Chapter 3:

Maximum-Likelihood and Bayesian Parameter Estimation

Introduction

- Data availability in a Bayesian framework
  - We could design an optimal classifier if we knew:
    - $P(\omega_i)$ (priors)
    - $P(x \mid \omega_i)$ (class-conditional densities)

    Unfortunately, we rarely have this complete information
- Design a classifier from a training sample
  - No problem with prior estimation
  - Samples are often too small for class-conditional estimation (large dimension of feature space!)

- Apriori information about the problem
  - Normality of $P(x \mid \omega_i)$

$$P(x \mid \omega_i) \sim N(\mu_i, \Sigma_i)$$

  - Characterized by 2 parameters

- Estimation techniques
  - Maximum-Likelihood (ML) and Bayesian estimation
  - Results are nearly identical, but the approaches are different

- Parameters in ML estimation are fixed but unknown

- Bayesian methods view the parameters as random variables having some known distribution

- Best parameters are obtained by maximizing the probability of obtaining the samples observed

- In either approach, we use P(ωi | x) for our classification rule!

Maximum-Likelihood Estimation

- Has good convergence properties as the sample size increases
- Simpler than any other alternative techniques

- General principle

  - Assume we have c classes and
  - $P(x \mid \omega j) \sim N(\mu j, \Sigma j)$
  - $P(x \mid \omega j) \equiv P(x \mid \omega j, \theta j)$ where:

$$\theta_j = (\mu_j, \Sigma_j) = (\mu_j^1, \mu_j^2, ..., \sigma_j^{11}, \sigma_j^{22}, \mathrm{cov}(x_j^m, x_j^n)...)$$

Maximum-Likelihood Estimation

- Use the information provided by the training samples to estimate $\theta = (\theta 1, \theta 2, \ldots, \theta c)$, each θi ($i = 1, 2, \ldots, c$) is associated with each category
- Suppose that D contains n samples, x1, x2,…, xn

$$P(D \mid \theta) = \prod_{k=1}^{k=n} P(x_k \mid \theta) = F(\theta)$$

$P(D \mid \theta)$ is called the likelihood of θ w.r.t. the set of samples)

- ML estimate of θ is, by definition the value that maximizes $P(D \mid \theta)$
- "It is the value of θ that best agrees with the actually observed training sample"

