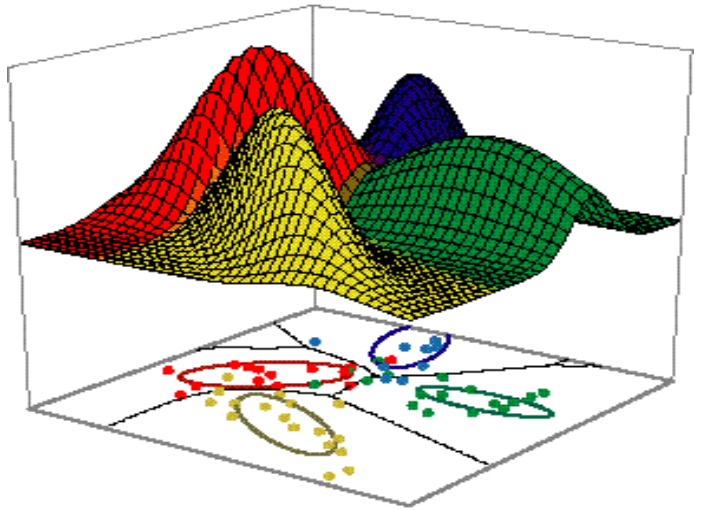
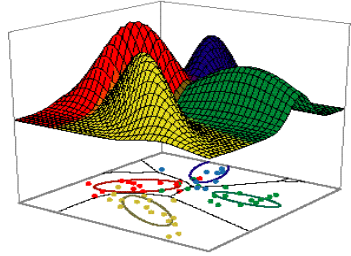
Part 6: Parameter Estimation



Parametric vs. Nonparametric
Maximum Likelihood Parameter Estimation
Bayesian Parameter Estimation

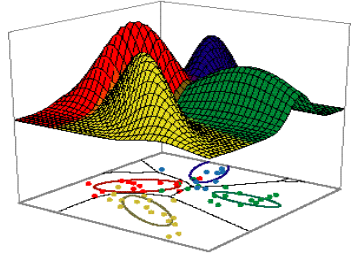
Some materials in these slides were taken from Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000, **Chapter 3.1-3.5.**

Introduction



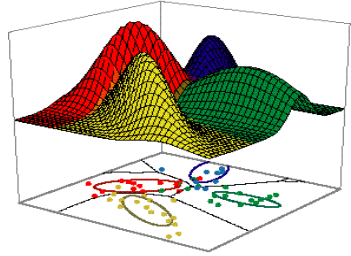
- Until now, have assumed that we know the distribution of all class-conditional density functions & priors
  - e.g., know that $p(x|\omega_1) \sim N(0.5,6.2)$
  - If we know this, we can create optimal classifiers.
- In reality, we rarely know this information!
  - How do we estimate these density functions from the data?
  - Samples are often too small for direct class-conditional estimation (large dimension of feature space!)

Introduction



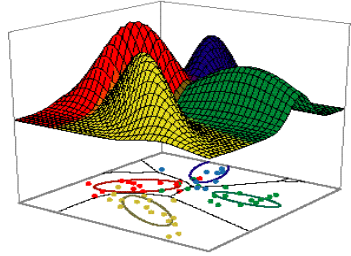
- 2 Fundamental Approaches:
  - Parameter Estimation:
    - We may know (or assume) form of distribution; estimate parameters of that distribution from the data → requires prior knowledge
    - e.g., assume that $p(x|\omega_1) \sim N(\mu,\sigma)$, now estimate parameters μ & σ from the data
      - Now only have to estimate 2 parameters instead of entire distribution.
  - Nonparametric Density Estimation:
    - Don't assume any particular form of distribution
    - Let the data determine the density function directly.
    - May not result in analytical solution

Parameter Estimation



- Typically discuss 2 approaches to estimating $p(x|\omega_1)$
  - Results are nearly identical, but the approaches are different
  - 1) Maximum Likelihood Parameter Estimation
    - Assume parameters, θ, have fixed but unknown values
    - Estimate the parameter values that make the data most likely to have been observed given the value of θ
  - 2) Bayesian Parameter Estimation
    - Assume that parameters, θ, are random variables with some known prior distribution.
    - Observations convert this to *a posteriori* density
    - Additional observations sharpen the *a posteriori* density and cause it to peak near actual values of θ
- In either approach, use $P(\omega_i \mid x)$ for classification rule!

