Lecture 5

1. Prior Probabilities
2. Class-Conditional Probabilities
3. Posterior Probabilities
4. Bayes Rule

Probability Decision Theory

- Bayesian Decision Theory is a fundamental statistical approach to the problem solving.
- It allows us to quantify the tradeoffs between various classification decisions using probability and the costs that accompany these decisions.
- We assume all relevant probability distributions are known.
  - Later, as part of the machine learning process, we will learn how to estimate these from data.
- Can we exploit prior knowledge using Bayesian Decision Theory?
- For example, in our fish classification problem:
  - Are the sequence of fish predictable? (statistics)
  - Is each class equally probable? (uniform priors)
  - What is the cost of an error? (risk, optimization)

Machine Learning Example: Sorting Fish

- **Sorting Fish:** incoming fish are sorted according to species using optical sensing (sea bass or salmon?)

- **Problem Analysis:**
  - Set up a camera and take some sample images to extract features
  - Consider features such as length, color, width, number and shape of fins etc.

↓

Sensing

↓

Segmentation

↓

Feature Extraction

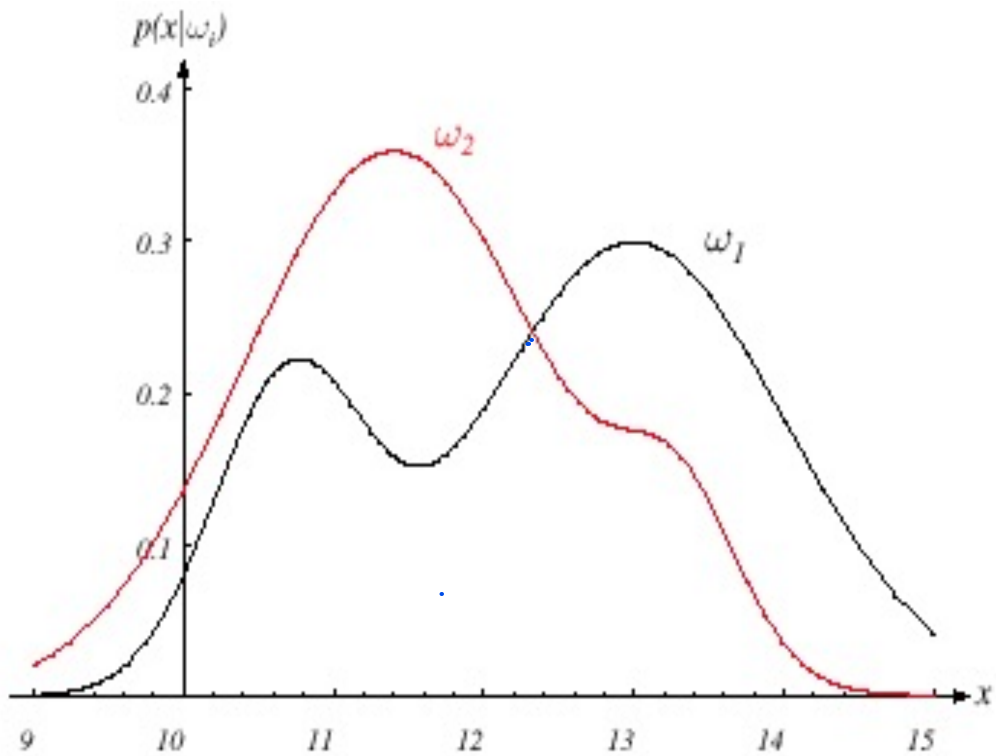
↓

Prior Probabilities

- State of nature is *prior* information
- Model as a random variable, ω:
  - $\omega = \omega_1$: the event that the next fish is a sea bass
  - category 1: sea bass; category 2: salmon
  - $P(\omega_1)$ = probability of category 1
  - $P(\omega_2)$ = probability of category 2
  - $P(\omega_1) + P(\omega_2) = 1$
    - Exclusivity: ω_1 and ω_2 share no basic events
    - Exhaustivity: the union of all outcomes is the sample space (either ω_1 or ω_2 must occur)
- If all incorrect classifications have an equal cost:
  - Decide ω_1 if $P(\omega_1) > P(\omega_2)$; otherwise, decide ω_2

Class-Conditional Probabilities

- **A decision rule with only prior information always produces the same result and ignores measurements.**
- **If $P(\omega_1) >> P(\omega_2)$, we will be correct most of the time.**
- **Probability of error: $P(E) = min(P(\omega_1),P(\omega_2))$.**
- **Given a feature, x (color), which is a continuous random variable, $p(x|\omega_1)$ is a class-conditional probability density function.**
- **$p(x|\omega_1)$ and $p(x|\omega_2)$ describe the difference in lightness between populations of sea and salmon.**
- **We often refer to these as likelihoods because they represent the likelihood of a value x given that it came from class ω_1 (or ω_2).**



Probability Functions

- **A probability density function represents a function of a continuous variable.**
- ***$p_x(x|\omega)$*, often abbreviated as *$p(x)$*, denotes a probability density function for the random variable X. Note that *$p_x(x|\omega)$* and *$p_y(y|\omega)$* can be two different functions.**
- ***$P(x|\omega)$* denotes a probability mass function, and must obey the following constraints:**

$$\boldsymbol{P(x) \geq 0}$$

$$\sum_{x \in X} \boldsymbol{P(x) = 1}$$

- **Probability mass functions are typically used for discrete random variables (which are summed) while densities describe continuous random variables (latter must be integrated).**
- **We may mix both discrete variables (related to the number of classes) and continuous variables (the probability of a feature vector).**

