Probabilistic Models: Introduction

CS 598: Deep Generative and Dynamical Models

Instructor: Arindam Banerjee

August 24, 2021

Overview: Probabilistic Models

- Probability Overview
- Bayesian Networks, Graphical Models
- Approximate Inference:
  - Markov Chain Monte Carlo (MCMC)
  - Variational Inference (VI)
- Expectation Maximization
- Dynamical Models
  - Filtering, Prediction, Smoothing
  - Examples: HMMs, KFs, DBNs
- Losses and Representation
  - Losses from generalized linear models
  - Beyond linear representations
- Scoring rules, Calibration

Probability Basics

- Sample space Ω of events
- Each "event" $\omega \in \Omega$ has an associated "measure"
  - Probability of the event $P(\omega)$
- Axioms of Probability:
  - $\forall \omega, P(\omega) \in [0,1]$
  - $P(\Omega) = 1$
  - $P(\omega_1 \cup \omega_2) = P(\omega_1) + P(\omega_2) - P(\omega_1 \cap \omega_2)$

- Note: We are being informal
- Some good references
  - Oliver Knill's book, great introduction: `https://abel.math.harvard.edu/~knill/books/KnillProbability.pdf`
  - David Williams' book, great exposure to the advanced stuff: `https://www.amazon.com/Probability-Martingales-Cambridge-Mathematical-Textbooks/dp/0521406056`

Random Variables

- Random variables are mappings of events (to real numbers)
  - Mapping $X : \Omega \mapsto \mathbb{R}$
  - Any event ω maps to $X(\omega)$
- Example:
  - Tossing a coin has two possible outcomes
  - Denoted by $\{H, T\} \mapsto \{1, 0\}$
  - Fair coin has uniform probabilities

$$P(X = 0) = \frac{1}{2} \qquad P(X = 1) = \frac{1}{2}$$

- Random variables (r.v.s) can be
  - Discrete, e.g., Bernoulli
  - Continuous, e.g., Gaussian

Distribution, Density

- For a continuous r.v.
  - Distribution function $F(x) = P(X \leq x)$
  - Corresponding density function $f(x)$, $f(x)dx = dF(x)$
  - Note that

$$F(x) = \int_{t=-\infty}^{x} f(t)dt$$

- For a discrete r.v.
  - Probability mass function $f(x) = P(X = x) = p(x)$
  - We will call this the probability of a discrete event
  - Distribution function $F(x) = P(X \leq x)$

Joint Distributions, Marginals

- For two continuous r.v.s X_1, X_2
  - Joint distribution $F(x_1, x_2) = P(X_1 \leq x_1, X_2 \leq x_2)$
  - Joint density function $f(x_1, x_2)$ can be defined as before
  - The marginal probability density

$$f(x_1) = \int_{x_2=-\infty}^{\infty} f(x_1, x_2) dx_2$$

- For two discrete r.v.s X_1, X_2
  - Joint probability $f(x_1, x_2) = P(X_1 = x_1, X_2 = x_2) = p(x_1, x_2)$
  - The marginal probability

$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

- Can be extended to joint distribution over several r.v.s
- Many *hard* problems involve computing marginals

Expectation

- The expected value of a r.v. X
  - For continuous r.v.s $\mathbb{E}[X] = \int_x xp(x)dx$
  - For discrete r.v. $\mathbb{E}[X] = \sum_i x_i p_i$
- Expectation is a linear operator

$$\mathbb{E}[aX + bY + c] = a\mathbb{E}[X] + b\mathbb{E}[Y] + c$$

- Expectation of a function of a r.v. X

$$\mathbb{E}[f(X)] = \int_x f(x)p(x)dx$$

Independence

- Joint probability $P(X_1 = x_1, X_2 = x_2)$
  - X_1, X_2 are different dice
  - X_1 denotes if grass is wet, X_2 denotes if sprinkler was on
- Two r.v.s are independent if

$$P(X_1 = x_1, X_2 = x_2) = P(X_1 = x_1)P(X_2 = x_2)$$

  - Two different dice are independent
  - If sprinkler was on, then grass will be wet $\Rightarrow$ dependent

Conditional Probability, Bayes Rule

	Grass Wet	Grass Dry
Sprinkler On	0.4	0.1
Sprinkler Off	0.2	0.3

- Inference problems:
  - Given 'grass wet' what is $P(\text{'sprinkler on'}|\text{'grass wet'})$
  - Given 'symptom' what is $P(\text{'disease'}|\text{'symptom'})$
- For any r.v.s X, Y, the conditional probability (forward model)
$$P(x|y) = \frac{P(x,y)}{P(y)}$$
- Since $P(x,y) = P(y|x)P(x)$, posterior probability (inference)
$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$
- Expressing 'posterior' in terms of 'conditional': Bayes Rule

Product Rule & Independence

- Product Rule:
  - For X_1, X_2, $P(X_1, X_2) = P(X_1)P(X_2|X_1)$
  - For X_1, X_2, X_3, $P(X_1, X_2, X_3) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)$
  - In general, the chain rule

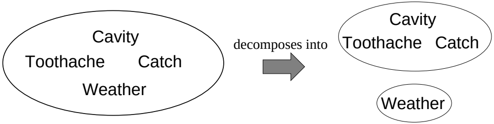
$$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1})$$

- Example: Joint distribution of n Boolean variables
  - Specification requires $2^n - 1$ parameters
- Recall Independence:
  - For X_1, X_2, $P(X_1, X_2) = P(X_1)P(X_2)$
  - In general

$$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P(X_i)$$

- Independence reduces specification to n parameters

Independence



- Consider 4 variables: Toothache, Catch, Cavity, Weather
- Independence implies

$$P(Toothache, Catch, Cavity, Weather) = P(Toothache, Catch, Cavity)P(Weather)$$

- Absolute independence helpful but rare

Conditional Independence

- X and Y are conditionally independent given Z
$$P(X, Y|Z) = P(X|Z)P(Y|Z)$$
- Example:
$$\begin{aligned} & P(Toothache, Catch|Cavity) \\ & \quad = P(Toothache|Cavity)P(Catch|Cavity) \end{aligned}$$
- Conditional Independence simplifies joint distributions
  - Often reduces from exponential to linear in n

$$P(X, Y, Z) = P(Z)P(X|Z)P(Y|Z)$$

Naive Bayes Model

- If $X_1, \ldots, X_n$ are independent given Y

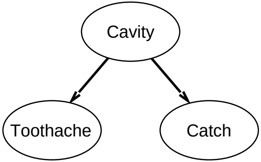
$$P(Y, X_1, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i | Y)$$

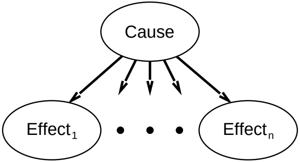
- Example:

$$P(Cavity, Toothache, Catch)$$
$$= P(Cavity) P(Toothache | Cavity) P(Catch | Cavity)$$

- More generally

$$P(Cause, Effect_1, \ldots, Effect_n) = P(Cause) \prod_{i=1}^{n} P(Effect_i | Cause)$$





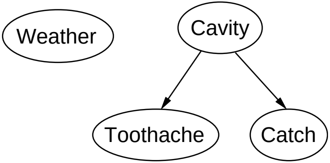
Bayesian networks

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

- Syntax
  - A set of nodes, one per variable
  - A directed, acyclic graph (link implies direct influence)
  - A conditional distribution for each node given its parents
- Conditional distributions
  - For each X_i, $P(X_i|Parents(X_i))$
  - In the form of a *conditional probability table* (CPT)
    - Distribution of X_i for each combination of parent values

Example

Topology of network encodes conditional independence assertions



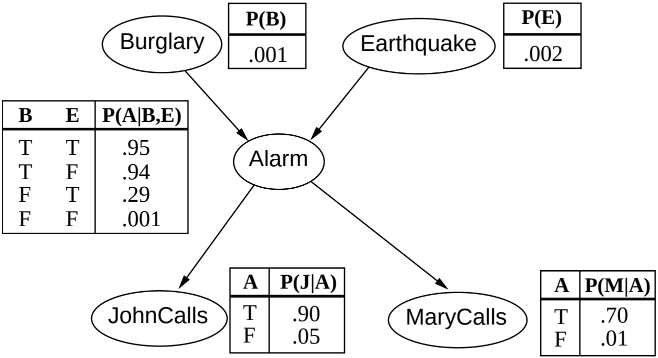
- *Weather* is independent of the other variables
- *Toothache*, *Catch* are conditionally independent given *Cavity*

Example

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

- Variables: *Burglar*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*
- Network topology reflects "causal" knowledge
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call

Example (Contd.)



P(B)
.001

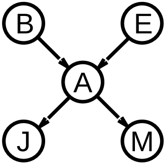
P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Compactness



- A CPT for Boolean X_i with k Boolean parents
  - 2^k rows for the combinations of parent values
  - Each row requires one number
- Each variable has no more than k parents
  - The complete network requires $O(n \cdot 2^k)$ numbers
  - Grows linearly with n
  - Full joint distribution requires $O(2^n)$
- Example: Burglary network
  - Full joint distribution requires $2^5 - 1 = 31$ numbers
  - Bayes net requires 10 numbers

Global semantics

- Full joint distribution
  - Can be written as product of local conditionals
- Example:

$$P(j, m, a, \neg b, \neg e) = P(\neg b)P(\neg e)P(a|\neg b, \neg e)P(j|a)P(m|a)$$

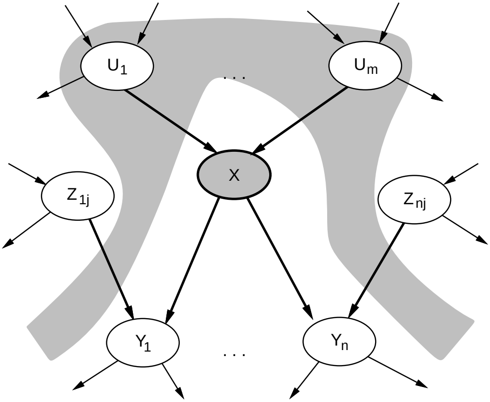
- Example:

$$P(j, \neg m, a, b, \neg e) = P(b)P(\neg e)P(a|b, \neg e)P(j|a)P(\neg m|a)$$

- Can we compute $P(b|j, \neg m)$?