Maximum Likelihood Estimation



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

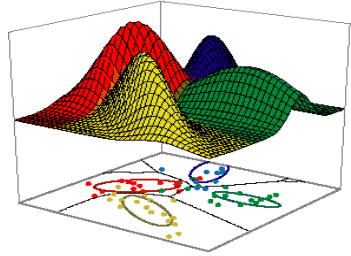
- General principle
  - Assume we have *c* classes and

    $P(x \mid \omega_j) \sim N(\mu_j, \Sigma_j)$

    $P(x \mid \omega_j) \equiv P(x \mid \omega_j, \theta_j)$ where:

$$\theta_j = (\mu_j, \Sigma_j) = (\mu_j^1, \mu_j^2, \ldots, \mu_j^d, \sigma_j^{11}, \sigma_j^{22}, \ldots, \sigma_j^{dd}, \mathrm{cov}(x_j^k, x_j^l) \ldots)$$

Maximum Likelihood Estimation



- Use the information provided by the training samples to estimate $\theta = (\theta_1, \theta_2, \ldots, \theta_c)$, each θ_i (i = 1, 2, …, *c*) is associated with a category

- Suppose that *D* contains *n* samples, $x_1, x_2, \ldots, x_n$ **drawn independently**

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

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

- ML estimate of θ is, by definition the value that maximizes $P(D \mid \theta)$. Referred to as: $\hat{\boldsymbol{\theta}}$

"It is the value of θ that best agrees with the actually observed training sample"

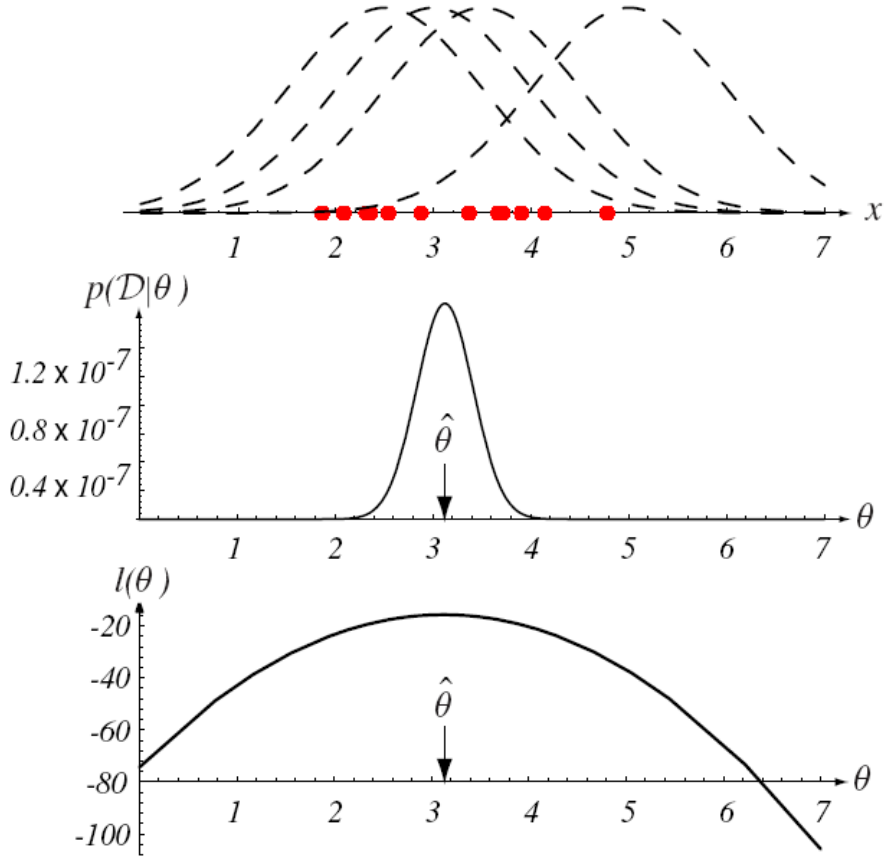
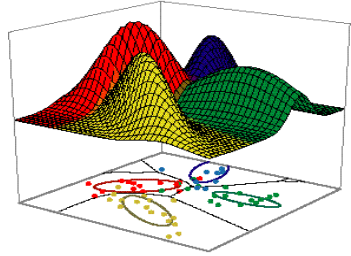


