Lecture 19 of 42

MAP and MLE continued, Minimum Description Length (MDL)

Wednesday, 28 February 2007

William H. Hsu
Department of Computing and Information Sciences, KSU
http://www.kddresearch.org

Readings for next class:
Chapter 5, Mitchell

Lecture Outline

• Read Sections 6.1-6.5, Mitchell
• Overview of Bayesian Learning
  – Framework: using probabilistic criteria to generate hypotheses of all kinds
  – Probability: foundations
• Bayes’s Theorem
  – Definition of conditional (posterior) probability
  – Ramifications of Bayes’s Theorem
    • Answering probabilistic queries
    • MAP hypotheses
• Generating Maximum A Posteriori (MAP) Hypotheses
• Generating Maximum Likelihood Hypotheses
• Next Week: Sections 6.6-6.13, Mitchell; Roth; Pearl and Verma
  – More Bayesian learning: MDL, BOC, Gibbs, Simple (Naïve) Bayes
  – Learning over text
Choosing Hypotheses

- **Bayes’s Theorem**
  \[ P(h | D) = \frac{P(D | h) P(h)}{P(D)} \]

- **MAP Hypothesis**
  - Generally want most probable hypothesis given the training data
  - Define: \( \text{arg max}_{x \in \Omega} f(x) \) as the value of \( x \) in the sample space \( \Omega \) with the highest \( f(x) \)
  - Maximum a posteriori hypothesis, \( h_{MAP} \)
    \[ h_{MAP} = \text{arg max}_{h \in H} P(h | D) = \text{arg max}_{h \in H} \frac{P(D | h) P(h)}{P(D)} \]

- **ML Hypothesis**
  - Assume that \( p(h_i) = p(h_j) \) for all pairs \( i, j \) (uniform priors, i.e., \( p_H \sim \text{Uniform} \))
  - Can further simplify and choose the maximum likelihood hypothesis, \( h_{ML} \)
    \[ h_{ML} = \text{arg max}_{h_i \in H} P(D | h_i) \]

Bayes’s Theorem: Query Answering (QA)

- **Answering User Queries**
  - Suppose we want to perform intelligent inferences over a database \( DB \)
    - Scenario 1: \( DB \) contains records (instances), some "labeled" with answers
    - Scenario 2: \( DB \) contains probabilities (annotations) over propositions
  - QA: an application of probabilistic inference

- **QA Using Prior and Conditional Probabilities: Example**
  - Query: *Does patient have cancer or not?*
  - Suppose: patient takes a lab test and result comes back positive
    - Correct + result in only 98% of the cases in which disease is actually present
    - Correct - result in only 97% of the cases in which disease is not present
    - Only 0.008% of the entire population has this cancer
  - \( \alpha = P(\text{false negative for } H_0 = \text{Cancer}) = 0.02 \) (NB: for 1-point sample)
  - \( \beta = P(\text{false positive for } H_0 = \text{Cancer}) = 0.03 \) (NB: for 1-point sample)
  - \[ P(\text{Cancer}) = 0.008, P(+ | \text{Cancer}) = 0.98 \]
  - \( P(\text{¬Cancer}) = 0.992, P(- | \text{¬Cancer}) = 0.97 \)
  - \( P(D) = 0.0076, P(D) \cdot P(\text{H}_A) = 0.0080 \Rightarrow \text{h}_{MAP} = H_A = \neg \text{Cancer} \)
Basic Formulas for Probabilities

- **Product Rule** (Alternative Statement of Bayes’s Theorem)
  \[ P(A|B) = \frac{P(A \land B)}{P(B)} \]
  - Proof: requires axiomatic set theory, as does Bayes’s Theorem

- **Sum Rule**
  \[ P(A \lor B) = P(A) + P(B) - P(A \land B) \]
  - Sketch of proof (immediate from axiomatic set theory)
    - Draw a Venn diagram of two sets denoting events A and B
    - Let \( A \lor B \) denote the event corresponding to \( A \lor B \)

- **Theorem of Total Probability**
  - Suppose events \( A_1, A_2, \ldots, A_n \) are mutually exclusive and exhaustive
    - Mutually exclusive: \( i \neq j \Rightarrow A_i \land A_j = \emptyset \)
    - Exhaustive: \( \sum P(A_i) = 1 \)
  - Then \( P(B) = \sum P(B|A_i)P(A_i) \)
  - Proof: follows from product rule and 3rd Kolmogorov axiom