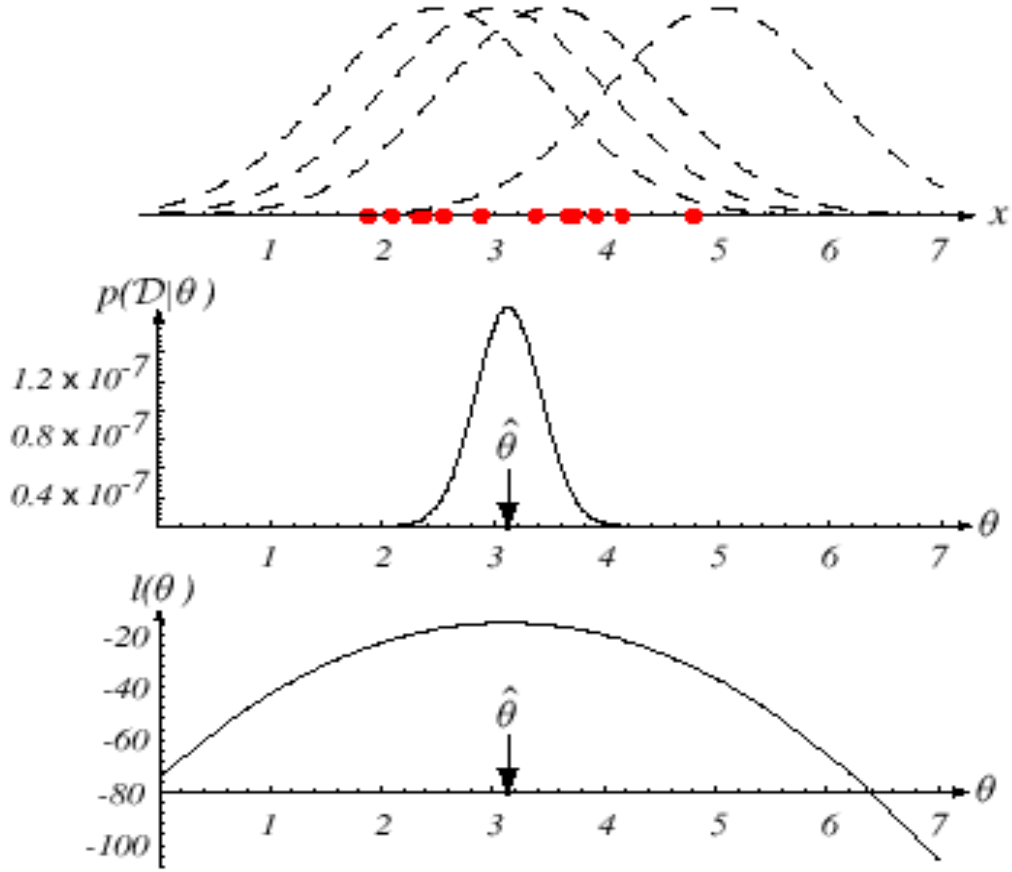


FIGURE 3.1. The top graph shows several training points in one dimension, known or assumed to be drawn from a Gaussian of a particular variance, but unknown mean. Four of the infinite number of candidate source distributions are shown in dashed lines. The middle figure shows the likelihood $p(\mathcal{D}|\theta)$ as a function of the mean. If we had a very large number of training points, this likelihood would be very narrow. The value that maximizes the likelihood is marked $\hat{\theta}$; it also maximizes the logarithm of the likelihood—that is, the log-likelihood $l(\theta)$, shown at the bottom. Note that even though they look similar, the likelihood $p(\mathcal{D}|\theta)$ is shown as a function of θ whereas the conditional density $p(x|\theta)$ is shown as a function of x. Furthermore, as a function of θ, the likelihood $p(\mathcal{D}|\theta)$ is not a probability density function and its area has no significance. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*.

- Optimal estimation
  - Let $\theta = (\theta 1, \theta 2, \ldots, \theta p)^t$ and let ∇_θ be the gradient operator

$$\nabla_\theta = \left[\frac{\partial}{\partial\theta_1}, \frac{\partial}{\partial\theta_2}, \ldots, \frac{\partial}{\partial\theta_p}\right]^t$$

  - We define $l(\theta)$ as the log-likelihood function $l(\theta) = \ln P(D \mid \theta)$
  - New problem statement: "Determine θ that maximizes the log-likelihood"

$$\hat{\theta} = \arg\max_\theta l(\theta)$$

- Set of necessary conditions for an optimum is:

$$\nabla_\theta l = \sum_{k=1}^{k=n} \nabla_\theta \ln P(x_k \mid \theta))$$

$$\nabla_\theta l = 0$$

Example of a specific case: Unknown μ

- P(xi | μ) ~ N(μ, Σ)
- (Samples are drawn from a multivariate normal population)

$$\ln P(x_k \mid \mu) = -\frac{1}{2}\ln\left[(2\pi)^d |\Sigma|\right] - \frac{1}{2}(x_k - \mu)^t \sum^{-1} (x_k - \mu)$$

$$and \quad \nabla_\mu \ln P(x_k \mid \mu) = \sum^{-1} (x_k - \mu)$$

- θ = μ therefore, the ML estimate for μ must satisfy:

$$\sum_{k=1}^{k=n} \Sigma^{-1}(x_k - \hat{\mu}) = 0$$

- Multiplying by Σ and rearranging, we obtain:

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{k=n} x_k$$

- Just the arithmetic average of the samples of the training data

- Conclusion:
  - If P(xk | ωj) (j = 1, 2, …, c) is supposed to be Gaussian in a d-dimensional feature space; then we can estimate the vector $\theta = (\theta 1, \theta 2, \ldots, \theta c)^t$ and perform an optimal classification!

Gaussian Case: Unknown μ and σ

- $\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$

$$l = \ln P(x_k \mid \theta) = -\frac{1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

$$\nabla_\theta l = \begin{pmatrix} \frac{\partial}{\partial \theta_1}(\ln P(x_k \mid \theta)) \\ \frac{\partial}{\partial \theta_2}(\ln P(x_k \mid \theta)) \end{pmatrix}$$

$$= \begin{cases} \frac{1}{\theta_2}(x_k - \theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} \end{cases}$$

- Summation:

$$\begin{cases} \sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2}(x_k - \theta_1) = 0 & (1) \\ -\sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{k=n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & (2) \end{cases}$$

- Combining (1) and (2):

$$\mu = \sum_{k=1}^{k=n} \frac{x_k}{n} \quad ; \quad \sigma^2 = \frac{\sum_{k=1}^{k=n} (x_k - \mu)^2}{n}$$

