Curve Fitting & Multisensory Integration

Using Probability Theory
Overheard at Porters

“You’re optimizing for the wrong thing!”
What would you say?

What makes for a good model?

Best performance?
Fewest assumptions?
Most elegant?
Coolest name?
Agenda

Motivation
Finding Patterns

Tools
Probability Theory

Applications
Curve Fitting
Multimodal Sensory Integration
The Motivation

FINDING PATTERNS
You have data... now find patterns
You have data... now find patterns

Unsupervised

\[ x = \text{data (training)} \]

\[ y(x) = \text{model} \]

Clustering
Density estimation
You have data... now find patterns

**Unsupervised**

\[ x = \text{data (training)} \]

\[ y(x) = \text{model} \]

- Clustering
- Density estimation

**Supervised**

\[ x = \text{data (training)} \]

\[ t = \text{(target vector)} \]

\[ y(x) = \text{model} \]

- Classification
- Regression
You have data... now find patterns

Unsupervised

\( x \) = data (training)

\( y(x) \) = model

Clustering
Density estimation

Supervised

\( x \) = data (training)

\( t \) = (target vector)

\( y(x) \) = model

Classification
Regression
Important Questions

What kind of model is appropriate?
What makes a model accurate?
Can a model be too accurate?
What are our prior beliefs about the model?
The Tools

PROBABILITY THEORY
Properties of a distribution

\[ x \] = event

\[ p(x) \] = prob. of event

1. \[ p(x) \geq 0 \]

2. \[ \int p(x) \, dx = 1 \]
Rules

**Sum**

\[ p(X) = \sum_{Y} p(X, Y) \]

\[ p(x) = \int p(x, y) \, dy \]

**Product**

\[ p(X, Y) = p(Y|X)p(X) \]

\[ p(x, y) = p(y|x)p(x) \]
Example (Discrete)
Bayes Rule (review)

\[ p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \]
# Probability Density vs. Mass Function

\[ \int_{a}^{b} f(x) \, dx \]

<table>
<thead>
<tr>
<th></th>
<th>PDF</th>
<th>PMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuous</td>
<td>discrete</td>
<td>discrete</td>
</tr>
<tr>
<td>Intuition: How much probability ( f(x) ) concentrated near ( x ) per length ( dx ), how dense is probability near ( x )</td>
<td>Intuition: Probability mass is same interpretation but from discrete point of view: ( f(x) ) is probability for each point, whereas in PDF ( f(x) ) is probability for an interval ( dx )</td>
<td></td>
</tr>
<tr>
<td>Notation: ( p(x) )</td>
<td>Notation: ( P(x) )</td>
<td></td>
</tr>
<tr>
<td>( p(x) ) can be greater than 1, as long as integral over entire interval is 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Expectation & Covariance

\[ \mathbb{E}[f] = \sum_x p(x) f(x) \]

\[ \mathbb{E}[f] = \int p(x) f(x) \, dx \]

\[ \text{var}[f] = \mathbb{E} \left[ (f(x) - \mathbb{E}[f(x)])^2 \right] \]

\[ \text{cov}[x, y] = \mathbb{E}_{x,y} \left\{ (x - \mathbb{E}[x]) (y - \mathbb{E}[y]) \right\} = \mathbb{E}_{x,y}[xy] - \mathbb{E}[x] \mathbb{E}[y] \]
CURVE FITTING

Application
Curve Fitting

Observed Data: Given $n$ points $(x,y)$
Curve Fitting

Observed Data: Given n points (x,t)

Textbook Example): generate x uniformly from range [0,1] and calculate target data t with \( \sin(2\pi x) \) function + noise with Gaussian distribution

Why?
Curve Fitting

Observed Data: Given n points (x,t)

Textbook Example): generate $x$ uniformly from range [0,1] and calculate target data $t$ with $\sin(2\pi x)$ function + noise with Gaussian distribution

Why?
Real data sets typically have underlying regularity that we are trying to learn.
Curve Fitting

Observed Data: Given n points (x,y)

Goal: use observed data to predict new target values t’ for new values of x’
Curve Fitting

Observed Data: Given n points \((x,y)\)
Can we fit a polynomial function with this data?

\[
y(x, w) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j
\]

What values for \(w\) and \(M\) fits this data well?
How to measure goodness of fit?

