Recap: How to Set the Potentials?

- Unary potentials
  - E.g. color model, modeled with a Mixture of Gaussians
    \[ \phi(x_i, y_k; \theta) = \log \sum_k \theta_k p(k|x_i) N(y_k; \bar{y}_k, \Sigma_k) \]
  - Learn color distributions for each label

- Pairwise potentials
  - Potts Model
    \[ \psi(x_i, x_j; \theta) = \theta \delta(x_i \neq x_j) \]
    - Simplest discontinuity preserving model.
    - Discontinuities between any pair of labels are penalized equally.
    - Useful when labels are unordered or number of labels is small.
  - Extension: “contrast sensitive Potts model”
    \[ \psi(x_i, x_j; y_j; \theta) = \theta \beta \delta(x_i \neq x_j) \]
    - Discourages label changes except in places where there is also a large change in the observations.

Recap: MRF Structure for Images

- Basic structure
  - Bayesian Networks + Applications
  - Markov Random Fields + Applications
  - Exact Inference
  - Approximate Inference

- Two components
  - Observation model
  - Neighborhood relations
  - Serve as smoothing terms.
  - Discourage neighboring pixels to have different labels.
  - This can either be learned or be set to fixed “penalties”.

Course Outline

- Fundamentals (2 weeks)
  - Bayes Decision Theory
  - Probability Density Estimation
- Discriminative Approaches (4 weeks)
  - Lin. Discriminants, SVMs, Boosting
- Generative Models (5 weeks)
  - Bayesian Networks + Applications
  - Markov Random Fields + Applications
  - Exact Inference
  - Approximate Inference
- Unifying Perspective (1 week)

Announcements

- Exercise 5 is available
  - Junction tree algorithm
  - st-mincut
  - Graph cuts
  - Sampling
  - MCMC

Recap: How to Set the Potentials?

- Unary potentials
  - E.g. color model, modeled with a Mixture of Gaussians
    \[ \phi(x_i, y_k; \theta) = \log \sum_k \theta_k p(k|x_i) N(y_k; \bar{y}_k, \Sigma_k) \]
  - Learn color distributions for each label

- Pairwise potentials
  - Potts Model
    \[ \psi(x_i, x_j; \theta) = \theta \delta(x_i \neq x_j) \]
    - Simplest discontinuity preserving model.
    - Discontinuities between any pair of labels are penalized equally.
    - Useful when labels are unordered or number of labels is small.
  - Extension: “contrast sensitive Potts model”
    \[ \psi(x_i, x_j; y_j; \theta) = \theta \beta \delta(x_i \neq x_j) \]
    - Discourages label changes except in places where there is also a large change in the observations.
Recap: Graph Cuts for Binary Problems

Recap: When Can s-t Graph Cuts Be Applied?

Converting an MRF into an s-t Graph

Recap: Simple Binary Image Denoising Model

Recap: s-t-Mincut Equivalent to Maxflow
Dealing with Non-Binary Cases

- Limitation to binary energies is often a nuisance. ⇒ E.g. binary segmentation only.
- We would like to solve also multi-label problems. ⇒ The bad news: Problem is NP-hard with 3 or more labels!
- There exist some approximation algorithms which extend graph cuts to the multi-label case:
  - \(\alpha\)-Expansion
  - \(\alpha\beta\)-Swap
- They are no longer guaranteed to return the globally optimal result.
  - But \(\alpha\)-Expansion has a guaranteed approximation quality (2-approx) and converges in a few iterations.

\(\alpha\)-Expansion Move

- Basic idea:
  - Break multi-way cut computation into a sequence of binary s-t cuts.

\(\alpha\)-Expansion Algorithm

1. Start with any initial solution
2. For each label "\(\alpha\)" in any (e.g. random) order:
   1. Compute optimal \(\alpha\)-expansion move (s-t graph cuts).
   2. Decline the move if there is no energy decrease.
3. Stop when no expansion move would decrease energy.

\(\alpha\)-Expansion Moves

- In each \(\alpha\)-expansion a given label "\(\alpha\)" grabs space from other labels

Example: Stereo Vision

GraphCut Applications: “GrabCut”

- Interactive Image Segmentation [Boykov & Jolly, ICCV’01]
  - Rough region cues sufficient
  - Segmentation boundary can be extracted from edges
- Procedure
  - User marks foreground and background regions with a brush.
  - This is used to create an initial segmentation which can then be corrected by additional brush strokes.
Approximate Inference

- Exact Bayesian inference is often intractable.
  - Often infeasible to evaluate the posterior distribution or to compute expectations w.r.t. the distribution.
  - E.g. because the dimensionality of the latent space is too high.
  - Or because the posterior distribution has a too complex form.
  - Problems with continuous variables
    - Required integrations may not have closed-form solutions.
  - Problems with discrete variables
    - Marginalization involves summing over all possible configurations of the hidden variables.
    - There may be exponentially many such states.