FIGURE 3.1. The top graph shows several training points in one dimension, known or assumed to be drawn from a Gaussian of a particular variance, but unknown mean. Four of the infinite number of candidate source distributions are shown in dashed lines. The middle figure shows the likelihood $p(\mathcal{D}|\theta)$ 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*. Copyright © 2001 by John Wiley & Sons, Inc.

Maximum Likelihood Estimation



- 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

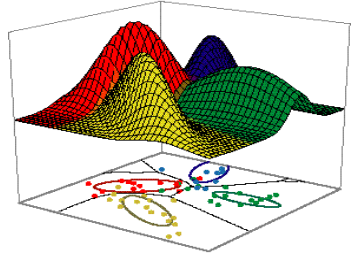
$$l(\theta) = \ln P(D \mid \theta)$$

  - New problem statement:
    determine θ that maximizes the log-likelihood

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

implicit dependence on D

Maximum Likelihood Estimation



Set of necessary conditions for an optimum is:

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

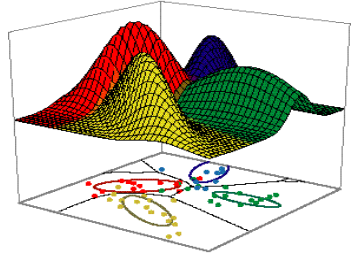
Since samples drawn IID

$$\nabla_\theta l = 0$$

Take the gradient, and set to zero

A solution could represent global min/max, local min/max, or point of inflection

Maximum Likelihood Estimation



- Example of a specific case: unknown μ

  - $p(x_i \mid \mu) \sim N(\mu, \Sigma)$
    (Samples are drawn from a multivariate normal population)

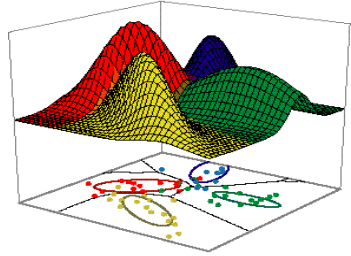
$$\ln p\,(x_k|\mu) = -\frac{1}{2}\ln[(2\pi)^d|\Sigma|] - \frac{1}{2}(x_k - \mu)^t\Sigma^{-1}(x_k - \mu)$$

$\theta = \mu$ therefore: $\nabla_\mu \ln p\,(x_k|\mu) = \Sigma^{-1}(x_k - \mu)$

- The ML estimate for μ must satisfy:

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

Maximum Likelihood Estimation



- Multiplying by Σ and rearranging, we obtain:

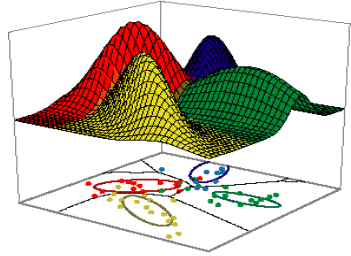
$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{k=n} x_k$$

Simply the arithmetic average of the samples of the training samples!

Conclusion:

If $p(x_k \mid \omega_j)$ (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!

