A Prototypical Course Project

• Find a problem where learning with graphical models seems promising. Typically this needs to be an area where data is easily accessible.

• Identify a baseline approach someone has used to analyze related data. Could involve graphical models, or could involve alternative ML methods.

• Propose, derive, implement, and validate a new approach requiring: A different graphical model structure compared to previous work, and/or different inference/learning algorithms than those tested in previous work.

• Interested in a different project style, perhaps more theoretical? Talk to me.

Project Timeline (more detailed instructions to follow)

• Oct. 23: Identify project teams, send brief description (few paragraphs). Team size? 3 students preferred, 2 is okay, 1 requires special permission.

• Nov. 6: Project proposal due (few pages). Requires an in-depth plan and preliminary results, not just problem area.

• Dec. 9: Project presentations (reading period). Report due following week.
Learning Graphical Model Structures

**Over-Fitting:** Maximum Likelihood always prefers fully-connected graphs

**Strategy 1:** Place hard limit on graph’s “structural complexity”
- Need to balance expressiveness, with learning & inference tractability
- Key example: Optimize over all (pairwise) tree-structure distributions

**Strategy 2:** Define penalized likelihood which encourages simpler graphs
- Interpretable as assigning a prior on models, and finding posterior mode
- Classic approach: Search over graph structures
- Modern approach: Optimization with penalties that encourage sparsity

**Strategy 3:** Bayesian model selection via marginal likelihoods of data
- Better in principle than simple penalties, but often intractable
- Revisit later in the course, once we’ve developed more sophisticated algorithms for approximate learning and inference
Learning Tree-Structured Graphical Models

(a) Original nearest–neighbor grid (observation nodes not shown). (b) Fully factored model employed by the naive mean field method. (c) An embedded tree, as might be exploited by a structured mean field method. (d) Another of this grid’s many embedded trees.

Recovering the true posterior marginals, minimization of $D(q||p)$ leads to tractable algorithms providing potentially useful approximations. In fact, as we discuss in later sections, this variational approach provides a flexible framework for developing richer approximations with increased accuracy. See [161], [311] for an alternative motivation of mean field methods based on conjugate duality.

Structured Mean Field

Results from the statistical physics literature guarantee that, for certain densely connected models with sufficiently homogeneous potentials, the naive mean field approximation becomes exact as the number of variables $N$ approaches infinity [337]. However, for sparse, irregular graphs like those considered by this thesis, its marginal estimates $q_i(x_i)$ can be extremely overconfident, underestimating the uncertainty of the true posterior $p(x_i|y)$. In addition, the mean field iteration of eqs. (2.107, 2.108) often gets stuck in local optima which differ substantially from the true posterior [98], [320]. Geometrically, these local optima arise because the set of pairwise marginals achievable via fully factorized densities is not convex [311].

Motivated by these issues, researchers have developed a variety of variational methods which extend and improve the naive mean field approximation [98], [161], [251], [311]. In particular, fully factorized approximations effectively remove all of the target graphical model’s edges. However, one can also consider structured mean field methods based on approximating families which directly capture more of the original graph’s structure (see Fig. 2.13). Optimization of these approximations is possible assuming exact inference in the chosen subgraphs is tractable [111], [252], [327], [335]. As we show in the following section, Markov chains and trees allow fast, exact recursive inference algorithms which form the basis for a variety of higher–order variational methods.

2.3.2 Belief Propagation

As discussed in Sec. 2.2.5, direct solution of learning and inference problems arising in graphical models is typically intractable. Sometimes, however, global inference tasks
Reminder: Information Theory

- The *entropy* is a natural measure of the inherent uncertainty (difficulty of compression) of some random variable:

\[ H(p_x) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) \geq 0 \quad \text{discrete entropy (concave)} \]

- The *mutual information* measures dependence between a pair of random variables:

\[ I(p_{xy}) = D(p_{xy} \mid \mid p_x p_y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \geq 0 \]

Zero if and only if variables are independent.
Pick any node as root, write directed factorization out to leaves:

\[ p(x) = p(x_s) \prod_{t \neq s} p(x_t | x_{\Gamma(t)}) = p(x_s) \prod_{t \neq s} \frac{p(x_t, x_{\Gamma(t)})}{p(x_{\Gamma(t)})} = p(x_s) \prod_{t \neq s} \frac{p(x_t, x_{\Gamma(t)})}{p(x_t)p(x_{\Gamma(t)})} p(x_t) \]

Grouping terms gives canonical undirected factorizations:

\[
p(x) = \prod_{s \in \mathcal{V}} p(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{p(x_s, x_t)}{p(x_s)p(x_t)} = \prod_{s \in \mathcal{V}} p(x_s)^{1-d_s} \prod_{(s,t) \in \mathcal{E}} p(x_s, x_t)
\]

\[ d_s = \text{number of neighbors for node } s \]
Factorizations for Pairwise Markov Trees