Bayes Formula

- **Suppose we know both $P(\omega_j)$ and $p(x|\omega_j)$, and we can measure x. How does this influence our decision?**

- **The joint probability of finding a pattern that is in category ω_j and that this pattern has a feature value of x is:**

$$p(\omega_j, x) = P(\omega_j | x)p(x) = p(x | \omega_j)P(\omega_j)$$

- **Rearranging terms, we arrive at Bayes Rule, also known as Bayes Formula:**

$$P(\omega_j | x) = \frac{p(x | \omega_j)P(\omega_j)}{p(x)}$$

- **The denominator term, which is known as the evidence, combines the two numerator terms (for the case of two categories):**

$$p(x) = \sum_{j=1}^{2} p(x | \omega_j)P(\omega_j)$$

- **This is the probability that a particular value of x can occur. It is difficult to calculate because it is the sum across all possible conditions.**

Posterior Probabilities

- **Bayes Rule:**

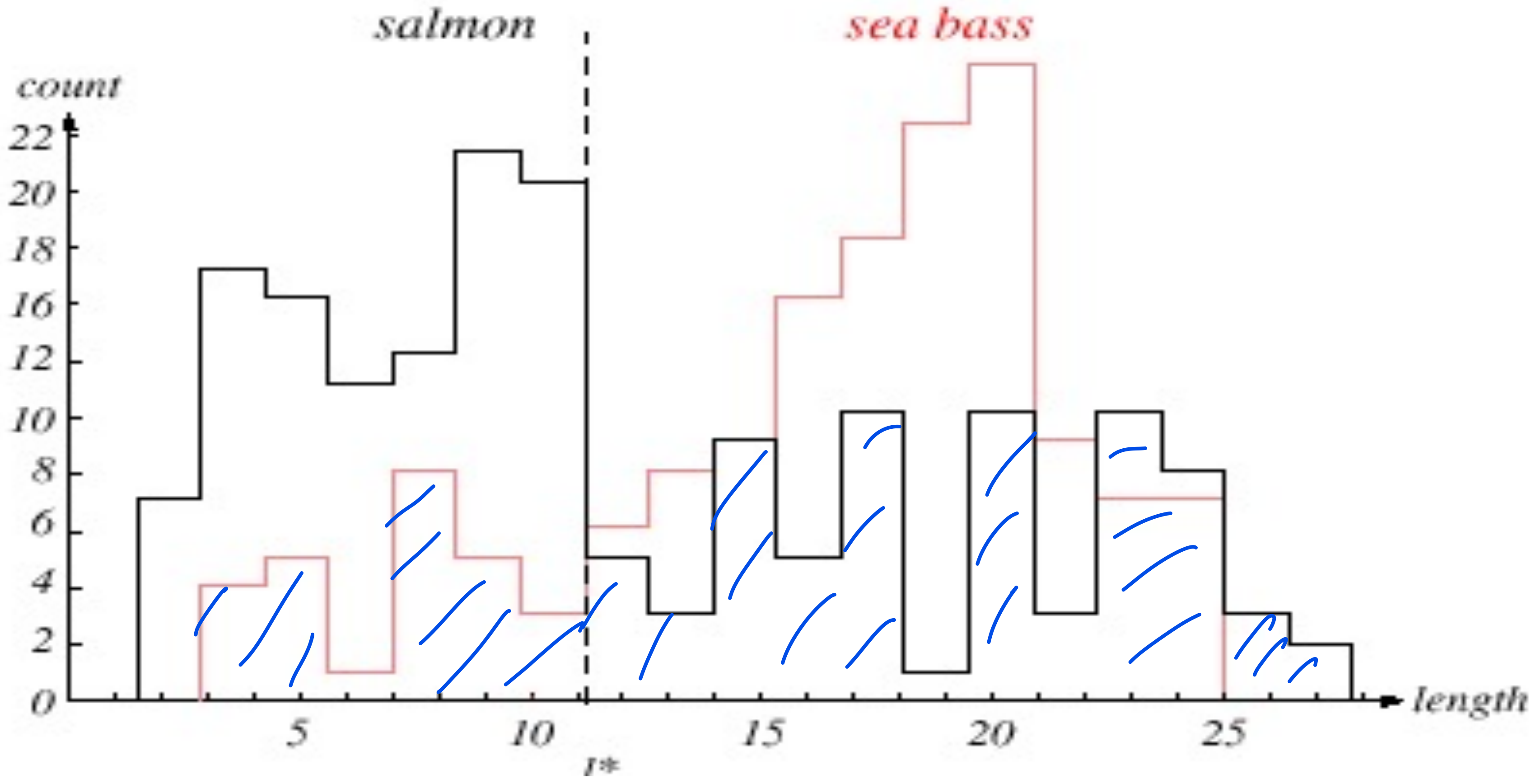
$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

can be expressed in words as:

