Machine Learning

4771

Instructors:
Adrian Weller and Ilia Vovsha

Lecture 3+4: Parametric Approaches to Statistical Inference

- Bayesian Decision Theory (Duda 2.1-2.4)
- Gaussian Distribution (Duda 2.5)
- Classification with Gaussians (Duda 2.6)
- Regression
- Polynomial Approximation (Bishop 1.1)
- Application to text classification.
- Multinomial and bag-of-words models.

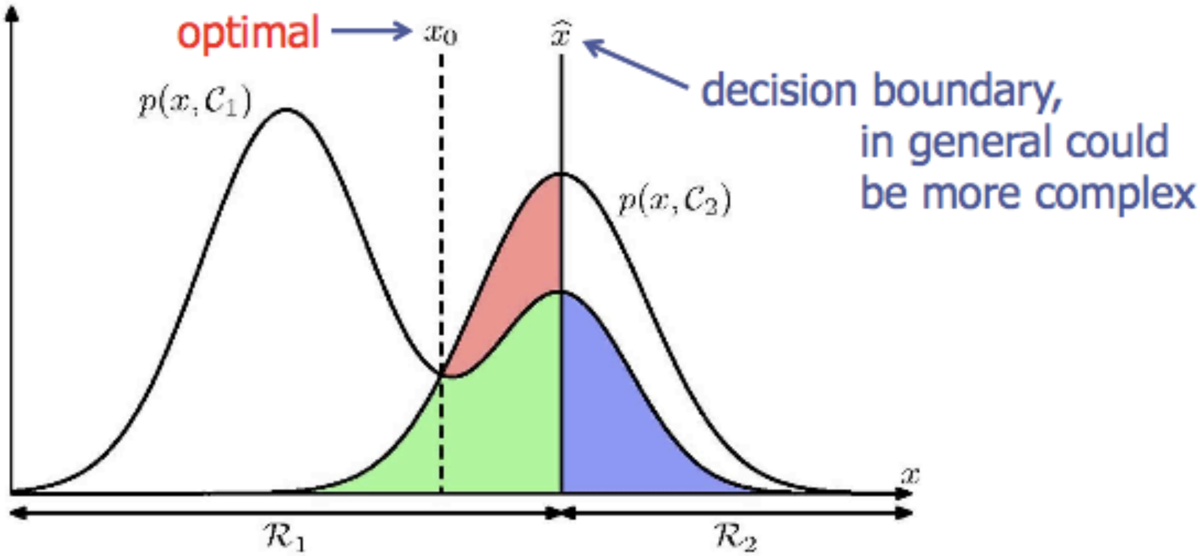
Bayesian Decision Theory

- Previously: dealing with uncertainty
- Assumption: probability values are known
- Given input & labels, obtain pdf
  - p(x) from $\mathcal{D}$
  - Difficult problem!
- Instead lets do prediction:
  - prediction ≈ decision (action)
  - Decision is trivial after inference

Bayesian Decision Theory

- Initially assume just 2 classes $\mathcal{C}_1, \mathcal{C}_2$
- Given input data x we want to determine which class is optimal
- Various possible criteria
- Need $p(\mathcal{C}_k \mid x)$ either directly (discriminative)
- Or by Bayes, (generative) $p(\mathcal{C}_k \mid x) = \frac{p(x, \mathcal{C}_k)}{p(x)} = \frac{p(x \mid \mathcal{C}_k) p(\mathcal{C}_k)}{p(x)}$
- Divide input space into *decision regions* $\mathcal{R}_1, \mathcal{R}_2$ separated by *decision boundaries* such that $x \in \mathcal{R}_k \Rightarrow \text{assign } \mathcal{C}_k$
- How might we choose boundaries?

Criterion 1: Minimize Misclassification Rate



assign class 1 but should be 2 — assign class 2 but should be 1

$$p(\text{mistake}) = p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1)$$

$$= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, \mathrm{d}\mathbf{x}.$$

$$p(x, \mathcal{C}_k) = p(\mathcal{C}_k \mid x)\, p(x)$$

Criterion 2: Minimize Expected Loss

Example loss matrix:

classify medical images as 'cancer' or 'normal'

