CPSC 540: Machine Learning
Non-Parametric Bayes

Mark Schmidt
University of British Columbia

Winter 2019
Last Time: Monte Carlo vs. Variational Inference

Two main strategies for approximate inference:

1. **Monte Carlo** methods:
   - Approximate $p$ with empirical distribution over samples,
   
   \[
   p(x) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[x^i = x].
   \]
   - Turns inference into sampling. Default method is Metropolis-Hastings.

2. **Variational** methods:
   - Approximate $p$ with “closest” distribution $q$ from a tractable family,
   
   \[ p(x) \approx q(x). \]
   - E.g., Gaussian, independent Bernoulli, or tree UGM.
   - Turns inference into optimization. Default method minimizes reverse KL divergence.
Variational vs. Monte Carlo

Monte Carlo vs. variational methods:
- Variational methods are typically more complicated.
- Variational methods are not consistent.
  - $q$ does not converge to $p$ if we run the algorithm forever.
- But variational methods often give better approximation for the same time.
  - Although MCMC is easier to parallelize.
- Variational methods typically have similar cost to MAP.

Combinations of variational inference and stochastic methods:
- **Stochastic variational inference (SVI)**: use SGD to speed up variational methods.
- **Variational MCMC**: use Metropolis-Hastings where variational $q$ can make proposals.
I’ve overviewed the “classic” view of variational methods that they minimize KL.

Modern view: write exact inference as constrained convex optimization (bonus).
- Based on convex conjugate, writing inference as maximizing entropy with constraints.
- Different methods correspond to different function/constraints approximations.
- There are also convex relaxations that approximate with linear programs.

For an overview of this and all things variational, see:
people.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf
Outline

1. Non-Parametric Bayes
2. GANs and VAEs
A stochastic process is an infinite collection of random variables \( \{ x^i \} \).

Non-parametric Bayesian methods use priors defined on stochastic processes:
- Allows extremely-flexible prior, and posterior complexity grows with data size.
- Typically set up so that samples from posterior are finite-sized.

The two most common priors are Gaussian processes and Dirichlet processes:
- Gaussian processes define prior on space of functions (universal approximators).
- Dirichlet processes define prior on space of probabilities (without fixing dimension).
Recall the partitioned form of a multivariate Gaussian

\[ \mu = [\mu_x, \mu_y], \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}, \]

and in this case the marginal \( p(x) \) is a \( \mathcal{N}(\mu_x, \Sigma_{xx}) \) Gaussian.

Generalization of this to infinite set of variables is Gaussian processes (GPs):
- Any finite set from collection follows a Gaussian distribution.
To date kriging has been used in a variety of disciplines, including the following:

- Environmental science\(^5\)
- Hydrogeology\(^6\)\(^7\)\(^8\)
- Mining\(^9\)\(^10\)
- Natural resources\(^11\)\(^12\)
- Remote sensing\(^13\)
- Real estate appraisal\(^14\)\(^15\)

and many others.

Mauna Loa, CO2, GP model fit on data until Dec 2003. 95% predicted confidence.
Gaussian Processes

- GPs are specified by a **mean function** $m$ and **covariance function** $k$,

  $$m(x) = \mathbb{E}[f(x)], \quad k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T].$$

- Any finite sample $f(x)$ from a GP follows a $\mathcal{N}(m(x), k(x, x))$ distribution.
  - Analogous to partitioned Gaussian where $m(x) = \mu_x$ and $k(x, x) = \Sigma_{xx}$.

- We write that

  $$f(x) \sim \text{GP}(m(x), k(x, x')).$$

- As an example, we could have a zero-mean and linear covariance GP,

  $$m(x) = 0, \quad k(x, x') = x^T x'.$$
Regression Models as Gaussian Processes

As an example, predictions made by linear regression with Gaussian prior

\[ f(x) = w^T \phi(x), \quad w \sim \mathcal{N}(0, \Sigma), \]

are a Gaussian process with mean function

\[ \mathbb{E}[f(x)] = \mathbb{E}[w^T \phi(x)] = \mathbb{E}[w]^T \phi(x) = 0. \]

and covariance function

\[ \mathbb{E}[f(x)f(x)^T] = \phi(x)^T \mathbb{E}[ww^T] \phi(x') = \phi(x)\Sigma\phi(x') = k(x, x'). \]
We can view a Gaussian process as a prior distribution over smooth functions.

Most common choice of covariance is RBF.

Is this related to using RBF kernels or the RBFs as the bases?
  - Yes, this is Bayesian linear regression plus the kernel trick.
Gaussian Process Model Selection

- So why do we care?
  - We can get estimate of uncertainty in the prediction.
  - We can use marginal likelihood to learn the kernel/covariance.