Maximum Likelihood Estimation

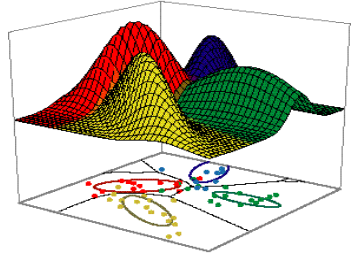


- ML Estimation:
  - 1D Gaussian Case: *unknown μ and σ*
    $\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$
  - Consider for a single point x_k:

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

$$\nabla_\theta l = \begin{pmatrix} \frac{\partial}{\partial\theta_1}\left(\ln P\left(x_k|\theta\right)\right) \\ \frac{\partial}{\partial\theta_2}\left(\ln P\left(x_k|\theta\right)\right) \end{pmatrix} = 0 \rightarrow \begin{cases} \frac{1}{\theta_2}\left(x_k - \theta_1\right) = 0 \\ -\frac{1}{2\theta_2} + \frac{\left(x_k - \theta_1\right)^2}{2\theta_2^2} = 0 \end{cases}$$

Maximum Likelihood Estimation



Summation over all x_k:

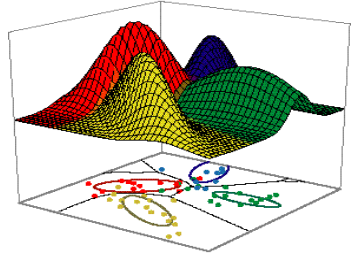
$$\begin{cases} \sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \hat{\theta}_1) = 0 & (1) \\ -\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{k=n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & (2) \end{cases}$$

Combining (1) and (2), one obtains:

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

Multivariate: $\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k \qquad \hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})\ (x_k - \hat{\mu})^t$

Maximum Likelihood Estimation



- Bias
  - Bias of an estimate is the systematic/deterministic error
    - i.e., diff between true value, θ, and expected value of its estimate, $\hat{\theta}$.
  - ML estimate for σ^2 is biased

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

  - An elementary unbiased estimator for Σ is:

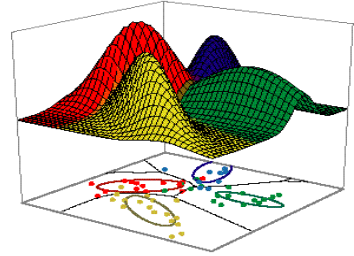
$$C = \frac{1}{n-1}\sum_{k=1}^{k=n}(x_k - \bar{x})^2$$

