Marrying Graphical Models & Deep Learning

Presenter: Shijia Wang

Max Welling[1]

[1]University of Amsterdam

Deep Learning (DLSS) and Reinforcement Learning (RLSS) Summer School, Montreal 2017

Outline

Outline

Overview

- often time learning problem is seen as an optimization problem
- but it is more than just optimization
- optimize maximum log likelihood (unsupervised and supervised)
- optimize minimal loss (supervised)
- draw independent and identically distributed (iid) from data sets
- optimal parameters are different due to different draws
- does not make sense to optimize further on a data set because you are overfitting

Bias Variance Tradeoff

- overfitting and generalization
- bias is the systematic error
- variance is the sampling error
- error $=$ bias $+$ variance $+$ irreducible error
- bias hard to estimate since it depends on the true distribution
- variance easy to estimate since it only depends on estimator
- irreducible error from $N(0, \sigma)$
- more complex model - higher variance but lower bias
- less complex model - higher bias but lower variance

Outline

Bayes Net

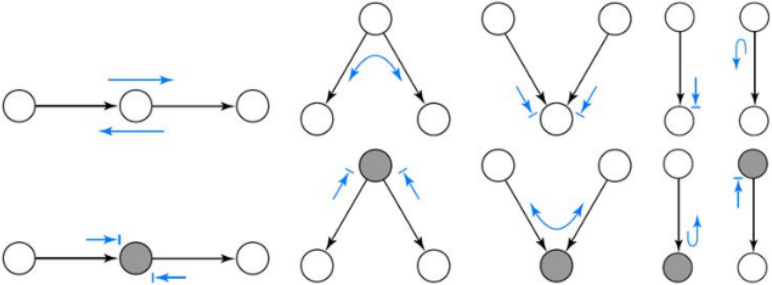
- good idea of what variables and relationships mean rather than with neural net
- write down a joint probability distribution over all random variables
- displays conditional probabilities
- P(all) $= \prod$ P(child — parents)
- "explain away" - the fact that a child is true given that a parent is true

Bayes ball algorithm

- used to determine conditional relationships
- if 2 variables are independent there can be no path between them
- if conditional upon a node, stop path

Bayes ball algorithm

An undirected path is active if a Bayes ball travelling along it never encounters the "stop" symbol: $\longrightarrow\!|$

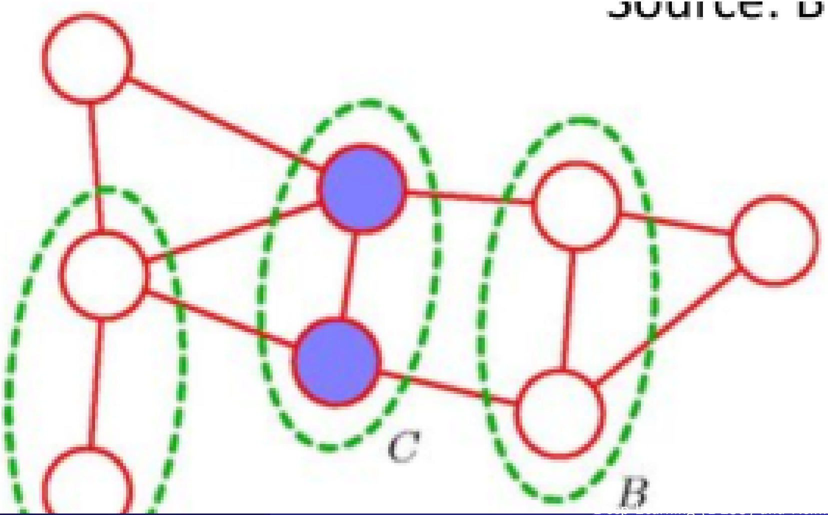


If there are no active paths from X to Y when $\{Z_1, \ldots, Z_k\}$ are shaded, then $X \perp\!\!\!\perp Y \mid \{Z_1, \ldots, Z_k\}$.

Outline

Markov Random Fields

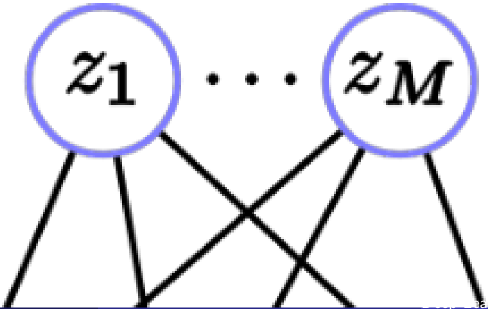
- for independence all paths must be blocked
- maximal clique: largest completely connect subgraphs



Outline

Latent Variable Models

- observed random variables and introduce stochastic latent variables
- don't know what exactly the latent variables are
- distribution over observed variable x
- marginalize over P(X) = sum of $P(X|Z)P(Z)$
- problem is $P(Z|X)$ is intractable for most nontrivial models