$$posterior = \frac{likelihood \times prior}{evidence}$$

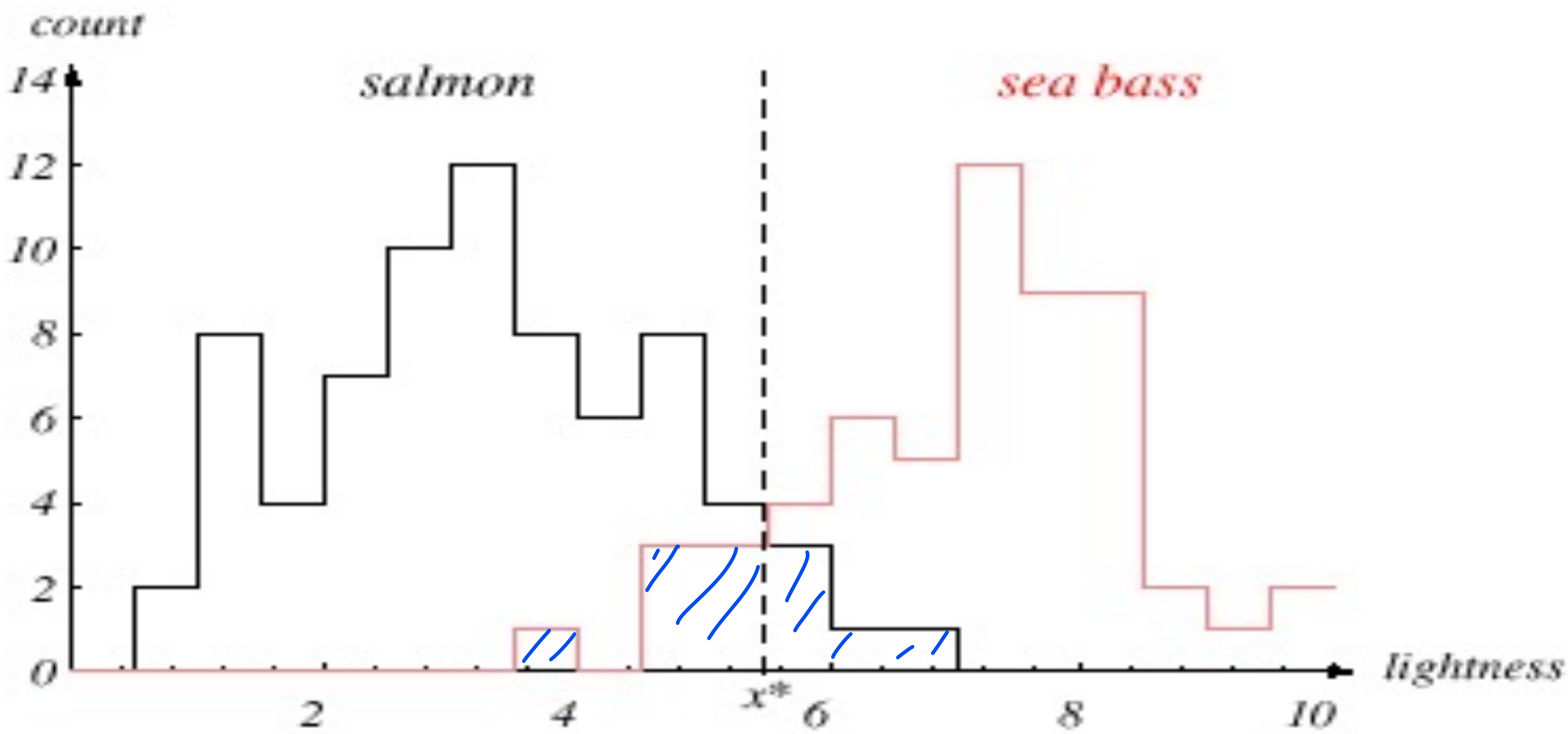
- **By measuring *x*, we can convert the prior probability, $P(\omega_j)$, into a posterior probability, $P(\omega_j|x)$.**
- **Evidence can be viewed as a scale factor and is often ignored in optimization applications (e.g., speech recognition).**
- **Bayes Rule allows us to train a system by collecting data in what is called a supervised mode (e.g., collect a sea bass sample and measure its length, speak a specific set of words and measure the feature vectors).**

