10-708 24/06/2021

U3 - Approximate inference: - Monte Carlo and seq. methods

- continue U2

(*) variational principle (V.P.)

Ex: Highly technical → review material with assigned readings.
- contents cannot be assimilated immediately in particular.

- Recap: Algebraic problems / probabilistic inference reformulated as variational inference
- define solution to partition function (quantity of interest) in variational manner.
- Normally computed (part. function) via summation / integration (exponentially complex)

(*) $A(\theta) = \sup_{\mu \in M} \{\theta^T \mu - A^*(\mu)\}$

- inner product of dual pair
- conjugate dual

- θ corresponds to canonical param in exp family
- μ — " — mean parameters
- M - marginal polytope

(A1)

- μ - satisfies constraints as expectation of sufficient statistics $\mu = \mathbb{E}_{\mu}$
- A^* - negative entropy of original distribution

(*) Solution only exists to this problem when mean params satisfy constraints; that they lie within a domain called marginal polytope

(*) Marginal polytope - convex combination of sufficient statistics

(*) In discrete case → convex hull
cont. case → — " —

(*) This is key; use two-node Ising Model as example. and review.

(*) Example - 2 node Ising model

(*) Note Minkowski-Weyl rep of marginal polytope

(*) optimum → stationary points

(*) more complex problems; we cannot use above variational principle to derive results of a qualitatively exact nature (that is the point of introducing V.M.)

(*) V.P. is a vehicle for approximations

(*) variational principle

(*) understand MF approxim. in context of V.P.

(*) note: from readings, introducing V.P. in this way shows/reformulates the intractability of summation/inference

- also opens possibilities on dealing with intractability of summation
→ dealing with intractability of marginal polytope and entropy.

* Approximation schemes can be viewed as approx. the marginal polytope and entropy terms.

(*) MF approx:-

i) Redefine domain of mean parameters to be in a subspace of the true marginal polytope M

ii) Redefine A^* (negative entropy) in a way that is ~~eas~~ comp. tractable

(*) Formally: MF approx uses:-

i) non-convex inner bound $M_{MF} \subset M$

ii) exact form of entropy $A^* = H$

(*) Bethe approx and LBP:-

i) polyhedral outer bound $M_{BETHE} \supset M$

ii) non-convex Bethe approx. → $A^* = H \approx H_{BETHE}$

(?) Ex: States that entropy is approx with Bethe free energy (surely he means Bethe entropy?)

(*) Mean-Field Approx.

- tractable subgraphs

- MF approx relies on using a tractable subgraph approx. to original graph.
- for a graph with effective canonical params on each edge $\Omega := \{\theta \in \mathbb{R}^d \mid A(\theta) < +\infty\}$
- set Ω is defined in such way that the range of log-normaliser is bounded.
- Exp family with sufficient statistics ϕ defined on graph G:-
  - $\mathcal{M}(G;\phi) := \{\mu \in \mathbb{R}^d \mid \exists p \text{ s.t. } \mathbb{E}_p[\phi(X)] = \mu\}$
- define a subgraph F_0 with no edges (still a graph - it has nodes)
- corresponding canonical parameter set:-

$$\Omega(F_0) := \{\theta \in \Omega \mid \theta_{(s,t)} = 0 \ \forall (s,t) \in E\}$$

- $\theta_{ij} = 0$
- what are corresponding mean params μ_{ij} (expected suff statistic defined on pair (i,j))
- $\theta_{ij} = 0$
- As $\mu_{ij} = p(x_i, x_j)$, $\theta_{ij} = 0 \Rightarrow \mu_{ij} = p(x_i)p(x_j)$ ($\theta_{ij} = 0$ (lack of edges) induces independence)

- define another subgraph T (properties wazy)

$$\Omega(T) := \{\theta \in \Omega \mid \theta_{(s,t)} = 0 \ \forall (s,t) \notin E(T)\}$$

(iii): You can always define a subgraph approximation - an approximation to a graph by removing edges

(iii): can also give geometric approx to marginal polytope

(*) Above is high-level treatment.