Bias

- ML estimate for σ^2 is biased, i.e., Expected value of sample variance over all datasets of size *n* is not equal to true variance

$$E\left[\frac{1}{n}\Sigma(x_i - \bar{x})^2\right] = \frac{n-1}{n}\sigma^2 \neq \sigma^2$$

- An elementary unbiased estimator for Σ is:

$$\underbrace{C = \frac{1}{n-1}\sum_{k=1}^{k=n}(x_k - \mu)(x_k - \hat{\mu})^t}_{\textit{Sample}\ \text{covariance matrix}}$$

- If an estimator is unbiased for all distributions, then it is called *absolutely unbiased*

- If the estimator tends to become unbiased as the number of samples become very large, then the estimator is *asymptotically unbiased*

- Bayesian Estimation (BE)
- Bayesian Parameter Estimation: General Estimation
- Bayesian Parameter Estimation: Gaussian Case

Bayesian Estimation

- Bayesian learning to pattern classification problems
  - In MLE, θ was supposed fix
  - In BE, θ is a random variable
  - The computation of posterior probabilities P(ωi | x) lies at the heart of Bayesian classification
  - Goal: Compute P(ωi | x, D)
  - Given the sample D, Bayes formula can be written

$$P(\omega_i \mid x, \mathrm{D}) = \frac{P(x \mid \omega_i, \mathrm{D}).P(\omega_i \mid \mathrm{D})}{\sum_{j=1}^{c} P(x \mid \omega_j, \mathrm{D}).P(\omega_j \mid \mathrm{D})}$$

Bayesian Estimation …

- To demonstrate the preceding equation, use:

$$P(x,\mathsf{D}\mid\omega_i) = P(x\mid\mathsf{D}\omega_i).P(\mathsf{D}\mid\omega_i)$$

$$P(x\mid\mathsf{D}) = \sum_j P(x,\omega_j\mid\mathsf{D})$$

$P(\omega_i) = P(\omega_i\mid\mathsf{D})$ (Training sample provides this!)

Thus:

$$P(\omega_i\mid x,\mathsf{D}) = \frac{P(x\mid\omega_i,\mathsf{D}_i).P(\omega_i)}{\sum_{j=1}^{c} P(x\mid\omega_j,\mathsf{D}).P(\omega_j)}$$

Bayesian Parameter Estimation: Gaussian Case

- Goal: Estimate θ using the a-posteriori density $P(\theta|D)$
  - The univariate case: $P(\mu \mid D)$
  - μ is the only unknown parameter

$$p(x \mid \mu) \sim N(\mu, \sigma^2)$$
$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$

  - (μ_0 and σ_0 are known!)

$$P(\mu \mid \mathrm{D}) = \frac{P(\mathrm{D} \mid \mu).P(\mu)}{\int P(\mathrm{D} \mid \mu).P(\mu)\,d\mu}$$

$$= \alpha \prod_{k=1}^{k=n} P(x_k \mid \mu).P(\mu)$$

$$P(\mu \mid \mathrm{D}) \sim N(\mu_n, \sigma_n^2)$$

$$\mathrm{P}(\mu) \sim \mathrm{N}(\mu_0, \sigma_0^2)$$

$$\begin{aligned}
p(\mu|\mathcal{D}) &= \alpha \prod_{k=1}^{n} \overbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2\right]}^{p(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right]}^{p(\mu)} \\
&= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{\mu-x_k}{\sigma}\right)^2 + \left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right)\right] \\
&= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right],
\end{aligned}$$

$$
\begin{aligned}
p(\mu|\mathcal{D}) &= \alpha \prod_{k=1}^{n} \overbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2\right]}^{p(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right]}^{p(\mu)} \\
&= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{\mu-x_k}{\sigma}\right)^2 + \left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right)\right] \\
&= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right],
\end{aligned}
$$

$$
p(\mu|\mathcal{D}) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right]
$$

$$
\frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma_2}\,\bar{x}_n + \frac{\mu_0}{\sigma_0^2},
$$

$$
\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}
$$

$$\bar{x}_n = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2+\sigma^2}\right)\bar{x}_n + \frac{\sigma^2}{n\sigma_0^2+\sigma^2}\mu_0$$