Length As A Discriminator



- **Conclusion: Length is a poor discriminator**

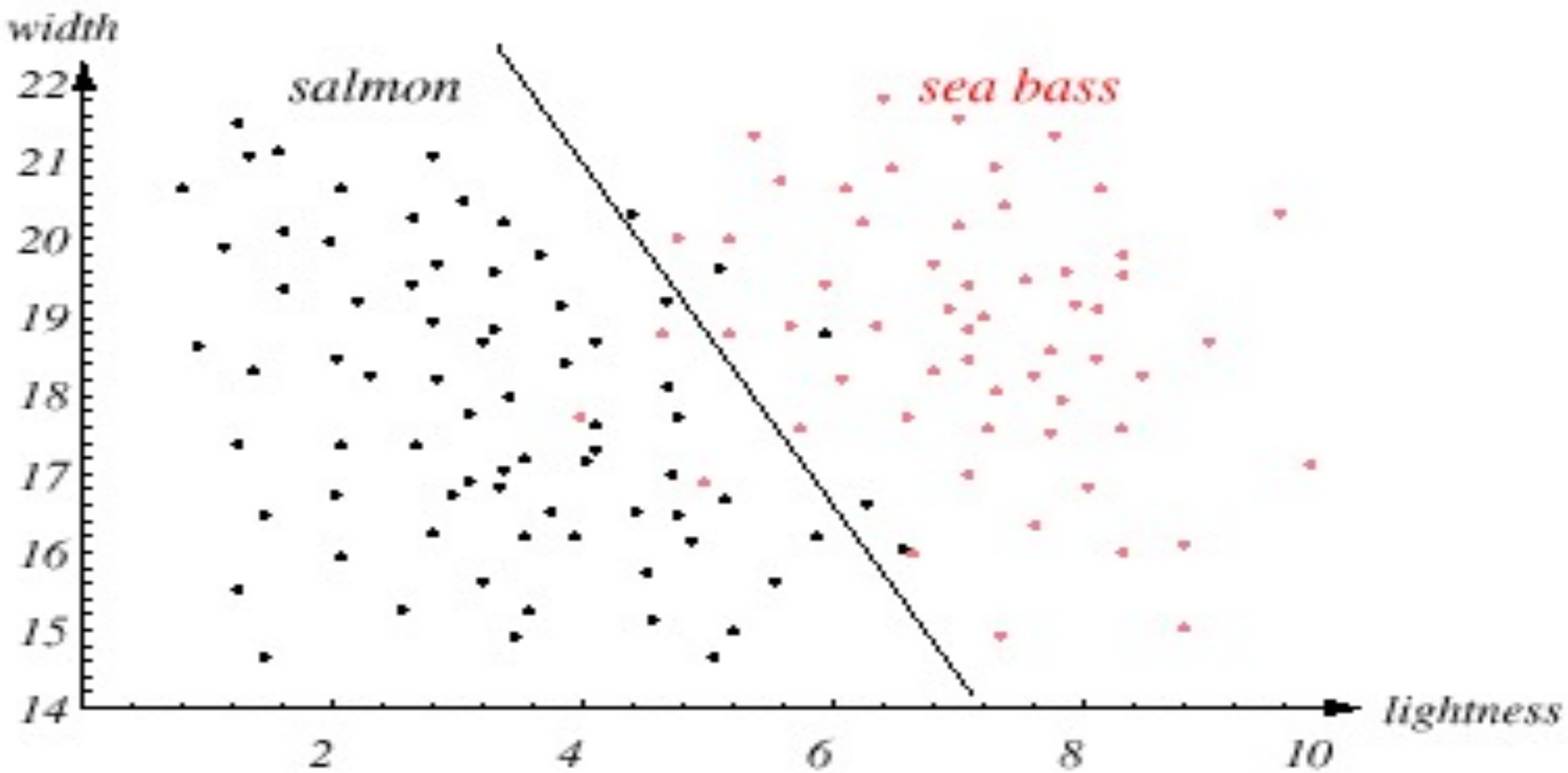
Add Another Feature



- **Lightness is a better feature than length because it reduces the misclassification error.**
- **Can we combine features in such a way that we improve performance? (Hint: correlation)**

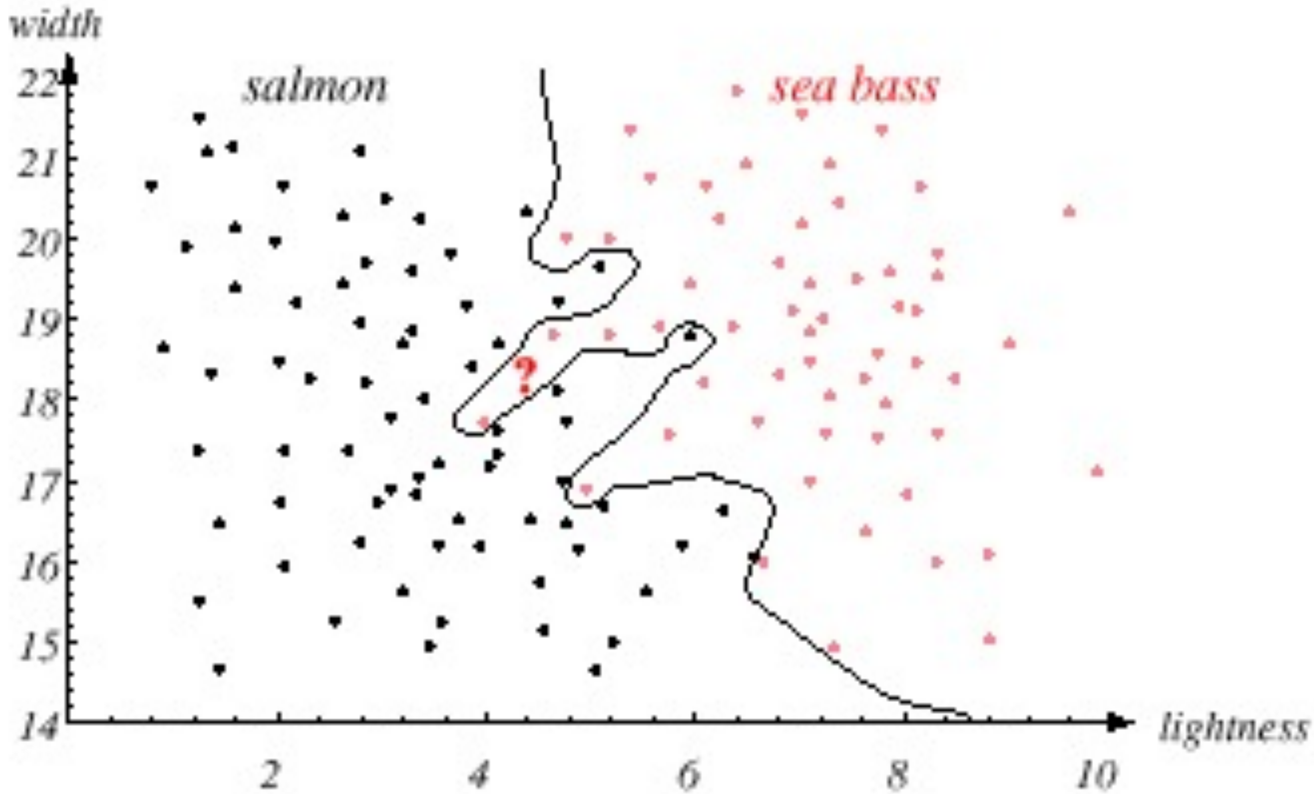
Width And Lightness

- Treat features as a N-tuple (two-dimensional vector)
- Create a scatter plot
- Draw a line (regression) separating the two classes



