EECS 545: Machine Learning

Lecture 5. Linear models of classification & generative models

Honglak Lee

1/24/2011

Outline

- Recap: Linear models of classification
  - Discriminant functions
  - Logistic regression
- Exponential Family distribution
- Probabilisitic Generative models
  - Gaussian Discriminant Analysis
  - Naive Bayes

Classification

Classification

- The task of classification:
  - Given an input vector **x**, assign it to one of K distinct classes C_k where $k = 1, . . . K$
- Representing the assignment:
  - For K=2
    - Let t=1 mean that **x** is in C_1.
    - Let t=0 mean that **x** is in C_2.
  - For K>2,
    - Use 1-of-K coding, e.g., $\mathbf{t} = (0, 1, 0, 0, 0)^T$
      - (This would also work for K=2, of course.)

Learning the Classifier

- From input vectors x = $\{x_1, \ldots x_N\}$
  – and corresponding target values $\mathbf{t} = \{t_1, \ldots t_N\}$.
- 1. Discriminant functions

  Learn a function $y(\mathbf{x})$ that maps $\mathbf{x}$ onto some C_j.
- 2. Learn the distributions $p(C_k \mid \mathrm{x})$.

  (a) Learn model parameters from the training set.

  Discriminative models

  (b) Learn class densities $p(x \mid C_k)$ and priors $p(C_k)$

  Generative models

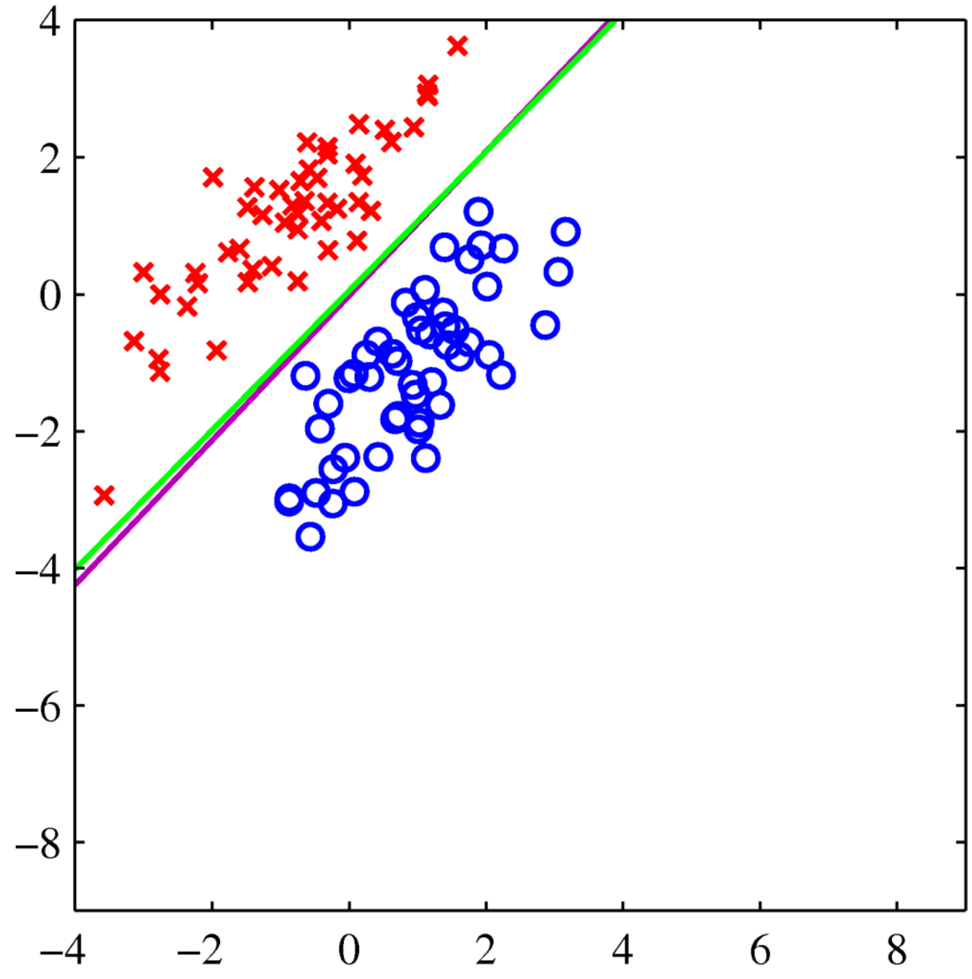
Discriminant functions

Discriminating two classes

- Specify a weight vector w and a bias w_0.

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

- Assign x to C_1 if
  $$y(\mathbf{x}) \geq 0$$
  – and to C_0 otherwise.
- How to pick w?



Fisher’s Linear Discriminant

- Use w to project x to one dimension.

$$\text{if } \mathbf{w}^T\mathbf{x} \geq -w_0 \text{ then } C_1 \text{ else } C_0$$

- Select w that best separates the classes.

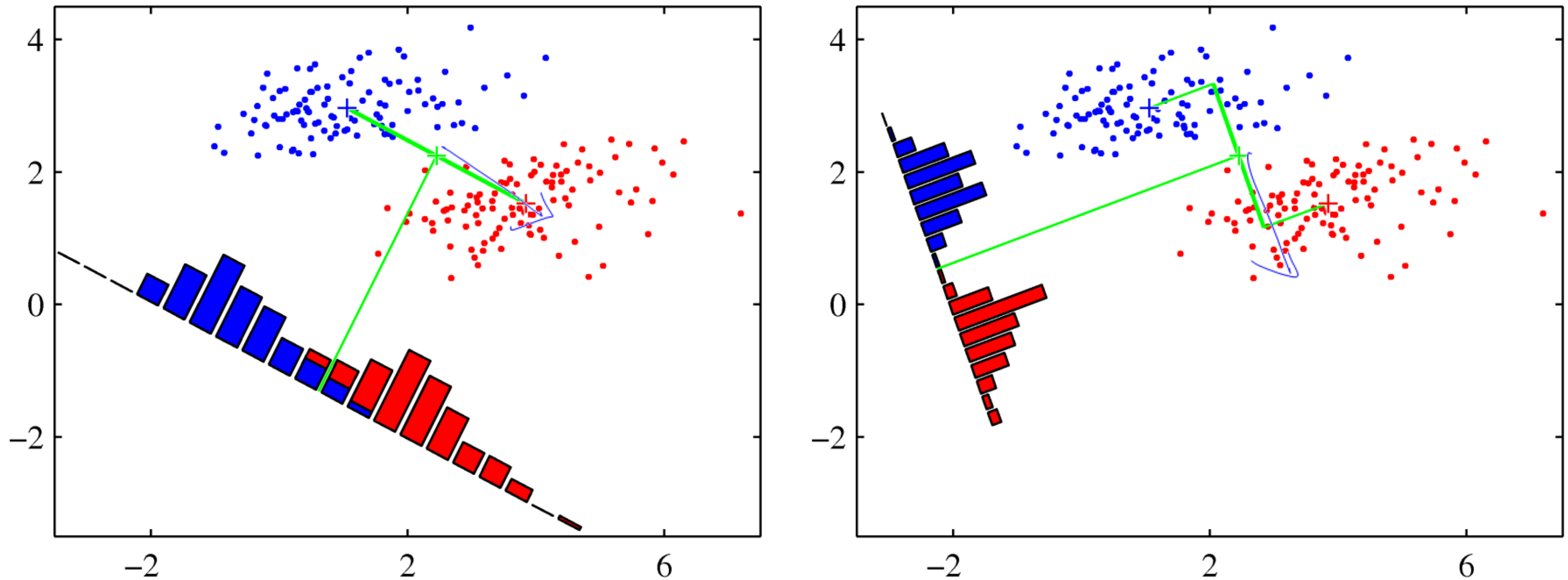
- What does that mean? Simultaneously,
  - Maximize class separation
  - Minimize class variances

