CS 5806 Machine Learning II

Lecture 9 - Statistical Learning 2: Maximum Likelihood Estimation & Maximum A-Posteriori
September 25th, 2023
Hoda Eldardiry

Recommended Reading

- [7] Sec 20.1, 20.2
- [8] Sec. 2.2, 3.2
- References are listed on canvas /pages/textbook-resources

Conditional, Chain & Bayes

Conditional

$$P(A\,|\,B) = P(A \wedge B)/P(B)$$

Conditional **Joint** Marginal

Chain

$$P(A \wedge B) = P(A\,|\,B)P(B)$$

Joint **Conditional** Marginal

Bayes

$$P(B\,|\,A) = [P(A\,|\,B)P(B)]\ /\ P(A)$$

Conditional **Conditional** Marginals

Using Bayes Rule for inference

- **Form a hypothesis about the world based on what we observe**
    - Prior of hypothesis: captures prior belief/uncertainty (before observing)
    - Observe data e
    - Likelihood: probability of observing e given H
- **Bayes rule: states belief of hypothesis H, given evidence e**
- **Posterior: after observing, reduce uncertainty, revised distribution**

Prior probability

Likelihood

$$P(H|e) = \frac{P(e|H)P(H)}{P(e)}$$

Posterior probability

Normalizing constant

Bayesian Learning

- **ML from a statistical perspective:**
  Use Bayes rule to reduce uncertainty
  Find best/correct hypothesis

Likelihood   Prior probability

$$P(H|e) = \frac{P(e|H)P(H)}{P(e)}$$

Posterior probability   Normalizing constant

- — Start with a **Prior** P(H)
  — Observe dataset/**evidence** $e = \langle e_1, e_2, \ldots, e_N \rangle$
  — **Likelihood distribution** P(e|H) Likelihood of obtaining a certain dataset (or evidence) given each hypothesis
  — Use Bayes rule to compute a posterior (encodes what's learned)

$$P(H\,|\,\mathbf{e}) = k\ P(\mathbf{e}\,|\,H)P(H)$$

- Assume equal probability of e for each hypothesis (normalizing constant k)

Bayesian Prediction

- To make a prediction about unknown quantity X

$$P(X \mid \mathbf{e}) = \sum_i P(X \mid \mathbf{e}, h_i) P(h_i \mid \mathbf{e})$$

h_i: model

$$= \sum_i P(X \mid h_i) P(h_i \mid \mathbf{e})$$

predictions

predictions of individual hypothesis

weight: posterior/belief

weighted average of predictions of individual hypotheses

- Posterior of h is used as a weight of h's prediction
- Prediction: weighted average of predictions of individual hypothesis
- Hypotheses serve as "intermediaries" between raw data & prediction

Bayesian Learning

- **Bayesian learning properties**
  – **Optimal** (i.e. given prior, no other prediction is correct more often than the Bayesian one)
  – **No overfitting** (all hypotheses considered and weighted)

- **Limitation**
  – When hypothesis space is large, may be intractable
  – sum (or integral) over hypotheses often intractable

- **Solution**
  – "approximate" Bayesian learning

Maximum a posteriori (MAP)

- The first approximation is MAP
- Key idea
  — Instead of working directly with posterior distribution
  — Pick hypothesis with highest probability in the posterior
  — Then make predictions based on chosen hypothesis
- Intuition
  — Problem: too many hypotheses
  — Instead of considering all & taking a weighted combination
  — Only take hypothesis with highest probability
  — Assuming it may be good enough
  — Then make prediction based on that hypothesis

Maximum a posteriori (MAP)

- **MAP:**
  Make prediction based on <u>most probable hypothesis</u> h_{MAP}

most probable hypothesis
max a-posteriori hypothesis

$$h_{MAP} = argmax_{h_i} P(h_i \,|\, \mathbf{e})$$

approximate prediction

$$P(X \,|\, \mathbf{e}) \approx P(X \,|\, h_{MAP})$$

- **Bayesian learning**
  Make prediction based on <u>all hypotheses weighted by their (posterior) probability</u>

MAP properties

- **Computation/Accuracy tradeoff**
  MAP relies only on one hypothesis h_{MAP}
  + + simpler computationally
  - - MAP prediction less accurate than Bayesian prediction
    [maybe chosen hypothesis is not the best: fitting data very well or overfit]

- **Convergence**
  MAP & Bayesian predictions converge as data increases

- **Overfitting**
  — may not be able to get rid of overfitting completely
  — but we can control overfitting using the prior
  — to avoid hypotheses that are: flexible, complex, can easily capture noise in data
  — prior can be used to penalize such hypotheses [give them lower prior probability]
  — give simpler hypotheses a higher prior probability
  — there is still a chance of overfitting since we only return one hypothesis
  — if the returned hypothesis still captures some noise in the data, then there will be overfitting

- **Tractability**
  + + working with a single hypothesis is a big win computationally
  - - finding hypothesis with highest probability requires solving an optimization problem:
    look for hypothesis that maximizes: $$h_{MAP} = argmax_h P(h \mid \mathbf{e})$$
  - - finding h_{MAP} may be intractable
    [optimization may be difficult]
  Did not solve the tractability problem
  We just changed it!

