CPSC 440: Advanced Machine Learning

Bayesian Statistics

Mark Schmidt

University of British Columbia

Winter 2021

Last Time: Conditional Random Fields (CRFs)

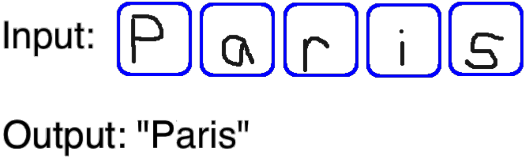
- Conditional random fields: supervised learning method for structured y variables.
    - Models conditional density of y given fixed x values.
- Example is logistic regression with an Ising dependence:

$$p(y_1, y_2, \dots, y_k \mid x_1, x_2, \dots, x_k) \propto \exp\left(\sum_{c=1}^{k} y_c w^T x_c + \sum_{(c,c') \in E} y_c y_{c'} v\right),$$

- Does not need to model any dependencies between features x.

Modeling OCR Dependencies

- What dependencies should we model for this problem?



  - $\phi(y_c, x_c)$: potential of individual letter given image.
  - $\phi(y_{c-1}, y_c)$: dependency between adjacent letters ('q-u').
  - $\phi(y_{c-1}, y_c, x_{c-1}, x_c)$: adjacent letters and image dependency.
  - $\phi_c(y_{c-1}, y_c)$: inhomogeneous dependency (French: 'e-r' ending).
  - $\phi_c(y_{c-2}, y_{c-1}, y_c)$: third-order and inhomogeneous (English: 'i-n-g' end).
  - $\phi(y \in \mathcal{D})$: is y in dictionary $\mathcal{D}$?

Tractability of Discriminative Models

- Features can be very complicated, since we just condition on the x_c, .
- Given the x_c, tractability depends on the conditional UGM on the y_c.
  - Inference tasks will be fast or slow, depending on the y_c graph.
- Besides "low treewidth", some other cases where exact computation is possible:
  - Semi-Markov chains (allow dependence on time you spend in a state).
    - For example, in rain data the seasons will be approximately 3 months.
  - Context-free grammars (allows potentials on recursively-nested parts of sequence).
  - Sum-product networks (restrict potentials to allow exact computation).
  - "Dictionary" feature is non-Markov, but exact computation still easy.
- We can alternately use our previous approximations:
  1. Pseudo-likelihood (what we used).
  2. Monte Carlo approximate inference (eventually better but probably much slower).
  3. Variational approximate inference (fast, quality varies).

Outline

Motivation: Controlling Complexity

- For many machine learning, we need very complicated models.
  - We require multiple forms of regularization to prevent overfitting.
- In 340 we saw two ways to reduce overfitting of a model:
  - Model averaging (ensemble methods).
  - Regularization (linear models).
- Bayesian methods combine both of these.
  - Average over models, weighted by posterior (which includes regularizer).
  - Allows you to fit extremely-complicated models without overfitting.

Most Frequent Keywords at International Confernce on Machine Learning

Bayesian learning includes:

- Gaussian processes.
- Approximate inference.
- Bayesian nonparametrics.

Why Bayesian Learning?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - For example, samples (x^i, y^i) with $x^i \in \mathbb{R}^d$ and $y^i \in \{-1, 1\}$.
  - Find "best" w by minimizing NLL with a regularizer to "prevent overfitting".

$$\hat{w} \in \underset{w}{\text{argmin}} - \sum_{i=1}^{n} \log p(y^i \mid x^i, w) + \frac{\lambda}{2}\|w\|^2.$$

  - Predict labels of *new* example $\tilde{x}$ using single weights $\hat{w}$,

$$\hat{y} = \text{sgn}(\hat{w}^T \tilde{x}).$$

- But data was random, so weight $\hat{w}$ is a random variables.
  - This might put our trust in a $\hat{w}$ where posterior $p(\hat{w} \mid X, y)$ is tiny.

- Bayesian approach: "all parameters are nuissance parameters".
  - Treat w as random and predict based on rules of probability.

Problems with MAP Estimation

- Does MAP make the right decision?
  - Consider three hypothesese $\mathcal{H} = \{\text{“lands''}, \text{“crashes''}, \text{“explodes''}\}$ with posteriors:

$$p(\text{“lands''} \mid D) = 0.4, \quad p(\text{“crashes''} \mid D) = 0.3, \quad p(\text{“explodes''} \mid D) = 0.3.$$

  - The MAP estimate is “plane lands”, with posterior probability 0.4.
    - But probability of dying is 0.6.
    - If we want to live, MAP estimate doesn't give us what we should do.