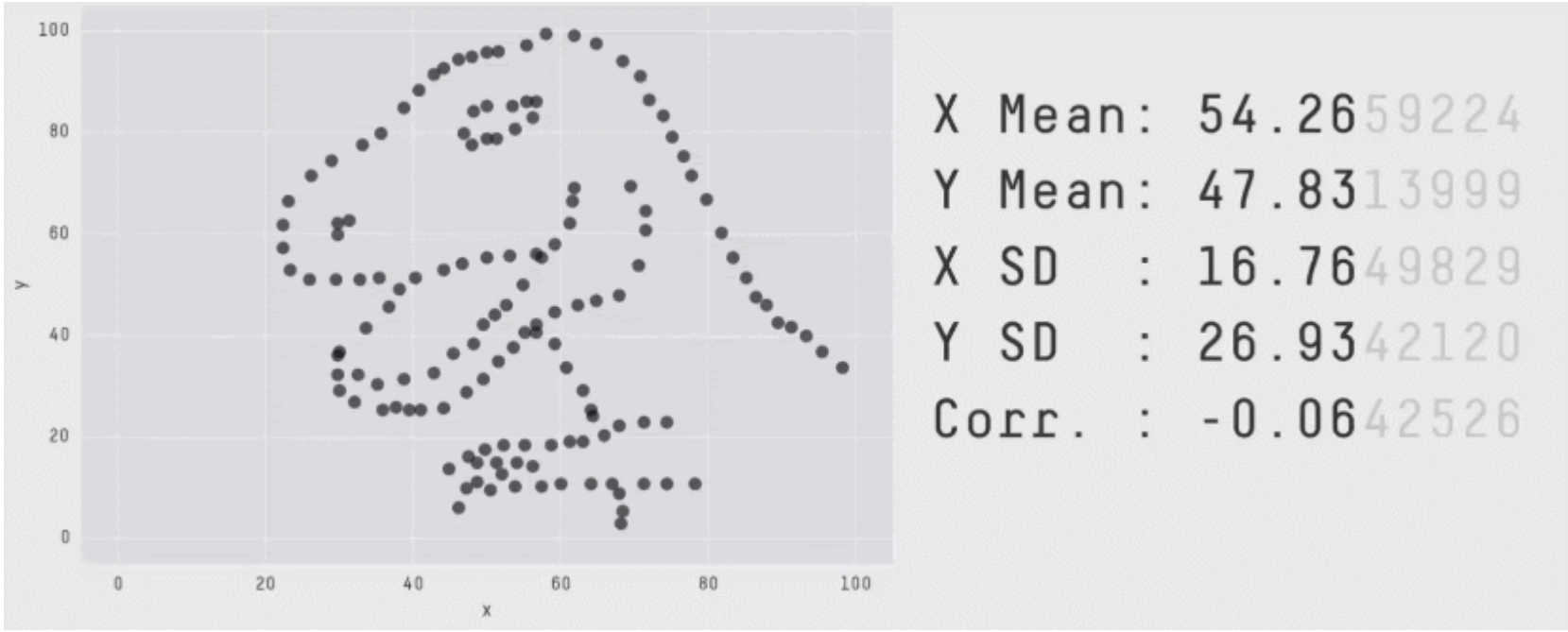
Sample covariance

$$C = \frac{1}{n-1}\sum_{k=1}^{k=n}(x_k - \hat{\mu})(x_k - \hat{\mu})^t$$

Sample covariance matrix (multivariate)

Warning: Visualize Data Before Assuming Distribution

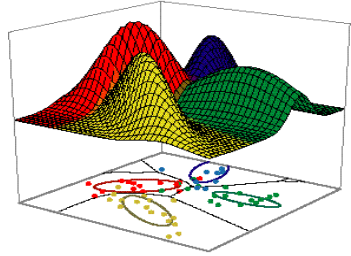




Justin Matejka and George Fitzmaurice. **Same stats, different graphs: Generating datasets with varied appearance and identical statistics through simulated annealing.** In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pages 1290–1294. ACM, 2017.

Slide from Lemai.ai, Daniel Shapiro

Bayesian Estimation

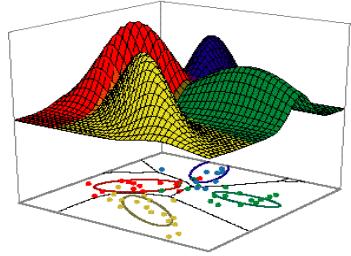


- (aka Bayesian learning when used for pattern classification)
- Assume form of density is known, parameter θ is not
  - In MLE, θ was assumed to be fixed, but unknown
  - In BE θ is a random variable, training data allows us to convert dist on this variable into a posterior probability density $P(\theta| x)$
  - Either way, p(x) is not known, but $p(x|\theta)$ is completely known
- Ultimate goal: compute $P(\omega_i | x, D)$
  - Given the sample *D*, Bayes' formula can be written

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

  - Use information from *D* to help determine class-conditional and prior probabilities. Will henceforth assume priors are known or easily estimated (simplify notation…)

Bayesian Estimation

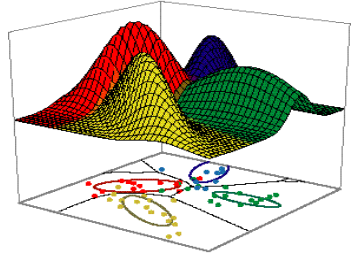


- Assume that samples in D_i have no influence on $p(x|w_j,D)$ if $i \neq j$, so:

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

- Allows us to work with each class separately

Bayesian Estimation



- Gaussian Case
  - Goal: Estimate θ using the *a posteriori* density P(θ | *D*)

