Variational Inference for NLP

29.11.2018.

- github
- learn a distribution over observed and unobserved data
- supervised – with unown probability (mass/density)

  $X \sim Cat(\pi_1, \dots \pi_k) \rightarrow$ k genders ...

  or $X \sim \mathcal{N}(\underbrace{\mu, \sigma^2}) \rightarrow$ hight of population

  estimate params that assign maximum likelihood

  – for every problem we have <u>side info</u> & <u>observation</u>

  a sentence ϕ — its translation x

  $\phi \longrightarrow x$ N

  no parse trees of sn

  $\rightarrow$.

- have NNs predict parameters of our probabilistic model and proceed to estimate params θ of the NN

  $x|\phi \sim Cat(\pi_\theta(\phi))$ or $x|\phi \sim \mathcal{N}(\mu_\theta(\phi), \sigma_\theta(\phi)^2)$

- NNs – as task-driven feature extraction
  - they don't generate the data (they don't output sentiment), but parametrise distributions that by assumption govern data (they output probability per sentiment labels). $\rightarrow$ prediction is done by us, not learning & statistics or math.

- $\mathcal{L}(\theta|x^{(1:N)}) = \log \prod_{s=1}^{N} p(x^{(s)}|\theta) = \sum_{s=1}^{N} \log p(x^{(s)}|\theta)$

  $\rightarrow_{max}$ log-likelihood fn gives us something to chase

  $\rightarrow$ given the data, we see this params

  $\rightarrow$ gradient-based optimisation

- for large N, computing the gradient is inconvenient

$$\sum_{s=1}^{N} \frac{1}{N} N \nabla_\theta \log p(x^{(s)}|\theta)$$

uniform distribution

$$= \sum_{s=1}^{N} U(s|1/N) N \nabla_\theta \log$$

$$= \mathbb{E}_{s \sim U(1/N)} [N \nabla_\theta \log p(x^{(s)}|\theta)]$$

S selects data point uniformly at random

expected gradient → don't compute E N times, but use Monte Carlo to get a sample & get fewer Ns

$$\overset{MC}{\approx} \frac{1}{M} \sum_{m=1}^{M} N \nabla_\theta \log$$

$$s_i \sim U(1/N)$$

get unbiased gradient estimation.

- DL in MLP
  - MLE → which loss to optimise (eg. neg log-likelihood)
  - automatic differentiation (backprop)
  - stochastic opt powered by backprop

- when do we have intractable likelihood?
  - → latent variables
  - $p(x|\theta) = \sum_{c=1}^{k} Cat(c|\pi_1 \dots \pi_k) \mathcal{N}(x|\mu_\theta(c), \sigma_\theta(c)^2)$

    inconvinient: discrete latent variable, continuous observations

SLIDES impossible: continuous -"-, discrete -"-

- exact gradient is intractable

$\frac{\text{joint}}{\text{marginal}}$ = conditional 19/23

do math

get expected gradient

&

do ~~MC~~ sampling? we don't know the posterior $p(z|x,\theta)$ → hard to estimate. How?

- Why latent variable modeling then??
  - better handle on stat. assumptions
  - organise a massive collection of data
  - learn from unlabelled data, semi supervised learning
  - induce discrete rep.
  - uncertainty quantification
- Deep Gen. Models → probabilistic modes parametrised by NN

II. Variational Inference: the Basics

- Generative Model
  - Joint Distribution of x and z random variables

x → observed, z → latent unobserved

$$p(x,z) = p(x)p(z|x)$$
$$= p(z)p(x|z)$$
$$\text{assumption} = p(x) \cdot p(z)$$

- $p(x|z)$ is the likelihood
- $p(z)$ is the prior over z

$$\underbrace{p(z|x)}_{\text{posterior}} = \frac{p(x|z) \cdot p(z)}{\underbrace{p(x)}_{\text{marginal likelihood / evidence}}}$$

• We want to compute ~~the~~ the posterior over latent variables $p(z|x)$. Must compute

$$p(x) = \int p(x,z)\, dz$$

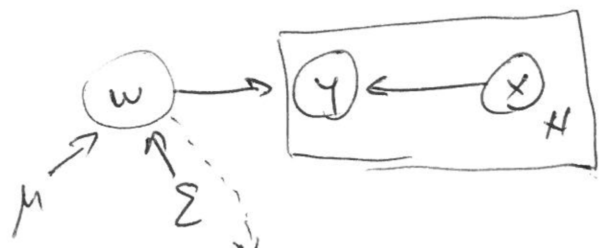
which is often intractable ⇒ approximate inference

accept that we can't solve this.

• Can't compute the posterior when:

(1) functional form of the posterior is unknown

- bayesian log-linear POS tagger



don't know distr. → assume its Gaussian.