⇒ We need to resort to approximation schemes.
**Topics of This Lecture**

- **Approximate Inference**
  - Variational methods
  - Sampling approximations

- **Sampling approaches**
  - Sampling from a distribution
  - Ancestral Sampling
  - Rejection Sampling
  - Importance Sampling

- **Markov Chain Monte Carlo**
  - Metropolis Chains
  - Metropolis-Hastings Algorithm
  - Gibbs Sampling

---

**Sampling Idea**

- **Objective:** Evaluate expectation of a function $f(z)$ w.r.t. a probability distribution $p(z)$.

\[ \mathbb{E}[f] = \int f(z)p(z)dz \]

- **Sampling idea**
  - Draw $L$ independent samples $z_l$ with $l = 1, \ldots, L$ from $p(z)$.
  - This allows the expectation to be approximated by a finite sum

\[ \mathbb{E}[f] \approx \frac{1}{L} \sum_{l=1}^{L} f(z_l) \]

- As long as the samples $z_l$ are drawn independently from $p(z)$, then
\[ \mathbb{E}[f] = \mathbb{E}[f] \]

\[ \Rightarrow \text{Unbiased estimate, independent of the dimension of } z! \]

---

**Sampling - Challenges**

- **Problem 1:** Samples might not be independent
  \[ \Rightarrow \text{Effective sample size might be much smaller than apparent sample size.} \]

- **Problem 2:**
  - If $f(z)$ is small in regions where $p(z)$ is large and vice versa, the expectation may be dominated by regions of small probability.
  \[ \Rightarrow \text{Large sample sizes necessary to achieve sufficient accuracy.} \]

---

**Sampling from a Gaussian**

- **Given:** 1-dim. Gaussian pdf (probability density function) $p(x|\mu, \sigma^2)$ and the corresponding cumulative distribution:

\[ F_{\mu,\sigma^2}(x) = \int_{-\infty}^{x} p(x|\mu, \sigma^2)dx \]

- To draw samples from a Gaussian, we can invert the cumulative distribution function:

\[ u \sim \text{Uniform}(0, 1) \Rightarrow F_{\mu,\sigma^2}^{-1}(u) \sim p(x|\mu, \sigma^2) \]

---

**Parametric Density Model**

- **Example:**
  - A simple multivariate (d-dimensional) Gaussian model

\[ p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \]

\[ \Rightarrow \text{This is a "generative" model in the sense that we can generate samples } x \text{ according to the distribution.} \]

---

**Sampling from a pdf (Transformation method)**

- **In general,** assume we are given the pdf $p(x)$ and the corresponding cumulative distribution:

\[ F(x) = \int_{-\infty}^{x} p(z)dz \]

- To draw samples from this pdf, we can invert the cumulative distribution function:

\[ u \sim \text{Uniform}(0, 1) \Rightarrow F^{-1}(u) \sim p(x) \]
Ancestral Sampling

- Generalization of this idea to directed graphical models.
  - Joint probability factorizes into conditional probabilities:
    \[ p(x) = \prod_k p(x_k | p_{\text{pa}k}) \]
  - Ancestral sampling
    - Assume the variables are ordered such that there are no links from any node to a lower-numbered node.
    - Start with lowest-numbered node and draw a sample from its distribution.
    - Cycle through each of the nodes in order and draw samples from the conditional distribution (where the parent variable is set to its sampled value).
  \[ \hat{x}_n \sim p(x_n | p_{\text{pa}_n}) \]

Discussion

- Transformation method
  - Limited applicability, as we need to invert the indefinite integral of the required distribution \( p(x) \).
- More general
  - Rejection Sampling
  - Importance Sampling

Rejection Sampling

- Assumptions
  - Sampling directly from \( p(x) \) is difficult.
  - But we can easily evaluate \( p(x) \) up to some normalization factor \( Z_p \):
    \[ p(x) = \frac{1}{Z_p} \tilde{p}(x) \]
- Idea
  - We need some simpler distribution \( q(z) \) (called proposal distribution) from which we can draw samples.
  - Choose a constant \( k \) such that: \( \forall z : kq(z) \geq \tilde{p}(z) \)