    (recall that ML maximized P($\mathcal{D}$|θ) instead)
  - The univariate case: P(μ | *D*)

    μ is the only unknown parameter

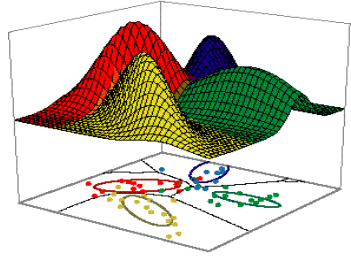
$$p(x \mid \mu) \sim N(\mu, \sigma^2)$$

$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$ ← *Prior distribution of the unknown mean μ.*

(μ_0 and σ_0 are known!)

Imagine value is drawn for μ from a population governed by p(μ). Once drawn, it becomes the true value for μ and completely determines density of x. Now *n* Samples *D* = {x_1, …, x_n} are drawn independently from resulting population.

Bayesian Estimation



$$p(\mu \mid \mathsf{D}) = \frac{P(\mathsf{D} \mid \mu)\, p(\mu)}{\int P(\mathsf{D} \mid \mu)\, p(\mu)\, d\mu} \qquad (1)$$

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

α depends on D, but is indep of μ

- Reproducing density p(μ|*D*) is the product of two Normal distributions. Results in a new Normal:

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

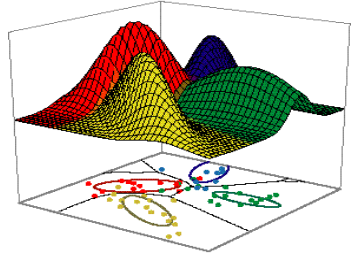
- Equating (1) and (2) yields *(skipping math on p93)*:

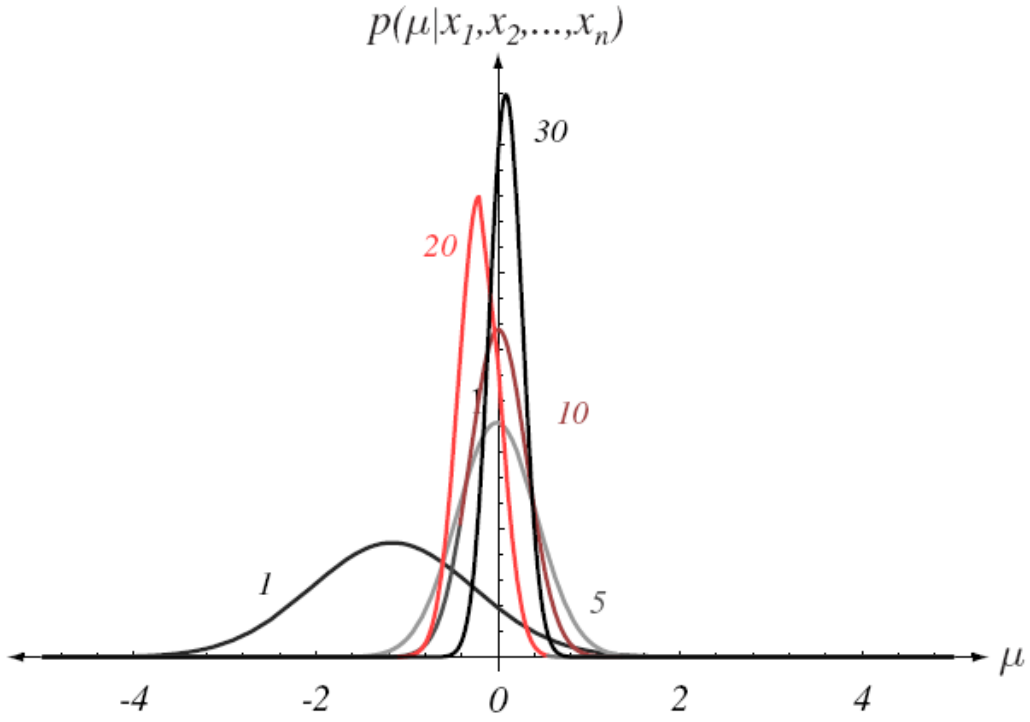
$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \cdot \mu_0$$

μ_n is best guess for μ after n points. Weighted combination of sample mean, and prior.