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

Read Bishop book

Fisher's Linear Discriminant

- Maximizing separation alone doesn't work.
  - Minimizing class variance is a big help.



Read Bishop book

Objective function

- We want to maximize the “distance between classes”

$$m_2 - m_1 \equiv \mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)$$

- While minimizing the “distance within each class”

$$s_1^2 + s_2^2 \equiv \sum_{n \in C_1} (\mathbf{w}_1^T \mathbf{x}_n - m_k)^2 + \sum_{n \in C_2} (\mathbf{w}_2^T \mathbf{x}_n - m_k)^2$$

- Objective function: $$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

- Can be solved via eigenvalue problem.

Read Bishop book

Probabilistic discriminative models: logistic regression

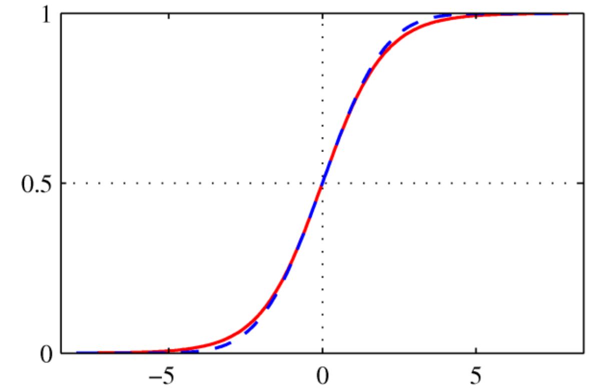
Main idea of probabilistic discriminative models

- Model decision boundary as a function of input x
  - Learn P(Ck|x) over data (e.g., maximum likelihood)
  - Directly predict class labels from inputs
- we will also cover probabilistic generative models
  - Learn P(Ck,x) over data (maximum likelihood) and then use Bayes’ rule to predict P(Ck|x)

Sigmoid and Logit functions

- The *logistic sigmoid* function is:

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$



- Its inverse is the *logit* function:

$$\ln \frac{P(y=1|x)}{P(y=0|x)} \quad a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

"log-odds"

- Generalizes to *normalized exponential*, or *softmax*.

$w_i^T x$

$\|$

w_i: vector for class i

$$p_i = \frac{\exp(q_i)}{\sum_j \exp(q_j)}$$

Likelihood function

- Depending on the label y, the likelihood of x is defined as:

$$
\begin{aligned}
P(t=1|x,w) &= \sigma(w^T\phi(x)) \\
P(t=0|x,w) &= 1-\sigma(w^T\phi(x))
\end{aligned}
$$

- Therefore:

$$
P(t|x,w) = \sigma(w^T\phi(x))^y \left(1-\sigma(w^T\phi(x))\right)^{1-y}
$$

- Likelihood of data: $\{\langle\phi(\mathbf{x}_n), t_n\rangle\}$ where $t_n \in \{0,1\}$

$$
p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n}
$$

Logistic Regression

- For a data set $\{\langle\phi(\mathbf{x}_n), t_n\rangle\}$ where $t_n \in \{0, 1\}$
- the likelihood function is

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n}$$

- where

$$y_n = p(C_1|\phi(\mathbf{x}_n)) = \sigma(\mathbf{w}^T \phi(\mathbf{x}_n))$$

- Define an error function $E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w})$
  - (Minimizing *E*(**w**) maximizes likelihood.)

Logistic Regression: gradient descent

- Taking the gradient of *E*(w) gives us

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi(\mathbf{x}_n)$$

  – Recall

$$y_n = p(C_1|\phi(\mathbf{x}_n)) = \sigma(\mathbf{w}^T\phi(\mathbf{x}_n))$$

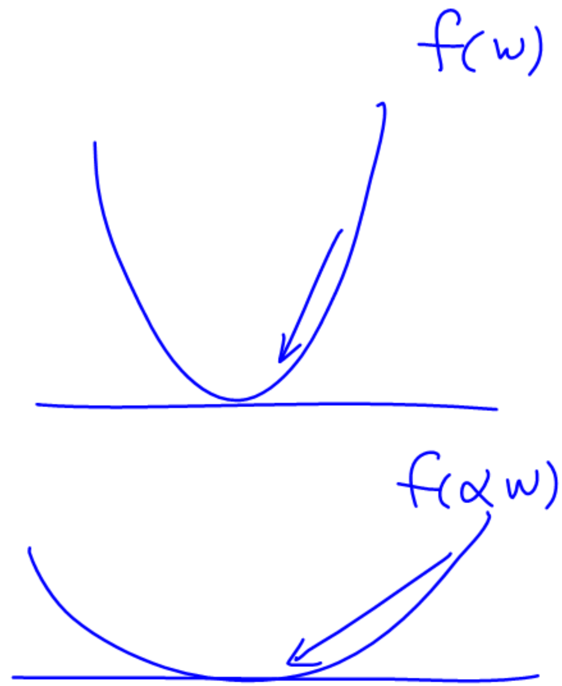
    - This is essentially the same gradient expression that appeared in linear regression with least-squares.
    - Note the error term between model prediction and target value: $\sigma(\mathbf{w}^T\phi(\mathbf{x}_n)) - t_n$

Newton's method

- Goal: Minimizing a general function $l(w)$ (one dimensional case)
  - Approach: solve for $f(w) = \frac{\partial l(w)}{\partial w} = 0$
  - So, how to solve this problem?
- Newton's method:
  - Repeat until convergence:

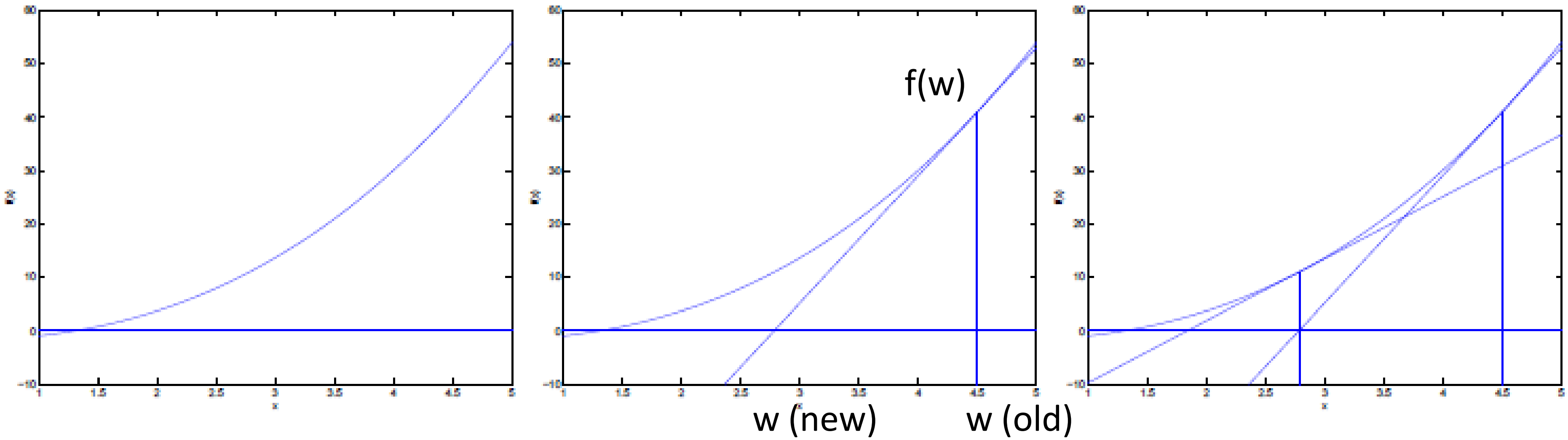
$$w := w - \frac{f(w)}{f'(w)}$$

$\alpha = 0.1$



Newton's method

- Interatively solve until we get $f(w) = 0$.



- Geometric intuition

$$w := w - \frac{f(w)}{f'(w)}$$

Current value

"Slope"

0.

Newton's method

- Convering $l'(w) = f(w)$
  - Repeat until convergence:

$$w := w - \frac{l'(w)}{l'(w)}$$

- This method can be also extended for multivariate case:

$$w := w - H^{-1} \nabla_w l$$

gradient

where H is a Hessian matrix

$$H_{ij}(w) = \frac{\partial^2 l(w)}{\partial w_i \partial w_j}$$

Note: We already did this for least squares problem!

Logistic Regression

- For linear regression, least-squares has a closed-form solution:

$$\mathbf{w}_{ML} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}$$

- Generalizes to weighted-least-squares with an *N*x*N* diagonal weight matrix R.

$$\mathbf{w}_{WLS} = (\mathbf{\Phi}^T\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{R}\mathbf{t}$$

- But, because $\nabla E(\mathbf{w}) = 0$ is non-linear,
- there is no exact solution. Must iterate.

Iterative Solution

- Apply Newton-Raphson method to iterate to a solution w to $\nabla E(\mathbf{w}) = 0$
- This involves least-squares with weights R:

$$R_{nn} = y_n(1 - y_n)$$

- Since R depends on w (and vice versa), we get *iterative reweighted least squares* (IRLS)

$$\mathbf{w}^{(new)} = (\mathbf{\Phi}^T \mathbf{R} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{R} \mathbf{z}$$

– where

$$\mathbf{z} = \mathbf{\Phi} \mathbf{w}^{(old)} - \mathbf{R}^{-1}(\mathbf{y} - \mathbf{t})$$

Bayesian Logistic Regression

- Possible, but computationally intractable.
  - Likewise the predictive distribution.
- The Laplace Approximation is helpful.
  - Given a distribution $p(z)$, take the Taylor series of $\ln p(z)$, at a point (the mode) where the linear term vanishes.
  - Use the quadratic term to define a Gaussian.

Exponential family distributions

Motivation

- We considered a binary classification problem where P(y|x) is a Bernoulli distribution
- We are interested in more general distribution
  - E.g., integer variables $y \in \{0,1,2,\ldots,\infty\}$
  - E.g., multinomial variables $y \in \{0,1,2,\ldots,K\}$
  - Q. is there a general way of parameterizing these distributions?
- Approach: exponential family distribution

Exponential family distribution

- Exponential family distribution

$$p(x|\eta) \quad = \quad h(\mathbf{x})g(\eta)\exp(\eta^T\mathbf{u}(\mathbf{x}))$$

  - η: natural parameters
  - x: data
  - u(x): sufficient statistic

The Exponential Family

- Distribution

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

  - η: natural parameters
  - x: data
  - u(x): sufficient statistic
- Normalization:

$$g(\boldsymbol{\eta}) \underline{\int h(\mathbf{x}) \exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\} \mathrm{d}\mathbf{x}} = 1$$

  - so g(η) can be interpreted as a normalization coefficient.

The Exponential Family (2.1)

- The Bernoulli Distribution

$$\begin{cases} P(x=1|\mu) = \mu \\ P(x=0|\mu) = 1-\mu \end{cases}$$

$$\begin{aligned} p(x|\mu) &= \mathrm{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x} \\ &= \exp\{x \ln \mu + (1-x)\ln(1-\mu)\} \\ &= (1-\mu)\exp\left\{\ln\left(\frac{\mu}{1-\mu}\right)x\right\} \end{aligned}$$

$x \in \{0,1\}$

$x[\ln \mu - \ln(1-\mu)]$

$+ \ln(1-\mu)$

η

- Comparing with the general form we see that

$$\eta = \ln\left(\frac{\mu}{1-\mu}\right) \quad \text{and so} \quad \underbrace{\mu = \sigma(\eta) = \frac{1}{1+\exp(-\eta)}}_{\text{Logistic sigmoid}}.$$

log-odds

The Exponential Family (2.2)

- The Bernoulli distribution can hence be written as

$$p(x|\eta) = \boxed{\sigma(-\eta)}\exp(\eta x)$$

$$\sigma(-\eta) = \frac{1}{1+\exp(\eta)} \qquad \exp(\eta x) = \begin{cases} \dfrac{\exp(\eta)}{1+\exp(\eta)} = \sigma(\eta) & x=1 \\ \dfrac{1}{1+\exp(\eta)} = 1-\sigma(\eta) & x=0 \end{cases}$$