- Bayesian approach considers all models: says don't take plane.
- Bayesian decision theory: accounts for costs of different errors.

MAP vs. Bayes

- MAP (regularized optimization) approach maximizes over w:

$$\hat{w} \in \underset{w}{\operatorname{argmax}}\, p(w \mid X, y)$$
$$\equiv \underset{w}{\operatorname{argmax}}\, p(y \mid X, w)p(w) \qquad \text{(Bayes' rule, } w \perp X\text{)}$$
$$\hat{y} \in \underset{y}{\operatorname{argmax}}\, p(y \mid \tilde{x}, \hat{w}).$$

- Bayesian approach predicts by integrating over possible w:

$$\begin{aligned} p(\tilde{y} \mid \tilde{x}, X, y) &= \int_w p(\tilde{y}, w \mid \tilde{x}, X, y)dw && \text{marginalization rule} \\ &= \int_w p(\tilde{y} \mid w, \tilde{x}, X, y)p(w \mid \tilde{x}, X, y)dw && \text{product rule} \\ &= \int_w p(\tilde{y} \mid w, \tilde{x})p(w \mid X, y)dw && \tilde{y} \perp X, y \mid \tilde{x}, w \end{aligned}$$

- Considers all possible w, and weights prediction by posterior for w.

Motivation for Bayesian Learning

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability (and possibly error costs).
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.
  3. Elegant approaches for model selection and model averaging.
     - E.g., optimize λ or optimize grouping of w elements.
  4. Easy to relax IID assumption.
     - E.g., hierarchical Bayesian models for data from different sources.
  5. Bayesian optimization: fastest rates for some non-convex problems.
  6. Allows models with unknown/infinite number of parameters.
     - E.g., number of clusters or number of states in hidden Markov model.

- Why isn't everyone using this?
  - Philosophical: Some people don't like that results depend on "subjective" prior.
  - Computational: Typically leads to nasty integration problems.

Coin Flipping Example: MAP Approach

- MAP vs. Bayesian for a simple coin flipping scenario:
  1. Our likelihood is a Bernoulli,
  $$p(H \mid \theta) = \theta.$$
  2. Our prior assumes that we are in one of two scenarios:
     - The coin has a 50% chance of being fair ($\theta = 0.5$).
     - The coin has a 50% chance of being rigged ($\theta = 1$).
  3. Our data consists of three consecutive heads: 'HHH'.

- What is the probability that the next toss is a head?
  - MAP estimate is $\hat{\theta} = 1$, since $p(\theta = 1 \mid HHH) > p(\theta = 0.5 \mid HHH)$.
  - So MAP says the probability is 1.

  - But MAP overfits: we believed there was a 50% chance the coin is fair.

Coin Flipping Example: Posterior Distribution

- Bayesian method needs posterior probability over θ,

$$p(\theta = 1 \mid HHH) = \frac{p(HHH \mid \theta = 1)p(\theta = 1)}{p(HHH)} \quad \text{(Bayes rule)}$$

$$\text{(marg and prod rule)} = \frac{p(HHH \mid \theta = 1)p(\theta = 1)}{p(HHH \mid \theta = 0.5)p(\theta = 0.5) + p(HHH \mid \theta = 1)p(\theta = 1)}$$

$$= \frac{(1)(0.5)}{(1/8)(0.5) + (1)(0.5)} = \frac{8}{9},$$

and similarly we have $p(\theta = 0.5 \mid HHH) = \frac{1}{9}$.

- So given the data, we should believe with probability $\frac{8}{9}$ that coin is rigged.
  - There is still a $\frac{1}{9}$ probability that it is fair that MAP is ignoring.

Coin Flipping Example: Posterior Predictive

- Posterior predictive gives probability of head given data and prior,

$$
\begin{aligned}
p(H \mid HHH) &= p(H, \theta = 1 \mid HHH) + p(H, \theta = 0.5 \mid HHH) \\
&= p(H \mid \theta = 1, HHH)p(\theta = 1 \mid HHH) \\
&+ p(H \mid \theta = 0.5, HHH)p(\theta = 0.5 \mid HHH) \\
&= p(H \mid \theta = 1)p(\theta = 1 \mid HHH) + p(H \mid \theta = 0.5)p(\theta = 0.5 \mid HHH) \\
&= (1)(8/9) + (0.5)(1/9) = 0.94.
\end{aligned}
$$

- So the correct probability given our assumptions/data is 0.94, and not 1.
  - Though with a different prior we would get a different answer.