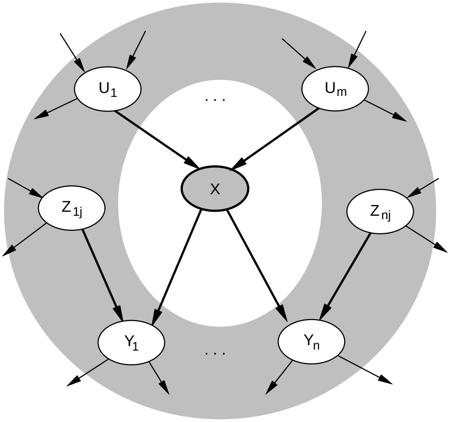
Local semantics

Each node is conditionally independent of its nondescendants given its parents

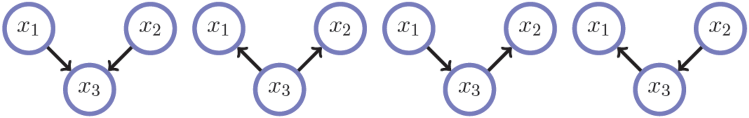


Markov blanket

Each node is conditionally independent of all others given its Markov blanket, i.e., parents + children + children's parents



Conditional Independence in BNs

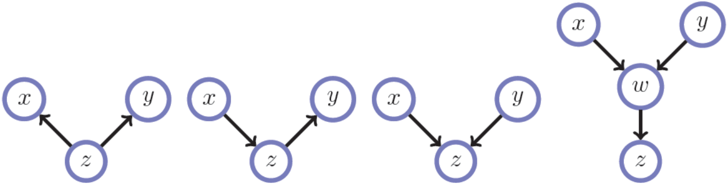


(a) (b) (c) (d)

Which BNs support $x_1 \perp x_2 | x_3$

- For (a), x_1, x_2 are dependent, x_3 is a *collider*
- For (b)-(d), $x_1 \perp x_2 | x_3$

Conditional Independence (Contd.)



(a) (b) (c) (d)

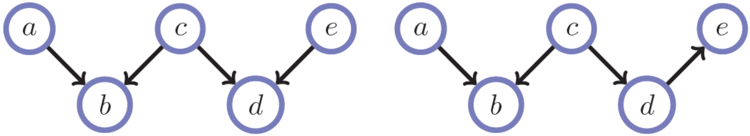
Which BNs support $x \perp y | z$

- For (a)-(b), z is not a collider, so $x \perp y | z$
- For (c), z is a collider, so x and y are conditionally dependent
- For (d), w is a collider, and z is a descendent of w, so x and y are conditionally dependent

d-connection, d-separation

- Definition (d-connection): X, Y, Z be disjoint sets of vertices in a directed graph G. X, Y is d-connected by Z iff $\exists$ an undirected path U between some $x \in X, y \in Y$ such that
  - for every collider C on U, either C or a descendent of C is in Z, and
  - no non-collider on U is in Z
- Otherwise X and Y are d-separated by Z
- If Z d-separates X and Y, then $X \perp Y | Z$ for all distributions represented by the graph

Conditional Independence (Contd.)

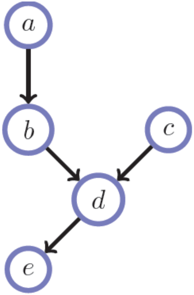


(a) (b)

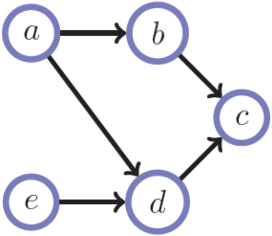
Examples

- For (a), $a \perp e | b$; but a, e are dependent given $\{b, d\}$
- For (b) a and e are dependent given b; c and e are unconditionally dependent

Conditional Independence: More Examples



(a)



(b)

- For (a), Is $a \perp c | e$? Is $a \perp e | b$? Is $a \perp e | c$?
- For (b), Is $a \perp e | d$? Is $a \perp e | c$? Is $a \perp c | b$

Constructing Bayesian networks

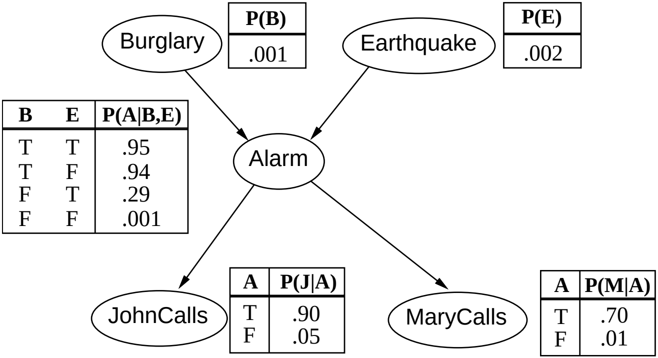
- Hard problem in general: Structure learning
- Choose an ordering of variables $X_1, \ldots, X_n$
- For $i = 1$ to n
  - Add X_i to the network
  - Select parents from $X_1, \ldots, X_{i-1}$ such that

$$P(X_i | Parents(X_i)) = P(X_i | X_1, \ldots, X_{i-1})$$

This choice of parents guarantees global semantics

$$\begin{aligned} P(X_1, \ldots, X_n) &= \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1}) \\ &= \prod_{i=1}^{n} P(X_i | Parents(X_i)) \end{aligned}$$

Example: Burglary Network, Causal Order



P(B)
.001

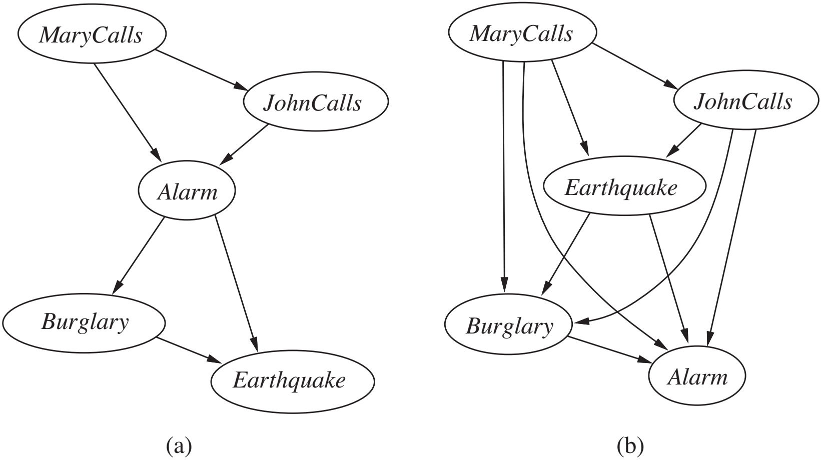
P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Example: Burglary Network, Other Orders

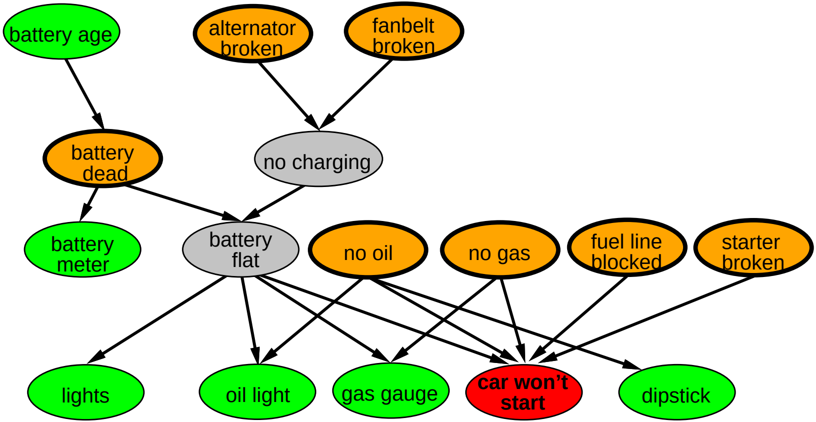


Example: Car diagnosis

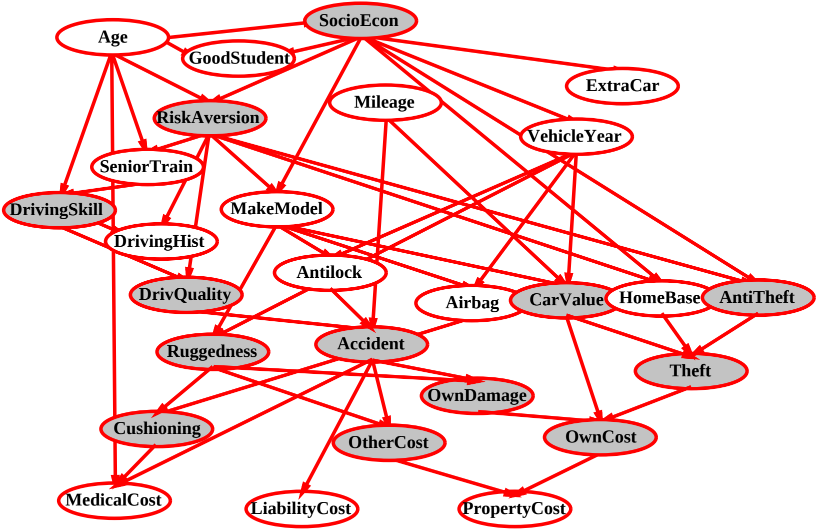
Initial evidence: car won't start

Testable variables (green), "broken, so fix it" variables (orange)

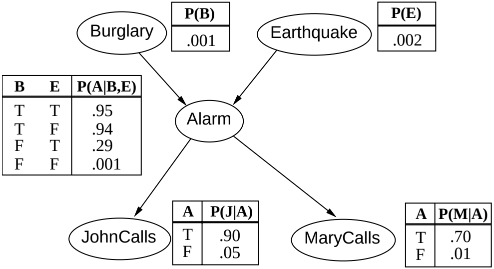
Hidden variables (gray) ensure sparse structure, reduce parameters



Example: Car insurance

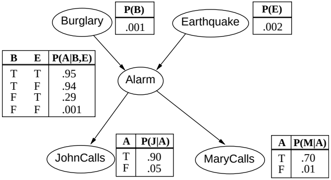


Inference



How can we compute $P(b|j, \neg m)$?