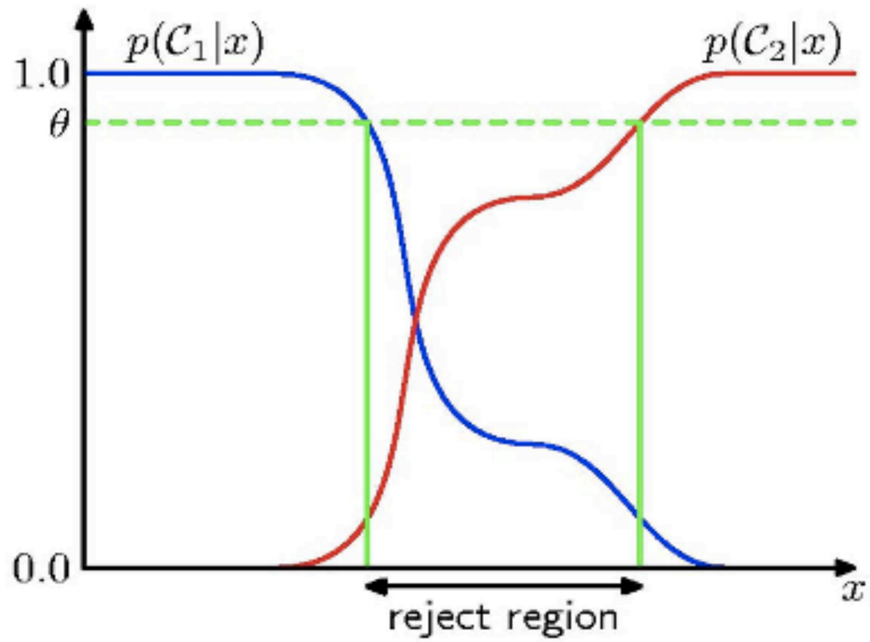
Truth \ Decision	cancer	normal
cancer	0	1000
normal	1	0

Now $\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}$

Choose regions $\mathcal{R}_j$ to minimize Expected Loss

$$\mathbb{E}[L] = \sum_k L_{kj} p(\mathcal{C}_k | \mathbf{x})$$

Reject Option



If $p(\mathcal{C}_1 \mid x) \approx p(\mathcal{C}_2 \mid x)$ then less confident about assigning class.

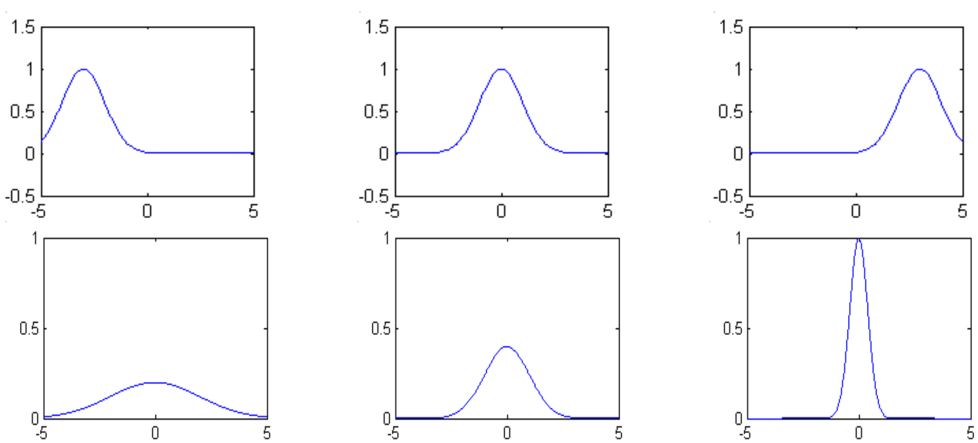
One possibility is to reject or refuse to assign.

Gaussian Distribution

- Most popular continuous distribution (why?)
- Recall 1-dimensional form:
  - ➢ Mean parameter μ translates Gaussian left & right
  - ➢ Variance parameter σ^2 widens or narrows the Gaussian

$$p\left(x \mid \mu\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(x-\mu\right)^2\right)$$

$$p\left(x \mid \mu, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right)$$



Note: $\int_{x=-\infty}^{\infty} p\left(x\right) dx = 1$

Multivariate Gaussian

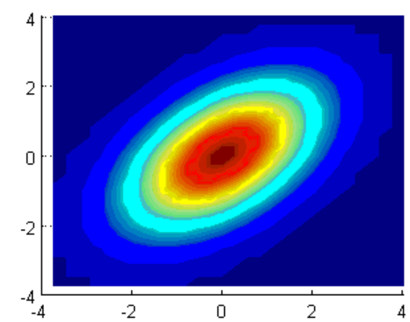
- Gaussian can extend to D-dimensions

- Gaussian mean parameter μ vector, it translates the bump
- Covariance matrix Σ stretches and rotates bump

$$p\left(\vec{x} \mid \vec{\mu}, \Sigma\right) = \frac{1}{\left(2\pi\right)^{D/2}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}\left(\vec{x}-\vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}-\vec{\mu}\right)\right)$$

$$\vec{x} \in \mathbb{R}^D, \vec{\mu} \in \mathbb{R}^D, \Sigma \in \mathbb{R}^{D\times D}$$