Minimize an error function

\[ E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 \]

Sum of squares of error
Overfitting

\[ y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \]
Combatting Overfitting

1. Increase data points
Combatting Overfitting

1. Increase data points
   Data observations may have just been noisy
   With more data, can see if data variation is due to noise or if is part of underlying relationship between observations
Combatting Overfitting

1. Increase data points

   Data observations may have just been noisy

   With more data, can see if data variation is due to noise or if is part of underlying relationship between observations

2. Regularization

   Introduce penalty term

   Trade off between good fit and penalty

   Hyperparameter, $\lambda$, is input to model. Hyperparam will reduce overfitting, in turn reducing variance and increasing bias (difference between estimated and true target)

\[
\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2} \|w\|^2
\]
How to check for overfitting?

Training and validation subset
heuristic: data points > (multiple)(# parameters)

Training vs Testing
don’t touch test set until you are actually evaluating experiment!!
Cross-validation

1. Use portion, \((S-1)/S\), for training (white)
2. Assess performance (red)
3. Repeat for each run
4. Average performance scores

4-fold cross-validation (\(S=4\))
Cross-validation

When to use?
Cross-validation

When to use?
Validation set is small. If very little data, use $S=$ number of observed data points
Cross-validation

When to use?
Validation set is small. If very little data, use $S=$number of observed data points

Limitations:
- Computationally expensive - # of training runs increases by factor of $S$
- Might have multiple complexity parameters - may lead to training runs that is exponential in # of parameters
Alternative Approach:

Want an approach that depends only on size of training data rather than # of training runs

e.g.) Akaike information criterion (AIC), Bayesian information criterion (BIC, Sec 4.4)
Akaike Information Criterion (AIC)

Choose model with largest value for

$$\ln p(D|w_{ML}) - M$$

- best-fit log likelihood
- # of adjustable model parameters
Gaussian Distribution

\[ \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\} \]

This satisfies the two properties for a probability density! (what are they?)
Likelihood Function for Gaussian

Assumption: data points $x$ drawn \textit{independently} from same Gaussian distribution defined by unknown mean and variance parameters, i.e. independently and identically distributed (i.i.d)
Curve Fitting (ver. 1)

Assumption:
1. Given value of \( x \), corresponding target value \( t \) has a Gaussian distribution with mean \( y(x, w) \)

\[
y(x, w) = w_0 + w_1x + w_2x^2 + \ldots + w_Mx^M = \sum_{j=0}^{M} w_jx^j
\]

2. Data \{x, t\} drawn independently from distribution:

\[
\text{likelihood} = p(t|x, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, w), \beta^{-1})
\]
Maximum Likelihood Estimation (MLE)

Log likelihood: \[ \ln p(t|x, w, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \left\{ y(x_n, w) - t_n \right\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) \]

What does maximizing the log likelihood look similar to?
Maximum Likelihood Estimation (MLE)

Log likelihood: \( \ln p(t|x, w, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) \)

What does maximizing the log likelihood look similar to?

wrt \( w \): \[ E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 \]
Maximum Posterior (MAP)

Simpler example: use Gaussian of form

$$p(w|\alpha) = \mathcal{N}(w|0, \alpha^{-1}I) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}w^Tw\right\}$$

With Bayes’ can calculate posterior:

$$p(w|x, t, \alpha, \beta) \propto p(t|x, w, \beta)p(w|\alpha)$$
Maximum Posterior (MAP)

Determine $w$ by maximizing posterior distribution

Equivalent to taking negative log of:

$$p(w|x, t, \alpha, \beta) \propto p(t|x, w, \beta)p(w|\alpha)$$

and combining the Gaussian & log likelihood function from earlier...
Maximum Posterior (MAP)

Minimum of $\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\alpha}{2} w^T w$

What does this look like?
Maximum Posterior (MAP)

Minimum of \[
\beta \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\alpha}{2} w^T w
\]

What does this look like?

\[
\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2} \|w\|^2
\]

What is regularization parameter?
Maximum Posterior (MAP)

Minimum of

\[ \beta \sum_{n=1}^{N} \left\{ y(x_n, w) - t_n \right\}^2 + \frac{\alpha}{2} w^T w \]

What does this look like?

\[ \tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \left\{ y(x_n, w) - t_n \right\}^2 + \frac{\lambda}{2} \| w \|^2 \]

What is regularization parameter?  \[ \lambda = \alpha / \beta \]
Bayesian Curve Fitting

However...