Graphical Models: Two (Three) Problems of Interest



- Structure learning
  - Given samples, find undirected/directed dependency structure
  - Not causality, but statistical (in)dependence
- Parameter (conditional probability) estimation
  - Given samples and structure, estimate conditional probabilities
  - 'Easy' without latent variables
- Inference
  - Given observed samples or components
  - Infer properties of latent variable distribution

Overview: Probabilistic Models

- Probability Overview
- Bayesian Networks, Graphical Models
- Approximate Inference:
  - Markov Chain Monte Carlo (MCMC)
  - Variational Inference (VI)
- Expectation Maximization
- Dynamical Models
  - Filtering, Prediction, Smoothing
  - Examples: HMMs, KFs, DBNs
- Losses and Representation
  - Losses from generalized linear models
  - Beyond linear representations
- Scoring rules, Calibration

Inference and Estimation Problems

- Joint distribution of a latent variable model (LVM)
$$p_\theta(\mathrm{x},\mathrm{z}) = p_\theta(\mathrm{z})p_\theta(\mathrm{x}|\mathrm{z}) \ ,$$
  - x denotes the observed variable
  - z denotes the latent variable
  - θ denotes the parameters
- Problems of interest
  - Compute marginal or conditional distributions
$$p_\theta(\mathrm{x}) = \int_{\mathrm{z}} p_\theta(\mathrm{x},\mathrm{z})dz \qquad\qquad p_\theta(\mathrm{z}|\mathrm{x}) = \frac{p_\theta(\mathrm{x},\mathrm{z})}{p_\theta(\mathrm{x})}$$
  - Estimate θ by optimizing a function of $p_\theta(\mathrm{x})$
- Problems need to (approximately) compute high-d integrals

Monte Carlo Principle

- Target density $p(x)$ on a high-dimensional space
- Draw i.i.d. samples $\{x_i\}_{i=1}^n$ from $p(x)$
- Construct empirical point mass function

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}(x)$$

- One can approximate integrals/sums by

$$I_n(f) = \frac{1}{n} \sum_{i=1}^{n} f(x_i) \xrightarrow[n \to \infty]{a.s.} I(f) = \int_x f(x) p(x) dx$$

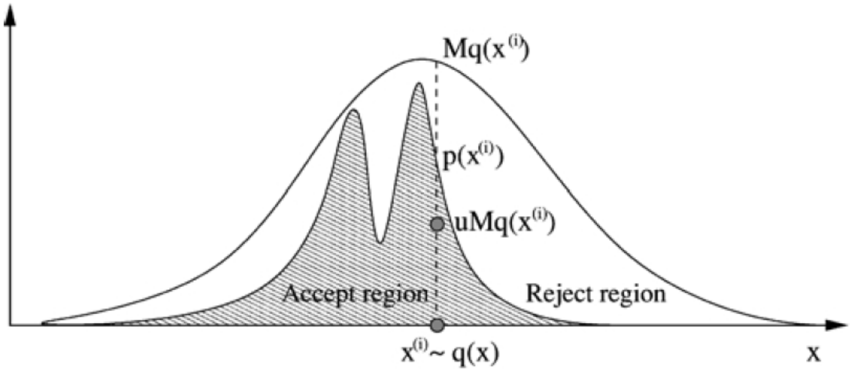
- Unbiased estimate $I_n(f)$ converges by strong law
- For finite σ_f^2, central limit theorem implies

$$\sqrt{n}(I_n(f) - I(f)) \underset{n \to \infty}{\Longrightarrow} \mathcal{N}(0, \sigma_f^2)$$

Rejection Sampling

- Target density $p(x)$ is known, but hard to sample
- Use an easy to sample *proposal distribution* $q(x)$
- $q(x)$ satisfies $p(x) \leq Mq(x), M < \infty$
- Algorithm: For $i = 1, \cdots, n$
  - Sample $x_i \sim q(x)$ and $u \sim \mathcal{U}(0,1)$
  - If $u < \frac{p(x_i)}{Mq(x_i)}$, accept x_i, else reject
- Issues:
  - Tricky to bound $p(x)/q(x)$ with a reasonable constant M
  - If M is too large, acceptance probability is small

Rejection Sampling (Contd.)



Importance Sampling

- For a proposal distribution $q(x)$, with $w(x) = p(x)/q(x)$

$$I(f) = \int_x f(x)w(x)q(x)dx$$

- $w(x)$ is the importance weight
- Monte Carlo estimate of $I(f)$ based on samples $x_i \sim q(x)$

$$\hat{I}_n(f) = \sum_{i=1}^{n} f(x_i)w(x_i)$$

- The estimator is unbiased, and converges to $I(f)$ a.s.

Importance Sampling (Contd.)

- Choose $q(x)$ that minimizes variance of $\hat{I}_n(f)$
$$\text{var}_q(f(x)w(x)) = E_q[f^2(x)w^2(x)] - I^2(f)$$
- Applying Jensen's and optimizing, we get
$$q^*(x) = \frac{|f(x)|p(x)}{\int |f(x)|p(x)dx}$$
- Efficient sampling focuses on regions of high $|f(x)|p(x)$
- Super efficient sampling, variance lower than even $q(x) = p(x)$

Markov Chains

- Use a Markov chain to explore the state space
- Markov chain in a discrete space is a process with
$$p(x_i | x_{i-1}, \ldots, x_1) = T(x_i | x_{i-1})$$
- After t steps, probability of being in state x_i
$$p_t(x_i) = \sum_{x_{i'}} p_{t-1}(x_{i'}) T(x_i | x_{i'})$$
- A chain is homogenous if T is invariant over time $\forall i$
- MC has reached stationary distribution if $p_t(x_i) = p_{t-1}(x_i), \forall i$
- MC will stabilize into a stationary distribution if
  - Irreducible, transition graph is connected
  - Aperiodic, does not get trapped in cycles

Markov Chains (Contd.)

- Sufficient condition to ensure $p(x)$ is the stationary distribution
$$p(x_{i'})T(x_i|x_{i'}) = p(x_i)T(x_{i'}|x_i)$$
- Detailed balance equation implies invariant (stationary) distribution
$$\sum_{x_{i'}} p(x_{i'})T(x_i|x_{i'}) = \sum_{x_{i'}} p(x_i)T(x_{i'}|x_i) = p(x_i)$$
- MCMC samplers, stationary distribution = target distribution
- Design $T(\cdot|\cdot)$ to get stationary distribution $p(x)$
- Sampling from $p(x)$ by running the MC to convergence

Markov Chains (Contd.)

- Random walker on the web
  - Irreducibility, should be able to reach all pages
  - Aperiodicity, do not get stuck in a loop
- PageRank used $T = L + E$
  - $L =$ link matrix for the web graph
  - $E =$ uniform random matrix, to ensure irreducibility, aperiodicity
- Invariant distribution $p(x)$ represents rank of webpage x
- Continuous spaces, T becomes an integral kernel K

$$\int_{x_i} p(x_i) K(x_{i+1}|x_i) dx_i = p(x_{i+1})$$

- Stationary $p(x)$ is the corresponding eigenfunction

The Metropolis-Hastings Algorithm

- Most popular MCMC method
- Based on a proposal distribution $q(x^*|x)$
- Algorithm: For $i = 0, \ldots, (n-1)$
  - Sample $u \sim \mathcal{U}(0,1)$
  - Sample $x^* \sim q(x^*|x_i)$
  - Then

$$x_{i+1} = \begin{cases} x^* & \text{if } \; u < A(x_i, x^*) = \min\left\{1, \frac{p(x^*)q(x_i|x^*)}{p(x_i)q(x^*|x_i)}\right\} \\ x_i & \text{otherwise} \end{cases}$$

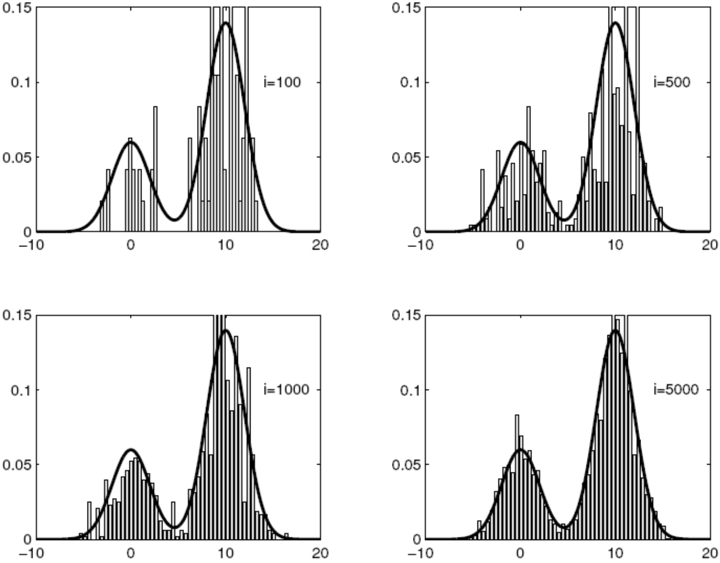
- The transition kernel is

$$K_{MH}(x_{i+1}|x_i) = q(x_{i+1}|x_i)A(x_i, x_{i+1}) + \delta_{x_i}(x_{i+1})r(x_i)$$

where $r(x_i)$ is the term associated with rejection

$$r(x_i) = \int_x q(x|x_i)(1 - A(x_i, x))dx$$

The Metropolis-Hastings Algorithm (Contd.)



The Metropolis-Hastings Algorithm (Contd.)

- By construction

$$p(x_i)K_{MH}(x_{i+1}|x_i) = p(x_{i+1})K_{MH}(x_i|x_{i+1})$$

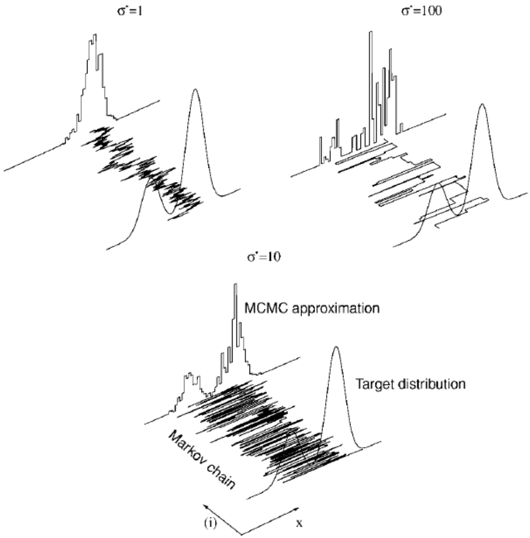
- Implies $p(x)$ is the invariant distribution
- Basic properties
  - Irreducibility, ensure support of q contains support of p
  - Aperiodicity, ensured since rejection is always a possibility
- Independent sampler: $q(x^*|x_i) = q(x^*)$ so that

$$A(x_i, x^*) = \min\left\{1, \frac{p(x^*)q(x_i)}{q(x^*)p(x_i)}\right\}$$

- Metropolis sampler: symmetric $q(x^*|x_i) = q(x_i|x^*)$

$$A(x_i, x^*) = \min\left\{1, \frac{p(x^*)}{p(x_i)}\right\}$$

The Metropolis-Hastings Algorithm (Contd.)