Decision Theory

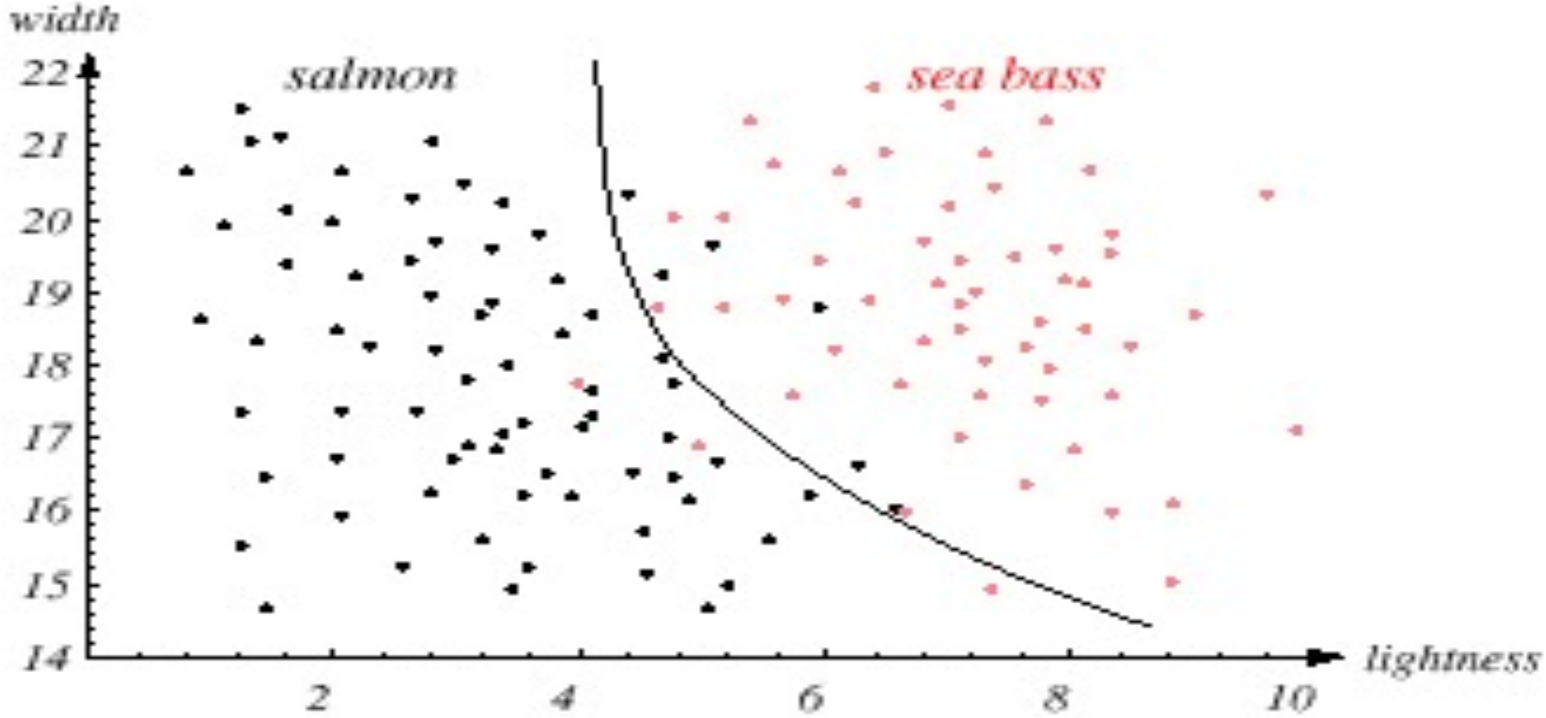
- **Can we do better than a linear classifier?**