(2) Very hard computation

Factorial HMMs

- complexity of inference $O(L^{2M} \cdot T)$

#chains

len. chain

seq. len.

LDA - assume posterior is

categorical distr. (per word)

Dirichlet distr. (for docs)

⇒ Rule of Thumb ⇒ assume the posterior is in the same family as the prior.

•

• Variational Inference

$p(z|x)$ is not computable

$\Rightarrow$ approximat by auxiliary distr. $q(z)$

Requirement: choose $q(z)$ as close as possible to $p(z|x)$

KL divergence

$$KL(q(z) \| p(z|x)) = \mathbb{E}_{q(z)}\left[\log\left(\frac{q(z)}{p(z|x)}\right)\right]$$

continuous $\int q(z)\log(\quad)dz$

discrete $\sum_z q(z)\log$

assume support of q is included in support of p.

- properties: $KL \geq 0$. (we want it 0)

$-KL \leq 0$

• VI derivation I.

$$\log p(x) = \log \int p(x,z)\,dz = \ldots = \underbrace{\log\left(\mathbb{E}_{q(z)}\left[\frac{p(x,z)}{q(z)}\right]\right) \geq \mathbb{E}_{q(z)}[\log(\,)]}_{\text{magic.}}$$

Found a lower bound for log-likelihood:

$$\log p(x) \geq \mathbb{E}_{q(z)}\left[\log\left(\frac{p(z|x)\,p(x)}{q(z)}\right)\right]$$

$$= \int q(z)\log\left(\frac{p(z|x)}{q(z)}\right)dz + \log p(x)$$

$$= -KL(q(z)\|p(z|x)) + \log p(x)$$

The ~~gap~~ introduced gap by the assumption between the actual log-likelihood and approx. is exactly

- VI derivation II.

- minimize KL

$$\min_{q(z)} KL = \max_{q(z)} -KL$$

key: ignore $\log p(x)$ because its constant

$$\max_{q(z)} \int q(z) \log \frac{p(z|x)}{q(z)} dz = \ldots = \max_{q(z)} \underbrace{\mathbb{E}_{q(z)}[\log(p(x,z))] + H(q(z))}_{\text{ELBO}}$$

ELBO
evidence lower bound

fit the data

regularizer basically so we don't go far from 0.

- Performing VI:

(1) Maximize regularized expected log-density

(2) Optimize generative model:

$$\max_{p(x,z)} E_{q(z)}[\log(p(x,z))] + \underbrace{H(q(z))}_{\text{constant. we found it}}$$

- Do (1) & (2) iteratively
- EM Algorithm - fix one thing

- Mean Field Inference
  - how to choose q? From a good family, must be tractable.
  - make all latent variables independent under $q(z)$
  - approx. posterior $q(s,z) = \prod_{t=1}^{T} q(s_t) q(z_t)$

poor approx. bcz we assume everything is independent.

Make some latent variables independent

- posterior inference is intractable cuz marginal likelihood $p(x)$
  $\Rightarrow$ VI approx. (start with mean field approx.) David Blei

III. Deep Generative Models

- $p(x, z|\theta) = p(z)\,p(x|z,\theta)$

  $\theta \rightarrow$ NN params

  marginal likelihood $p(x|\theta)$ is intractable.

- Wake-sleep Algorithm
  - generalize latent variables to NN
    (1) a generation network to model the data (θ)
    (2) an inference (recognition) network to model the latent variable (λ param.)
  - binary hidden units, "hard EM" fashion

  (1) Wake phase
    - update gen. params θ

  (2) Sleep phase
    - update λ, but can't get proper gradient $\Rightarrow$ must use fictional data $\Rightarrow$ we can go in the wrong direction
  - its supervised learning with made up labels (by inference n.)

  - pros: - conditionaly independent layer-wise updates
    - amortised inference

  cons: - trained on different objectives
    - λ are updated on fake data $\hat{x}$
    - just gets worse :(

- Turn to VI

$$\log p(x|\theta) \geq \text{ELBO} \qquad \text{\#math on slides}$$

$$\arg\max_{\theta,\lambda} \underbrace{E_{q(z|x,\lambda)}[\log p(x|z,\theta)]}_{\text{approx. by sampling}} - \underbrace{KL(q(z|x,\lambda)\,||\,p(z))}_{\text{easy, analytical true}}$$

- Generator Network ~~Gener~~ Gradient

$$\frac{\partial}{\partial \theta}\left\{ E_{q}\ldots[\log\ ] - \cancel{KL} \right\}$$

constant

$$= \mathbb{E}_{q(z|x,\lambda)}\left[\frac{\partial}{\partial \theta} \log p(x|z,\theta)\right]$$

$$\overset{MC}{\approx} \frac{1}{S}\sum_{i=1}^{S} \frac{\partial}{\partial \theta}$$

we can do it easily with backprop

- Inference Network

  - KL part is analytical computation
  - focus again on:

$$\frac{\partial}{\partial \lambda} \mathbb{E}_{q(z|x,\lambda)}[\log p(x|z,\theta)]$$

→ gradient is stuck, cant compute ⇒ reparametrisation trick

- $h: z \rightarrow \epsilon$, ϵ doesn't depend on λ.