Rejection Sampling - Discussion

- Limitation: high-dimensional spaces
  - For rejection sampling to be of practical value, we require that \( kq(z) \) be close to the required distribution, so that the rate of rejection is minimal.
- Artificial example
  - Assume that \( p(x) \) is Gaussian with covariance matrix \( \sigma_x^2 I \)
  - Assume that \( q(z) \) is Gaussian with covariance matrix \( \sigma_z^2 I \)
  - Obviously: \( \sigma_z^2 \geq \sigma_x^2 \)
  - In \( D \) dimensions: \( k = (\sigma_z / \sigma_x)^D \)
    - Assume \( \sigma_z \) is just 1% larger than \( \sigma_x \)
    - \( D = 1000 \Rightarrow k = 1.01^{1000} \gg 20,000 \)
    - And \( p(\text{accept}) \leq \frac{1}{20000} \)
  - Often impractical to find good proposal distributions for high dimensions!

Importance Sampling

- Approach
  - Approximate expectations directly (but does not enable to draw samples from \( p(x) \) directly).
  - Goal: \[ \mathbb{E}[f] = \int f(x)p(x)dx \]
- Simplistic strategy: Grid sampling
  - Discretize \( x \)-space into a uniform grid.
  - Evaluate the integrand as a sum of the form
    \[ \mathbb{E}[f] \approx \sum_{i=1}^{N} f(x^{(i)})p(x^{(i)})dx \]
  - But: number of terms grows exponentially with number of dimensions!
Importance Sampling

- Idea
  - Use a proposal distribution \( q(z) \) from which it is easy to draw samples.
  - Express expectations in the form of a finite sum over samples \( \{z^{(i)}\} \) drawn from \( q(z) \).

  \[
  \mathbb{E}[f] = \int f(x)p(x)dx = \int f(x) \frac{p(x)}{q(x)} q(x)dx \approx \frac{1}{L} \sum_{i=1}^{L} p(z^{(i)}) f(z^{(i)})
  \]
  - with importance weights

  \[
  r_i = \frac{p(z^{(i)})}{q(z^{(i)})}
  \]

  - Ratio of normalization constants can be evaluated

  \[
  \frac{Z_p}{Z_q} = \frac{1}{Z_q} \int p(x)dx = \frac{\int p(x^{(i)}) q(x^{(i)})/q(z^{(i)}) dx}{\int q(x^{(i)}) dx} \approx \frac{1}{L} \sum_{i=1}^{L} r_i
  \]
  - and therefore

  \[
  \mathbb{E}[f] \approx \frac{L}{\sum_{i=1}^{L} r_i} \sum_{i=1}^{L} r_i f(z^{(i)})
  \]
  - with

  \[
  w_i = \frac{r_i}{\sum_{i=1}^{L} r_i} = \frac{p(z^{(i)}) q(z^{(i))}}{\sum_{i=1}^{L} p(z^{(i)}) q(z^{(i))}}
  \]

Importance Sampling - Discussion

- Observations
  - Success of importance sampling depends crucially on how well the sampling distribution \( q(z) \) matches the desired distribution \( p(z) \).
  - Often, \( p(z)/f(z) \) is strongly varying and has a significant proportion of its mass concentrated over small regions of \( z \)-space.
  - Weights \( r_i \) may be dominated by a few weights having large values.
  - Practical issue: if none of the samples falls in the regions where \( p(z)/f(z) \) is large.
  - The results may be arbitrary in error.
  - And there will be no diagnostic indication (no large variance in \( r \))!
  - Key requirement for sampling distribution \( q(z) \):
    - Should not be small or zero in regions where \( p(z) \) is significant!

Topics of This Lecture

- Approximate Inference
  - Variational methods
  - Sampling approaches
- Sampling approaches
  - Sampling from a distribution
    - Metropolis Algorithm
    - Metropolis-Hastings Algorithm
    - Gibbs Sampling
- Markov Chain Monte Carlo
  - Markov Chains
  - Metropolis Algorithm
  - Metropolis-Hastings Algorithm
  - Gibbs Sampling

Independent Sampling vs. Markov Chains

- So far
  - We’ve considered two methods, Rejection Sampling and Importance Sampling, which were both based on independent samples from \( q(z) \).
  - However, for many problems of practical interest, it is difficult or impossible to find \( q(z) \) with the necessary properties.
- Different approach
  - We abandon the idea of independent sampling.
  - Instead, rely on a Markov Chain to generate dependent samples from the target distribution.
  - Independence would be a nice thing, but it is not necessary for the Monte Carlo estimate to be valid.
Markov Chains – Properties

- **Sufficient (but not necessary) condition to ensure that a Markov chain \( \sum_{m=1}^{\infty} z_m^\tau \) tends to \( p(z) \) as \( \tau \to \infty \):**
  - The new candidate sample \( z_{\tau+1} \) is accepted with probability
    \[ A(z_{\tau}^\tau, z_{\tau+1}^{\tau+1}) = \min \left( 1, \frac{p(z_{\tau+1}^{\tau+1})}{p(z_{\tau}^\tau)} \right) \]

- **Implementation:**
  - Choose random number \( u \) uniformly from unit interval \((0,1)\).
  - Accept sample if \( A(z_{\tau}^\tau, z_{\tau+1}^{\tau+1}) > u \).