- Write kernel in terms of parameters, use empirical Bayes to learn kernel.

- Hierarchical approach: put a hyper-prior of types of kernels.

- Application: Bayesian optimization of non-convex functions:
  - Gradient descent is based on a Gaussian (quadratic) approximation of $f$.
  - Bayesian optimization is based on a Gaussian process approximation of $f$.
    - Can approximate non-convex functions.
Recall the basic mixture model:

\[ p(x \mid \theta) = \sum_{c=1}^{k} \pi_c p(x \mid \theta_c). \]

Non-parametric Bayesian methods allow us to consider infinite mixture model,

\[ p(x \mid \theta) = \sum_{c=1}^{\infty} \pi_c p(x \mid \theta_c). \]

Common choice for prior on \( \pi \) values is Dirichlet process:

- Also called “Chinese restaurant process” and “stick-breaking process”.
- For finite datasets, only a fixed number of clusters have \( \pi_c \neq 0 \).
- But don’t need to pick number of clusters, grows with data size.
Dirichlet Process

- Gibbs sampling in Dirichlet process mixture model in action: https://www.youtube.com/watch?v=0Vh7qZY9sPs

- We could alternately put a prior on $k$:
  - “Reversible-jump” MCMC can be used to sample from models of different sizes.
    - AKA “trans-dimensional” MCMC.

- There are a variety of interesting variations on Dirichlet processes
  - Beta process (“Indian buffet process”).
  - Hierarchical Dirichlet process.
  - Polya trees.
  - Infinite hidden Markov models.
Bayesian Hierarchical Clustering

Hierarchical clustering of \{0, 2, 4\} digits using classic and Bayesian method:

http://www2.stat.duke.edu/~kheller/bhcnew.pdf  (y-axis represents distance between clusters)
Bayesian Hierarchical Clustering

- Hierarchical clustering of newsgroups using classic and Bayesian method:

http://www2.stat.duke.edu/~kheller/bhcnew.pdf (y-axis represents distance between clusters)
Summary of Part 1

- **Non-Parametric Bayes** puts probabilities over infinite spaces.
  - Gaussian processes are priors over continuous functions.
  - Dirichlet processes are priors over probability mass functions.

- Part 2: new generative deep learning methods.
Modern view of variational inference:
- Formulate inference problem as constrained optimization.
- Approximate the function or constraints to make it easy.
We will again consider log-linear models:

\[ P(X) = \frac{\exp(w^T F(X))}{Z(w)}, \]

but view them as exponential family distributions,

\[ P(X) = \exp(w^T F(X) - A(w)), \]

where \( A(w) = \log(Z(w)) \).

Log-partition \( A(w) \) is called the cumulant function,

\[ \nabla A(w) = \mathbb{E}[F(X)], \quad \nabla^2 A(w) = \mathbb{V}[F(X)], \]

which implies convexity.
Convex Conjugate and Entropy

- The convex conjugate of a function $A$ is given by
  \[ A^*(\mu) = \sup_{w \in W} \{ \mu^T w - A(w) \}. \]

- E.g., if we consider for logistic regression
  \[ A(w) = \log(1 + \exp(w)), \]
  we have that $A^*(\mu)$ satisfies $w = \log(\mu) / \log(1 - \mu)$.
    - When $0 < \mu < 1$ we have
      \[
      A^*(\mu) = \mu \log(\mu) + (1 - \mu) \log(1 - \mu) \\
      = -H(p_\mu),
      \]
      negative entropy of binary distribution with mean $\mu$.
    - If $\mu$ does not satisfy boundary constraint, $\sup$ is $\infty$. 

Non-Parametric Bayes GANs and VAEs
Convex Conjugate and Entropy
Convex Conjugate and Entropy

- More generally, if $A(w) = \log(Z(w))$ then
  \[ A^*(\mu) = -H(p_{\mu}), \]
  subject to boundary constraints on $\mu$ and constraint:
  \[ \mu = \nabla A(w) = \mathbb{E}[F(X)]. \]

- Convex set satisfying these is called marginal polytope $\mathcal{M}$.