I'm lost

- Gaussian Transformation for
  - we can write any ~~standard~~ Gaussian with $\mathcal{N}(0,1)$

$$h(z,\lambda) = \frac{z-\mu(\lambda)}{\sigma(\lambda)} = \epsilon \sim \mathcal{N}(0,1)$$

replace with standard Gaussian so $\frac{\partial}{\partial \lambda}$ can go in:

$\int \mathcal{N}(z|\mu,\sigma^2) \log p(x|z) dz$

⇓

$\mathcal{N}(\epsilon|0,1) \log p(x|z) d\epsilon$

$$\int \underbrace{q(\epsilon)}_{\text{standard Gauss.}} \frac{\partial}{\partial \lambda} \log\Big(p(x|\ \underbrace{\overbrace{h^{-1}(\epsilon,\lambda)}^{z}}_{\text{shift}},\theta)\Big) d\epsilon$$

- now we ~~have~~ can sample

- Unigram Document Model
  - slides

$$h = relu(W_1 z + b_1)$$

$$f(z, \theta) = \underbrace{softmax}_{\text{dim. of vocabulary (probabilities)}}(W_2 h + b_2)$$

Gen. net

$$\theta = \{W_1, b_1, W_2, b_2\}$$

Inference model:

average emb. of doc.

$$s = \sum_{i=1}^{N} E_{x_i}$$

$$h = relu(M_1 s + c_1)$$

$$\mu(x_1^N, \lambda) = M_2 h + c_2$$

$$\sigma(x_1^N, \lambda) = softplus(M_3 h + c_3)$$

$$\lambda = \{E, M_1, M_2, M_3, c_1, c_2, c_3\}$$

- VAE trains both networks with the same objective
  → solves wake-sleep paper

IV. DGM: Discrete Latent Variable

- reparametrisation gradient

$$\ldots \quad - \frac{\partial}{\partial \theta} KL(q(z|\lambda) \,||\, \underbrace{p(z|\theta)}_{\text{depends on } \theta\text{, not constant anymore.}})$$

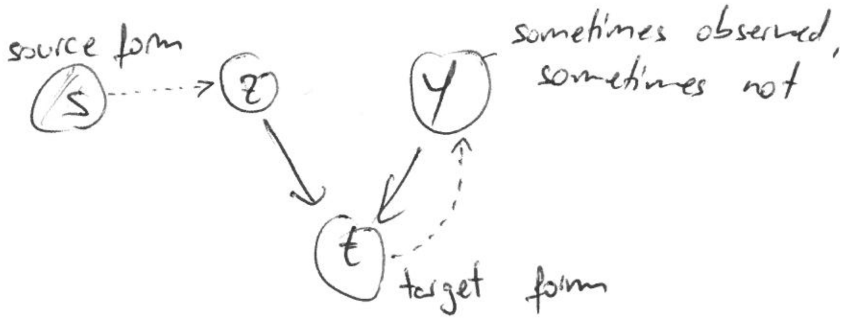
However, it's easy and stuff exist that solves it easily.

- Gotta do more manipulation to do it for discrete variables (in latent space)
  depends on the problem you're working on
  - Compute the Jacobian to transform the variables
    → transformation will expand/shrink the space, must control with | |

- Example – cobine cont. & disc. latent variables
  → partially observed data

- Morphological Reinflection task
  - plays → played

  z – lemma (real vector)
  y – morph. information – gender of a word etc. (discrete vector)



simple posterior approximation

IV.

What we know?

- DGMs
- objective - lowerbond on log-likelihood → can't be computed exactly → MC
  - MC is not differentiable → score function estimator
    → reparameterised gradient

- Gamma distribution - distr. of Gaussian σ
- Derichlet - distr. over categorical distributions
- retractable ex. If we have Gaussian → reparametrisation
- ADVI - Automatic Differentation variational inference

VI. Normalising Flows

- map transform complex distributions to simpler ones (NFs)
- we need a more flexible environment

→ not just for known distributions → ~~...~~
ones we know
the reparametrisation for
↓
express the density of
variable Y in terms of a var X.
Assume that a differentiable, inveribl
mapping $h: X \to Y$ exists.

- → Let's ~~...~~ learn the transformation h (or its inverse)
- our data x has unknown cont. density $p(x)$. Can't handcraft alikelihood → word embeddings