Mixtures of MCMC Kernels

- Powerful property of MCMC: Combination of Samplers
- Let K_1, K_2 be kernels with invariant distribution p
  - Mixture kernel $\alpha K_1 + (1-\alpha)K_2, \alpha \in [0,1]$ converges to p
  - Cycle kernel $K_1 K_2$ converges to p
- Mixtures can use global and local proposals
  - Global proposals explore the entire space (with probability α)
  - Local proposals discover finer details (with probability $(1-\alpha)$)
- Example: Target has many narrow peaks
  - Global proposal gets the peaks
  - Local proposals get the neighborhood of peaks (random walk)

Cycles of MCMC Kernels

- Split a multi-variate state into blocks
- Each block can be updated separately
- Convergence is faster if correlated variables are blocked
- Transition kernel is given by

$$K_{MHCycle}(x^{(i+1)}|x^{(i)}) = \prod_{j=1}^{n_b} K_{MH(j)}(x_{b_j}^{(i+1)}|x_{b_j}^{(i)}, x_{-[b_j]}^{(i+1)})$$

$$x_{-[b_j]}^{(i+1)} = \{x_{b_1}^{(i+1)}, \dots, x_{b_{j-1}}^{(i+1)}, x_{b_{j+1}}^{(i)}, \dots, x_{b_{n_b}}^{(i)}\}$$

- Trade-off on block size
  - If block size is small, chain takes long time to explore the space
  - If block size is large, acceptance probability is low
- Gibbs sampling effectively uses block size of 1

The Gibbs Sampler

- For a d-dimensional vector x, assume we know
$$p(x_j|x_{-j}) = p(x_j|x_1,\ldots,x_{j-1},x_{j+1},\cdots,x_d)$$
- Gibbs sampler uses the following proposal distribution
$$q(x^*|x^{(i)}) = \begin{cases} p(x_j^*|x_{-j}^{(i)}) & \text{if } \; x_{-j}^* = x_{-j}^{(i)} \\ 0 & \text{otherwise} \end{cases}$$
- The acceptance probability
$$A(x^{(i)},x^*) = \min\left\{1, \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})}\right\} = 1$$
- Deterministic scan: All samples are accepted

The Gibbs Sampler (Contd.)

- Initialize $x^{(0)}$. For $i = 0, \ldots, (N-1)$
  - Sample $x_1^{(i+1)} \sim p(x_1 | x_2^{(i)}, x_3^{(i)} \ldots, x_d^{(i)})$
  - Sample $x_2^{(i+1)} \sim p(x_1 | x_1^{(i+1)}, x_3^{(i)} \ldots, x_d^{(i)})$
  - $\cdots$
  - Sample $x_d^{(i+1)} \sim p(x_d | x_1^{(i+1)}, \ldots, x_{d-1}^{(i+1)})$
- Possible to have MH steps inside a Gibbs sampler
- For $d = 2$, Gibbs sampler is the data augmentation algorithm
- For Bayes nets, the conditioning is on the Markov blanket

$$p(x_j | x_{-j}) \propto p(x_j | x_{pa(j)}) \prod_{k \in ch(j)} p(x_k | pa(k))$$

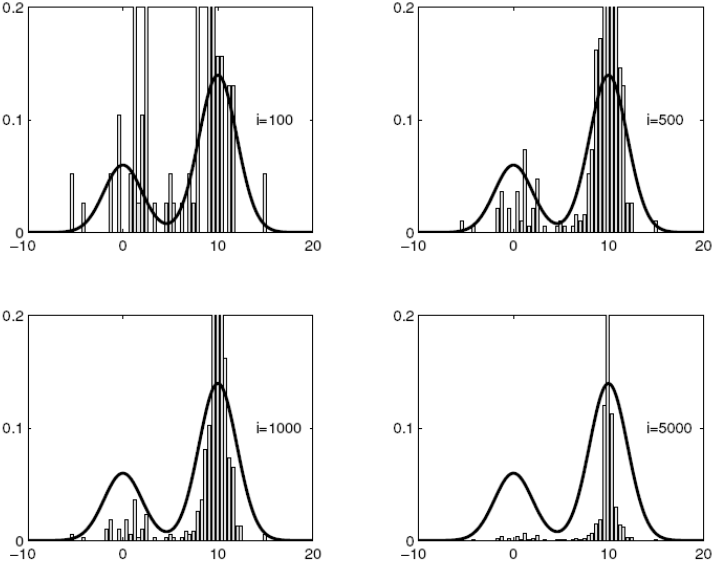
Simulated Annealing: Finding Modes

- Problem: To find global maximum of $p(x)$
- Initial idea: Run MCMC, estimate $\hat{p}(x)$, compute max
- Issue: MC may not come close to the mode(s)
- Simulate a *non-homogenous* Markov chain
- Invariant distribution at iteration i is $p_i(x) \propto p^{1/T_i}(x)$
- Sample update follows

$$x_{i+1} = \begin{cases} x^* & \text{if} \quad u < A(x_i, x^*) = \min\left\{1, \frac{p^{\frac{1}{T_i}}(x^*)q(x_i|x^*)}{p^{\frac{1}{T_i}}(x_i)q(x^*|x_i)}\right\} \\ x_i & \text{otherwise} \end{cases}$$

- T_i decreases following a cooling schedule, $\lim_{i\to\infty} T_i = 0$
- Cooling schedule needs proper choice, e.g., $T_i = \frac{1}{C\log(i+T_0)}$

Simulated Annealing (Contd.)



Latent Variable Models, Redux

- Joint distribution of a latent variable model (LVM)

$$p_\theta(x, z) = p_\theta(z) p_\theta(x|z) \ ,$$

  - x denotes the observed variable
  - z denotes the latent variable
  - θ denotes the parameters
- Problems of interest
  - Compute marginal or conditional distributions

$$p_\theta(x) = \int_z p_\theta(x, z) dz \qquad\qquad p_\theta(z|x) = \frac{p_\theta(x, z)}{p_\theta(x)}$$

  - Estimate θ by optimizing a function of $p_\theta(x)$
- Problems need to compute high-d integrals

Variational Inference (VI): Warm Up

- Construct a distribution $q_\phi(\mathrm{z}|\mathrm{x})$ with parameters ϕ
- Choose family q and parameters ϕ to approximate true posterior
$$q_\phi(\mathrm{z}|\mathrm{x}) \approx p_\theta(\mathrm{z}|\mathrm{x})$$
- Ideally: Choose q to minimize some divergence $D(q_\phi(\mathrm{z}|\mathrm{x}), p_\theta(\mathrm{z}|\mathrm{x}))$
  - Challenge: Do not know $p_\theta(\mathrm{z}|\mathrm{x})$ explicitly
- Inference model $q_\phi(z|x)$
  - Also called recognition model, or *encoder*
  - ϕ are called the *variational parameters*
- Generative model $p_\theta(\mathrm{x}|\mathrm{z})$, also called *decoder*

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\log p_\theta(\mathrm{x}) &= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}[\log p_\theta(\mathrm{x})] \\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}\left[\log\left(\frac{p_\theta(\mathrm{x},\mathrm{z})}{p_\theta(\mathrm{z}|\mathrm{x})}\right)\right] \\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}\left[\log\left(\frac{p_\theta(\mathrm{x},\mathrm{z})}{q_\phi(\mathrm{z}|\mathrm{x})}\frac{q_\phi(\mathrm{z}|\mathrm{x})}{p_\theta(\mathrm{z}|\mathrm{x})}\right)\right] \\
&= \underbrace{\mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}\left[\log\left(\frac{p_\theta(\mathrm{x},\mathrm{z})}{q_\phi(\mathrm{z}|\mathrm{x})}\right)\right]}_{\mathcal{L}_{\theta,\phi}(\mathrm{x})\ \text{(ELBO)}} + \underbrace{\mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}\left[\log\left(\frac{q_\phi(\mathrm{z}|\mathrm{x})}{p_\theta(\mathrm{z}|\mathrm{x})}\right)\right]}_{D_{KL}(q_\phi(\mathrm{z}|\mathrm{x})\|p_\theta(\mathrm{z}|\mathrm{x}))}
\end{aligned}
$$

Maximize the ELBO, lower bound to $\log p_\theta(\mathrm{x})$

$$
\mathcal{L}_{\theta,\phi}(\mathrm{x}) = \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}[\log p_\theta(\mathrm{x},\mathrm{z}) - \log q_\phi(\mathrm{z}|\mathrm{x})]
$$

Mean Field VI

- Inference is done based on a dataset $\{x_i, i = 1, \ldots, n\}$
- Mean field VI assumes a tractable inference model

$$q_\phi(\mathrm{z}|\mathrm{x}) = \prod_{i=1}^{n} q_{\phi_i}(z_i|x_i)$$

  - Naive mean field, fully factorized distribution over $\{z_{ij}\}$
- Each component typically belongs to some exponential family
- Optimize over the *free* variational parameters $\{\phi_i, i = 1, \ldots, n\}$
  - Need to optimize each ϕ_i, can be slow for large datasets
- The fully-factorized assumption may be inaccurate