- **What is wrong with this decision surface? (Hint: generalization)**

Generalization and Risk Revisited

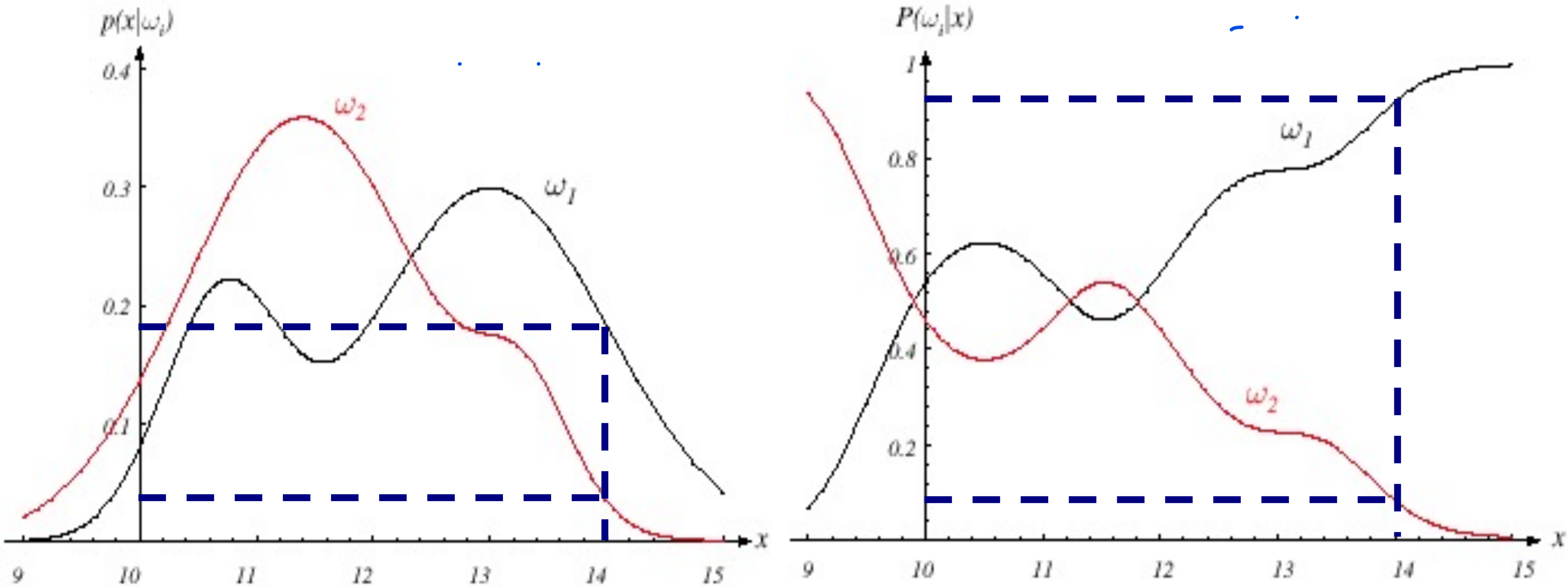
- **Why might a smoother decision surface be a better choice? (Hint: Occam's Razor).**



- **This course investigates how to find such "optimal" decision surfaces and how to provide system designers with the tools to make intelligent trade-offs.**

Posteriors Sum To 1.0

- Two-class fish sorting problem *($P(\omega_1) = 2/3$, $P(\omega_2) = 1/3$)*:



- For every value of *x*, the posteriors sum to 1.0.
- At *x=14*, the probability *x* is in category ω_1 is 0.92.
- The probability *x* is in ω_2 is 0.08.
- Likelihoods and posteriors are related via Bayes Rule.

Summary

- **Probability Decision Theory:** allows us to quantify the tradeoffs between various classification decisions using probability and the costs that accompany these decisions.
- **Prior Probabilities:** reflect our knowledge of the problem, which comes from "subject matter expertise."
- **Likelihoods:** A model that assesses the probability a specific feature vector could have occurred from a specific class.
- **Posterior Probabilities:** the probability a class occurred given a specific feature vector (converts a measurement to a probability that it came from a specific class).
- **Bayes Rule**: factors a posterior into a combination of a likelihood, prior and the evidence. Is this the only appropriate engineering model?

Univariate Normal Distribution

- **A normal or Gaussian density is a powerful model for modeling continuous-valued feature vectors corrupted by noise due to its analytical tractability.**
- **Univariate normal distribution:**

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

12

x = data/observation

μ → mean , σ = STD

$P(x \mid \omega_1)$

$P(x \mid \omega_2)$

where the mean and covariance are defined by:

$$\mu \equiv E[x] = \int_{-\infty}^{\infty} x p(x) dx$$

$$\sigma^2 \equiv E[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx$$

Likelihood:

$P(x \mid \mu_1, \sigma_1)$

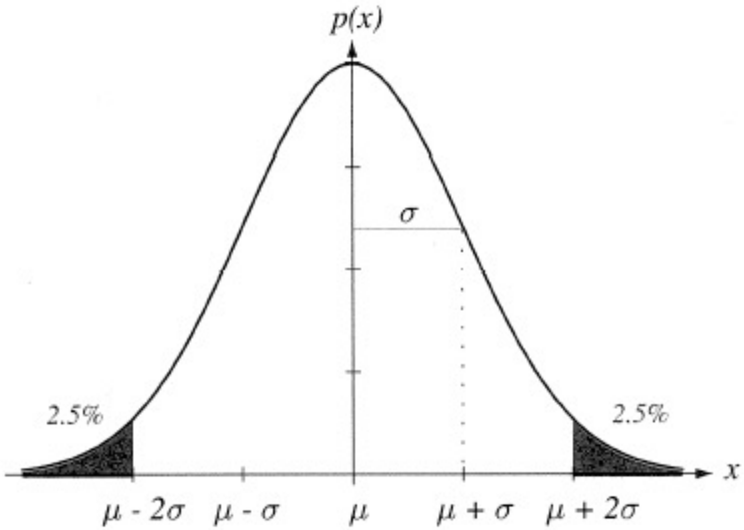
$P(x \mid \mu_2, \sigma_2)$

- **The entropy of a univariate normal distribution is given by:**

$$H(p(x)) = -\int_{-\infty}^{\infty} p(x) \ln p(x) dx = \frac{1}{2}\log(2\pi e \sigma^2)$$