MAP and ML Hypotheses: A Pattern Recognition Framework

- **Pattern Recognition Framework**
  - Automated speech recognition (ASR), automated image recognition
  - Diagnosis

- **Forward Problem**: One Step in ML Estimation
  - Given: model \( h \), observations (data) \( D \)
  - Estimate: \( P(D | h) \), the “probability that the model generated the data”

- **Backward Problem**: Pattern Recognition / Prediction Step
  - Given: model \( h \), observations \( D \)
  - Maximize: \( P(h(X) = x | h, D) \) for a new \( X \) (i.e., find best \( x \))

- **Forward-Backward (Learning) Problem**
  - Given: model space \( H \), data \( D \)
  - Find: \( h \in H \) such that \( P(h | D) \) is maximized (i.e., MAP hypothesis)

- **More Info**
  - Emphasis on a particular \( H \) (the space of hidden Markov models)
Bayesian Learning Example: Unbiased Coin [1]

- **Coin Flip**
  - Sample space: \( \Omega = \{\text{Head}, \text{Tail}\} \)
  - Scenario: given coin is either fair or has a 60% bias in favor of Head
    - \( h_1 \) = fair coin: \( P(\text{Head}) = 0.5 \)
    - \( h_2 \) = 60% bias towards Head: \( P(\text{Head}) = 0.6 \)
  - Objective: to decide between default (null) and alternative hypotheses
- **A Priori (aka Prior) Distribution on \( H \)**
  - \( P(h_1) = 0.75, P(h_2) = 0.25 \)
  - Reflects learning agent’s prior beliefs regarding \( H \)
  - Learning is revision of agent’s beliefs
- **Collection of Evidence**
  - First piece of evidence: \( d = \) a single coin toss, comes up Head
  - Q: What does the agent believe now?
  - A: Compute \( P(d) = P(d | h_1) P(h_1) + P(d | h_2) P(h_2) \)
    - \( P(\text{Head}) = 0.5 \cdot 0.75 + 0.6 \cdot 0.25 = 0.375 + 0.15 = 0.525 \)
    - This is the probability of the observation \( d = \text{Head} \)
- **Bayesian Learning**
  - Now apply Bayes’s Theorem
    - \( P(h_1 | d) = P(d | h_1) P(h_1) / P(d) = 0.375 / 0.525 = 0.714 \)
    - \( P(h_2 | d) = P(d | h_2) P(h_2) / P(d) = 0.15 / 0.525 = 0.286 \)
    - Belief has been revised downwards for \( h_1 \), upwards for \( h_2 \)
    - The agent still thinks that the fair coin is the more likely hypothesis
  - Suppose we were to use the ML approach (i.e., assume equal priors)
    - Belief is revised upwards from 0.5 for \( h_1 \)
    - Data then supports the bias coin better

Bayesian Learning Example: Unbiased Coin [2]

- **Bayesian Inference**: Compute \( P(d) = P(d | h_1) P(h_1) + P(d | h_2) P(h_2) \)
  - \( P(\text{Head}) = 0.5 \cdot 0.75 + 0.6 \cdot 0.25 = 0.375 + 0.15 = 0.525 \)
  - This is the probability of the observation \( d = \text{Head} \)
- **Bayesian Learning**
  - Now apply Bayes’s Theorem
    - \( P(h_1 | d) = P(d | h_1) P(h_1) / P(d) = 0.375 / 0.525 = 0.714 \)
    - \( P(h_2 | d) = P(d | h_2) P(h_2) / P(d) = 0.15 / 0.525 = 0.286 \)
    - Belief has been revised downwards for \( h_1 \), upwards for \( h_2 \)
    - The agent still thinks that the fair coin is the more likely hypothesis
  - Suppose we were to use the ML approach (i.e., assume equal priors)
    - Belief is revised upwards from 0.5 for \( h_1 \)
    - Data then supports the bias coin better
- **More Evidence**: Sequence \( D \) of 100 coins with 70 heads and 30 tails
  - \( P(D) = (0.5)^{70} \cdot (0.5)^{30} \cdot 0.75 + (0.6)^{70} \cdot (0.4)^{30} \cdot 0.25 \)
  - Now \( P(h_1 | d) \ll P(h_2 | d) \)
Brute Force MAP Hypothesis Learner