$$\sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2+\sigma^2}.$$

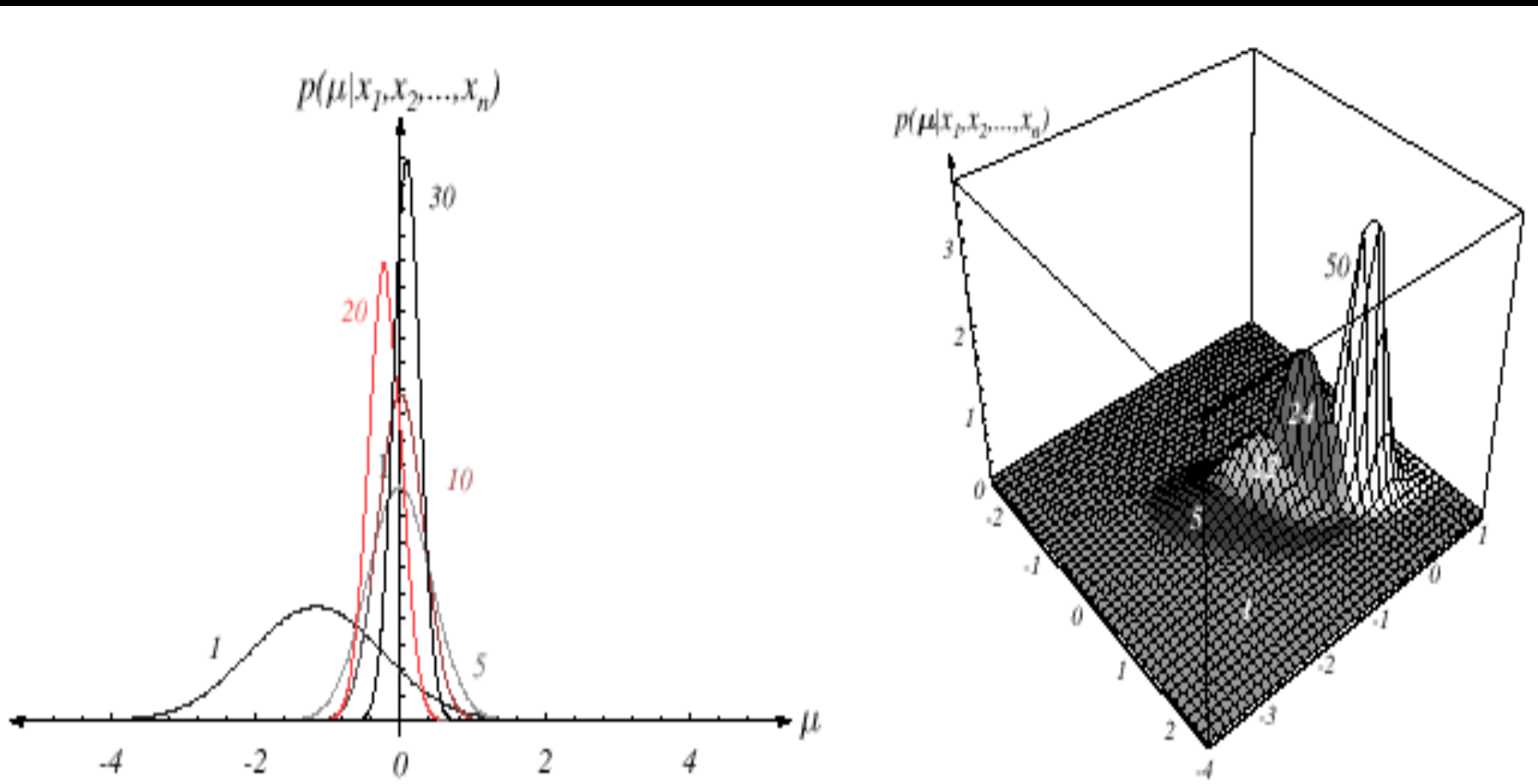


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Univariate Case

– The univariate case P(x | D)
  - P(μ | D) computed
  - P(x | D) remains to be computed!

$$P(x \mid \mathrm{D}) = \int P(x \mid \mu).P(\mu \mid \mathrm{D})d\mu \text{ is Gaussian}$$

  - It provides: $P(x \mid \mathrm{D}) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$

  - (Desired class-conditional density P(x | Dj, ωj))
  - Therefore: P(x | Dj, ωj) together with P(ωj)
  - And using Bayes formula for classification

Bayesian Parameter Estimation: General Theory

- P(x | D) computation can be applied to any situation in which the unknown density can be parameterized: the basic assumptions are:
  - The form of P(x | θ) is assumed known, but the value of θ is not known exactly
  - Our knowledge about θ is assumed to be contained in a known prior density P(θ)
  - The rest of our knowledge θ is contained in a set D of n random variables x1, x2, …, xn that follows P(x)