- Pairwise MRFs defined by non-negative potentials:
  \[ p(x) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t) \]

- A canonical parameterization via marginals:
  \[ p(x) = \prod_{s \in \mathcal{V}} q_s(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{q_{st}(x_s, x_t)}{q_s(x_s)q_t(x_t)} \]

- Suppose we choose a **self-consistent** set of non-negative discrete marginals:
  \[ \sum_{x_s} q_s(x_s) = 1, \quad \sum_{x_t} q_{st}(x_s, x_t) = q_s(x_s) \text{ for all } x_s. \]

- The canonical parameterization then exactly matches these marginals:
  \[ q_s(a) = \sum_{x | x_s = a} p(x) \quad q_{st}(a, b) = \sum_{x | x_s = a, x_t = b} p(x) \]
Learning with a Fixed Tree Structure

- **A canonical parameterization** via valid marginals:

\[
p(x) = \prod_{s \in V} q_s(x_s) \prod_{(s,t) \in E} \frac{q_{st}(x_s, x_t)}{q_s(x_s)q_t(x_t)}
\]

\[
\sum_{x_t} q_{st}(x_s, x_t) = q_s(x_s) \text{ for all } x_s.
\]

- Suppose we have \(L\) observations \(\{x^{(\ell)}\}_{\ell=1}^L\) with **empirical distribution**:

\[
\hat{p}_s(a) = \frac{1}{L} \sum_{\ell=1}^L \delta_a(x_s^{(\ell)})
\]

\[
\hat{p}_{st}(a, b) = \frac{1}{L} \sum_{\ell=1}^L \delta_a(x_s^{(\ell)}) \delta_b(x_t^{(\ell)})
\]

- The (normalized) **maximum likelihood** parameter estimation objective is:

\[
\frac{1}{L} \sum_{\ell=1}^L \log p(x^{(\ell)}) = \frac{1}{L} \sum_{\ell=1}^L \left[ \sum_{s \in V} \log q_s(x_s^{(\ell)}) + \sum_{(s,t) \in E} \log \frac{q_{st}(x_s^{(\ell)}, x_t^{(\ell)})}{q_s(x_s^{(\ell)})q_t(x_t^{(\ell)})} \right]
\]
• A canonical parameterization via valid marginals:

\[
p(x) = \prod_{s \in \mathcal{V}} q_s(x_s) \prod_{(s, t) \in \mathcal{E}} \frac{q_{st}(x_s, x_t)}{q_s(x_s)q_t(x_t)}
\]

\[
\sum_{x_t} q_{st}(x_s, x_t) = q_s(x_s) \text{ for all } x_s.
\]

• This is an exponential family distribution with statistics (features):

\[
\phi_{s; a}(x) = \delta_a(x_s) \text{ for all } s \in \mathcal{V}, a = 1, \ldots, K_s.
\]

\[
\phi_{st; ab}(x) = \delta_a(x_s)\delta_b(x_t) \text{ for all } (s, t) \in \mathcal{E}, a = 1, \ldots, K_s, b = 1, \ldots, K_t.
\]

• Maximum likelihood estimates match expected values of the statistics, and thus exactly match the empirical marginal distributions:

\[
q_s(x_s) = \hat{p}_s(x_s) \text{ for all } s \in \mathcal{V}.
\]

\[
q_{st}(x_s, x_t) = \hat{p}_{st}(x_s, x_t) \text{ for all } (s, t) \in \mathcal{E}.
\]
Learning with a Fixed Tree Structure

\( p(x) = \prod_{s \in \mathcal{V}} q_s(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{q_{st}(x_s, x_t)}{q_s(x_s)q_t(x_t)} \)

\( q_s(a) = \hat{p}_s(a) = \frac{1}{L} \sum_{\ell=1}^{L} \delta_a(x_s^{(\ell)}) \)

\( q_{st}(a, b) = \hat{p}_{st}(a, b) = \frac{1}{L} \sum_{\ell=1}^{L} \delta_a(x_s^{(\ell)})\delta_b(x_t^{(\ell)}) \)

- Optimum ML model matches empirical marginals for all single variables, and empirical \textit{pairwise marginals for all neighboring variables}
- For these optimal parameters, the corresponding training log-likelihood is:

\[
\frac{1}{L} \sum_{\ell=1}^{L} \log p(x^{(\ell)}) = - \sum_{s \in \mathcal{V}} \hat{H}_s + \sum_{(s,t) \in \mathcal{E}} \hat{I}_{st}
\]

\[
\hat{H}_s = - \sum_{x_s} \hat{p}_s(x_s) \log \hat{p}_s(x_s)
\]