- **Intuitive Idea:** Produce Most Likely $h$ Given Observed $D$

- **Algorithm Find-MAP-Hypothesis ($D$)**
  
  1. FOR each hypothesis $h \in H$
  
  Calculate the conditional (i.e., posterior) probability:

  \[ P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \]

  2. RETURN the hypothesis $h_{MAP}$ with the highest conditional probability

  \[ h_{MAP} = \arg \max_{h \in H} P(h \mid D) \]

Relation to Concept Learning

- **Usual Concept Learning Task**
  
  - Instance space $X$
  - Hypothesis space $H$
  - Training examples $D$

- **Consider Find-S Algorithm**
  
  - Given: $D$
  
  - Return: most specific $h$ in the version space $VS_{H,D}$

- **MAP and Concept Learning**
  
  - Bayes’s Rule: Application of Bayes’s Theorem
  
  - What would Bayes’s Rule produce as the MAP hypothesis?

- **Does Find-S Output A MAP Hypothesis?**
Bayesian Concept Learning and Version Spaces

- **Assumptions**
  - Fixed set of instances <x₁, x₂, ..., xₘ>
  - Let D denote the set of classifications: D = <c(x₁), c(x₂), ..., c(xₘ)>
- **Choose P(D | h)**
  - P(D | h) = 1 if h consistent with D (i.e., ∀ xᵢ, h(xᵢ) = c(xᵢ))
  - P(D | h) = 0 otherwise
- **Choose P(h) ~ Uniform**
  - Uniform distribution: P(h) = \frac{1}{|H|}
  - Uniform priors correspond to "no background knowledge" about h
  - Recall: maximum entropy
- **MAP Hypothesis**
  \[
P(h | D) = \begin{cases} 
  \frac{1}{|\text{VS}_{h,D}|} & \text{if } h \text{ is consistent with } D \\
  0 & \text{otherwise} 
\end{cases}
\]

Evolution of Posterior Probabilities

- **Start with Uniform Priors**
  - Equal probabilities assigned to each hypothesis
  - Maximum uncertainty (entropy), minimum prior information

  \[P(h) \rightarrow P(h | D₁) \rightarrow P(h | D₁, D₂)\]

- **Evidential Inference**
  - Introduce data (evidence) D₁: belief revision occurs
    - Learning agent revises conditional probability of inconsistent hypotheses to 0
    - Posterior probabilities for remaining h ∈ VS_{h,D} revised upward
  - Add more data (evidence) D₂: further belief revision
Characterizing Learning Algorithms by Equivalent MAP Learners

Inductive System

Training Examples $D$

Hypothesis Space $H$

Candidate Elimination Algorithm

Output hypotheses

Equivalent Bayesian Inference System

Training Examples $D$

Hypothesis Space $H$

Brute Force MAP Learner

Output hypotheses

Prior knowledge made explicit

Problem Definition

- Target function: any real-valued function $f$
- Training examples $<x_i, y_i>$ where $y_i$ is noisy training value
  - $y_i = f(x_i) + e_i$
  - $e_i$ is random variable (noise) i.i.d. ~ Normal ($0$, $\sigma$), aka Gaussian noise
- Objective: approximate $f$ as closely as possible

Solution

- Maximum likelihood hypothesis $h_{ML}$
- Minimizes sum of squared errors (SSE)

$$h_{ML} = \arg \min_{h \in H} \sum_{i=1}^{m} (y_i - h(x_i))^2$$
Maximum Likelihood: Learning A Real-Valued Function [2]

- Derivation of Least Squares Solution
  - Assume noise is Gaussian (prior knowledge)
  - Max likelihood solution: 
    \[ h_{\text{ML}} = \arg \max_{h} p(D | h) \]
    \[ = \arg \max_{h} \prod_{i=1}^{n} p(d_i | h) \]
    \[ = \arg \max_{h} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(d_i - h(x_i))^2}{2\sigma^2}} \]

- Problem: Computing Exponents, Comparing Reals - Expensive!
- Solution: Maximize Log Prob

\[
    h_{\text{ML}} = \arg \max_{h} \sum_{i=1}^{n} \ln \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2} \left( \frac{d_i - h(x_i)}{\sigma} \right)^2 \\
    = \arg \max_{h} \sum_{i=1}^{n} \left( \frac{d_i - h(x_i)}{\sigma} \right)^2 \\
    = \arg \min_{h} \sum_{i=1}^{n} (d_i - h(x_i))^2
\]