Bayesian Parameter Estimation: General Theory…

- The basic problem is: "Compute the posterior density P(θ | D)"
- then "Derive P(x | D)"
- Using Bayes formula: $$P(\theta \mid \mathrm{D}) = \frac{P(\mathrm{D} \mid \theta).P(\theta)}{\int P(\mathrm{D} \mid \theta).P(\theta)d\theta},$$

- And by independence assumption:

$$P(\mathrm{D} \mid \theta) = \prod_{k=1}^{k=n} P(x_k \mid \theta)$$

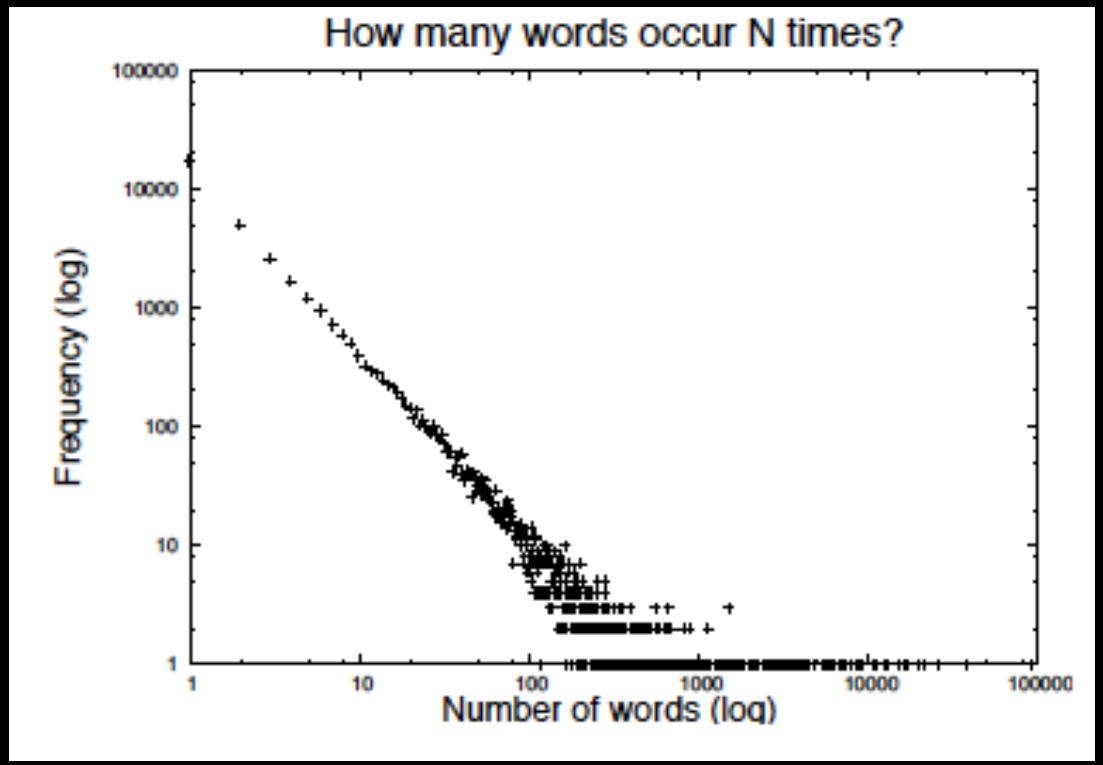
Three Sources of Error in Classification

- Bayes or Indistinguishability Error
  - Due to overlapping densities. Inherent property of given feature set
- Model Error
  - Due to an incorrect model. Can only be eliminated if designer specifies true model that generated the data.
- Estimation Error
  - Parameters are estimated from a finite sample.

Smoothing

How much probability does the model assign to strings/ pixel values it hasn't seen before?

- An empirical fact about language:
- A small number of events occur with high frequency
- A large number of events occur with low frequency



Zipf's Law

Smoothing

- Add-1 (Laplace) smoothing: Assume every seen or unseen event occurred once more than it did in the training data

$$\text{MLE} \quad P(w_i) = \frac{C(w_i)}{\sum_j C(w_j)} = \frac{C(w_i)}{N}$$

$$\text{Add One} \quad P(w_i) = \frac{C(w_i)+1}{\sum_j (C(w_j)+1)} = \frac{C(w_i)+1}{N+V}$$

- Disadvantage of Add-1 Smoothing
  - Takes away too much probability mass from seen events.
  - Assigns too much total probability mass to unseen events.