- where

$$\begin{aligned} u(x) &= x \\ h(x) &= 1 \\ g(\eta) &= 1-\sigma(\eta) = \sigma(-\eta). \end{aligned}$$

$$\sigma(\eta) \qquad 1-\sigma(\eta) = \sigma(-\eta)$$

The Exponential Family (3.1)

$x = [x_1, x_2, .., x_M]$

- The Multinomial Distribution

$x=1 \Leftrightarrow [1, 0, \dots, 0]$

$x=j \Leftrightarrow [0, 0 \dots, 1, 0 \dots 0]$ ← j-th coord

$\mu_1+\mu_2+\cdots+\mu_M = 1$

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{M} \mu_k^{x_k} = \exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} = h(\mathbf{x})g(\boldsymbol{\eta})\underbrace{\exp\left(\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right)}$$

$\Leftrightarrow p(x_j = 1|\mu) = \mu_j$

- where, $\mathbf{x} = (x_1, \dots, x_M)^{\mathrm{T}}$, $\boldsymbol{\eta} = (\eta_1, \dots, \eta_M)^{\mathrm{T}}$ and

$$\begin{aligned} \eta_k &= \ln \mu_k \\ \mathbf{u}(\mathbf{x}) &= \mathbf{x} \\ h(\mathbf{x}) &= 1 \\ g(\boldsymbol{\eta}) &= 1. \end{aligned}$$

NOTE: The μ_k parameters are not independent since the corresponding μ_k must satisfy

$$\sum_{k=1}^{M} \mu_k = 1.$$

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_M \end{bmatrix} = \begin{bmatrix} \ln \mu_1 \\ \ln \mu_2 \\ \vdots \\ \ln \mu_M \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}$$

The Exponential Family (3.2)

$\mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{M-1} \end{bmatrix}$

- Let $\mu_M = 1 - \sum_{k=1}^{M-1} \mu_k$. This leads to

$$\eta_k = \ln\left(\frac{\mu_k}{\underbrace{1 - \sum_{j=1}^{M-1} \mu_j}_{\mu_M}}\right) \text{ and } \mu_k = \underbrace{\frac{\exp(\eta_k)}{1 + \sum_{j=1}^{M-1} \exp(\eta_j)}}_{\text{Softmax}}.$$

- Here the μ_k parameters are independent. Note that

$$0 \leqslant \mu_k \leqslant 1 \text{ and } \sum_{k=1}^{M-1} \mu_k \leqslant 1.$$

The Exponential Family (3.3)

- The Multinomial distribution can then be written as

$$p(\mathbf{x}|\boldsymbol{\mu}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left(\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right)$$

- where

$$\begin{aligned}
\boldsymbol{\eta} &= (\eta_1, \ldots, \eta_{M-1}, 0)^{\mathrm{T}} \\
\mathbf{u}(\mathbf{x}) &= \mathbf{x} \\
h(\mathbf{x}) &= 1 \\
g(\boldsymbol{\eta}) &= \left(1 + \sum_{k=1}^{M-1} \exp(\eta_k)\right)^{-1}.
\end{aligned}$$

The Exponential Family (4)

- The Gaussian Distribution

$$
\begin{aligned}
p(x|\mu,\sigma^2) &= \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}\mu^2\right\} \\
&= h(x)g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(x)\right\}
\end{aligned}
$$

- where

$$\boldsymbol{\eta} = \begin{pmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{pmatrix} \qquad h(\mathbf{x}) = (2\pi)^{-1/2}$$

$$\mathbf{u}(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix} \qquad g(\boldsymbol{\eta}) = (-2\eta_2)^{1/2} \exp\left(\frac{\eta_1^2}{4\eta_2}\right).$$

ML for the Exponential Family (1)

- From the definition:

$$\nabla_{\eta} \left[\underline{g(\boldsymbol{\eta})} \int h(\mathbf{x}) \underline{\exp\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\}}\, \mathrm{d}\mathbf{x} \right] = 1 \quad = 0$$

∇_{η} (*)

$$= \mathbb{E}[u(x)]$$

$$\int p(x|\eta)\, u(x)\, dx$$

  – Taking derivative w.r.t. eta:

$$\Rightarrow \underbrace{\nabla g(\boldsymbol{\eta}) \int h(\mathbf{x}) \exp\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\}\, \mathrm{d}\mathbf{x}}_{1/g(\boldsymbol{\eta})} + \underbrace{g(\boldsymbol{\eta}) \int h(\mathbf{x}) \exp\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\} \boxed{\mathbf{u}(\mathbf{x})}\, \mathrm{d}\mathbf{x}}_{\mathbb{E}[\mathbf{u}(\mathbf{x})]} = 0$$

(*)

- Thus

$$\underline{-\nabla \ln g(\boldsymbol{\eta}) = \mathbb{E}[\mathbf{u}(\mathbf{x})]}$$

ML for the Exponential Family (2)

- Given a data set, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ the likelihood function is given by

$$\boxed{p(\mathbf{X}|\boldsymbol{\eta})} = \left(\prod_{n=1}^{N} h(\mathbf{x}_n)\right) \boxed{g(\boldsymbol{\eta})}^{N} \exp\left\{\boldsymbol{\eta}^{\mathrm{T}} \sum_{n=1}^{N} \mathbf{u}(\mathbf{x}_n)\right\}.$$

- Thus we have (by taking gradient w.r.t. eta)

$$-\nabla \ln g(\boldsymbol{\eta}_{\mathrm{ML}}) = \underbrace{\frac{1}{N} \sum_{n=1}^{N} \mathbf{u}(\mathbf{x}_n)}_{\text{Sufficient statistic}}$$

prior of η

$p(\eta)$

Conjugate priors

- For any member of the exponential family, there exists a prior

$$\text{"}P(\eta) = \text{"} \quad p(\boldsymbol{\eta}|\boldsymbol{\chi}, \nu) = f(\boldsymbol{\chi}, \nu)\boxed{g(\boldsymbol{\eta})}^{\nu} \exp\left\{\nu \boldsymbol{\eta}^{\mathrm{T}} \boldsymbol{\chi}\right\}.$$

- Combining with the likelihood function, we get

$$p(\boldsymbol{\eta}|\mathbf{X}, \boldsymbol{\chi}, \nu) \propto g(\boldsymbol{\eta})^{\boxed{\nu+N}} \exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\left(\sum_{n=1}^{N} \mathbf{u}(\mathbf{x}_n) + \boxed{\nu\boldsymbol{\chi}}\right)\right\}.$$

Prior corresponds to ν “pseudo-observations” with value χ.

Exponential Family distribution and Generalized Linear Models (GLMs)

- Intuition: we want to model the exponential family distribution P(y) by parameterizing by $\eta = w^T x$.