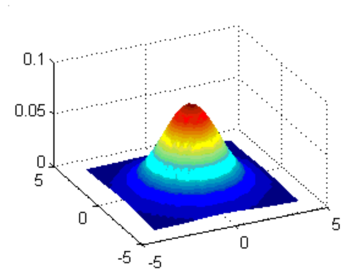
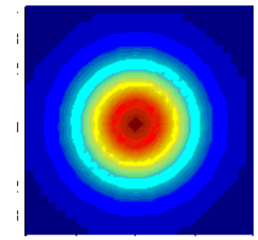
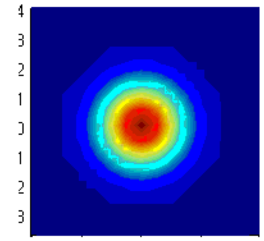
- Mean is any real vector
- Max and expectation = μ
- Variance parameter is now Σ matrix
- Covariance matrix is positive definite
- Covariance matrix is symmetric
- Need matrix inverse (inv)
- Need matrix determinant (det)
- Need matrix trace operator (trace)



Multivariate Gaussian

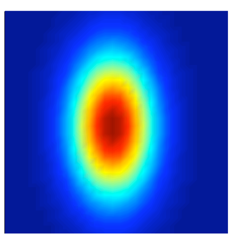
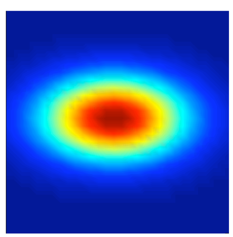
- Spherical:

$$\Sigma = \sigma^2 I = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

- Diagonal Covariance:
  - ➢ Dimensions of x are independent
  - ➢ Product of multiple 1d Gaussians

$$p\left(\vec{x} \mid \vec{\mu}, \Sigma\right) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}\vec{\sigma}(d)} \exp\left(-\frac{\left(\vec{x}(d)-\vec{\mu}(d)\right)^2}{2\vec{\sigma}(d)^2}\right)$$

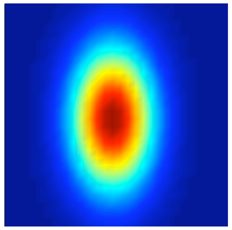



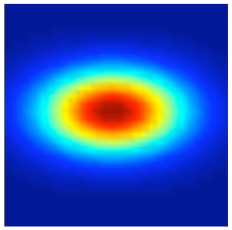
$$\Sigma = \begin{bmatrix} \vec{\sigma}(1)^2 & 0 & 0 & 0 \\ 0 & \vec{\sigma}(2)^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \vec{\sigma}(D)^2 \end{bmatrix}$$

Multivariate Gaussian

- Diagonal Covariance:
  - Dimensions of x are independent
  - Product of multiple 1d Gaussians

$$p\left(\vec{x} \mid \vec{\mu}, \Sigma\right) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}\vec{\sigma}(d)} \exp\left(-\frac{\left(\vec{x}(d)-\vec{\mu}(d)\right)^2}{2\vec{\sigma}(d)^2}\right)$$





$$\Sigma = \begin{bmatrix} \vec{\sigma}(1)^2 & 0 & 0 & 0 \\ 0 & \vec{\sigma}(2)^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \vec{\sigma}(D)^2 \end{bmatrix}$$

- The surface is an ellipsoid.
  - Eigenvectors of covariance = principle axes
  - Eigenvalues of covariance = length

MLE for Gaussian

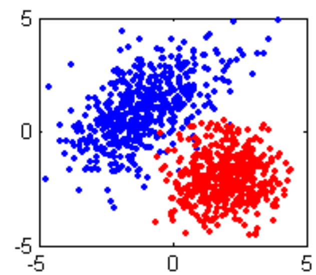
- Have IID samples as vectors i=1..N: $\mathcal{D} = \left\{ \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \right\}$
- How do we recover the mean and covariance parameters?
- Standard approach: Maximum Likelihood (IID)
- Maximize probability of data given model (likelihood)

$$
\begin{aligned}
p\left(\mathcal{D} \mid \theta\right) &= p\left(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \mid \theta\right) \\
&= \prod_{i=1}^{N} p\left(\vec{x}_i \mid \vec{\mu}_i, \Sigma_i\right) \quad \textit{independent Gaussian samples} \\
&= \prod_{i=1}^{N} p\left(\vec{x}_i \mid \vec{\mu}, \Sigma\right) \quad \textit{identically distributed}
\end{aligned}
$$