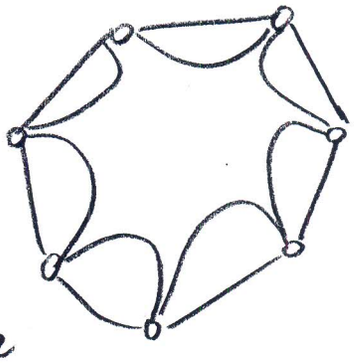
(*) Mean Field methods

- In more detail :-
- For a given tractable subgraph F, a subset of canonical params is :-

$$M(F;\phi) := \{\tau \in \mathbb{R}^d \mid \tau = \mathbb{E}_\theta[\phi(X)] \text{ for some } \theta \in \Omega(F)\}$$

- $\Omega(F)$ defines a new space for canonical params. corresp. to θ constrained by subgraph
- Inner approx (to the full marg. polytope) :-

$$M(F;\phi)^\circ \subseteq M(G;\phi)^\circ$$



- Recall marginal polytope is convex
- The mf approximation of marginal polytope lies within the marginal polytope, because it uses a subset (of canonical parameters associated with the tractable subgraph F)
- For MF approx, we define the (:) parameters to lie within the mf approx to marginal polytope :-
- Relaxed problem :

$$\max_{\tau \in M_F(G)} \{\langle \tau, \theta \rangle - A_F^*(\tau)\}$$

*) Compute dual of the partition function i.e. entropy of a distribution that corresponds to a tractable subgraph with no edges

$$A_F^* = A^*|_{M_F(G)}$$

*) Subgraph with no edge :-

$P(X_1, \ldots, X_N)$ approximated by $Q_F(X_{1\ldots})$ where Q represents joint over graph with no edges

*) For graph with no edges :-

$$Q_F(X_1 \ldots, X_N) = \prod_{n=1}^{N} q(X_n)$$

(*) To compute $A^*_{\theta EF} = H_Q - \sum H(q(x_i))$ (A2) - nicer see - tidy

(*) complex models e.g. topic models ; H_P is not tractable

(*) stepping back;

we have a relaxed problem with an analytic exp for $A^*_F(\tau)$

and easier exp for approx. to marginal polytope $M_F(G)$

(*) naive MF approx for Ising

- (F2) - review example

- solve: take derivative wrt τ to get MF update equations

MF approx sol. to topic models etc is a similar method. flavour

generates iterative solution

(*) Geometry of MF approx

- MF opt. is always non-convex

- Full space is convex ⟶ subsets/space within cannot be convex (marginal polytope)

(*) strict subset

(*) will an approximation that is equivalent to true solution result?

(*) No. of issues:-

i) True solutions lie within $M(G) - M_F(G)$ i.e. outside of mne approx.

ii) Non-convexity ⟶ not guaranteed global optima

(*) Benefit: Simple algorithm that offers solution to a complex prob. in an iterative fashion

---

(*) sum-product / BP.

- Key equations:- M.P.P., marginals

Minimises Bethe free energy

*) Tree-GMS

*) Bethe approx yields exact solution for tree-structured G.MS.

*) In this context the marginal polytope ⟶ true polytope

(A4) - review equations

*) local consistency + global consistency constraints (for trees) on marginal probabilities / mean parameters

*) Local consistency ⇏ global consistency generally

(*) Decomposition of entropy for trees

(A5) Renew the variational form on trees

(*) Lagrangian deriv.

- Reparametrisation of constrained minim. of Bethe free energy for tree-structured GMs yields exact inference; and B.P. update equations.

B.P. on arbitrary graphs

- 2 main difficulties of variational formulation:-
- In earlier MF approx, we relied on inner approx. of marginal polytope
- Variational form:

$$A(\theta) = \sup_{\mu \in \mathcal{M}} \{\theta^T \mu - A^*(\mu)\}$$

- Here, use a tree-based outer bound on the marginal polytope $\mathcal{M}$:-

$$\mathbb{L}(G) = \left\{ \tau \geq 0 \mid \sum_{x_s} \tau_s(x_s) = 1, \sum_{x_t} \tau_{st}(x_s, x_t) = \tau_s(x_s) \right\}$$

- only impose local consistency on vectors/ fns $\tau_s(x_s)$ and $\tau_{st}(x_s, x_t)$ which are pseudomarginals

Q: Is the space $\mathbb{L}(G)$ large or smaller than the marginal polytope M?