MLE and MAP are not fully Bayesian because they involve using point estimates for $w$. 
Curve Fitting (ver. 2)

Given training data \{x, t\} and new point x, predict the target value t

Assume parameters $\alpha$ and $\beta$ are fixed

Evaluate predictive distribution: $p(t|x, x, t)$
Bayesian Curve Fitting

Fully Bayesian approach requires integrating over all values of $w$, by applying sum and product rules of probability

$$p(t|x, x, t) = \int p(t|x, w)p(w|x, t) \, dw$$
Bayesian Curve Fitting

$$p(t | x, x, t) = \int p(t | x, w) p(w | x, t) \, dw$$

$$\mathcal{N}(t | y(x, w), \beta^{-1})$$

This posterior is Gaussian and can be evaluated analytically (Sec 3.3)
Bayesian Curve Fitting

Predictive is Gaussian of form

$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}(t|m(x), s^2(x))$$

with mean and variance and matrix

$$m(x) = \beta \phi(x)^T S \sum_{n=1}^{N} \phi(x_n) t_n$$

$$s^2(x) = \beta^{-1} + \phi(x)^T S \phi(x).$$

$$S^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^{N} \phi(x_n) \phi(x)^T$$
Bayesian Curve Fitting

Need to define $\phi_i(x) = x^i$ for $i = 0, \ldots, M$

Mean and variance depend on $x$ as a result of marginalization

$$m(x) = \beta \phi(x)^T S \sum_{n=1}^{N} \phi(x_n) t_n$$

$$s^2(x) = \beta^{-1} + \phi(x)^T S \phi(x).$$

(not the case in MLE/MAP $\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2} ||w||^2$)
Application

MULTIMODAL SENSORY INTEGRATION
Two Models

Visual Capture
Two Models
Visual Capture

MLE

(a)
probability of location given visual signal
optimal location estimate given both signals
probability of location given auditory signal

(b)
probability of location given visual signal
optimal location estimate given both signals
probability of location given auditory signal

location
Procedure

(a) 

(b) 

(c) 

![Graph showing the percent of trials perceived as "light" vs. comparison visual angle for auditory and visual noise conditions.](chart.png)
Final Result

\[ w_v = \frac{1}{\sigma_v^2} \] and \[ w_a = \frac{1}{\sigma_a^2} \]
The Math (MLE Model)

Likelihood

\[
p(R|\mu, \sigma^2) = \prod_{t=1}^{T} p_t r_t^{i} (1 - p_t)^{1-r_t}
\]

\[
p_t = p(r_t|\mu, \sigma^2)
\]
The Math (MLE Model)

Likelihood

\[ p(R|\mu, \sigma^2) = \prod_{t=1}^{T} p_t^{r_t}(1 - p_t)^{1-r_t} \]

\[ p_t = p(r_t|\mu, \sigma^2) \]

\[ w_v = \frac{1/\sigma_v^2}{1/\sigma_v^2 + 1/\sigma_a^2} \quad \text{and} \quad w_a = \frac{1/\sigma_a^2}{1/\sigma_v^2 + 1/\sigma_a^2} \]
The Math (“Bayesian” Model)

\[ p(R|\mu, \sigma^2) \ p(\mu, \sigma^2) \]

Likelihood * Prior

Uniform

Inverse Gamma
The Math (Empirical)

Likelihood:

\[ p(\mathcal{R}|\mu, \sigma^2) = \prod_{t=1}^{T} p_t^r(1 - p_t)^{1-r_t} \]

Logistic function:

\[ p_t = p(r_t = 1|w_v, w_a) = \frac{1}{1 + \exp[-(L_c - L_s)/\tau]} \]

Location estimates:

\[ L^c = w_v L^c_v + w_a L^c_a \]
\[ L^s = w_v L^s_v + w_a L^s_a \]

Weight constraint:

\[ w_v + w_a = 1 \]
Final Result
Two Models (Prediction)

Visual Capture

MLE

Visual Weight

Visual Noise

Visual Weight

Visual Noise