- **Note:**
  - New candidate samples always accepted if \( p(z_{\tau}^\tau) \geq p(z_{\tau+1}^{\tau+1}) \).
  - I.e. when new sample has higher probability than the previous one.
  - The algorithm sometimes accepts a state with lower probability.

- **Property:**
  - When \( q(x_{\tau+1}|x_\tau) \geq 0 \) for all \( x \), the distribution of \( x_\tau \) tends to \( p(x) \) as \( \tau \to \infty \).

- **Example: Sampling from a Gaussian**
  - Proposal: Gaussian with \( \sigma = 0.2 \).
  - **Green:** accepted samples
  - **Red:** rejected samples

**Markov Chains - Properties**

- **Invariant distribution**
  - A distribution is said to be invariant (or stationary) w.r.t. a Markov chain if each step in the chain leaves that distribution invariant.
  - Transition probabilities:
    \[ T(x_{\tau}^\tau, x_{\tau+1}^{\tau+1}) = p(x_{\tau+1}^{\tau+1}|x_{\tau}^\tau) \]
  - Distribution \( p(x) \) is invariant if:
    \[ p(x) = \sum_{x'} T(x', x) p(x') \]

- **Detailed balance**
  - Sufficient (but not necessary) condition to ensure that a distribution is invariant:
    \[ p'(x) T'(x', x) = p'(x') T'(x', x) \]
  - A Markov chain which respects detailed balance is reversible.
Gibbs Sampling

- **Example**
- Assume distribution \( p(z_1, z_2, z_3) \).
- Replace \( z_1^{(t)} \) with new value drawn from \( z_1^{(t+1)} \sim p(z_1 | z_2^{(t)}, z_3^{(t)}) \)
- Replace \( z_2^{(t)} \) with new value drawn from \( z_2^{(t+1)} \sim p(z_2 | z_1^{(t)} , z_3^{(t)}) \)
- Replace \( z_3^{(t)} \) with new value drawn from \( z_3^{(t+1)} \sim p(z_3 | z_1^{(t)}, z_2^{(t+1)}) \)
- And so on...

MCMC - Metropolis-Hastings Algorithm

- **Properties**
  - We can show that \( p(z) \) is an invariant distribution of the Markov chain defined by the Metropolis-Hastings algorithm.
  - We show detailed balance:
    \[
    p(z) q(z', z) \mathbb{A}(z', z) = \min \left( \frac{p(z') q(z, z')}{p(z) q(z', z)} \mathbb{A}(z, z') \right)
    \]
  - If you can compute (and sample from) the conditionals, you can apply Gibbs sampling.
  - The algorithm is completely parameter free.
  - Can also be applied to subsets of variables.

- **Approach**
  - MCMC-algorithm that is simple and widely applicable.
  - May be seen as a special case of Metropolis-Hastings.

- **Idea**
  - Sample variable-wise: replace \( z_i \) by a value drawn from the distribution \( p(z_i | z_{\neq i}) \).
  - This means we update one coordinate at a time.
  - Repeat procedure either by cycling through all variables or by choosing the next variable.

- **Note**
  - When the proposal distributions are symmetric, Metropolis-Hastings reduces to the standard Metropolis algorithm.
Gibbs Sampling

- Example
  - 20 iterations of Gibbs sampling on a bivariate Gaussian.
  - Note: strong correlations can slow down Gibbs sampling.

Summary: Approximate Inference

- Exact Bayesian inference often intractable.
- Rejection and Importance Sampling
  - Generate independent samples.
  - Impractical in high-dimensional state spaces.
- Markov Chain Monte Carlo (MCMC)
  - Simple & effective (even though typically computationally expensive).
  - Scales well with the dimensionality of the state space.
  - Issues of convergence have to be considered carefully.
- Gibbs Sampling
  - Used extensively in practice.
  - Parameter free
  - Requires sampling conditional distributions.

References and Further Reading

- Sampling methods for approximate inference are described in detail in Chapter 11 of Bishop’s book.
- Another good introduction to Monte Carlo methods can be found in Chapter 29 of MacKay’s book (also available online: http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html)