- If $A$ is convex (and LSC), $A^{**} = A$. So we have
  \[ A(w) = \sup_{\mu \in \mathcal{U}} \{w^T \mu - A^*(\mu)\}. \]
  and when $A(w) = \log(Z(w))$ we have
  \[ \log(Z(w)) = \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_{\mu})\}. \]

- We’ve written inference as a convex optimization problem.
The maximum likelihood parameters \( w \) satisfy:

\[
\min_{w \in \mathbb{R}^d} -w^T F(D) + \log(Z(w)) = \min_{w \in \mathbb{R}^d} -w^T F(D) + \sup_{\mu \in \mathcal{M}} \{ w^T \mu + H(p_\mu) \} \quad \text{(convex conjugate)}
\]

\[
= \min_{w \in \mathbb{R}^d} \sup_{\mu \in \mathcal{M}} \{ -w^T F(D) + w^T \mu + H(p_\mu) \}
\]

\[
= \sup_{\mu \in \mathcal{M}} \min_{w \in \mathbb{R}^d} \{ -w^T F(D) + w^T \mu + H(p_\mu) \} \quad \text{(convex/concave)}
\]

which is \(-\infty\) unless \( F(D) = \mu \) (e.g., maximum likelihood \( w \)), so we have

\[
\min_{w \in \mathbb{R}^d} -w^T F(D) + \log(Z(w)) = \max_{\mu \in \mathcal{M}} H(p_\mu),
\]

subject to \( F(D) = \mu \).

- Maximum likelihood \( \Rightarrow \) maximum entropy + moment constraints.
We wrote inference as a convex optimization:

$$\log(Z) = \sup_{\mu \in \mathcal{M}} \{ w^T \mu + H(p_\mu) \},$$

Did this make anything easier?
- Computing entropy $H(p_\mu)$ seems as hard as inference.
- Characterizing marginal polytope $\mathcal{M}$ becomes hard with loops.

Practical variational methods:
- Work with approximation to marginal polytope $\mathcal{M}$.
- Work with approximation/bound on entropy $A^*$.

Notation trick: we put everything “inside” $w$ to discuss general log-potentials.
Mean Field Approximation

- **Mean field** approximation assumes

\[ \mu_{ij,st} = \mu_{i,s}\mu_{j,t}, \]

for all edges, which means

\[ p(x_i = s, x_j = t) = p(x_i = s)p(x_j = t), \]

and that variables are independent.

- Entropy is simple under mean field approximation:

\[ \sum_X p(X) \log p(X) = \sum_i \sum_{x_i} p(x_i) \log p(x_i). \]

- Marginal polytope is also simple:

\[ \mathcal{M}_F = \{ \mu \mid \mu_{i,s} \geq 0, \sum_s \mu_{i,s} = 1, \mu_{ij,st} = \mu_{i,s}\mu_{j,t} \}. \]
Entropy of Mean Field Approximation

- Entropy form is from distributive law and probabilities sum to 1:

\[
\sum_X p(X) \log p(X) = \sum_X p(X) \log(\prod_i p(x_i))
\]

\[
= \sum_X p(X) \sum_i \log(p(x_i))
\]

\[
= \sum_i \sum_X p(X) \log p(x_i)
\]

\[
= \sum_i \sum_X \prod_j p(x_j) \log p(x_i)
\]

\[
= \sum_i \sum_X p(x_i) \log p(x_i) \prod_{j \neq i} p(x_j)
\]

\[
= \sum_{i \, x_i} p(x_i) \log p(x_i) \sum_{x_j \, j \neq i \neq i} \prod_{j \neq i} p(x_j)
\]

\[
= \sum_{i \, x_i} p(x_i) \log p(x_i) \prod_{j \neq i} p(x_j)
\]

\[
= \sum_{i \, x_i} p(x_i) \log p(x_i).
\]
Mean Field as Non-Convex Lower Bound

- Since $\mathcal{M}_F \subseteq \mathcal{M}$, yields a lower bound on $\log(Z)$:
  $$\sup_{\mu \in \mathcal{M}_F} \{w^T \mu + H(p_\mu)\} \leq \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\} = \log(Z).$$

- Since $\mathcal{M}_F \subseteq \mathcal{M}$, it is an inner approximation:

![Diagram](image)

- Constraints $\mu_{ij,st} = \mu_{i,s} \mu_{j,t}$ make it non-convex.
- Mean field algorithm is coordinate descent on $w^T \mu + H(p_\mu)$ over $\mathcal{M}_F$. 
Mean field is weird:
- Non-convex approximation to a convex problem.
- For learning, we want upper bounds on $\log(Z)$.

Structured mean field:
- Cost of computing entropy is similar to cost of inference.
- Use a subgraph where we can perform exact inference.

http://courses.cms.caltech.edu/cs155/slides/cs155-14-variational.pdf
Structured Mean Field with Tree

- More edges means better approximation of $\mathcal{M}$ and $H(p_\mu)$:

$$\text{original } G \quad (\text{Naïve}) \text{ MF } H_0 \quad \text{structured MF } H_s$$

- Fixed points of loopy correspond to using “Bethe” approximation of entropy and “local polytope” approximation of “marginal polytope”.

- You can design better variational methods by constructing better approximations.

http://courses.cms.caltech.edu/cs155/slides/cs155-14-viariational.pdf