(*) Key nuance here:-
The tree structured GM and local consistency conditions inspire constraints; but we are not working in that setting, this is an arbitrary graph.

(*) There are global constraints not present here; hence $\mathbb{L}(G)$ is large (?)

(W): Don't understand

(*) For arbitrary distri (graphs), $A^*(\mu)$ is also complicated

(*) Approximate $A^*(\mu)$ with Bethe free energy (?) (solely entropy)

$$-A^*(\tau) \approx H_{Bethe}(\tau) = \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st})$$

(*) H_{Bethe} only requires pairwise and singleton sufficient statistics

(*) Bethe variational Problem

(marg. polyt.) (entropy)

combining $\mathbb{L}(G)$ and $A^*(\tau)$ i.e. approx of M and $A^*(\mu)$:-

Bethe V.P.:

$$\max_{\tau \in \mathbb{L}(G)} \left\{ \langle \theta, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st}) \right\}$$

(*) MF approx uses $M \approx M_F(G)$ and $A^*(\mu) = H_q$
(mne approx of marginal polytope) (true entropy)

(*) Ex: Why do we use true entropy H_q for $A^*(\mu)$ in MF approx; whereas here we use Bethe approx to entropy?

(W): As we use approx. P with an MF approx. Q, as product of marginal q distri (fully factorised) ✓ correct

(*) In loopy BP, the approximation being used was NOT explicit in the same manner as MF approx.

(*) For loopy B.P., we only requested queries

$\mathbb{E}[\phi(X_i)]$ and $\mathbb{E}[\phi(X_i, X_j)]$; with no regard for what else lies in full distri. $p(x)$

singleton exp. pairwise exp

(*) Treat the above, corresponding to mean params. of singleton and pairwise and define a space based on local consistency constraints, then define approximation to entropy only using these 2 statistics

(*) Bypasses explicit representation of approx. of p, $\tilde{p}$; but the full distri has more edges and structure in entropy; we are aware of this; but set aside.

(A6) - Review rest of results here

(*) Geometry of B.P. (outer approx)

- for any graph, $\mathcal{M}(G) \subseteq \mathbb{L}(G)$, with equality $\mathcal{M}(G) = \mathbb{L}(G)$ if and only if the graph is a tree

- concerning entropy; $H_{Bethe} \hat{=} H$

If graph is a tree $H_{Bethe} = H$

(A7) Tidy entropy functionals

(*) Yields insight on behavior of B.P.

- may or may not converge

(*) (A8): Review BP convergence in context of outer approx to marginal polytope geometrically

---

(*) Inexactness of Bethe entropy approx.

(A9): Review

(*) Remark + Summary

Kikuchi variants rarely used nowadays

10-708 24/06/2020

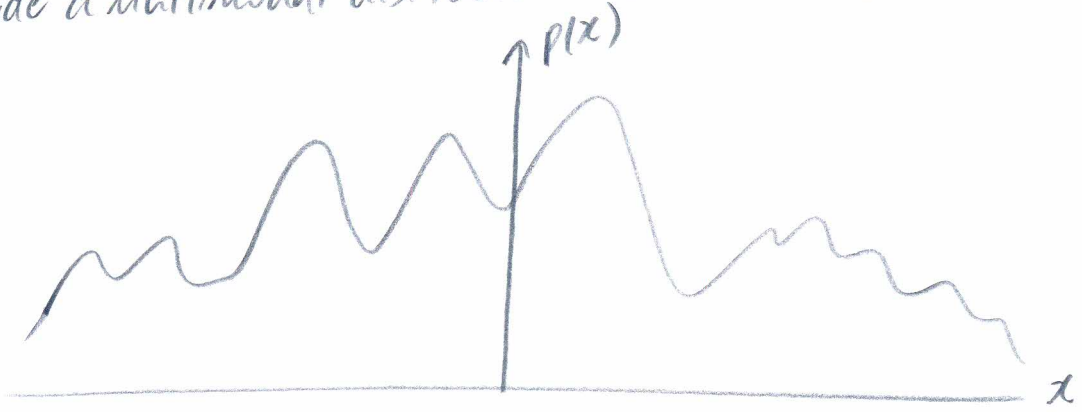
L13 - Approx Inference: Monte Carlo and sequential Monte Carlo