Learning to Predict Probabilities

- Application: Predicting Survival Probability from Patient Data
- Problem Definition
  - Given training examples \( <x_i, d_i> \), where \( d_i \in H = \{0, 1\} \)
  - Want to train neural network to output a probability given \( x_i \) (not a 0 or 1)
- Maximum Likelihood Estimator (MLE)
  - In this case can show:
    \[ h_{\text{ML}} = \arg \max_{h} \sum_{i=1}^{n} [d_i \ln h(x_i) + (1-d_i) \ln(1-h(x_i))] \]
  - Weight update rule for a sigmoid unit
    \[
    \Delta w_{\text{start-layer, end-layer}} = \sum_{i=1}^{n} (d_i - h(x_i)) \cdot x_i_{\text{start-layer, end-layer}}
    \]
    \[
    w_{\text{start-layer, end-layer}} = w_{\text{start-layer, end-layer}} + \Delta w_{\text{start-layer, end-layer}}
    \]
Most Probable Classification of New Instances

- **MAP and MLE: Limitations**
  - Problem so far: “find the most likely hypothesis given the data”
  - Sometimes we just want the best classification of a new instance \( x \), given \( D \)

- **A Solution Method**
  - Find best (MAP) \( h \), use it to classify
  - *This may not be optimal, though!*
  - Analogy
    - Estimating a distribution using the *mode* versus the *integral*
    - One finds the maximum, the other the area

- **Refined Objective**
  - Want to determine the *most probable classification*
  - Need to *combine* the prediction of all hypotheses
  - Predictions must be *weighted by their conditional probabilities*
  - Result: *Bayes Optimal Classifier (next time…)*

Minimum Description Length (MDL) Principle: Occam’s Razor

- **Occam’s Razor**
  - Recall: prefer the shortest hypothesis - an inductive bias
  - Questions
    - Why short hypotheses as opposed to an arbitrary class of rare hypotheses?
    - What is special about minimum description length?
  - Answers
    - MDL approximates an *optimal coding strategy* for hypotheses
    - *In certain cases*, this coding strategy maximizes conditional probability
  - Issues
    - How exactly is “minimum length” being achieved (length of what)?
    - When and why can we use “MDL learning” for MAP hypothesis learning?
    - *What does “MDL learning” really entail (what does the principle buy us)?*

- **MDL Principle**
  - Prefer \( h \) that minimizes *coding length of model plus coding length of exceptions*
  - *Model*: encode \( h \) using a *coding scheme* \( C_1 \)
  - *Exceptions*: encode the *conditioned* data \( D \mid h \) using a *coding scheme* \( C_2 \)
MDL and Optimal Coding: Bayesian Information Criterion (BIC)

- **MDL Hypothesis**
  \[ h_{\text{MDL}} = \arg \min_{h \in H} L_C(h) + L_{C_2}(D \mid h) \]
  - e.g., \( H \) = decision trees, \( D \) = labeled training data
  - \( L_C(h) \) = number of bits required to describe tree \( h \) under encoding \( C_1 \)
  - \( L_{C_2}(D \mid h) \) = number of bits required to describe \( D \) given \( h \) under encoding \( C_2 \)
  - NB: \( L_{C_2}(D \mid h) = 0 \) if all \( x \) classified perfectly by \( h \) (need only describe exceptions)
  - Hence \( h_{\text{MDL}} \) trades off tree size against training errors

- **Bayesian Information Criterion**
  \[ BIC(h) = \log P(D \mid h) + \log P(h) \]
  - \( h_{\text{MAP}} = \arg \max_{h \in H} [\log P(D \mid h) \cdot P(h)] \)
  - Interesting fact from information theory: the optimal (shortest expected code length) code for an event with probability \( p \) is \(-\log(p)\) bits
  - Interpret \( h_{\text{MAP}} \) as total length of \( h \) and \( D \) given \( h \) under optimal code
  - \( \text{BIC} = \text{MDL} \) (i.e., \( \arg \max \) of BIC is \( \arg \min \) of MDL criterion)
  - Prefer hypothesis that minimizes \( \text{length}(h) + \text{length(misclassifications)} \)