$P(y|\eta) = P(y|w^T x)$

- From exponential distribution, the prediction function is $E[y|\eta] = E[y|w^T x]$.
- Terminology:
  - Canonical response function: $E[y|\eta] = E[y|w^T x]$

Examples of GLMs: Logistic regression

- From Bernoulli distribution

$$P(y|\eta) = \sigma(-\eta)\exp(\eta y)$$

$$\begin{aligned} u(y) &= y \\ h(y) &= 1 \\ g(\eta) &= 1 - \sigma(\eta) = \sigma(-\eta) \end{aligned}$$

$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(N)}, y^{(N)})\}$

- With $\eta = w^T x$, we have:

$$P(y|\boxed{w^T x}) = \sigma(-w^T x)\exp(y w^T x) = \begin{cases} \sigma(w^T x) & \text{if } y = 1 \\ \sigma(-w^T x) = 1 - \sigma(w^T x) & \text{if } y = 0 \end{cases}$$

- Canonical response function: $E[y|w^T x] = \sigma(w^T x)$.

Probabilistic generative models

Learning the Classifier

- Goal: Learn the distributions $p(C_k \mid \mathrm{x})$.
  (a) Learn model parameters from the training set: i.e., try to predict $p(C_k \mid \mathrm{x})$ directly from x

  Discriminative models

  (b) Learn class densities $p(x \mid C_k)$ and priors $p(C_k)$

  Generative models

Comparing the Approaches

- The *generative* approach is model-based, and makes it possible to generate synthetic data from $p(\mathrm{x} \mid C_k)$.
  - Training data to estimate $p(\mathbf{x} \mid C_k)$ may be easier to find.

- The *discriminative* approach will typically have fewer parameters to estimate.
  - Linear versus quadratic in the dimension of the input.

Probabilistic Generative Models

- Bayes' theorem reduces the classification problem $p(C_k \mid \mathrm{x})$ to simpler problems . . .

- Density estimation problems are easy to learn from labeled training data.
  - $p(C_k)$
  - $p(\mathbf{x} \mid C_k)$
- Maximum likelihood parameter estimation.

Probabilistic Generative Models

- For two classes, Bayes' theorem says:

$$\underline{p(C_1|\mathbf{x})} = \frac{p(\mathbf{x}|C_1)p(C_1)}{\underbrace{p(\mathbf{x}|C_1)p(C_1)}_{p(x,C_1)} + \underbrace{p(\mathbf{x}|C_2)p(C_2)}_{p(x,C_2)}}$$

$p(\mathbf{x}|C_1)p(C_1) = p(x, C_1)$

$p(x,C_1) + p(x,C_2) = p(x)$

- Use *log odds*:

$$a = \ln \frac{p(C_1|\mathbf{x})}{p(C_2|\mathbf{x})} = \ln \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)} = \frac{p(x,C_1)}{p(x,C_2)}$$

- To define the posterior via the *sigmoid*:

$$p(C_1|\mathbf{x}) = \frac{1}{1+\exp(-a)} = \sigma(a)$$

Gaussian Discriminative Models

Gaussian Discriminant Analysis

- Prior distribution
  - $p(C_k)$: Constant (e.g., Bernoulli)

$$\begin{cases} P(C_1) = \phi \\ P(C_2) = 1-\phi \end{cases}$$

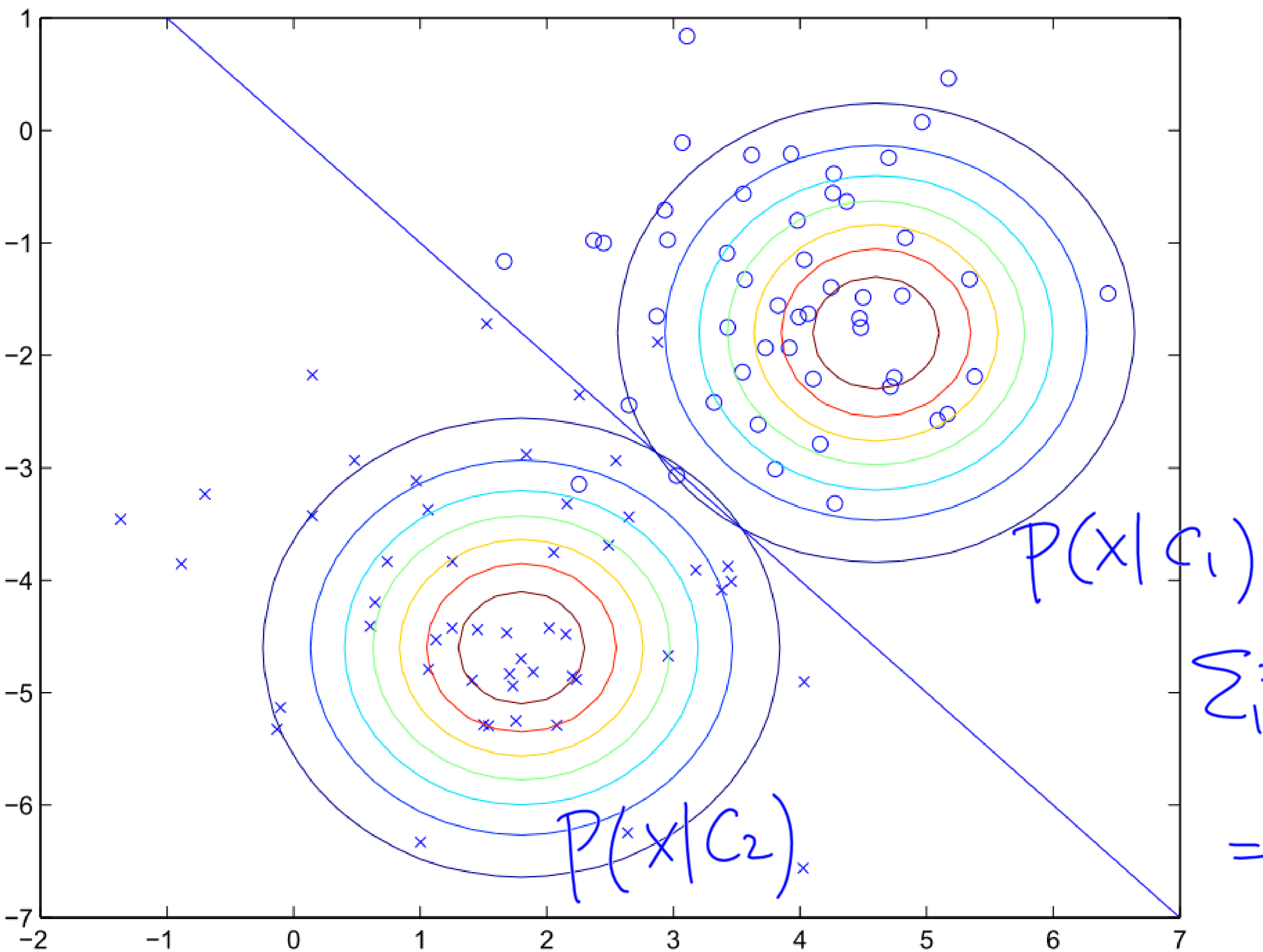
- Likelihood
  - $P(x|C_k)$: Gaussian distribution

$$\begin{cases} P(x|C_1) = N(\mu_1, \Sigma) \\ P(x|C_2) = N(\mu_2, \Sigma) \end{cases}$$

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\mu_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\mu_k) \right\}$$

- Classification: use Bayes rule (previous slide)

Gaussian Discriminant Analysis



$\Sigma_1 = \Sigma_2$

$= I_2$

$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Class-Conditional Densities

- Suppose we model $p(\mathrm{x} \mid C_k)$ as Gaussians with the **<u>same covariance</u>** matrix.