(*) How to represent a joint dist.

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

- Sample based representation:

- consider a multimodal distribution (and density):-



(?) - How to integrate?

- It may not be feasible (analytically)? to integrate this.
- Sol: draw samples from distr
- computationally: will use a uniform PRNG to draw a sample $u(0,1)$ and transform to get Gaussian

(*) Draw N samples $x^{(i)} \sim P$:-

$i = 1, \ldots, N$

compute $\mathbb{E}_p[f(X)] = \dfrac{\sum_{i=1}^{N} f(x^{(i)})}{N}$

} Monte Carlo methods

(*) original distr P expressed as a set of samples

(*) Monte Carlo methods

(*) replace integration with summation (simple-based averages)

(*) Asymptotically exact → more samples drawn ($N \to \infty$), arbitrarily close to expectation we wish to approx.

Challenges:

1) How to draw samples from distri?
   - How to do this efficiently?
     from an arbitrary
2) How to make better use of samples?
   - asymptote - how many samples can we draw ...

(*) Naive sampling

$P(X_A, \ldots, X_M) = P(X_B)P(X_A|X_B) \ldots$

$\underline{X}^{(k)} = \begin{bmatrix} X_A \\ \vdots \\ X_M \end{bmatrix} \sim P$

- How to draw samples?
  - traverse graph
- (*) CK holds through realisation of sample in BN
  - sample r.v.s. from B.N.
- (*) Rows of table correspond to a 'sample' from B.N.
- (*) computation of queries: check samples where condition is met.
- (*) Rare events ⟶ conditioning on rare events?
- (*) shows the limitations of sampling

(*) monte carlo methods (rejection sampling)

- attempt to get around above deficiencies
- $\pi(X) = \frac{\pi'(X)}{Z}$

$\pi(X)$ - difficult to sample from
$\pi'(X)$ - easy to evaluate (eg. unnormalised clique pot. product)

- sample from simple distri $Q(X)$

(2): $x^* \sim Q(x)$ (draw samples from Q)

- Accept x^* with probability $\frac{\pi'(x^*)}{kQ(x^*)}$ k-arbitrary constant

(*) Ratio of unnormalised part of given (evaluatable distri) and a constant multiplied by proposal distribution.

(*) With these two steps:-

$$x^* \sim \pi(x)$$

EX: How to prove this?

$$P(x^*) = Q(x^*)\frac{\pi'(x^*)}{kQ(x^*)} = \frac{k\pi'(x^*)}{k\int \pi'(x^*)dx^*} = \pi(x)$$

$$\int Q(x^*)\frac{\pi'(x^*)}{kQ(x^*)}dx^*$$

(*) A solution to sampling from a distri $\pi(x)$ which is difficult to normalise (using a proposal distrib)

Pitfall: (21): Geometric intuition/explanation of too many rejections (?)

(*) Rejection sampling

- conceptual experiment to illustrate rejection sampling
- Small differences between proposal distri and distri of interest caused by scaling constant can cause a large difference in rejection rate
- $k = \left(\frac{\sigma_q}{\sigma_p}\right)^d \approx \frac{1}{20,000}$

(*) The actual vol. of 2 distrs are vastly different in high-dim space $\Rightarrow$ reject samples most of time even though distrs are close

(*) solution: Don't use one Q-distri (proposal distri) to cover the target distri, but use a piece-wise envelope

(*) Adaptive rejection sampling is an example of this

- so rejection sampling doesn't work well in a high-dim space

---

(*) unnormalised importance sampling

- Draw, as before, samples $x^{(i)} \sim Q(x)$ from a proposal distri. N times
- instead of accepting/rejecting as earlier, compute weight based on new sample

$$w^{(i)} = \frac{P(x^{(i)})}{Q(x^{(i)})}$$

- store $\{x^{(i)}, w^{(i)}\}_{i=1}^{N}$ as representation of true distri.

samples

(*) Why is P suddenly available/tractable?