Mean and Variance

- **A normal distribution is completely specified by its mean and variance:**



- **The peak is at:**

$$\boxed{p(\mu) = \frac{1}{\sqrt{2\pi}\sigma}}$$

- **66% of the area is within one σ; 95% is within two σ; 99% is within three σ.**

$\mu \pm \sigma$ 66%
$\mu \pm 2\sigma$ 95%
$\mu \pm 3\sigma$ 99%

- **A normal distribution achieves the maximum entropy of all distributions having a given mean and variance.**
- **Central Limit Theorem: The sum of a large number of small, independent random variables will lead to a Gaussian distribution.**

Multivariate Normal Distributions

$d = 3$

$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in R^d$

- **A multivariate distribution is defined as:**

Mahalanobis Distance

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[\frac{-1}{2}\underbrace{(\mathbf{x}-\boldsymbol{\mu})^t \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}_{1\times 1}\right]$$

where μ represents the mean (vector) and Σ represents the covariance (matrix).

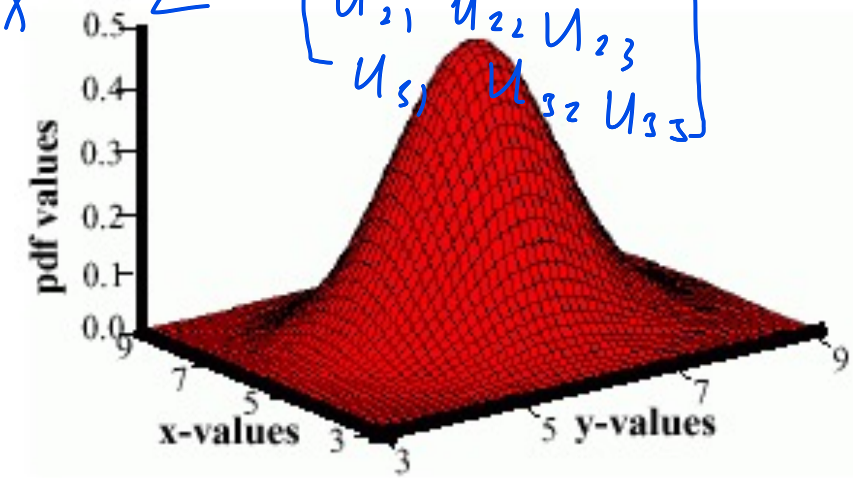
$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}$

- **Note the exponent term is really a dot product or weighted Euclidean distance.**

Covariance Matrix

$\Sigma = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix}$

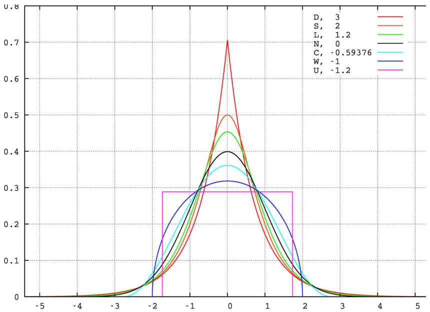
- **The covariance is always symmetric and positive semidefinite matrix.**
- **How does the shape vary as a function of the covariance?**



Multivariate Normal Distributions

- **Recall the definition of a normal distribution (Gaussian):**

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$



- **Why is this distribution so important in engineering?**
- **Mean:** $\boldsymbol{\mu} \equiv E[\mathbf{x}] = \int \mathbf{x}p(\mathbf{x})d\mathbf{x}$
- **Covariance:** $\Sigma \equiv E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^t] = \int (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^t p(\mathbf{x})d\mathbf{x}$

Prior Distributions

- We can encode out beliefs about what the values of the parameters could be using

$$P(\theta)$$

- Using Bayes' rule, we have

likelihood

Prior

$$P(\theta = \theta_0|\text{data}) = \frac{\mathrm{P}(\theta=\theta_0,data)}{P(data)} = \frac{P(data|\theta = \theta_0)P(\theta=\theta_0)}{P(data)}$$

$$= \sum_{\theta_1} P(data|\ \theta = \theta_1)\,P(\theta = \theta_1)$$

Maximum a-posteriori (MAP)

- Maximize the *posterior probability* of the parameter:

$$argmax_{\theta_0} \frac{P(data|\theta = \theta_0)P(\theta=\theta_0)}{P(data)}$$

$$= argmax_{\theta_0} P(data|\theta = \theta_0)P(\theta = \theta_0)$$

$$= argmax_{\theta_0} \log P(data|\theta = \theta_0) + \log P(\theta = \theta_0)$$

- The posterior of probability is the product of the prior and the data likelihood
- Represents the *updated* belief about the parameter, given the observed data

Aside: Bayesian Inference is a Powerful Idea

- You can think about anything like that. You have your prior belief $P(\theta)$, and you observe some new data. Now your belief about θ *must be* proportional to $P(\theta)P(data|\theta)$
  - But only if you are 100% sure that the likelihood function is correct!
  - Recall that the likelihood function is your model of the world – it represents knowledge about how the data is generated for given values of θ
  - Where do you get your original prior beliefs anyway?
- Arguably, makes more sense than Maximum Likelihood

Gaussian Residuals Models

- Log-likelihood:

$$\log P(data|\theta) = \sum -\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2} - \frac{\mathrm{m}}{2}\log(2\pi\sigma^2)$$