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \mu_k)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu_k) \right\}$$

- This gives us $p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$
- where

$$\mathbf{w} = \mathbf{\Sigma}^{-1}(\mu_1 - \mu_2)$$

- and

$$w_0 = -\frac{1}{2}\mu_1^T \mathbf{\Sigma}^{-1} \mu_1 + \frac{1}{2}\mu_2^T \mathbf{\Sigma}^{-1} \mu_2 + \ln \frac{p(C_1)}{p(C_2)}$$

Derivation

$P(C_1|x) = \sigma(a)$

log-odds

$$
\begin{aligned}
P(x, C_1) &= P(x|C_1)P(C_1) \\
&= \cancel{\frac{1}{(2\pi)^{D/2}}} \cancel{\frac{1}{|\Sigma|^{1/2}}} \exp\left\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right\} P(C_1) \\
P(x, C_2) &= P(x|C_2)P(C_2) \\
&= \cancel{\frac{1}{(2\pi)^{D/2}}} \cancel{\frac{1}{|\Sigma|^{1/2}}} \exp\left\{-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\right\} P(C_2)
\end{aligned}
$$

$a =$

$$\boxed{\log \frac{P(C_1|x)}{P(C_2|x)}} = \log \frac{P(C_1|x)}{1 - P(C_1|x)}$$

"Log-odds"

$-\frac{1}{2} x^T \Sigma^{-1} x$

$$
\begin{aligned}
&= \log \frac{\exp\left\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right\}}{\exp\left\{-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\right\}} + \log\frac{P(C_1)}{P(C_2)} \\
&= \left\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right\} - \left\{-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\right\} + \log\frac{P(C_1)}{P(C_2)} \\
&= (\mu_1 - \mu_2)^T \Sigma^{-1} x - \frac{1}{2}\mu_1\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2\Sigma^{-1}\mu_2 + \log\frac{P(C_1)}{P(C_2)} \\
&= \underbrace{\boxed{\left(\Sigma^{-1}(\mu_1 - \mu_2)\right)}}_{w}{}^T x + w_0
\end{aligned}
$$

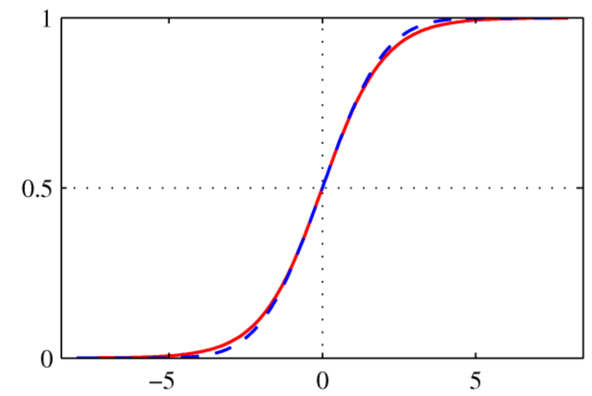
constant wrt x.

where $w_0 = -\frac{1}{2}\mu_1\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2\Sigma^{-1}\mu_2 + \log\frac{P(C_1)}{P(C_2)}$

Sigmoid and Logit functions

- The *logistic sigmoid* function is:

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$



  – with log-odds (*logit* function):

$$a = \log\left(\frac{\sigma}{1-\sigma}\right) = \left(\Sigma^{-1}(\mu_1 - \mu_2)\right)^T x + w_0$$

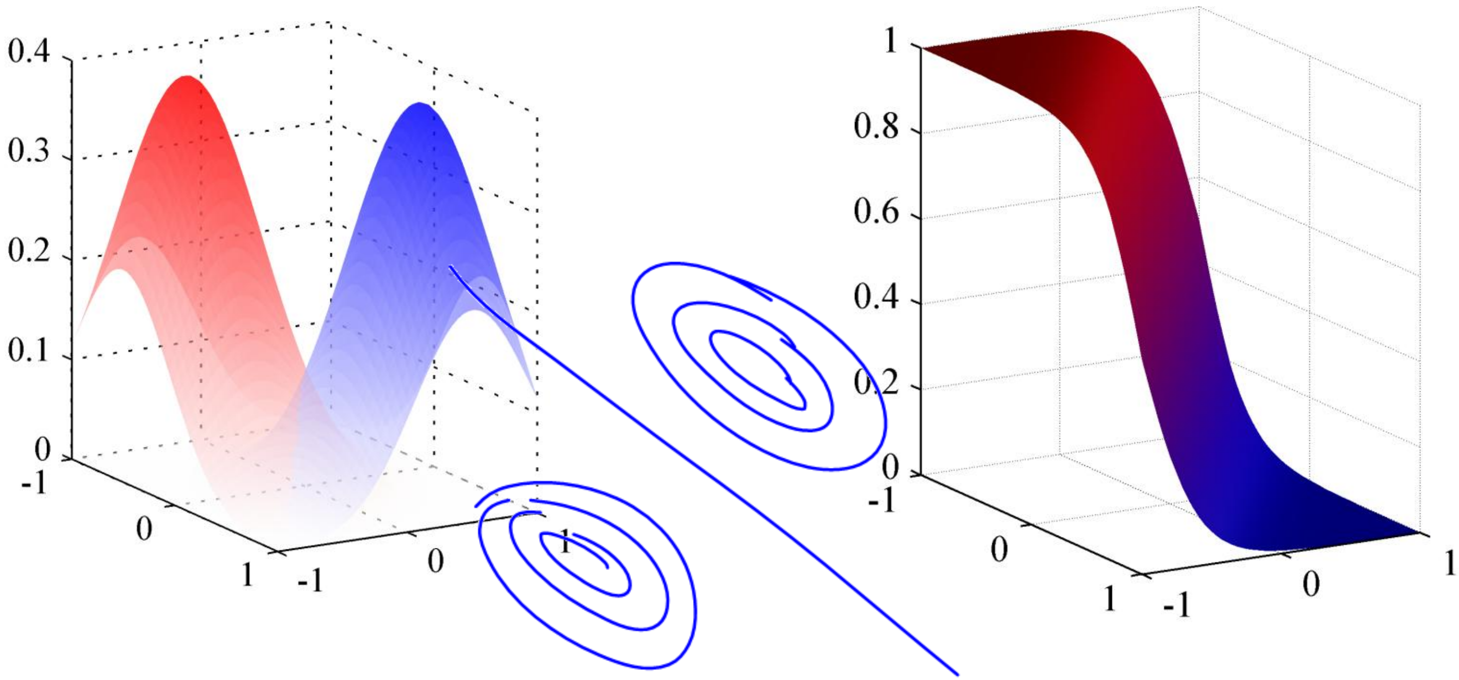
$$\text{where } w_0 = -\tfrac{1}{2}\mu_1\Sigma^{-1}\mu_1 + \tfrac{1}{2}\mu_2\Sigma^{-1}\mu_2 + \log\frac{P(C_1)}{P(C_2)}$$