Outline

Variational Inference

- try to find a target distribution p
- try to find an approximating distribution inside a family of fully tractable distributions
- approximates the distribution closest to the true distribution with some distance, possible KL divergence
- deterministic
- biased within the family
- local minima
- easy to assess convergence

Sampling

- take distribution and represent it with randomly sampled points
- compute expectation by evaluation the function at these points, sum them, then average them
- stochastic sample error variance to the estimation
- unbiased on average
- hard to mix between multiple modes
- hard to assess convergence

Outline

Generating Independent Samples

- Rejection Sampling - draw everything around and delete those not in the true distribution
  doesn't work in high dimension
- Importance Sampling - assign weights to the sampling in respect to their closeness with the true distribution
  doesn't work in high dimension
- don't work in high dimension due to the ratio of acceptable samples decrease to 0

MCMC - Markov Chain Monte Carlo

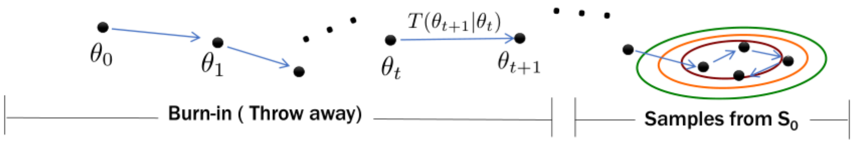
- work in high dimension
- draw first sample from a completely wrong distribution, have transition kernel so next sample is drawn from a different distribution
- drawn distributions converges to the true distribution
- first few are from wrong distribution, throw away "burn in"
- those in the correct distribution are collected
- Bayesian procedure to sample the parameters of a model given the data
- range of sampling is given by the maximum likelihood function to the true distribution

MCMC

- can think of this as training a neural net to get to correct distribution
- first samples are random thrown away
- transition probability object, self designed T pick next point given previous point
- property that T eventually will sample from the correct distribution
- mix very slowly - moves through the support of the posterior distribution slowly since two subsequent samples are highly correlated
- the faster it mixes, the faster the dependence decays, faster convergence

MCMC

Given target distribution S_0, design transitions s.t. $p_t(\theta_t) \rightarrow S_0$ as $t \rightarrow \infty$

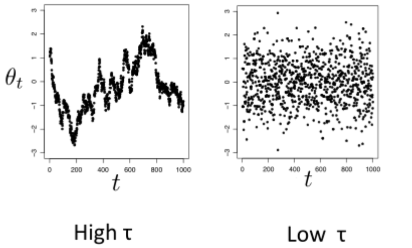


$$I = \langle f \rangle_{S_0} \approx \hat{I} = \frac{1}{T} \sum_{t=1}^{T} f(\theta_t)$$

$$\text{Bias}(\hat{I}) = \mathbb{E}[\hat{I} - I] = 0$$

$$\text{Var}(\hat{I}) = \tau \frac{\text{Var}(f)}{T}$$

Auto correlation time



MCMC

- autocorrelation is very high /tau
- bad mixing samplers pay a high time price
- bias variance decomposition
- if works, then bias should go to 0; unbiased estimator
- there is always a variance; τ autocorrelation coefficient
- $\tau = 1$ every sample is independent
- other wise if τ gets bigger every sample counts only a fraction of independent sample so you need to draw many more samples

Metropolis-Hastings

- design transition kernels
- simple, doesn't really work very well but convenient
- if at θ_t choose next from distribution q
- then evaluate probability of accepting new sample
- if accept, add sample to stack
- if reject add current sample to stack; 2 copies of current
- order O(N) expensive to compute; burn ins are slow
- variance is too high

Metropolis-Hastings

- design transition kernels $T(\theta_{t+1}|\theta_t)$
- Propose: $\theta' \sim q(\theta'|\theta_t)$
- Accept/Reject T: $P_a = min(1, \frac{q(\theta_t|\theta')}{q(\theta'|\theta_t)} \frac{S_0(\theta')}{S_0(\theta)})$
- first fraction says if it easy to come back to current state
- second fraction says if the new state more probable

Approximate MCMC

- not really afraid of bias
- can tradeoff bias to get the distribution we want
- want to compute in a finite short amount of time
- if relaxing on true distribution, we can draw a lot of samples with low variance but high bias
- normal MCMC procedure, can only draw fewer samples: high variance(slow) but low bias
- find some middle ground but setting bias to 0 may not be optimal

Stochastic Gradient Langevin Dynamics

- mimics SGD but when it gets close to the target distribution it will start sampling from the full posterior distribution
- adds a normally-distributed noise whose variance is 2 times the step size

Langevin

Welling & Teh 2011

Gradient Ascent

$$\Delta\theta_t = \frac{\epsilon}{2}\left(\nabla \log p(\theta_t) + \sum_{i=1}^{N} \nabla \log p(x_i;\theta_t)\right)$$