- Suppose we believe that $P(\theta_i) = N\left(0, \left(\frac{1}{2\lambda}\right)\right)$
  - I.e., the coefficients in θ will generally be in $[-1.5/\lambda, 1.5/\lambda]$
- $\log[P(data|\theta)P(\theta)]$ is $\log P(data|\theta) - \lambda|\theta|^2 + const$ (exercise)
- Maximize $\log[P(data|\theta)P(\theta)]$ to get the θ that you believe the most

Why $P(\theta_i) = N\left(0, \left(\frac{1}{2\lambda}\right)\right)$

- More on this later
- If the θ_i's are allowed to be arbitrarily large, the ratio of the influences of the different features over the decision boundary could be arbitrarily high
  - Difficult to believe that one of the features still matters, but it matters a 10000000 times less than some other feature
    - Easy to believe a feature doesn't matter at all, though
    - Only reasonable if the inputs are all on the same scale, and the output is on roughly the same scale as the inputs
  - Mostly when we fit coefficients, they don't get crazy high, so it's a reasonable prior belief

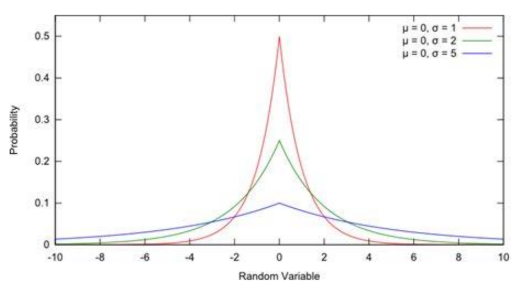
L2 Regularization

- L2 regularization:
  maximize: $\log P(data|\theta) - \lambda|\theta|^2 + const$
- "L2 regularization" because numerically, the cost function penalizes the L2 norm of θ
- A way of preventing overfitting
  - If we set λ to be very high, θ will just be 0: the performance on the training and test sets will be the same (and will be bad)
  - If we set λ to be moderately high, we won't let θ_i's be too large even if that leads to good performance on the training set. Idea: if the training set makes a θ_i very large, that probably won't be good for test set performance, since usually large θ_i 's lead to poor performance
- What about other norms?

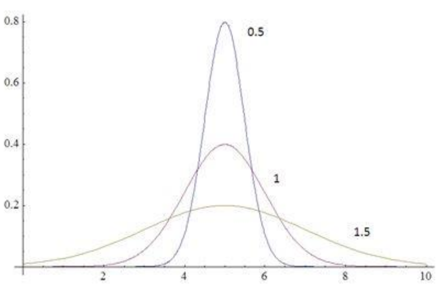
L1 Regularization

- Alternative: L1 regularization:
  maximize: $\log P(data|\theta) - \lambda|\theta|_1 + const$
- Equivalent to using a Laplacian prior:

L1 = Laplacian prior



L2 = Gaussian prior



- Encourages sparsity (feature selection)
  - Sparsity: most θ_i are zero

Image credit: http://slideplayer.com/slide/8012755/

L2 vs L1 regularization

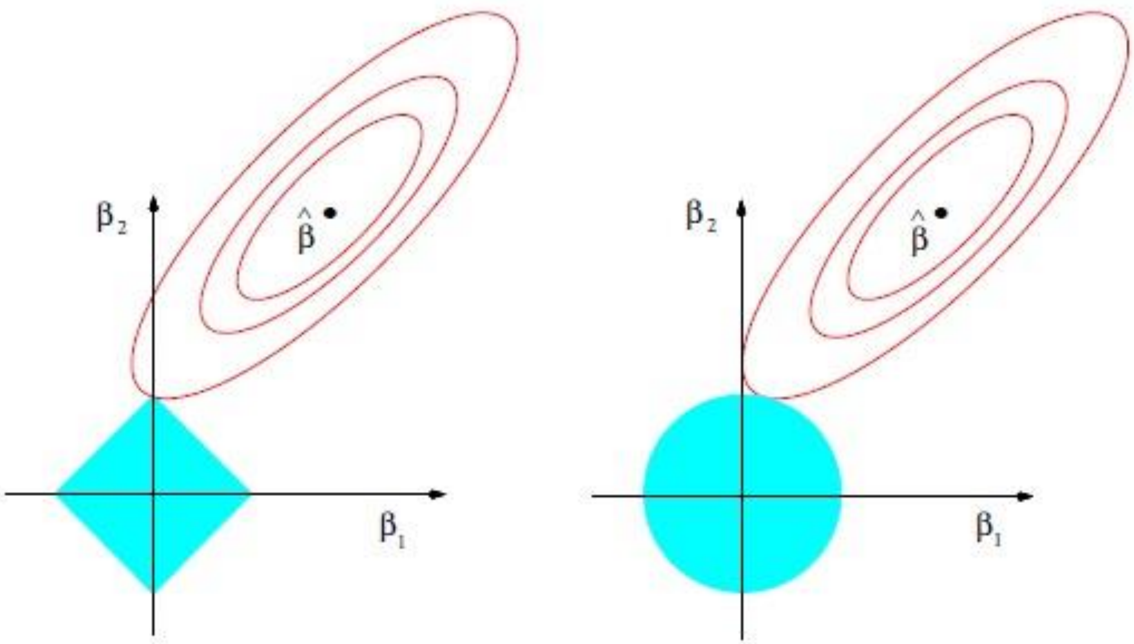


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Early Stopping

- Initialize the θs to small initial values
- Run Gradient Descent, but stop early
  - Before finding the minimum of the cost function applied to the training set