- Assume in simplest case that P can be eval (up to norm. constant)

(A2): Assess proof, picture.

- Proposal distri introduced as a dummy in proof
- (a) unnormalised as weights directly comp. from likelihood ratio (do not sum to one).

ex. what if I can only evaluate true distri up to a constant? (2)

e.g. no exact value of norm. constant (factor to be arbitrary α)

(*) Normalised importance sampling

- only can evaluate $p'(x) = \alpha P(x)$ (e.g. MRF)
- Get a second normalisation constant:-

(*) take expectation of the ratio

$$r(x) = \frac{p'(x)}{Q(x)} \Rightarrow \langle r(x) \rangle_Q = \int \frac{p'(x)}{Q(x)} Q(x)\,dx = \int p'(x)\,dx = \alpha$$

- expectation of ratio is normalisation constant $\alpha = \frac{r^{(i)}}{\sum_{i=1}^{N} r^{(i)}}$

- now draw $x^{(i)} \sim Q(x)$ $i = 1, \ldots, N$ (N samples)
- compute $r^{(i)}$ for each i^{th} sample.
- compute α via $\alpha = \sum_{i=1}^{N} r^{(i)}$

(A3) - confused here
- review derivations

Proof:

(*) compute expectations of function using samples drawn as above
(*) instructor presentation lost me. (*) don't need to know norm constant or part function of target

---

(*) normalised vs unnormalised importance sampling
(A4) - review at home (instructor does not emphasise).

(*) efficiency of likelihood weighting
- so far, we have used likelihood weighting → may not get 'true' answer in pecular/path scenarios
- consider practical implementation of MCMC. when do we stop sampling?
typically stop when stability/convergence observed as empir. strategy for controlling iteration/algo duration.
- in a technically equivalent way; can view this as terminating when variance ↓ of current solution.
- makes sense. (i.e. convergent behaviour through variance)
(*) so variance in a sense can be viewed as measure of closeness to (of current algo output) true answer
(*) in importance sampling; we can have a pathological case :-
(A5): Review; watching + record intuition
(*) example of poor proposal distn (note we do not know its poor); does not envelop true distn.
(*) hoping weights $w^{(i)}$ will correct this.
solution: use heavy tailed Q → (but many samples wasted)
- weighted resampling from $\{x^{(i)}, w^{(i)}\}_{i=1}^{N}$

- we resample from $\{w^{(i)}\}_{i=1}^{N}$ N' times where $N' >> N$.

(*) 'Amplify' distn using ~~resa~~ weighted resampling (A6): Review weighted resampling

---

(*) weighted resampling

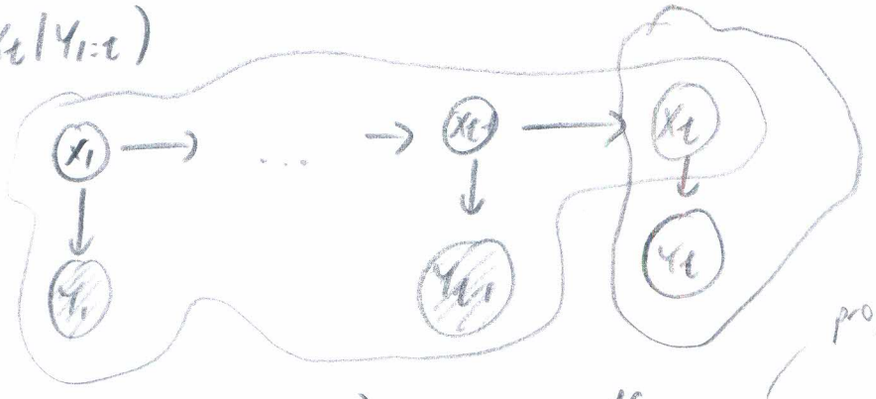
(*) Particle filtering

- use resampling idea to elegant, efficient inference
- make a fast sampling based algorithm for SSMs
- KF demanding to implement in high-dim space, or non-Gaussian (Taylor approx)