- Notice that there was no optimization of the parameter θ:
  - In Bayesian stats we condition on data and integrate over unknowns.

- In Bayesian stats/ML: "all parameters are nuissance parameters".

Coin Flipping Example: Discussion

Comments on coin flipping example:

- Bayesian prediction uses that HHH could come from fair coin.
- As we see more heads, posterior converges to 1.
  - MLE/MAP/Bayes usually agree as data size increases.
- If we ever see a tail, posterior of $\theta = 1$ becomes 0.

- If the prior is correct, then Bayesian estimate is optimal:
  - Bayesian decision theory gives optimal action incorporating costs.
- If the prior is incorrect, Bayesian estimate may be worse.
  - This is where people get uncomfortable about "subjective" priors.

- But MLE/MAP are also based on "subjective" assumptions.

Outline

Bayesian Model Averaging

- In 340 we saw that model averaging can improve performance.
  - E.g., random forests average over random trees that overfit.
- But should all models get equal weight?
  - What if we find a random decision stump that fits the data perfectly?
    - Should this get the same weight as deep random trees that likely overfit?
  - In science, research may be fraudulent or not based on evidence.
    - Should "vaccines cause autism" or "climate change denial" models get equal weight?
- In these cases, naive averaging may do worse.

Bayesian Model Averaging

- Suppose we have a set of m probabilistic classifiers w_j
  - Previously our ensemble method gave all models equal weights,

$$p(\tilde{y} \mid \tilde{x}) = \frac{1}{m}p(\tilde{y} \mid \tilde{x}, w_1) + \frac{1}{m}p(\tilde{y} \mid \tilde{x}, w_2) + \cdots + \frac{1}{m}p(\tilde{y} \mid \tilde{x}, w_m).$$

- Bayesian model averaging (following rules of probability) weights by posterior,

$$p(\tilde{y} \mid \tilde{x}) = p(w_1 \mid X, y)p(\tilde{y} \mid \tilde{x}, w_1) + p(w_2 \mid X, y)(\tilde{y} \mid \hat{x}, w_2) + \cdots + p(w_m \mid X, y)p(\tilde{y} \mid \tilde{x}, w_m).$$

- So we should weight by probability that w_j is the correct model.
  - Equal weights assume all models are equally probable and fit data equally well.

Bayesian Model Averaging

- Weights are posterior, so proportional to likelihood times prior:

$$p(w_j \mid X, y) \propto \underbrace{p(y \mid X, w_j)}_{\text{likelihood}} \underbrace{p(w_j)}_{\text{prior}}.$$

- Likelihood gives more weight to models that predict y well.
- Prior should gives less weight to models that are likely to overfit.

- This is how rules of probability say we should weight models.
  - It's annoying that it requires a "prior" belief over models.
  - You also need to know the normalizing constant for most interesting cases.
  - But as $n \to \infty$, all weight goes to "correct" model[s] w^* as long as $p(w^*) > 0$.

Digression: Bayes for Density Estimation and Generative/Discriminative

- We can use Bayesian approach for density estimation:
  - With data D and parameters θ we have:
    1. Likelihood $p(D \mid \theta)$.
    2. Prior $p(\theta)$.
    3. Posterior $p(\theta \mid D)$.

- We can also use Bayesian approach for supervised learning:
  - Generative approach (naive Bayes, GDA) are density estimation on X and y:
    1. Likelihood $p(y, X \mid w)$.
    2. Prior $p(w)$.
    3. Posterior $p(w \mid X, y)$.

  - Discriminative approach (logistic regression, neural nets) just conditions on X:
    1. Likelihood $p(y \mid X, w)$.
    2. Prior $p(w)$.
    3. Posterior $p(w \mid X, y)$.

Summary

- Bayesian statistics:
  - Optimal way to make predictions, given likelihood and prior.
  - Conditions on the data, integrates (rather than maximize) over posterior.
  - "All parameters are nuissance parameters".
- Posterior predictive distribution:
  - Probability of new data, given old data (integrating over parameters).
- Bayesian model averaging:
  - Model averaging based on rules of probability, rather than uniform weight.
- Next time: learning the prior?