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

σ_n measures uncertainty about μ. as n↑, σ_n ↓. Leads to a dirac dist about true μ → Bayesian Learning





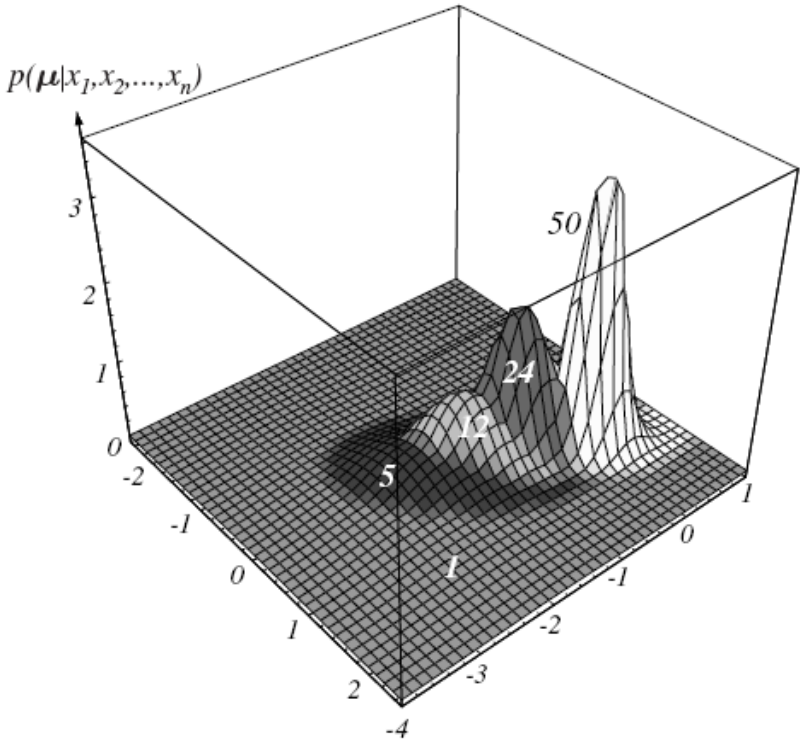
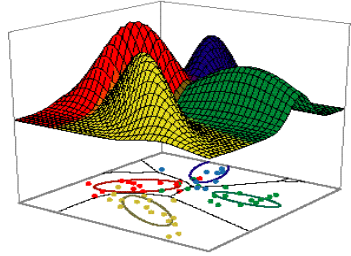


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001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

Bayesian Estimation



- The univariate case P(x | $\mathcal{D}$)
  - P(μ | *D*) computed
  - P(x | *D*) remains to be computed!

$$p(x \mid \mathrm{D}) = \int p(x \mid \mu)\, p(\mu \mid \mathrm{D})\, d\mu \quad \text{is Gaussian}$$

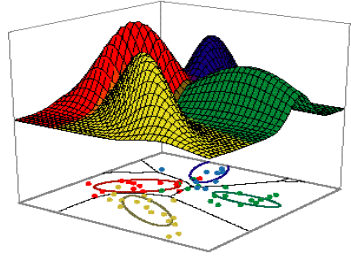
It provides: $p(x \mid \mathrm{D}) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$ *Variance increased due to uncertainty about mean.*

(This is desired class-conditional density P(x | ω_j , D_j))

Finally, using Bayes formula to combine P(x | ω_j , D_j) together with P(ω_j), we obtain the Bayesian classification rule:

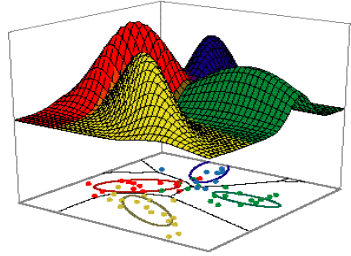
$$\underset{\omega_j}{Max}\left[P(\omega_j \mid x, \mathrm{D})\right] \equiv \underset{\omega_j}{Max}\left[P(x \mid \omega_j, \mathrm{D}_j).P(\omega_j)\right]$$

Bayesian Estimation : General Theory



- p(x | $\mathcal{D}$) computation can be applied to any situation in which the unknown density can be parameterized (not just univariate Gaussian)

- 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 $x_1, x_2, \ldots, x_n$ that follows p(x)



Bayesian Estimation : General Theory

The basic problem is:

"1) Compute the posterior density P(θ | $\mathcal{D}$)"

Using Bayes formula, we have:

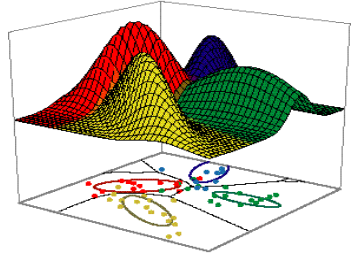
$$p(\theta \mid \mathsf{D}) = \frac{p(\mathsf{D} \mid \theta)\, p(\theta)}{\int p(\mathsf{D} \mid \theta)\, p(\theta) d\theta}$$

And by independence assumption: $P(\mathsf{D} \mid \theta) = \prod_{k=1}^{k=n} P(x_k \mid \theta)$

then "2) Derive P(x | $\mathcal{D}$)"

$$p(x \mid \mathsf{D}) = \int p(\mathsf{x} \mid \theta)\, p(\theta \mid D) d\theta$$

Recursive Bayesian Approach

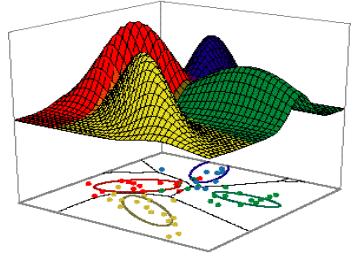


- Given: $P(\mathsf{D}\,|\,\theta) = \prod_{k=1}^{k=n} P(x_k\,|\,\theta)$
- We get: $p(\mathsf{D}^n\,|\,\theta) = p\left(x_n\,|\,\theta\right)p\left(D^{n-1}\,|\,\theta\right)$
- And: $p(\theta\,|\,\mathsf{D}^n) = \dfrac{p(\mathsf{x}_n\,|\,\theta)\,p(\theta\,|\,D^{n-1})}{\int p(\mathsf{x}_n\,|\,\theta)\,p(\theta\,|\,D^{n-1})\,d\theta}$

$$p(\theta\,|\,\mathsf{D}^0) = p\left(\theta\right)$$

- Recursive relationship.
  - Example of *incremental* or *on-line* learning
    - Learn as data is collected.
  - Many methods require all training data must be present before learning takes place.

ML vs. Bayes Methods



- For reasonable priors that include true solution, ML & Bayes equivalent as n → ∞
  - However, we are normally faced with limited data
  - If p(θ|D) is broad or asymmetric around ML estimate for θ, resulting p(x|D) will differ.
- Several criteria for comparing:
  - Computational complexity → ML preferred
    - Compute gradient rather than multidimensional integration.
  - Interpretability → ML often preferred.
    - Returns single best model, not weighted distribution of possible models.
    - However, Bayes makes bias/variance trade-off more explicit.
  - Prior information → Bayes preferred
    - If dependable prior information is available, Bayes can use it more completely.
    - If prior does not contain true model, ML will never find it. Bayes can (not strictly constrained by assumed parametric model)
    - Flat or uninformative priors make methods similar.
- Recall 3 sources of error in pattern classification
  - 1) Bayes or Indistinguishability Error: Can never be eliminated.
  - 2) Model Error: Model based on prior information. Generally not dependent on parameter estimation approach that will be taken.
  - 3) Estimation error: Need more training data → independent of P.E. method.