(*) Sketch of particle filtering

- use KF to establish recursive procedure

$p(x_t | y_{1:t})$



- Break down $p(x_t | y_{1:t})$ → 2 parts.

prop. 'weights'

$$p(x_t | y_{1:t}) = p(x_t | y_t, y_{1:t-1}) = \frac{p(x_t | y_{1:t-1}) p(y_t | x_t)}{\int p(x_t | y_{1:t-1}) p(y_t | x_t) dx_t}$$

- Inspect $p(x_t | y_{1:t-1})$ and use this as a proposal distrib. from which
- we sample
- Represent $p(x_t | y_{1:t})$ using above decomp:-

$$\left\{ x_t^{(i)} \sim p(x_t | y_{1:t-1}),\ w_t^{(i)} = \frac{p(y_t | x_t^{(i)})}{\sum_{i=1}^{N} p(y_t | x_t^{(i)})} \right\}$$

∴ Hence $p(x_t | y_{1:t})$ is expressed as weighted resamples from proposal distn.

(*) Goal:- use weighted resample representation of $p(x_t | y_{1:t})$ to recursively infer $p(x_{t+1} | y_{1:t+1})$

(*) sequential weighted resampler

time update: keep evidence; propagate forward with no new measurement

$$p(x_{t+1}|y_{1:t}) = \int \underbrace{p(x_{t+1}|x_t)}_{(i)} \underbrace{p(x_t|y_{1:t})}_{\hookrightarrow \text{ given by weighted res. rep.}} dx_t$$

· i) Given via model

(*) Replace integration with sampling :-

$$p(x_{t+1}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} p(x_{t+1}|x_t^{(i)})$$

- weighted sum of transition prob. cndt. on samples of previous states.

(*) similar to mixture model (sampling from).

(*) when new evidence :-

measurement update :- $p(x_{t+1}|y_{1:t+1}) = \dfrac{\overbrace{p(x_{t+1}|y_{1:t})}^{(*)} \overbrace{p(y_{t+1}|x_{t+1})}^{\text{reweights}}}{\int p(x_{t+1}|y_{1:t}) p(y_{t+1}|x_{t+1}) dx_{t+1}}$ (new prop.)

(*) note (*) is computed from time update

- Replace by weighted resampling version :-

$\Rightarrow p(x_{t+1}|y_{1:t+1})$ has representation :-

$$\left\{ x_{t+1}^{(i)} \sim p(x_{t+1}|y_{1:t}),\ w_{t+1}^{(i)} = \frac{p(y_{t+1}|x_{t+1}^{(i)})}{\sum_{i=1}^{N} p(y_{t+1}|x_{t+1}^{(i)})} \right\} \text{ reweighted}$$

(*) Particle filtering graph (illustrating resampling)

- Size of balls correspond to weights

(A2) - review

(*) Particle filtering
gives online results for predictive distri

(*) PF for SSMs (switching)

- Allows drawing from more complex distri e.g switching SSM
- multiple hidden distns.

(*) Rao-Blackwellised sampling

(RS): Review slides (not emphasised)

(*) Statistical properties of unnormalised/normalised importance sampling.

- unnormalised importance sampling is unbiased

i.e. $\mathbb{E}_Q[f(x)w(x)] = \mathbb{E}_P[f(x)]$

- normalised importance sampling is biased for finite samples e.g. $M=1$

$$\mathbb{E}_Q\left[\frac{f(x')r(x')}{\sum r(x')}\right] \neq \mathbb{E}_P[f(x)]$$