\textit{Empirical Entropy} & \textit{Empirical Mutual Info}

\[
\hat{I}_{st} = \sum_{x_s, x_t} \hat{p}_{st}(x_s, x_t) \log \frac{\hat{p}_{st}(x_s, x_t)}{\hat{p}_s(x_s)\hat{p}_t(x_t)}
\]
Learning with a Fixed Tree Structure

\[ p(x) = \prod_{s \in \mathcal{V}} q_s(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{q_{st}(x_s, x_t)}{q_s(x_s)q_t(x_t)} \]

\[ q_s(a) = \hat{p}_s(a) = \frac{1}{L} \sum_{\ell=1}^{L} \delta_a(x_s^{(\ell)}) \]

\[ q_{st}(a, b) = \hat{p}_{st}(a, b) = \frac{1}{L} \sum_{\ell=1}^{L} \delta_a(x_s^{(\ell)})\delta_b(x_t^{(\ell)}) \]

- Optimum ML model matches empirical marginals for all single variables, and empirical \textit{pairwise marginals for all neighboring variables.}
- For these optimal parameters, the corresponding training log-likelihood is:

\[ \frac{1}{L} \sum_{\ell=1}^{L} \log p(x^{(\ell)}) = - \sum_{s \in \mathcal{V}} \hat{H}_s + \sum_{(s,t) \in \mathcal{E}} \hat{I}_{st} \]

Data that is “more random” is harder to predict. Empirical Entropy & Empirical Mutual Info Likelihood increases when link strongly dependent variables.
Optimizing Tree Structures

\[ p(x) = \prod_{s \in \mathcal{V}} q_s(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{q_{st}(x_s, x_t)}{q_s(x_s)q_t(x_t)} \]

- Picking the max-likelihood marginals for any tree:
  \[ \frac{1}{L} \sum_{\ell=1}^{L} \log p(x^{(\ell)}) = - \sum_{s \in \mathcal{V}} \hat{H}_s + \sum_{(s,t) \in \mathcal{E}} \hat{I}_{st} \]
  \[ \hat{H}_s = - \sum_{x_s} \hat{p}_s(x_s) \log \hat{p}_s(x_s) \]
  \[ \hat{I}_{st} = \sum_{x_s, x_t} \hat{p}_{st}(x_s, x_t) \log \frac{\hat{p}_{st}(x_s, x_t)}{\hat{p}_s(x_s)\hat{p}_t(x_t)} \]

- Need to maximize the following objective over acyclic edge patterns:
  \[ f(\mathcal{E}) = \sum_{(s,t) \in \mathcal{E}} \hat{I}_{st} \quad \text{Weight each pair of nodes by its empirical mutual information, and find the tree which maximizes the sum of edge weights!} \]

- Example greedy algorithm (Kruskal):
  - Initialize with highest-weight edge (most-dependent pair of variables).
  - Iteratively add the highest-weight edge that is not yet included in the tree, and does not introduce any cycles. Stop when all variables connected.
Graph Structure Identification via Sparse Regression
Factor Graphs & Exponential Families

\[ p(x) = \frac{1}{Z(\theta)} \prod_{f \in \mathcal{F}} \psi_f(x_f \mid \theta_f) \]

**Graph structure selection is feature selection!**

Want to determine which factors (features) should be used, and which should be discarded (assigned zero weight).

- A factor graph is created from non-negative potential functions, often defined as
  \[ \psi_f(x_f \mid \theta_f) = \exp \left\{ \theta_f^T \phi_f(x_f) \right\} \]

Local exponential family

\[ \phi_f(x_f) \in \mathbb{R}^{d_f} \]

\[ p(x \mid \theta) = \exp \left\{ \sum_{f \in \mathcal{F}} \theta_f^T \phi_f(x_f) - \Phi(\theta) \right\} \]

Log normalization constant

\[ \Phi(\theta) = \log Z(\theta) \]
Sparse Feature Selection for Regression

Review of material from Brown CS142, Fall 2013
http://cs.brown.edu/courses/csci1420/calendar.html
See lecture slides from October 17 & October 22.
Sparse Learning for Undirected Models

\[
\log p(x \mid \theta) = \sum_{\ell=1}^{L} \sum_{f \in \mathcal{F}} \theta^T_f \phi_f(x^{(\ell)}_f) - L \Phi(\theta)
\]

\[
- \log p(\theta \mid x) = C - \left[ \sum_{\ell=1}^{L} \sum_{f \in \mathcal{F}} \theta^T_f \phi_f(x^{(\ell)}_f) - L \Phi(\theta) \right] + \lambda \|\theta\|_1
\]

- Standard software packages for \(L_1\)-regularized learning assume we can evaluate objective function and its gradient in closed form. This is possible assuming inference tractable in model with all features.

- Pseudo-likelihood estimators & variational estimators approximate true likelihood, but can scale to features where exact inference intractable.

- Can replace \(L1\) with fancier penalties to encourage blocks of parameters to simultaneously be set to zero.

- Learning directed graphs? Tricky unless global variable order is known.