Stochastic and Amortized VI

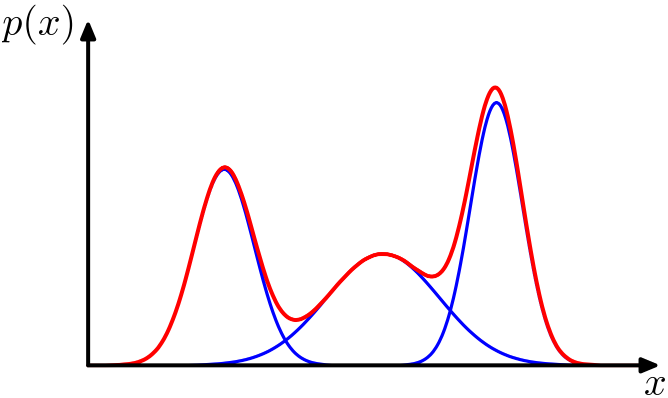
- Stochastic VI based on stochastic optimization
  - Update variational parameter by optimizing expectation
  - Use stochastic mini-batch instead of full-batch gradient descent
  - Work with noisy unbiased gradients
  - More discussions on gradient computation soon
- Amortized VI
  - Challenge: optimize ϕ_i for each $i = 1, \ldots, n$
  - Instead learn a mapping $\phi_i = f_\gamma(\mathrm{x}_i)$
  - More generally, posterior approximations with inference networks

$$q_{\phi_i}(\mathrm{z}_i|\mathrm{x}_i) = q_{f_\gamma(\mathrm{x}_i)}(\mathrm{z}_i|\mathrm{x}_i)$$

Overview: Probabilistic Models

- Probability Overview
- Bayesian Networks, Graphical Models
- Approximate Inference:
  - Markov Chain Monte Carlo (MCMC)
  - Variational Inference (VI)
- Expectation Maximization
- Dynamical Models
  - Filtering, Prediction, Smoothing
  - Examples: HMMs, KFs, DBNs
- Losses and Representation
  - Losses from generalized linear models
  - Beyond linear representations
- Scoring rules, Calibration

Simple LVMs: Finite Mixture Models



Mixture of Gaussians

- The probability density function is given by

$$p(\mathrm{x}|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathrm{x}|\mu_k, \Sigma_k)$$

- Set of parameters $\Theta = \{\{\pi_k\}, \{\mu_k\}, \{\Sigma_k\}\}$
- π is a discrete distribution: relative proportions of each component

$$0 \leq \pi_k \leq 1 \qquad\qquad \sum_{k=1}^{K} \pi_k = 1$$

- Each component is a multi-variate Gaussian

$$\mathcal{N}(\mathrm{x}|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|} \exp\left(-(\mathrm{x}-\mu_k)^T \Sigma_k^{-1}(\mathrm{x}-\mu_k)\right)$$

- To generate a sample x from the mixture model
  - Sample mixture component $z \sim \pi$
  - Sample $\mathrm{x} \in \mathbb{R}^d$ from the z^{th} component $\mathrm{x} \sim \mathcal{N}(\mu_z, \Sigma_z)$
- An alternative viewpoint: z is a 1-of-K binary vector

$$p(\mathrm{x}) = \sum_{\mathrm{z}} p(\mathrm{z}) p(\mathrm{x}|\mathrm{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathrm{x}|\mu_k, \Sigma_k)$$

- The posterior distribution

$$p(z_k|\mathrm{x}) = \frac{p(z_k) p(\mathrm{x}|z_k)}{p(\mathrm{x})} = \frac{\pi_k \mathcal{N}(\mathrm{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathrm{x}|\mu_j, \Sigma_j)}$$

Maximum Likelihood Estimation

- Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be drawn i.i.d. from MoG
- The log-likelihood of the observations

$$\log p(\mathcal{X}|\pi, \mu, \Sigma) = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}$$

- Optimizing directly w.r.t. (π, μ, Σ) is difficult
  - log works on sum, not on individual Gaussians
  - Closed form solution cannot be obtained
- Expectation Maximization (EM)
  - Powerful family of iterative update algorithm
  - Applicable for learning mixture models
  - Has applications beyond mixture models

EM for Gaussian Mixtures

- At the optimum, gradient w.r.t. (π, μ, Σ) should vanish
- Taking derivative w.r.t. μ_k and setting it to 0

$$
\begin{aligned}
0 &= -\sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathrm{x}_n|\mu_k, \Sigma_k)}{\sum_j \sum_{k=1}^{K} \pi_j \mathcal{N}(\mathrm{x}_n|\mu_j, \Sigma_j)} \Sigma_k(\mathrm{x}_n - \mu_k) \\
&= -\sum_{n=1}^{N} p(z_k|\mathrm{x}_n)\Sigma_k(\mathrm{x}_n - \mu_k)
\end{aligned}
$$

- A direct simplification gives (let $N_k = \sum_{n=1}^{N} p(z_k|\mathrm{x}_n)$)

$$
\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|\mathrm{x}_n)\mathrm{x}_n
$$

EM for Gaussian Mixtures (Contd.)

- Taking derivative w.r.t. Σ_k

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|\mathrm{x}_n)(x_n - \mu_k)(\mathrm{x}_n - \mu_k)^T$$

- Constrained optimization for π_k with Lagrangian

$$\log p(\mathcal{X}|\pi, \mu, \Sigma) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

- A direct calculation gives

$$\pi_k = \frac{N_k}{N}$$

EM for Gaussian Mixtures: Algorithm

- Initialize π, μ, Σ
- Till Convergence

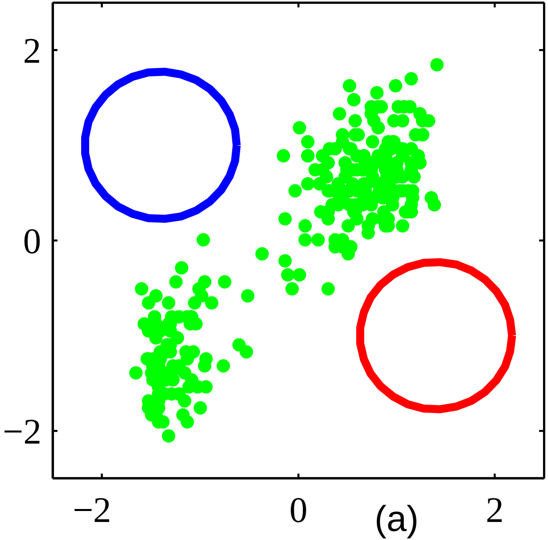
E-step Evaluate the posterior probabilities

$$p(z_k|\mathrm{x}_n) = \frac{\pi_k \mathcal{N}(\mathrm{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathrm{x}|\mu_j, \Sigma_j)}$$

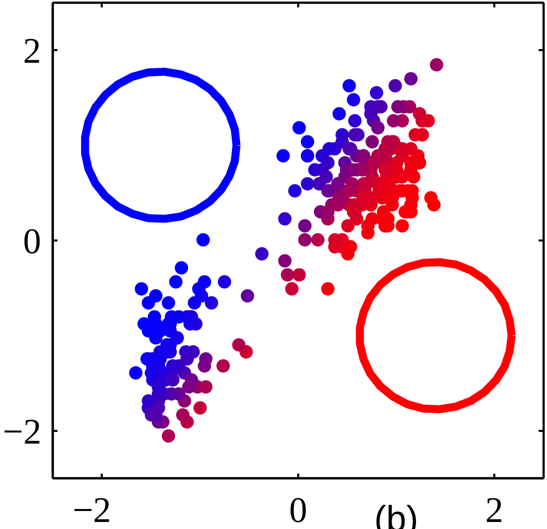
M-step Update the parameter values

$$\begin{aligned}
\mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|\mathrm{x}_n)\mathrm{x}_n \\
\Sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|\mathrm{x}_n)(\mathrm{x}_n - \mu_k)(\mathrm{x}_n - \mu_k)^T \\
\pi_k &= \frac{N_k}{N}
\end{aligned}$$

EM on Gaussian Mixtures Example

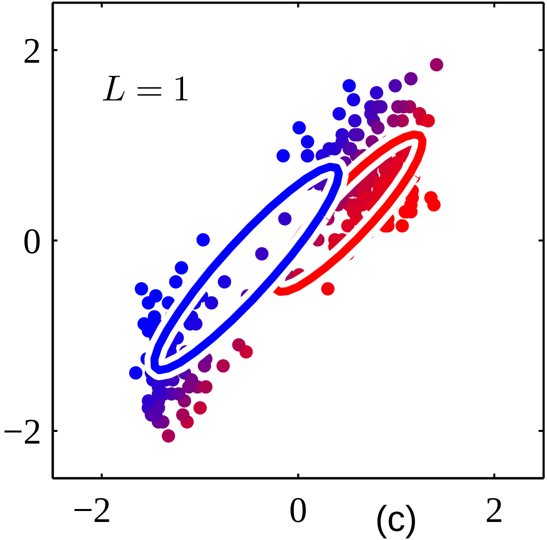


EM on Gaussian Mixtures Example

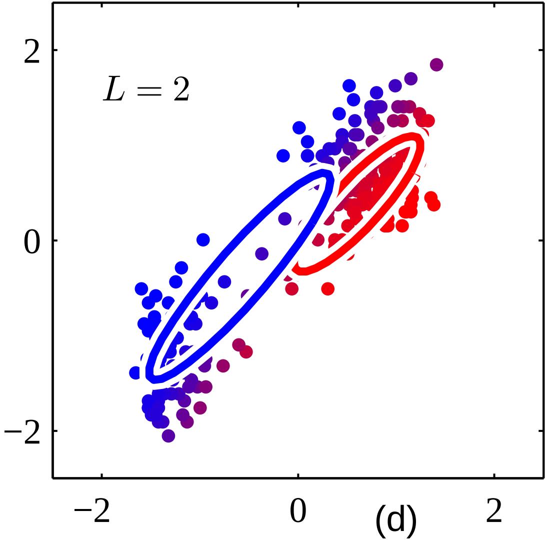


(b)