Maximum Likelihood (ML)

- The second approximation is ML
  — instead of finding hypothesis with highest probability a-posteriori (in the posterior)
  — find hypothesis that best fits data: hypothesis w/ highest likelihood of generating data (evidence **e**)
- ML simplifies MAP by assuming uniform prior i.e., $P(h_i) = P(h_j) \forall_{i,j}$ uniform prior assumption

$$h_{MAP} = argmax_h \underbrace{P(h)}_{\text{prior}} \underbrace{P(\mathbf{e} \mid h)}_{\text{likelihood}}$$

$$h_{ML} = argmax_h \underbrace{P(\mathbf{e} \mid h)}_{\text{likelihood}}$$

- Make prediction based on h_{ML} only: $P(X \mid \mathbf{e}) \approx P(X \mid h_{ML})$
- ML: an approximation because we eliminate prior
- ML: a simplification because no need to set prior
  prior: distribution over all hypotheses, if there are too many hypotheses, prior is complex to encode

Maximum Likelihood (ML)

- What does maximizing likelihood accomplish?
- There is only a finite amount of probability mass
(i.e. sum-to-one constraint)
- ML tries to allocate **as much** probability mass **as possible** to the things we have observed
- At the expense of the things we have not observed

ML properties

- **Computation/Prediction Accuracy tradeoff**
  ML ignores prior info & relies only on one hypothesis h_{ML}
  + + Simpler computation
  - - ML prediction less accurate than Bayesian & MAP predictions

- **Convergence**
  — ML, MAP & Bayesian predictions converge as data increases

- **Overfitting**
  — Subject to overfitting
  — Pick hypothesis that best fits data [risk: fit everything including noise]
  — No prior to penalize complex hypotheses that can exploit statistically insignificant data patterns

- **Tractability**
  — Still have an optimization problem
  — Could still be intractable
  — Finding h_{ML} is often easier than h_{MAP}
  — Many ML algorithms maximize likelihood (solving this optimization)

$$h_{ML} = argmax_h \sum_n \underbrace{logP(e_n|h)}_{\text{likelihood}}$$

Infinite Data?

- Bayesian learning, MAP & ML converge to same prediction given infinite data

- Bayesian Learning
  — start with a prior that captures current uncertainty
  — more data reduces uncertainty
  — more confidence in a hypothesis being the right one

- In the limit, w/ infinite data, we converge
  — Bayesian Learning: all mass of posterior distribution centers on one hypothesis that explains data well (or a few equally good hypotheses)
  — Same hypothesis has highest probability in the posterior
  — Same hypothesis is most consistent with data

- Since they are all equivalent, then given sufficient data, use ML

How much data is sufficient?

- Different problems require different amounts of data
- Large hypothesis space, need more data to find best hypothesis
- Small hypothesis space, less data may be okay
- Answering this Q requires a full course on learning theory!

Is this a realizable problem?

- Is there a true underlying function as a hypothesis inside our hypothesis space?
- We do not make this assumption
- Candy example:
  — we set 5 hypotheses for 5 types of bags
  — if store sells other types, then we don't have a realizable solution
- Discussed approaches will work
  by converging to hypothesis that is best at making a prediction within my space

Principles: MAP vs. ML

MAP Estimation
Choose hypothesis to **maximize posterior of hypothesis** given data

$$h_{MAP} = argmax_h P(h \,|\, \mathbf{e})$$

ML Estimation
Choose hypothesis to **maximize likelihood** of data

$$h_{ML} = argmax_h P(\mathbf{e} \,|\, h)$$

Principles: MAP vs. ML

MAP Estimation

Choose hypothesis to **maximize posterior of hypothesis** given data

$$h_{MAP} = argmax_h \underbrace{P(h)}_{\textbf{Prior}} \underbrace{P(\mathbf{e} \mid h)}_{\textbf{Likelihood}}$$

ML Estimation

Choose hypothesis to **maximize likelihood** of data

$$h_{ML} = argmax_h P(\mathbf{e} \mid h)$$

Bayesian, MAP & ML

	Bayesian	MAP	ML
	$P(H\|\mathbf{e}) = k\ P(\mathbf{e}\|H)P(H)$ $P(X\|\mathbf{e}) = \sum_i P(X\|h_i)P(h_i\|\mathbf{e})$	$h_{MAP} = argmax_{h_i} P(h_i\|\mathbf{e})$ $P(X\|\mathbf{e}) \approx P(X\|h_{MAP})$	$h_{ML} = argmax_h P(\mathbf{e}\|h)$ $P(X\|\mathbf{e}) \approx P(X\|h_{ML})$
Accuracy	**Optimal**	**Less accurate** than Bayesian	**Less accurate** than Bayesian & MAP
Overfitting	**No overfitting**	**Controlled overfitting**	**Overfitting**
Choosing *h*	All hypotheses are used	Solve optimization to find $\boldsymbol{h_{MAP}}$ (**Intractable**)	• Solve optimization to find $\boldsymbol{h_{ML}}$ **(Intractable)** • Finding $\boldsymbol{h_{ML}}$ easier than $\boldsymbol{h_{MAP}}$
Making predictions	**Intractable** to compute prediction	Prediction computed using one hypothesis	Prediction computed using one hypothesis

Conditional, Chain & Bayes

Conditional

$$P(A \mid B) = P(A \wedge B) / P(B)$$

Conditional **Joint** Marginal

Chain

$$P(A \wedge B) = P(A \mid B) P(B)$$

Joint **Conditional** Marginal

Bayes

$$P(B \mid A) = [P(A \mid B) P(B)] \ / \ P(A)$$

Conditional **Conditional** Marginals

Takeaways

- One view of what ML is trying to accomplish is **function approximation**
- The principle of **maximum likelihood estimation** provides **an alternate view of learning**
- **Probability distributions** can be used to **model real data** that occurs in the world