- Instead, work with maximum of log-likelihood

$$
\sum_{i=1}^{N} \log p\left(\vec{x}_i \mid \vec{\mu}, \Sigma\right) = \sum_{i=1}^{N} \log \frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2} \left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1} \left(\vec{x}_i - \vec{\mu}\right) \right)
$$

Classification with Gaussians



- Have two classes, each with its own Gaussian:
- The goal is to assign each example to one of the classes, while minimizing misclassification rate.

$$P(C_1|x) = p(x|C_1)P(C_1)$$
$$P(C_2|x) = p(x|C_2)P(C_2)$$
$$P(x|C_i) \sim N(\mu_i, \Sigma_i)$$

- We could use a discriminant function:

$$g_i(x) = \ln[p(x|C_i)P(C_i)] \;\; = \ln[p(x|C_i)] + \ln[P(C_i)]$$
$$g_i(x) = -\tfrac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \tfrac{1}{2}\ln|\Sigma_i| + \ln[P(C_i)]$$

Spherical Case

- Recall: $\Sigma = \sigma^2 I = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$

- We can simplify the discriminant:

$$|\Sigma_i| = \sigma^{2d}$$
$$\Sigma_i^{-1} = (\tfrac{1}{\sigma^2})I$$

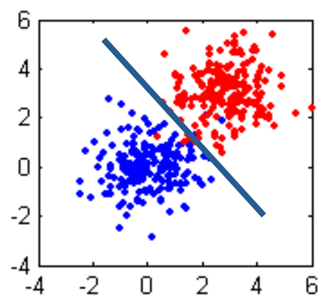
$$g_i(x) = -\tfrac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \tfrac{1}{2}\ln|\Sigma_i| + \ln[P(C_i)]$$

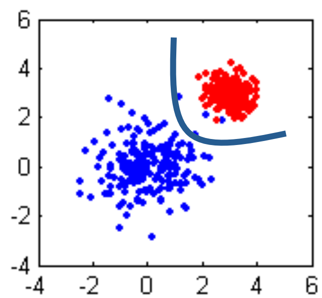
$$g_i(x) = \frac{-\|x-\mu_i\|^2}{2\sigma^2} + \ln[P(C_i)]$$

- Simple interpretation: if priors are equal, we have a *minimum distance classifier*
  - ➢ Euclidean distance to each mean

Arbitrary case

- Covariance matrices are different for each class
- Discriminant functions are quadratics
- See illustrations in Duda 2.6.3





Why so popular?

- Analytical tractability:
  - Gaussian family is *self-conjugate* (w.r.t Gaussian likelihood function)
  - Easy to manipulate
- Central Limit Theorem (CLT):
  - Given a sequence of iid random variables {X1,…,Xn}, their mean (assuming they have 'reasonable' properties, and there is enough of them) will be ≈ normally distributed
  - Convergence of mean to normal distribution
  - Model for many empirical processes
- Profound relation to Entropy:
  - For a given $\{\mu, \sigma^2\}$, Gaussian has the maximum entropy among all cont. pdfs
  - Entropy is a measure of randomness/unpredictability

What is MATLAB?

- MATLAB is a high-level language and interactive environment that allows one to solve science & engineering problems quickly using built-in functionality.
- High-level language:
  - User-friendly, easy to use, built-in functions (+)
  - Slower, less control (-)
- Interactive environment:
  - Graphical User Interface (GUI).
  - Visualization.
- Scripting language designed for “gluing together” computations.
- Object Oriented Programming (OOP) takes a backseat.
- Documentation is sufficient:
  - http://www.mathworks.com/help/techdoc/index.html
- Ideal for developing a prototype or a model, suitable for quick and dirty computation.
- Poor choice for a major commercial package.

MATLAB Overview

- See www.cs.columbia.edu->computing->Software->Matlab
- Online info to get started is available at:
  http://www.cs.columbia.edu/~coms4771/tutorials.html
- Basic functionality:
  - General: help, who, clear, %
  - Math: size, zeros, max, min, mean, norm, inv, sort
  - Control: if, for, while, end, return
  - Display: disp, figure, plot, hold on, fprintf
  - Input/Output: load, save, print
- Example code: for homework #1 we will use polyreg.m
- Discuss implementation details with TAs.