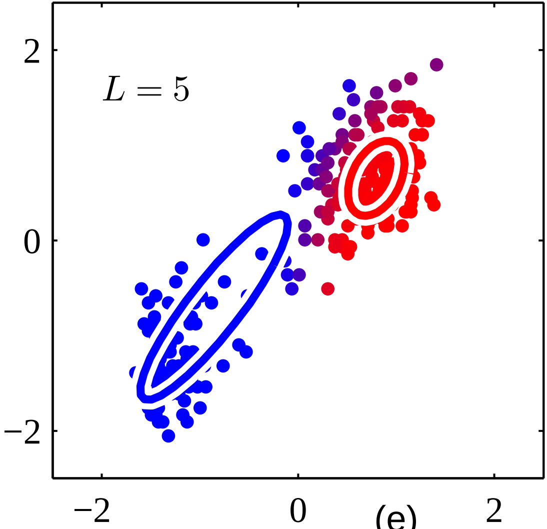
EM on Gaussian Mixtures Example



EM on Gaussian Mixtures Example

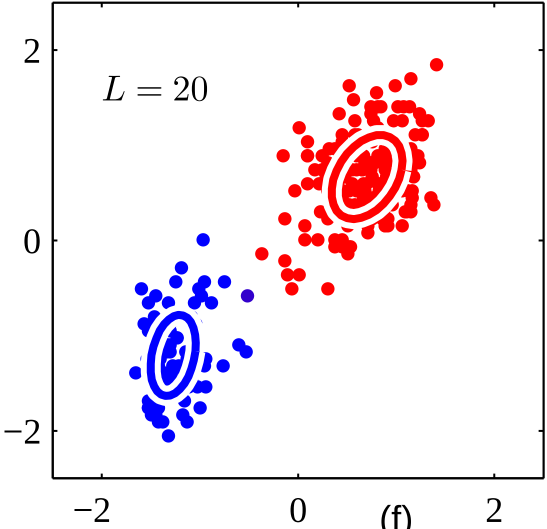


EM on Gaussian Mixtures Example



(e)

EM on Gaussian Mixtures Example



(f)

An Alternative View of EM

- Maximum likelihood in presence of latent variable

$$\log p(X|\theta) = \log \left\{ \sum_z p(X, Z|\theta) \right\}$$

- The marginal cannot be obtained in closed form
- $\{X, Z\}$ is the complete data, $\{X\}$ is the incomplete data
- Main Idea
    - We dont know Z, hence dont know $p(X, Z|\theta)$
    - We know $p(Z|X, \theta)$
    - Use expected value of $p(X, Z|\theta)$, expectation over $p(Z|X, \theta)$
- Expected value of the complete likelihood

$$Q(\theta, \theta^{old}) = \sum_z p(Z|X, \theta^{old}) \log p(X, Z|\theta)$$

- Compute θ^{new} by maximizing $Q(\theta, \theta^{old})$

The General EM Algorithm

- Choose initial value of parameters θ^{old}
- Till convergence

E-step Evaluate $p(Z|X, \theta^{old})$ [Recall: Inference network $q_\phi(\mathrm{z}|\mathrm{x})$]

M-step Evaluate θ^{new} given by

$$\theta^{new} = \mathrm{argmax}_\theta \sum_z p(Z|X, \theta^{old}) \log p(X, Z|\theta)$$

- Update $\theta^{old} \leftarrow \theta^{new}$

Gaussian Mixtures Revisited

- E-step evaluates the probabilities

$$p(z_k|\mathrm{x}_n) = \frac{\pi_k \mathcal{N}(\mathrm{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathrm{x}|\mu_j, \Sigma_j)}$$

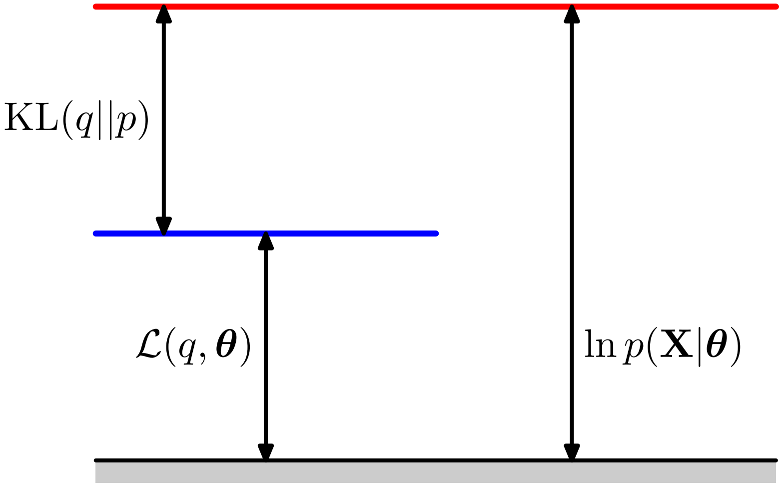
- M-step computes the new parameters

$$\begin{aligned}
\mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|\mathrm{x}_n)\mathrm{x}_n \\
\Sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|\mathrm{x}_n)(\mathrm{x}_n - \mu_k)(\mathrm{x}_n - \mu_k)^T \\
\pi_k &= \frac{N_k}{N}
\end{aligned}$$

Analysis of the EM Algorithm

- For any distribution $q(Z)$
$$\log p(X|\theta) = \mathcal{L}(q,\theta) + KL(q||p)$$
where
$$\begin{aligned} \mathcal{L}(q,\theta) &= \sum_z q(Z) \log \left\{ \frac{p(X,Z|\theta)}{q(Z)} \right\} \\ KL(q||p) &= \sum_z q(Z) \log \left\{ \frac{q(Z)}{p(Z|X,\theta)} \right\} \end{aligned}$$
- Since $KL(q||p) \geq 0$, we have a lower bound
$$\log p(X|\theta) \geq \mathcal{L}(q,\theta) = E_q[\log p(X,Z|\theta)] + H(q)$$
- Main Idea: Lower bound maximization

Analysis of EM

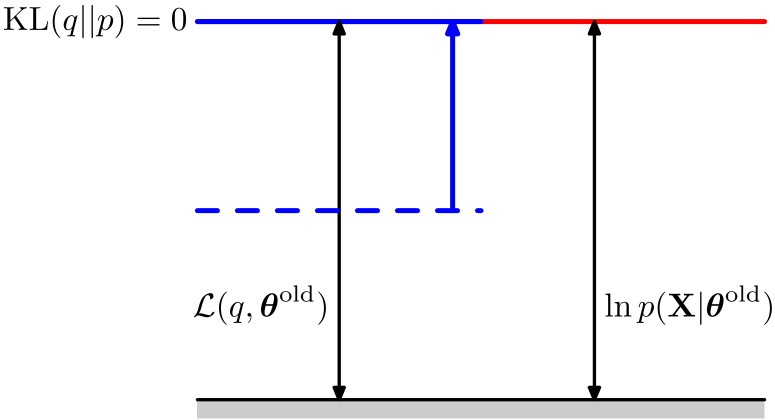


Analysis of the EM Algorithm (Contd.)

- $\log p(X|\theta) = \mathcal{L}(q, \theta) + KL(q||p)$
- The current parameter estimate θ^{old}
- The E-step:
  - Maximize $\mathcal{L}(q, \theta)$ w.r.t. q
  - The solution is $q(Z) = p(Z|X, \theta)$
  - We have $KL(q||p) = 0$, so that $\log p(X|\theta^{old}) = \mathcal{L}(q, \theta^{old})$
- The M-step:
  - Maximize $\mathcal{L}(q, \theta)$ w.r.t. θ
  - The new solution θ^{new}
  - The current q is not the optimal distribution, so $KL(q||p) \geq 0$
  - However,

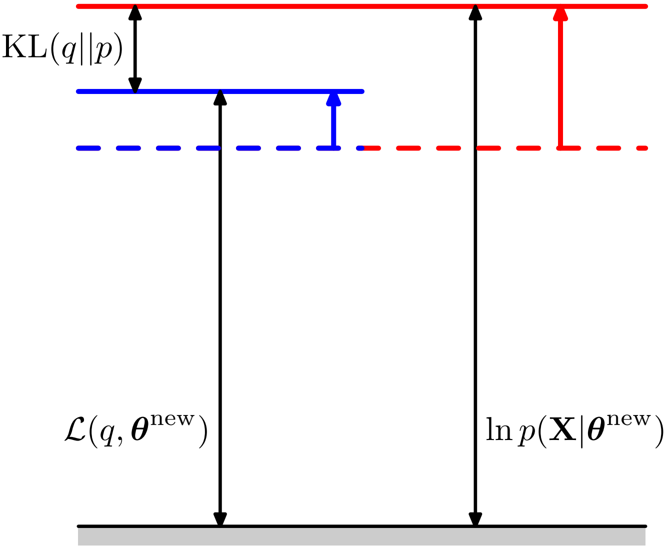
$$\log p(X|\theta^{new}) \geq \mathcal{L}(q, \theta^{new}) \geq \mathcal{L}(q, \theta^{old}) = \log p(X|\theta^{old})$$

The E-step



$\mathrm{KL}(q||p) = 0$
$\mathcal{L}(q, \boldsymbol{\theta}^{\mathrm{old}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\mathrm{old}})$

The M-step


$\mathrm{KL}(q||p)$
$\mathcal{L}(q, \boldsymbol{\theta}^{\mathrm{new}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\mathrm{new}})$

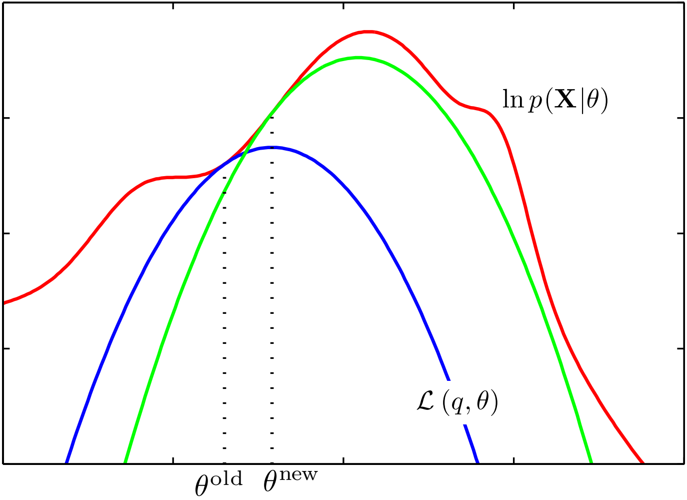
Variational EM

- $\log p(X|\theta) = E_q[\log p(X,Z|\theta)] + H(q) + KL(q||p)$
- The E-step:
  - Maximize $\mathcal{L}(q,\theta)$ w.r.t. q
  - The solution is $q(Z) = p(Z|X,\theta)$
  - For some models, $p(Z|X,\theta)$ cannot be obtained in closed form
  - Example: Latent Dirichlet Allocation, Bayesian Models, etc.
- Variational E-step:
  - Pick a parameterized family $q_\phi(Z)$
  - Choose variational parameter ϕ to minimize $KL(q_\phi||p)$
  - Same as maximizing lower bound to true the likelihood

$$\log p(X|\theta) \geq E_{q_\phi}[\log p(X,Z|\theta)] + H(q_\phi)$$

  - $KL(q_\phi||p)$ does not becomes zero, but progress is made
- M-step optimizes lower bound over θ
- Variational EM: Getting widely used for statistical models