Estimators

$$P(X) = \frac{N_X + f}{T + E_\Omega \cdot f},$$

- f is a fudge factor
- Expected Likelihood Estimator sets f = 0.5
- Laplace and add-one estimators are the same thing and set f = 1,
- add-tiny sets f = 1/T

http://kochanski.org/gpk/teaching/0401Oxford/GoodTuring.pdf

Good Turing Smoothing

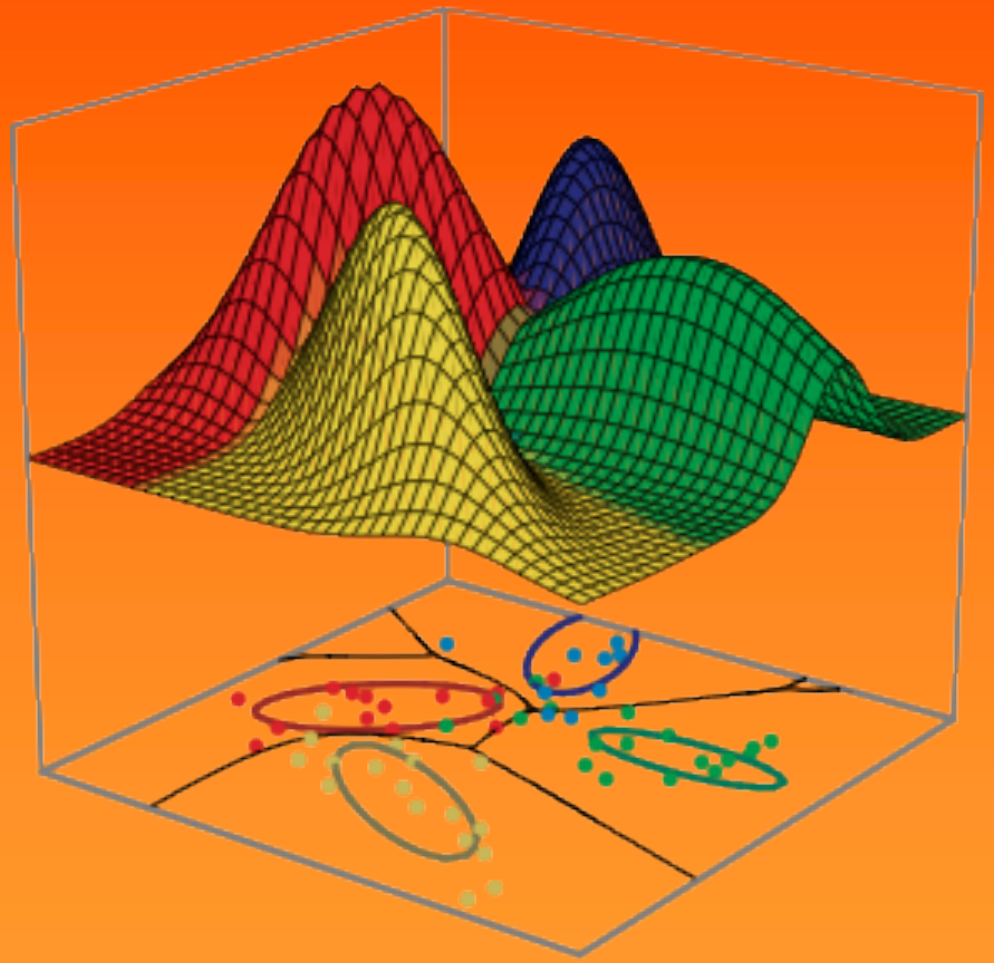
$$F_X = \frac{(N_X + 1)}{T} \cdot \frac{E(N_X + 1)}{E(N_X)}$$

- X is the event, NX is the number of times you have seen event X , T is the sample size and E (n) is an estimate of how many different events happened exactly n times. Translating that into text-analysis terms, X is a word, NX is the number of times you have seen word X , T is the size of the corpus and E(n) is an estimate of how many different words were observed exactly n times **3 .**

Good Turing Smoothing

- Take a corpus of 30000 English words, our universe is all English words, our event, **X , is the word "unusualness." The word** "unusualness" appears once, so **NX = 1. In a reasonable corpus, you might have** 10000 different words that appear once, so **E (1) = 10000, and you might have** 3000 words that appear twice, giving **E (2) = 3000. The Good-Turing estimate** of the probability of "unusualness" is then

$$P(unusualness) = \frac{2}{30000} \cdot \frac{3000}{10000},$$



Pattern Classification

All materials in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
with the permission of the authors and the publisher