Concluding Remarks on MDL

- **What Can We Conclude?**
  - Q: Does this prove once and for all that short hypotheses are best?
  - A: Not necessarily...
    - Only shows: if we find log-optimal representations for \( P(h) \) and \( P(D \mid h) \), then \( h_{\text{MAP}} = h_{\text{MDL}} \)
    - No reason to believe that \( h_{\text{MDL}} \) is preferable for arbitrary codings \( C_1, C_2 \)
  - Case in point: practical probabilistic knowledge bases
    - Elicitation of a full description of \( P(h) \) and \( P(D \mid h) \) is hard
    - Human implementor might prefer to specify relative probabilities

- **Information Theoretic Learning: Ideas**
  - Learning as compression
    - Abu-Mostafa: complexity of learning problems (in terms of minimal codings)
    - Wolff: computing (especially search) as compression
  - (Bayesian) model selection: searching \( H \) using probabilistic criteria
Bayesian Classification

- Framework
  - Find most probable classification (as opposed to MAP hypothesis)
  - $f : X \rightarrow V$ (domain = instance space, range = finite set of values)
  - Instances $x \in X$ can be described as a collection of features $x = (x_1, x_2, ..., x_n)$
  - Performance element: Bayesian classifier
    - Given: an example (e.g., Boolean-valued instances: $x_i \in B$)
    - Output: the most probable value $v_j \in V$ (NB: priors for $x$ constant wrt $v_{MAP}$)

- Parameter Estimation Issues
  - Estimating $P(v_j)$ is easy: for each value $v_j$ count its frequency in $D = \{<x, f(x)>\}$
  - However, it is infeasible to estimate $P(x \mid x_1, x_2, ..., x_n \mid v_j)$: too many 0 values
  - In practice, need to make assumptions that allow us to estimate $P(x \mid D)$

Bayes Optimal Classifier (BOC)

- Intuitive Idea
  - $h_{MAP}(x)$ is not necessarily the most probable classification!
  - Example
    - Three possible hypotheses: $P(h_1 \mid D) = 0.4$, $P(h_2 \mid D) = 0.3$, $P(h_3 \mid D) = 0.3$
    - Suppose that for new instance $x$, $h_1(x) = +$, $h_2(x) = -$, $h_3(x) = -$
    - What is the most probable classification of $x$?
  - Bayes Optimal Classification (BOC)
    - $v^*_{BOC} = \arg \max_{v_j \in V} \sum_{h \in H} |P(v_j \mid h) \cdot P(h \mid D)|$
      - Example
        - $P(h_1 \mid D) = 0.4$, $P(- \mid h_1) = 0$, $P(+ \mid h_1) = 1$
        - $P(h_2 \mid D) = 0.3$, $P(- \mid h_2) = 1$, $P(+ \mid h_2) = 0$
        - $P(h_3 \mid D) = 0.3$, $P(- \mid h_3) = 1$, $P(+ \mid h_3) = 0$
    - Result: $v^*_{BOC} = \arg \max_{v_j \in V} \sum_{h \in H} |P(v_j \mid h) \cdot P(h \mid D)| = -$
Terminology

- Introduction to Bayesian Learning
  - Probability foundations
    - Definitions: subjectivist, frequentist, logicist
    - (3) Kolmogorov axioms
- Bayes's Theorem
  - Prior probability of an event
  - Joint probability of an event
  - Conditional (posterior) probability of an event
- Maximum A Posteriori (MAP) and Maximum Likelihood (ML) Hypotheses
  - MAP hypothesis: highest conditional probability given observations (data)
  - ML: highest likelihood of generating the observed data
  - ML estimation (MLE): estimating parameters to find ML hypothesis
- Bayesian Inference: Computing Conditional Probabilities (CPs) in A Model
- Bayesian Learning: Searching Model (Hypothesis) Space using CPs

Summary Points

- Introduction to Bayesian Learning
  - Framework: using probabilistic criteria to search H
  - Probability foundations
    - Definitions: subjectivist, objectivist; Bayesian, frequentist, logicist
    - Kolmogorov axioms
- Bayes’s Theorem
  - Definition of conditional (posterior) probability
  - Product rule
- Maximum A Posteriori (MAP) and Maximum Likelihood (ML) Hypotheses
  - Bayes’s Rule and MAP
  - Uniform priors: allow use of MLE to generate MAP hypotheses
  - Relation to version spaces, candidate elimination
- Next Week: 6.6-6.10, Mitchell; Chapter 14-15, Russell and Norvig; Roth
  - More Bayesian learning: MDL, BOC, Gibbs, Simple (Naïve) Bayes
  - Learning over text