Auxiliary Function Viewpoint of EM



Overview: Probabilistic Models

- Probability Overview
- Bayesian Networks, Graphical Models
- Approximate Inference:
  - Markov Chain Monte Carlo (MCMC)
  - Variational Inference (VI)
- Expectation Maximization
- Dynamical Models
  - Filtering, Prediction, Smoothing
  - Examples: HMMs, KFs, DBNs
- Losses and Representation
  - Losses from generalized linear models
  - Beyond linear representations
- Scoring rules, Calibration

Dynamical Models: Outline

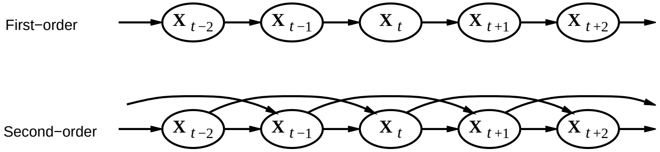
- Time and uncertainty
- Inference: filtering, prediction, smoothing
- Examples: Hidden Markov Models (HMMs), Kalman Filters (KFs), Dynamic Bayesian Networks (DBNs)

Time and uncertainty

- The world changes
  - Rational agent needs to track and predict
  - Example: Car diagnosis Vs Diabetes
- Consider state and evidence variables over time
- X_t = set of unobservable state variables at time t
  - Example: $BloodSugar_t$, $StomachContents_t$, etc.
- E_t = set of observable evidence variables at time t
  - Example: $MeasuredBloodSugar_t$, $FoodEaten_t$, etc.
- Time can be discrete or continuous
- Notation: $X_{a:b} = X_a, X_{a+1}, \ldots, X_{b-1}, X_b$

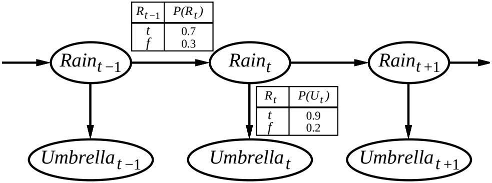
Markov Processes (Markov Chains)

- Construct a Bayes net from these variables: Parents?
- Markov Assumption X_t depends on bounded subset of $X_{0:t-1}$
  - First-order: $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$
  - Second-order: $P(X_t|X_{0:t-1}) = P(X_t|X_{t-2}, X_{t-1})$



- Sensor Markov assumption: $P(E_t|X_{0:t}, E_{0:t-1}) = P(E_t|X_t)$
- Stationary process:
  - Transition model $P(X_t|X_{t-1})$ fixed for all t
  - Sensor model $P(E_t|X_t)$ fixed for all t

Example



- First-order Markov assumption often not true in real world
- Possible fixes:
  - Increase order of Markov process
  - Augment state, e.g., add $Temp_t$, $Pressure_t$
- Example: Robot Motion
  - Augment position and velocity with $Battery_t$

Inference Tasks

- *Filtering*: $P(X_t | \mathrm{e}_{1:t})$
  - Belief state is input to the decision process
- *Prediction*: $P(X_{t+k} | \mathrm{e}_{1:t})$ for $k > 0$
  - Evaluation of possible state sequences
  - Like filtering without the evidence
- *Smoothing*: $P(X_k | \mathrm{e}_{1:t})$ for $0 \leq k < t$
  - Better estimate of past states
  - Essential for learning
- *Most likely explanation*: $\arg\max_{\mathrm{x}_{1:t}} P(\mathrm{x}_{1:t} | \mathrm{e}_{1:t})$
  - Example: Speech recognition, Decoding from noisy channel

Filtering

- Aim: A *recursive* state estimation algorithm

$$P(X_{t+1}|\mathrm{e}_{1:t+1}) = f(\mathrm{e}_{t+1}, P(X_t|\mathrm{e}_{1:t}))$$

- From Bayes rule

$$\begin{aligned} P(X_{t+1}|\mathrm{e}_{1:t+1}) &= P(X_{t+1}|\mathrm{e}_{1:t}, \mathrm{e}_{t+1}) \\ &= \alpha P(\mathrm{e}_{t+1}|X_{t+1}, \mathrm{e}_{1:t}) P(X_{t+1}|\mathrm{e}_{1:t}) \\ &= \alpha P(\mathrm{e}_{t+1}|X_{t+1}) P(X_{t+1}|\mathrm{e}_{1:t}) \end{aligned}$$

Filtering (Contd.)

- We have
$$P(X_{t+1}|\mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1}|X_{t+1}) P(X_{t+1}|\mathbf{e}_{1:t})$$
- First term $P(e_{t+1}|X_{t+1})$ is evidence conditional probability (known)
- Expanding the second term
$$\begin{aligned} P(X_{t+1}|\mathbf{e}_{1:t+1}) &= \alpha P(\mathbf{e}_{t+1}|X_{t+1}) \sum_{\mathbf{x}_t} P(X_{t+1}|\mathbf{x}_t, \mathbf{e}_{1:t}) P(\mathbf{x}_t|\mathbf{e}_{1:t}) \\ &= \alpha P(\mathbf{e}_{t+1}|X_{t+1}) \sum_{\mathbf{x}_t} P(X_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t}) \end{aligned}$$
- Recursive filtering
  - $p(\mathbf{x}_t|\mathbf{e}_{1:t})$ is the previous filtering term (recursion, known)
  - $p(X_{t+1}|\mathbf{x}_t)$ is state transition probability (known)
  - Need to do marginalization $\sum_{\mathbf{x}_t} \cdots$ (high-d integration)

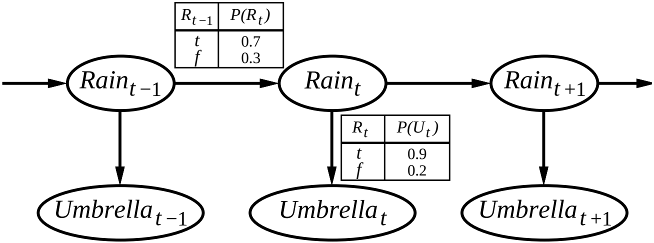
Prediction

- Prediction is similar to filtering
  - Without new evidence
- Filtering does one step prediction
- For prediction

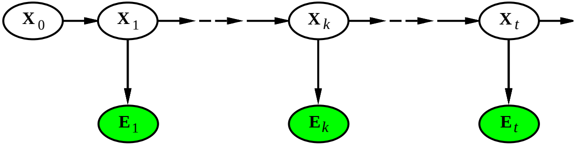
$$P(X_{t+k+1}|\mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} P(X_{t+k+1}|X_{t+k})P(X_{t+k}|\mathbf{e}_{1:t})$$

- How far in the future can we predict?
  - After evidence stops, prediction is running a Markov Chain
  - $\lim_{k\to\infty} P(X_{t+k}|\mathbf{e}_{1:t})$ converges to the *stationary distribution*
  - Prediction gets harder, uncertainty increases
  - Example: Weather forecasting for 2 days, 1 week, 4 weeks

Umbrella Example



Smoothing



- Divide evidence $e_{1:t}$ into $e_{1:k}$, $e_{k+1:t}$

$$\begin{aligned} P(X_k|e_{1:t}) &= P(X_k|e_{1:k}, e_{k+1:t}) \\ &= \alpha P(X_k|e_{1:k}) P(e_{k+1:t}|X_k, e_{1:k}) \\ &= \alpha P(X_k|e_{1:k}) P(e_{k+1:t}|X_k) \\ &= \alpha f_{1:k} b_{k+1:t} \end{aligned}$$

- Forward message $f_{1:k}$ is filtering

Smoothing (Contd.)

- Backward message computed by a backwards recursion:

$$\begin{aligned}
P(\mathbf{e}_{k+1:t}|X_k) &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t}|X_k, \mathbf{x}_{k+1}) P(\mathbf{x}_{k+1}|X_k) \\
&= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1}) P(\mathbf{x}_{k+1}|X_k) \\
&= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1}) P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1}) P(\mathbf{x}_{k+1}|X_k)
\end{aligned}$$

- $\mathbf{b}_{k+1:t} = P(\mathbf{e}_{k+1:t}|X_k) = \alpha \mathit{Backward}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$
- The smoothed probability

$$P(X_k|\mathbf{e}_{1:t}) = \alpha \mathbf{f}_{1:k} \mathbf{b}_{k+1:t}$$

Most Likely Explanation

- Most likely sequence $\neq$ sequence of most likely states
- Most likely path to each X_{t+1}

$$\max_{\mathbf{x}_1 \ldots \mathbf{x}_t} P(\mathbf{X}_1, \ldots, \mathbf{X}_t, \mathbf{X}_{t+1} | \mathbf{e}_{1:t+1})$$

$$= P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(P(\mathbf{X}_{t+1} | \mathbf{X}_t) \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} P(\mathbf{X}_1, \ldots, \mathbf{X}_{t-1}, \mathbf{X}_t | \mathbf{e}_{1:t}) \right)$$

- Identical to filtering, except $\mathbf{f}_{1:t}$ replaced by

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} P(\mathbf{X}_1, \ldots, \mathbf{X}_{t-1}, \mathbf{X}_t | \mathbf{e}_{1:t}),$$

- $\mathbf{m}_{1:t}(i)$ gives the probability of the most likely path to state i.
- Update has sum replaced by max, giving the *Viterbi algorithm*:

$$\mathbf{m}_{1:t+1} = P(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(P(\mathbf{X}_{t+1} | \mathbf{X}_t) \mathbf{m}_{1:t} \right)$$

Overview: Probabilistic Models

- Probability Overview
- Bayesian Networks, Graphical Models
- Approximate Inference:
  - Markov Chain Monte Carlo (MCMC)
  - Variational Inference (VI)
- Expectation Maximization
- Dynamical Models
  - Filtering, Prediction, Smoothing
  - Examples: HMMs, KFs, DBNs
- Losses and Representation
  - Losses from generalized linear models
  - Beyond linear representations
- Scoring rules, Calibration

