iPMR

Our Journey

Graphical Models	Decision Theory	Learning Probabilistic Models	Mixture and Factor Models	Markov Models	Approximate Inference

- Exponential Family
- Gaussian Distribution
  - Factor Models
- Gaussian Mixture Models
- Boltzmann Machine
- Deep Learning Methods

Our Journey

Graphical Models	Decision Theory	Learning Probabilistic Models	Mixture and Factor Models	Markov Models	Approximate Inference

- Gibbs Sampling Quick Summary

Quick Summary on Sampling

- If you did this in MLPR this is revision.

- Suppose we have an expectation we wish to compute: i.e. an integral

$$A = \langle f(\theta) \rangle_P = \int d\theta P(\theta) f(\theta)$$

  This occurs often: compute mean of distribution. Compute error for distribution. Compute best prediction for a distribution etc.
- Cannot compute it. But can sample (draw, get artificial data) from $P(\theta)$.
- Use

$$A \approx \tilde{A} = \frac{1}{N_S} \sum_{i=1}^{N_S} f(\theta_i)$$

  where θ_i are samples from $P(\theta)$.
- This is a Monte-Carlo approximation.

- But how do we get samples? Use properties of Markov Chains:
  - Ergodicity: a Markov chain is ergodic if you would expect to get from each state to any other state in finite time, and if it is acyclic: its return time to any state is not always divisible by a number > 1.
  - Reversibility: a Markov chain is reversible iff it satisfies detailed balance: for some distribution P_B: $P_B(\theta)P_T(\phi|\theta) = P_B(\phi)P_T(\theta|\phi)$
  - Equilibrium Distribution: an ergodic Markov chain has a unique equilibrium distribution $P_\infty(\theta)$ such that

$$P_\infty(\theta) = \int d\theta' \; P_T(\theta|\theta')P_\infty(\theta')$$

  - An ergodic reversible Markov chain satisfying detailed balance wrt P_B has P_B as its unique equilibrium distribution.

How?

- Did not know how to sample from a distribution $P(\theta)$.
- Idea: Use a Markov chain. Design so $P(\theta)$ is equilibrium distribution.
- Run Markov chain sampling 'for long enough' to get samples from equilibrium distribution.
- How to design Markov chain? Ensure satisfies detailed balance wrt. $P(\theta)$,
- Sampling from a chain:
- Initialise state θ_0. Compute $P_T(\theta_1|\theta_0)$. Sample from this to get θ_1. Repeat ad infinitum (or until you get bored).
- Markov Chain Monte-Carlo (MCMC)

Gibbs Sampling

- Markov chain: Adapt θ_i keeping all $\theta_{j \neq i}$ fixed. i.e.
- Choose i uniformly from $i = 1, 2, \ldots, D$. Set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$. Then sample $\theta_{t+1,i}$ from the conditional probability $P(\theta_{t+1,i} | \theta_{t+1,\neq i})$ where $\theta_{t+1,\neq i}$ denotes the set $\{\theta_{t+1,j} | j \neq i\}$.
- Repeat.
- Can cycle through i either (this is not reversible, but can be shown to have a unique equilibrium distribution)

Matlab Demos

Sampling

- End Of Summary. Questions.

Our Journey

Graphical Models	Decision Theory	Learning Probabilistic Models	Mixture and Factor Models	Markov Models	Approximate Inference

- Exponential Family
- Gaussian Distribution
  - Factor Models
- Gaussian Mixture Models
- Boltzmann Machine
- Deep Learning Methods

The Gaussian

- Remember the good old Gaussian

$$P(\mathbf{x}) = \frac{1}{Z}\exp(-E(\mathbf{x}))$$

where

$$\begin{aligned} E(\mathbf{x}) &= \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Lambda}(\mathbf{x}-\boldsymbol{\mu}) \\ &= \frac{1}{2}\mathbf{x}^T\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \text{const} \end{aligned}$$

- x is real valued.
- Does it have to be in these equations?
- What happens to Z if it isn't?

The Boltzmann Machine

- The Boltzmann Machine has the form

$$P(\mathbf{x}) = \frac{1}{Z}\exp(-E(\mathbf{x}))$$

where

$$E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{W}\mathbf{x} + \mathbf{b}^T\mathbf{x}$$

$$x_i \in \{0, 1\}$$

- but where x is a binary vector
- What is Z?
- So what does a Boltzmann Machine do?
- What sort of information can be captured? Discuss…

The Energy Function

- The Energy function E determines the regions of high and low probabilities.
- In 2D:
- In High D:

- Some model features:

- Q: Show that if $x_i \in \{-1, 1\}$ that is also a Boltzmann Machine
- Q: Show that W might as well be symmetric.
- Q: Show that W might as well be positive definite…
- …or W might as well have zero diagonal.

- Q: Show that if $x_i \in \{-1, 1\}$ that is also a Boltzmann Machine

- Q: Show that W might as well be symmetric.

- Q: Show that W might as well be positive definite…

- …or W might as well have zero diagonal.

Fully Visible Model

- Consider the case where x is all visible.
- If *b*=0 and $W = \sum_n \mathbf{x}^n (\mathbf{x}^n)^T$
- Then what is the Energy function like?

- But x could be split into visible units y and hidden units h.
- Then what form does P(x) take?

Learning

- Given data $D = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$
- How do we learn the parameters of the Boltzmann Machine?

Learning

- But this doesn't really work…
- Why?

- Various issues:
  - Signal to noise problems
  - Sampling error induces random walk behaviour
  - The gradient gets small in the tails of the sigmoids.

The Restricted Boltzmann Machine

- The Restricted Boltzmann Machine has the form

$$P(\mathbf{x}) = \frac{1}{Z}\exp(-E(\mathbf{x}))$$

where

$$E(\mathbf{x}) = \mathbf{v}^T\mathbf{W}\mathbf{h} + \mathbf{a}^T\mathbf{v} + \mathbf{b}^T\mathbf{h}$$

- What is its graphical structure?

- What are its conditional independence relationships?

Why is this a benefit?

- How do we do learning in Restricted Boltzmann Machines?

Representations

- What are the hidden units in an RBM?

Stacked RBMs

- Here is a cheat.
- Having learnt an RBM. We have a mapping from visible to hidden units.
- Given the visibles we can obtain a hidden representation.
- In fact we could just focus on this representation as a summary for the data.
- And we could learn another RBM for that representation
- And so on

Graphically

- How does this work pictorially?
- The result is a deep belief network.

Representation Learning

- Issue: Machine learning is dependent on representation.
- Need method of learning good representations.
- Method needs to be unsupervised.
- Representations are hierarchical.

- Deep Learning does representation learning.

Deep Networks in Action

- Top performing methods in many scenarios.