Langevin Dynamics

$$\Delta\theta_t = \frac{\epsilon}{2}\left(\nabla \log p(\theta_t) + \sum_{i=1}^{N} \nabla \log p(x_i;\theta_t)\right) + \mathbb{N}(0,\epsilon)$$

$$\downarrow$$

Metropolis-Hastings Accept Step

Stochastic Gradient Ascent

$$\Delta\theta_t = \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_{t_i};\theta_t)\right)$$

$$\sum_{t=1}^{\infty} \epsilon_t = \infty \qquad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty$$

e.g. $\epsilon_t = \frac{a}{(b+t)^\gamma}$

Stochastic Gradient Langevin Dynamics

$$\Delta\theta_t = \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_{t_i};\theta_t)\right) + \mathbb{N}(0,\epsilon_t)$$

$$\sum_{t=1}^{\infty} \epsilon_t = \infty \qquad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty$$

~~Metropolis-Hastings Accept Step~~

Noise

- when far from the optimal distribution, unbiased but noisy estimator of the full gradient variance dominates
- when closer to the optimal distribution, the added normal noise dominate

$$\Delta\theta_t = \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \underbrace{\frac{N}{n}\sum_{i=1}^{n}\nabla \log p(x_{t_i};\theta_t)}_{\mathcal{N}\left(\sum_{i=1}^{N}\nabla \log p(x_i;\theta_t),\, V(\theta_t, n)\right)}\right) + \mathbb{N}(0, \epsilon_t)$$

Variational Inference

- choose tractable family of distributions
- try to fit choosen distribution as the true distribution
- minimize KL-Divergence of choosen distribution and true distribution
- $Q(Z|X)$ tractable approximate
- $P(Z|X)$ true distribution - maximize over Φ in $\sum_Z Q(Z|X,\Phi)(logP(X|Z,\Theta)P(Z) - logQ(Z|X,\Phi))$

Expectation Maximization

- way to train a model that has latent variable Z
- X is observed
- $logP(X|\Theta) = log \sum_Z P(X|Z,\Theta)P(Z) \geq \sum_Z Q(Z|X,\Phi)(logP(X|Z,\Theta)P(Z) - logQ(Z|X,\Phi)) = B(\Theta,\Phi)$
- E-step: $argmax_\Phi B(\Theta,\Phi)$ inference train Q distribution
- M-step: $argmax_\Theta B(\Theta,\Phi)$ for P distribution

Connection between Deep Net and Graphical Models

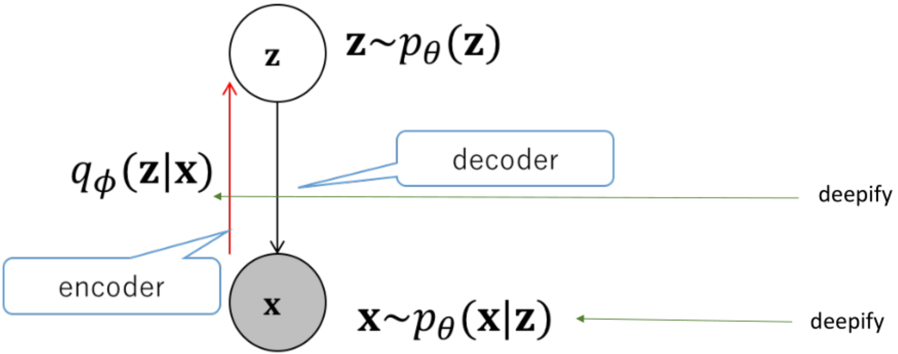
- in graphical models every node is a random variable
- in neural nets every node is an activation function
- for every conditional distribution we can insert a neural net

"Deepify" Operator

- for a graphical model with conditional distributions and replace those with a deep neural networks
- latent variable model: replace generative and recognition models with deep neural networks

Variational Autoencoder

- a distribution of $p(z|x)$
- another distribution of $p(x|z)$
- substitute conditional probabilities with deep network
- may have high variance so use a method to reparameterize the hidden state



Outline

Advantages of Generative Models

- Inject expert knowledge in the form of conditional probabilities
- Model causal relations generalize much better and are much more stable in predictions
- Interpretable
- Data efficient due to expert knowledge
- More robust to domain shift due to model causal relations
- Facilitate un/semi-supervised learning blackbox

Outline

Advantages of Discriminative Models

- Flexible map from input to target (low bias) but high variance
- Efficient training algorithms available
- Solve the problem you are evaluating on.
- Very successful and accurate

Autoencoder

- Use 2 neural networks
- the classifier is a discriminative model
- the generator is a generative model to insert bias

Conclusion

- Optimization is important in getting god solutions
- deep learning is also statistics not just optimization
- deep learning can be combined with classical graphical models
- a lot we do not know about deep learning