Losses and Representations: Warm Up

- Typically work with a set of samples $\{(\mathrm{x}_i, y_i), i = 1, \ldots, n\}$
  - Samples assumed to be i.i.d.
- Many problems we will consider

$$\min_{\theta} \sum_{i=1}^{n} L(y_i, f_\theta(\mathrm{x}_i))$$

- L is the loss, e.g., square loss, log loss, hinge loss, etc.
  - Losses as surrogates to target risk, e.g., hinge loss, log loss
  - Losses from statistical assumptions, e.g., square loss, log loss
- $f_\theta(\cdot)$ is the predictor, with suitable representation
  - Classical (linear) approach: $f_\theta(\mathrm{x}) = \theta^T \mathrm{x}$
  - Modern approach: deep representations

Least Squares Regression

- Objective function

$$\min_{\theta} \sum_{i=1}^{n} (y_i - f_\theta(\mathrm{x}_i))^2$$

- Statistical modeling assumptions: $P(Y|\mathrm{x})$
  - Conditional expectation is (a function of) the predictor

$$\mathbb{E}[Y|\mathrm{x}] = f_\theta(\mathrm{x})$$

  - Responses drawn from this conditional Gaussian, with fixed variance

$$y_i \sim \mathcal{N}(\mathbb{E}[Y|\mathrm{x}_i], \sigma^2) = \mathcal{N}(f_\theta(\mathrm{x}_i), \sigma^2)$$

- Maximum likelihood estimation $\equiv$ least squares objective

Logistic Regression

- For 2-class classification with $y_i \in \{0, 1\}$, objective function

$$\min_{\theta} \sum_{i=1}^{n} \left\{ y_i f_\theta(\mathrm{x}_i) - \log(1 + \exp(f_\theta(\mathrm{x}_i))) \right\}$$

- Statistical modeling assumptions: $P(Y \mid \mathrm{x})$
  - Conditional expectation is a function of the predictor

$$\log \frac{P(1|\mathrm{x})}{P(0|\mathrm{x})} = f_\theta(\mathrm{x}) \;\Rightarrow\; P(1|\mathrm{x}) = \mathbb{E}[Y|\mathrm{x}] = \sigma(f_\theta(\mathrm{x}))\,,\;\; \sigma(a) = \frac{1}{1 + \exp(-a)}$$

  - Response drawn from this conditional Bernoulli

$$y_i \sim \mathrm{Bern}(\mathbb{E}[Y|\mathrm{x}_i]) = \mathrm{Bern}(\sigma(f_\theta(\mathrm{x}_i)))$$

- Maximum likelihood estimation $\equiv$ log-loss (cross-entropy) objective

Exponential Family, Link Function

- Exponential family distributions
$$p_\eta(y) = \exp(\langle y, \eta \rangle - \psi(\eta)) p(y)$$
- Examples: Gaussian, Bernoulli, gamma, categorical, Dirichlet, Poisson, ...
- ψ is the log-partition function, convex, differentiable
- Expectation: $\mathbb{E}[Y] = \nabla \psi(\eta)$, the link function $\lambda(\cdot)$
- Example: for Bernoulli, $\psi(\eta) = \log(1 + \exp(\eta))$, so
$$\mathbb{E}[Y] = \nabla \psi(\eta) = \frac{\exp(\eta)}{1 + \exp(\eta)} = \frac{1}{1 + \exp(-\eta)} = \sigma(\eta)$$
- For logistic regression, model $Y|\mathrm{x}$ with $\eta = f_\theta(\mathrm{x})$, so
$$\mathbb{E}[Y|\mathrm{x}] = \sigma(f_\theta(\mathrm{x}))$$

Generalized (Linear) Models

- Conditional distribution of response y given covariates x

$$p_\eta(y|\mathrm{x}) = \exp(\langle y, \eta(\mathrm{x})\rangle - \psi(\eta(\mathrm{x})))p(y|\mathrm{x})$$

- Examples: least squares regression (continous), logistic regression (categorical, classification), Poisson regression (count), ...
- Representation: $\eta(\mathrm{x}) = f_\theta(\mathrm{x})$
  - Classical GLMs: $\eta(\mathrm{x}) = \theta^T \mathrm{x}$
- Statistical modeling assumptions: $\mathbb{P}(Y \mid \mathrm{x})$
  - Conditional expectation is the link function λ of the predictor

$$\mathbb{E}[Y|\mathrm{x}] = \nabla\psi(\eta(\mathrm{x})) = \lambda(f_\theta(\mathrm{x}))$$

  - Response drawn from this conditional exponential family

Overview: Probabilistic Models

- Probability Overview
- Bayesian Networks, Graphical Models
- Approximate Inference:
  - Markov Chain Monte Carlo (MCMC)
  - Variational Inference (VI)
- Expectation Maximization
- Dynamical Models
  - Filtering, Prediction, Smoothing
  - Examples: HMMs, KFs, DBNs
- Losses and Representation
  - Losses from generalized linear models
  - Beyond linear representations
- Scoring rules, Calibration

Scoring Rules

- Scoring rules measure accuracy of probabilistic forecasts
  - Example: Weather forecast, 25% chance of rain
- Probabilistic forecast P, true outcome x, scoring rule $S(P, x)$
  - Higher $S(P, x)$ means more accurate
- True outcome $X \sim Q$, expected score $S(P, Q) = \mathbb{E}_{X \sim Q}[S(P, X)]$
- Scoring rule is *proper* if $S(Q, Q) \geq S(P, Q)$, for all P, Q
  - Forecaster should try to use $P = Q$ for the forecasts
- Expected loss (or divergence): $d(P, Q) = S(Q, Q) - S(P, Q)$
  - For proper scoring rules, $d(P, Q) \geq 0$
  - "Better" forecasts P have smaller loss (or divergence)

Fitting Models using Scoring Rules

- Fitting parametric model P_θ given samples $X_1, \ldots, X_n$
- Measure goodness-of-fit by mean score

$$\mathcal{S}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} S(P_\theta, X_i)$$

- Choose a suitable (strictly) proper scoring rule, and estimate

$$\hat{\theta}_n = \underset{\theta}{\text{argmax}} \; \frac{1}{n} \sum_{i=1}^{n} S(P_\theta, X_i)$$

- Compare with maximum likelihood estimation:

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(X_i)$$

- Question: Is $S(P_\theta, X_i) = \log p_\theta(X_i)$ a proper scoring rule?

Scoring Rule: Examples (1 of 3)

- Quadratic or Brier score: Discrete distribution with m possible

$$S(\mathrm{p}, i) = -\sum_{j=1}^{m} (p_j - \delta_{ij})^2 = 2p_i - \sum_{j=1}^{m} p_j^2 - 1$$

$$d(\mathrm{p}, \mathrm{q}) = \sum_{j=1}^{m} (p_j - q_j)^2 = \|\mathrm{p} - \mathrm{q}\|_2^2$$

- Spherical score: For any $\alpha > 1$ (special case $\alpha = 2$)

$$S(\mathrm{p}, i) = \frac{p_i^{\alpha-1}}{\left(\sum_{j=1}^{m} p_j^{\alpha}\right)^{(\alpha-1)/\alpha}} \qquad \left(\frac{p_i}{\|\mathrm{p}\|_2}\right)$$

$$d(\mathrm{p}, \mathrm{q}) = \left(\sum_{j=1}^{m} q_j^{\alpha}\right)^{1/\alpha} - \frac{\sum_{i=1}^{m} p_j q_j^{\alpha-1}}{\left(\sum_{j=1}^{m} q_j^{\alpha}\right)^{\alpha-1/\alpha}} \qquad \left(\|\mathrm{q}\|_2 - \frac{\langle \mathrm{p}, \mathrm{q}\rangle}{\|\mathrm{q}\|_2}\right)$$

Scoring Rule: Examples (2 of 3)

- Logarithmic score:

$$S(\mathrm{p}, i) = \log p_i$$

$$d(\mathrm{p}, \mathrm{q}) = \sum_{j=1}^{m} q_j \log \frac{q_j}{p_j} = KL(\mathrm{q}, \mathrm{p})$$

- Continuous ranked probability score (CRPS): Forecast distribution F, $Z, Z' \sim F$

$$CRPS(F, x) = -\int_{-\infty}^{\infty} (F(z) - \mathbb{1}[z \geq x])^2 dz = \frac{1}{2}\mathbb{E}_F|Z - Z'| - \mathbb{E}_F|Z - x|$$

$$d(F, G) = \int_{-\infty}^{\infty} (F(z) - G(z))^2 dz$$

Scoring Rule: Examples (3 of 3)

- Hyvarinen score: Based on gradient of log-likelihood w.r.t. location ξ, rather than model parameter θ:

$$\psi(\xi;\theta) = \nabla_\xi \log p_\theta(\xi) = \begin{bmatrix} \frac{\partial \log p(\xi;\theta)}{\partial \xi_1} \\ \vdots \\ \frac{\partial \log p(\xi;\theta)}{\partial \xi_p} \end{bmatrix}$$

- For data distribution P_x, score $\psi_x(\xi) = \nabla_\xi \log p_x(\xi)$
- The loss or divergence:

$$\begin{aligned} d(P_\theta, P_x) &= \frac{1}{2}\mathbb{E}_{P_x}\left[\|\psi(\xi,\theta) - \psi_x(\xi)\|_2^2\right] \\ &= \mathbb{E}_{P_x}\left[\sum_{i=1}^{p}\left\{\frac{\partial^2 \log p(\xi;\theta)}{\partial \xi_i^2} + \frac{1}{2}\left(\frac{\partial \log p(\xi;\theta)}{\partial \xi_i}\right)^2\right\}\right] \end{aligned}$$

Calibrated Forecasts

- Assessing quality of probabilistic forecasts
  - Example: 25% chance of rain
- Sequential probabilistic forecasts
  - Forecaster observes a sequence of events $y_t \in K$, e.g., $K = \{1, 2, \ldots, m\}$
  - They predict $\mathrm{p}_{t+1} \in \Delta(K)$ (simplex), may depend on $y_{1:t}$
- Calibration: probability predictions match the outcome frequency
  - Consider all (past) days with "25% chance of rain" forecast
  - Estimate the fraction of these days it rained
  - Fraction should be ≈ 0.25
- Should be true for all predicted probabilities