihood estimate:  $\hat{\theta} = \frac{\text{Count}(Y = 1)}{n}$
- What happens in the extreme case  $n=1$ ?

$$\mathcal{D} = \{(y_1)\} \quad , \quad y_1 = 1$$

$$\Rightarrow \hat{\theta} = \frac{1}{1} = 1$$

## Prior over parameters (case c=2)

- As prior for our class probabilities, have assumed

$$P(Y = 1) = \theta$$

- Maximum likelihood estimate:  $\hat{\theta} = \frac{\text{Count}(Y = 1)}{n}$
- May want to put prior distribution  $P(\theta)$  and compute posterior distribution

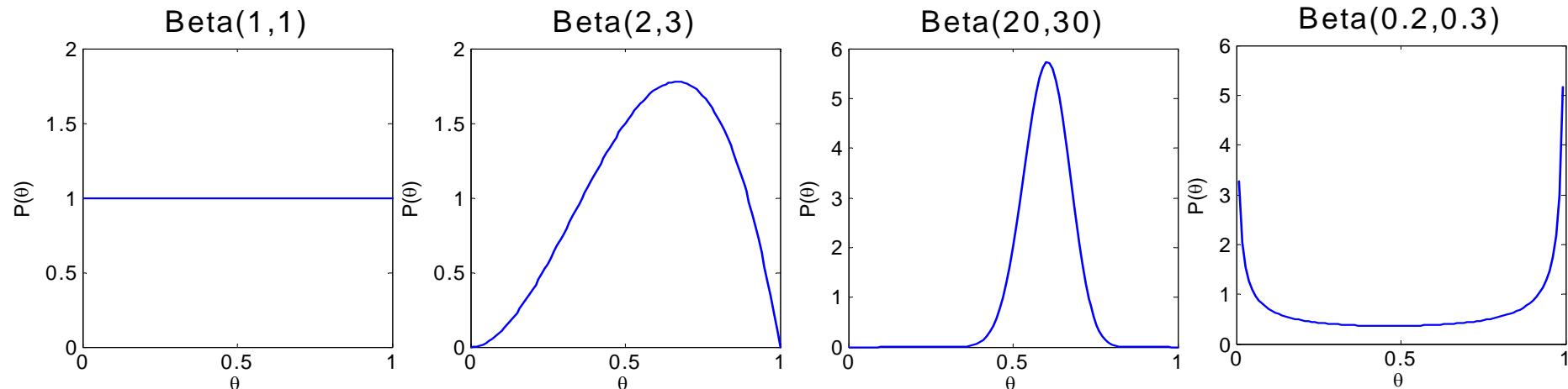
$$P(\theta | y_1, \dots, y_n) = \frac{1}{Z} P(\theta) \cdot P(y_{1:n} | \theta)$$

$$Z = \int P(\theta) P(y_{1:n} | \theta) d\theta$$

- In which cases can we do this efficiently?

# Beta prior over parameters

$$\text{Beta}(\theta; \alpha_+, \alpha_-) = \frac{1}{B(\alpha_+, \alpha_-)} \theta^{\alpha_+ - 1} (1 - \theta)^{\alpha_- - 1}$$



Assume obs data  $D$  with  $n_+$  times  $Y=+1$ ,  $n_-$  times  $Y=-1$

$$\begin{aligned} p(\theta | D) &= \frac{1}{Z} p(\theta) p(D|\theta) = \frac{1}{Z} \theta^{\alpha_+ - 1} (1-\theta)^{\alpha_- - 1} \cdot \theta^{n_+} \cdot (1-\theta)^{n_-} \\ &= \frac{1}{Z} \theta^{\alpha_+ + n_+ - 1} (1-\theta)^{\alpha_- + n_- - 1} = \text{Beta}(\theta; \alpha_+ + n_+, \alpha_- + n_-) \end{aligned}$$

# Conjugate distributions

- A pair of prior distributions and likelihood functions is called **conjugate** if the posterior distribution remains in the same family as the prior
- **Example:** Beta priors and Binomial likelihood
  - **Prior:**  $\text{Beta}(\theta; \alpha_+, \alpha_-)$
  - **Observations:** Suppose we observe  $n_+$  positive and  $n_-$  negative labels
  - **Posterior:**  $\text{Beta}(\theta; \alpha_+ + n_+, \alpha_- + n_-)$
- Thus  $\alpha_+, \alpha_-$  act as „**pseudo-counts**“
- **MAP estimate:**

$$\hat{\theta} = \arg \max_{\theta} P(\theta \mid y_1, \dots, y_n; \alpha_+, \alpha_-) = \frac{\alpha_+ + n_+ - 1}{\alpha_+ + n_+ + \alpha_- + n_- - 2}$$

# Conjugate priors

Prior / Posterior	Likelihood function
Beta	Bernoulli/Binomial
Dirichlet	Categorical/Multinomial
Gaussian (fixed covariance)	Gaussian
Gaussian-inverse Wishart	Gaussian
Gaussian process	Gaussian

- Can use conjugate priors as regularizers
- (Almost) no computational cost
- How to choose hyperparameters??
  - crossvalidation ☺

# Summary: Generative vs Discriminative

---

- **Discriminative models**

- Model  $P(y|x)$ . Do not attempt to model  $P(x)$
- Cannot detect outliers (property of  $P(x)$ )
- Are typically more robust, since accurately modeling  $x$  may be difficult

- **Generative models**

- Model joint distribution  $P(x,y)$
- This is a strictly more ambitious goal!
- Can be more powerful (e.g., detect outliers) if model assumptions are met
- Are typically less robust against outliers

# What you need to be able to do

---

- Apply (Gaussian / Categorical) Naive Bayes classifiers
- Relate different Gaussian Bayes classifiers
  - Naive Bayes
  - Fisher's LDA
  - General GBCs
- Use pseudocounts / conjugate priors as regularizers
- Compute distributions over features, and use them for outlier / anomaly detection

# Representation/ features

Linear hypotheses; nonlinear hypotheses with nonlinear feature transforms, kernels, learn nonlinear features via neural nets

## Paradigm:

### Discriminative vs. generative

## Probabilistic / Optimization Model:

Likelihood \* Prior  
Loss-function + Regularization

Squared loss = Gaussian lik., 0/1,  
Perceptron, Hinge, cost sensitive,  
multi-class hinge, reconstruction  
error, logistic loss=Bernoulli lik.,  
cross-entropy loss=Categorical lik.

$L^2$  norm (=Gaussian prior),  
 $L^1$  norm (=Laplace prior),  
early stopping, dropout  
Categorical;  
Beta/Dirichlet priors

## Method:

Exact solution, Gradient Descent, (mini-batch) SGD,  
Reductions, Bayesian model averaging

## Evaluation metric:

Mean squared error, Accuracy, F1 score, AUC,  
Confusion matrices, compression performance,  
log-likelihood on validation set

## Model selection:

K-fold Cross-Validation, Monte Carlo CV,  
Bayesian model selection