- Generalizes to *normalized exponential*, or *softmax*.

$$p_i = \frac{\exp(q_i)}{\sum_j \exp(q_j)}$$

Linear Decision Boundaries

- With the same covariance matrices, the boundary $p(C_1 \mid x) = p(C_2 \mid x)$ is linear.
  - Different priors $p(C_1)$, $p(C_2)$ just shift it around.



Learning via maximum likelihood

- Given training data $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$, and a generative model ("shared covariance")

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{\frac{1}{2}} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1} (x-\mu_0))$$

$$p(x|y=1) = \frac{1}{(2\pi)^{\frac{1}{2}} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1} (x-\mu_1))$$

Learning via maximum likelihood

- Maximum likelihood estimation is (homework):

$$\phi = \frac{1}{N}\sum_{i=1}^{N} 1\{y_i = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{N} 1\{y_i = 0\} x_i}{\sum_{i=1}^{N} 1\{y_i = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{N} 1\{y_i = 1\} x_i}{\sum_{i=1}^{N} 1\{y_i = 1\}}$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu_{y_i})(x_i - \mu_{y_i})^T.$$

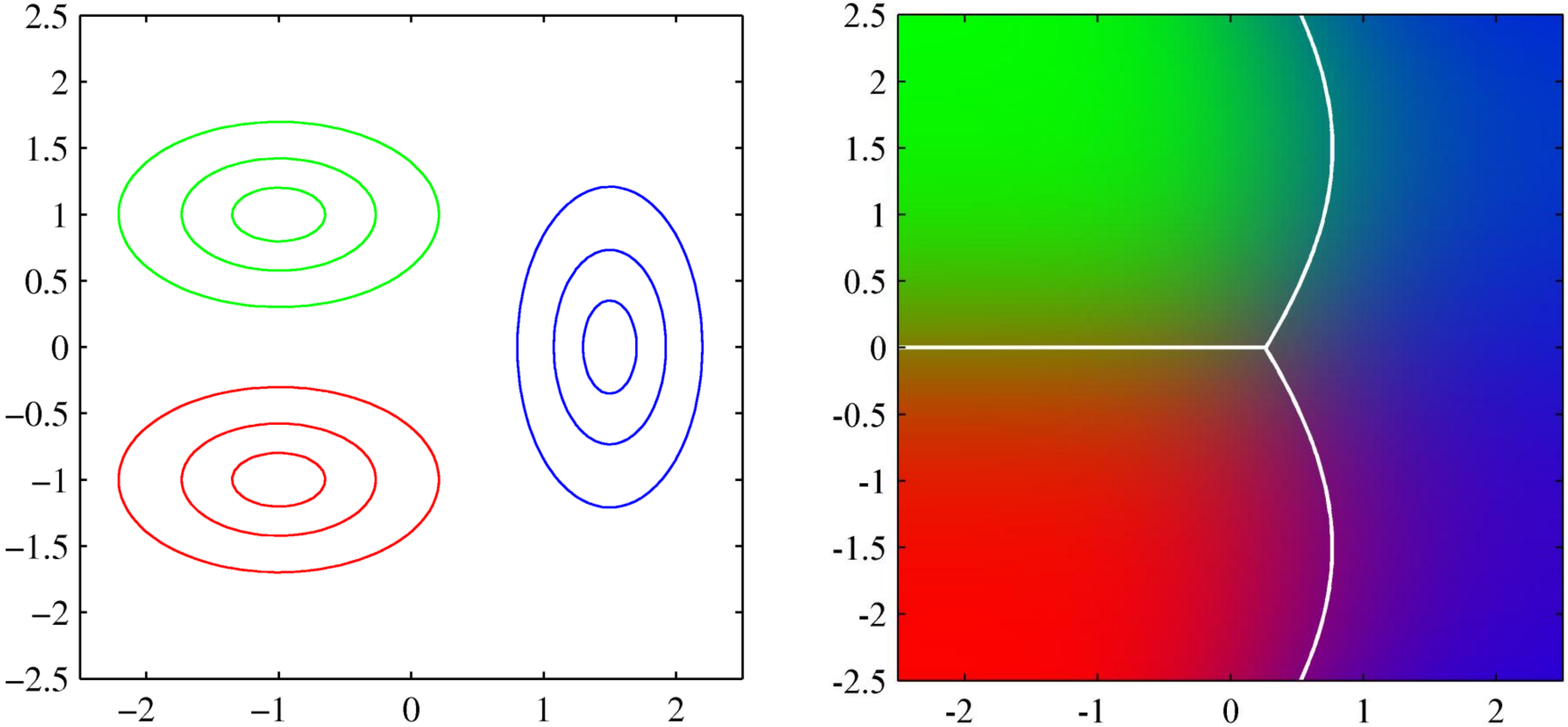
1. write down log-likelihood

2. Take derivatives

$$\begin{cases} \frac{\partial l}{\partial \phi} = 0 \\ \nabla_{\mu_0} l = 0 \\ \nabla_{\mu_1} l = 0 \\ \nabla_{\Sigma^{-1}} l = 0 \end{cases}$$

With Different Covariances

- Decision boundaries can be quadratic.



Comparison between GDA and Logistic regression

- Logistic regression:
  - For an M-dimensional feature space, this model has M parameters to fit.
- Gaussian Discriminative Analysis
  - 2M parameters for the means of $p(\mathbf{x} \mid C_1)$ and $p(\mathbf{x} \mid C_2)$
  - M(M+1)/2 parameters for the shared covariance matrix
- Logistic regression is has less parameters and is more flexible.
- GDA has a stronger modeling assumption, and works well when the distribution follows the assumption.

Naive Bayes classifier

Naive Bayes classifier

- Prior distribution
  - $p(C_k)$: Constant (e.g., Bernoulli)
- Likelihood

$x = [x_1, \dots, x_M]$

  - $P(x|C_k)$: factorized distribution (All x_j's are conditionally independent given the class label C_k.)

$$P(x_1, ..., x_M|C_k) = P(x_1|C_k) \cdots P(x_M|C_k) = \prod_{j=1}^{M} P(x_j|C_k)$$

- Classification: use Bayes rule

$P(C_1|x) = \sigma(a)$

$$P(C_1|x) \quad = \quad \frac{P(C_1, x)}{P(x)} = \frac{P(C_1, x)}{P(C_1, x) + P(C_2, x)}$$

$P(x) = P(C_1, x) + P(C_2, x)$

Equivalently, log-odds: $a = \dfrac{P(C_1|x)}{P(C_2|x)} = \dfrac{P(C_1) \prod_{j=1}^{M} P(x_j|C_1)}{P(C_2) \prod_{j=1}^{M} P(x_j|C_2)}$

Naive Bayes classifier

- When classifying, we can simply take the MAP (Maximum a Posteriori) estimation:

$$\arg\max_k P(C_k|x) = \arg\max_k P(C_k, x)$$

$$\parallel$$

$$\frac{P(C_k, x)}{P(x)} = \arg\max_k P(C_k) \prod_{j=1}^{M} P(x_j|C_k)$$

constant wrt k.

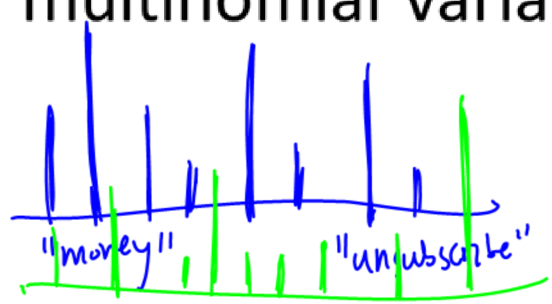
Example of Naive Bayes classifier

$X^{(i)} =$ word1 word2 ... word$_n$

$\{1,\dots,|V|\}$ " "

- Spam mail classification
- y=1 (spam), y=0 (non-spam)
- x_j :j-th word in the document $\in \{1, \dots, |V|\}$, where V is the vocabulary (of size n).
  - Multinomial variable = $\{0,1\}^{|V|}$ $= [1, 0, \dots, 0, 0]$
    $[0, 1, \dots, 0, 0]$
- Naive Bayes Assumption:
  - Given a class label y, each word in a document is a independent multinomial variable

$\sum \mu_k = 1$



Naive Bayes Spam classifier

- P(spam) = Bernoulli(ϕ)
- P(word|spam)=multinomial$(\mu_1^s, \ldots, \mu_N^s)$
- P(word|nonspam)=multinomial$(\mu_1^{ns}, \ldots, \mu_N^{ns})$
- Likelihood

$$
\begin{aligned}
& \prod_{i=1}^{N} P(x^{(i)}, y^{(i)}) \\
= & \prod_{i=1}^{N} P(x^{(i)}|y^{(i)})P(y^{(i)}) \\
= & \left(\prod_{i:y^{(i)}=1} P(x^{(i)}|y^{(i)})P(y^{(i)}) \right) \left(\prod_{i:y^{(i)}=0} P(x^{(i)}|y^{(i)})P(y^{(i)}) \right) \\
= & \left(\phi^{\boxed{N^{spam}}} \prod_{word\,j} (\mu_j^s)^{\boxed{N_j^{spam}}} \right) \left((1-\phi)^{\boxed{N^{nonspam}}} \prod_{word\,j} (\mu_j^{ns})^{\boxed{N_j^{nonspam}}} \right)
\end{aligned}
$$

Maximum likelihood estimation

- Log-likelihood

$$\begin{cases} \frac{\partial}{\partial \phi} l = 0 \\ \frac{\partial}{\partial \mu_j^s} l = 0 \\ \frac{\partial}{\partial \mu_j^{ns}} l = 0 \end{cases}$$

$$\begin{aligned} l &= \log \prod_{i=1}^{N} P(x^{(i)}, y^{(i)}) \\ &= N^{spam} \log \phi + \sum_{word\, j} N_j^{spam} \log \mu_j^s + N^{nonspam} \log(1-\phi) + \sum_{word\, j} N_j^{nonspam} \log \mu_j^{ns} \end{aligned}$$

- Maximum-likelihood
  - Take the derivative of log-likelihood w.r.t. the parameters, and set it to zero.

Maximum likelihood estimation

- From $$\frac{\partial l}{\partial \phi} = \frac{1}{\phi} N^{spam} - \frac{1}{1-\phi} N^{nonspam} = 0$$
  - We get $$\phi = \frac{N^{spam}}{N^{spam} + N^{nonspam}}$$

- Make the parameters μ independent:

$$\sum_{word\, j=1}^{M} N_j^{spam} \log \mu_j^s = \sum_{word\, j=1}^{M-1} N_j^{spam} \log \mu_j^s + N_M^{spam} \log(1 - \sum_{j=1}^{M-1} \mu_j^s)$$

$$\frac{\partial}{\partial \mu_j^s} \left(\sum_{word\, j=1}^{M} N_j^{spam} \log \mu_j^s \right) = \frac{N_j^{spam}}{\mu_j^s} - \frac{N_M^{spam}}{1 - \sum_{j=1}^{M-1} \mu_j^s} = 0$$

$$\boxed{\frac{N_j^{spam}}{\mu_j^s} = constant, \forall j}$$

  - We finally get $$\boxed{\mu_j^s = \frac{N_j^{spam}}{\sum_j N_j^{spam}}}$$

Maximum likelihood estimation

- Summary:

$$P(spam) = \phi = \frac{N^{spam}}{N^{spam} + N^{nonspam}}$$

$$P(word = j|spam) = \mu_j^s = \frac{N_j^{spam}}{\sum_j N_j^{spam}}$$

$$P(word = j|non - spam) = \mu_j^{ns} = \frac{N_j^{nonspam}}{\sum_j N_j^{nonspam}}$$

Laplace smoothing

- Main intuition: Put “imaginary” counts for each words
  - prevent zero probability estimates (overfitting)!
- E.g.: Adding “1” as imaginary count for each word

$$P(spam) = \phi = \frac{N^{spam}}{N^{spam} + N^{nonspam}}$$

$$P(word = j|spam) = \mu_j^s = \frac{N_j^{spam} + 1}{\sum_j N_j^{spam} + M}$$

$$P(word = j|non - spam) = \mu_j^{ns} = \frac{N_j^{nonspam} + 1}{\sum_j N_j^{nonspam} + M}$$

Next class

- Kernel methods