EE12A – Advanced Inference in Graphical Models

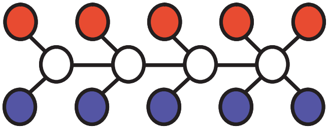
— Fall Quarter, Lecture 1 —

http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2014/

Prof. Jeff Bilmes

University of Washington, Seattle
Department of Electrical Engineering
http://melodi.ee.washington.edu/~bilmes

Sep 29th, 2014



Announcements

- Welcome to the class!

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`
- Lecturer: Prof. Jeff Bilmes, office hours Mondays 1:35-2:35pm, EEB-418.

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`
- Lecturer: Prof. Jeff Bilmes, office hours Mondays 1:35-2:35pm, EEB-418.
- Also available online by appointment (e.g., skype, google hangouts).

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`
- Lecturer: Prof. Jeff Bilmes, office hours Mondays 1:35-2:35pm, EEB-418.
- Also available online by appointment (e.g., skype, google hangouts).
- TA: Jounsup Park `<jsup517@uw.edu>`, office hours Tuesdays 12:00pm - 2:00pm, EEB-333.

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`
- Lecturer: Prof. Jeff Bilmes, office hours Mondays 1:35-2:35pm, EEB-418.
- Also available online by appointment (e.g., skype, google hangouts).
- TA: Jounsup Park `<jsup517@uw.edu>`, office hours Tuesdays 12:00pm - 2:00pm, EEB-333.
- On our web page (`http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2014/`), we will have announcements, readings, homework assignments, copies of these slides, bboard, and so on.

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`
- Lecturer: Prof. Jeff Bilmes, office hours Mondays 1:35-2:35pm, EEB-418.
- Also available online by appointment (e.g., skype, google hangouts).
- TA: Jounsup Park <jsup517@uw.edu>, office hours Tuesdays 12:00pm - 2:00pm, EEB-333.
- On our web page (`http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2014/`), we will have announcements, readings, homework assignments, copies of these slides, bboard, and so on.
- We'll have 3-5 homeworks this quarter. You'll have about a week to turn them in.

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`
- Lecturer: Prof. Jeff Bilmes, office hours Mondays 1:35-2:35pm, EEB-418.
- Also available online by appointment (e.g., skype, google hangouts).
- TA: Jounsup Park `<jsup517@uw.edu>`, office hours Tuesdays 12:00pm - 2:00pm, EEB-333.
- On our web page (`http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2014/`), we will have announcements, readings, homework assignments, copies of these slides, bboard, and so on.
- We'll have 3-5 homeworks this quarter. You'll have about a week to turn them in.
- Copies of lecture slides available on the web.

Class information

- Mon, Wed, 11:30-1:30 in PCAR-297 (this room).
- Lectures will also appear on youtube. See youtube channel `https://www.youtube.com/channel/UCvPnLF7oUh4p-m575fZcUxg`
- Lecturer: Prof. Jeff Bilmes, office hours Mondays 1:35-2:35pm, EEB-418.
- Also available online by appointment (e.g., skype, google hangouts).
- TA: Jounsup Park `<jsup517@uw.edu>`, office hours Tuesdays 12:00pm - 2:00pm, EEB-333.
- On our web page (`http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2014/`), we will have announcements, readings, homework assignments, copies of these slides, bboard, and so on.
- We'll have 3-5 homeworks this quarter. You'll have about a week to turn them in.
- Copies of lecture slides available on the web.
- Copies of (most) readings available on the web

Homework

- Again, about 3-5 this quarter.

Homework

- Again, about 3-5 this quarter.
- Problem sets: answer a question, prove a theorem, etc.

Homework

- Again, about 3-5 this quarter.
- Problem sets: answer a question, prove a theorem, etc.
- Alternatively programming projects, so you should be familiar with at least one programming language (e.g., C, C++, or matlab).

Homework

- Again, about 3-5 this quarter.
- Problem sets: answer a question, prove a theorem, etc.
- Alternatively programming projects, so you should be familiar with at least one programming language (e.g., C, C++, or matlab).
- All homework must be turned in electronically in PDF form via canvas at our assignment dropbox (https://canvas.uw.edu/courses/914697/assignments).

Final Project Possibility

- There will be a final project, consisting of a 4-page conference-style paper, and a 10-minute final presentation.
- There will be a few milestones (1-page project proposal, 1-page progress summaries) during the quarter. These are graded.
- The final project should be regarding graphical models - either as a user in an application, or as a researcher (i.e., new inference method, new proof, etc.).
- The date of the final project is tentatively Wednesday, December 10, 2014, 230-420 pm, PCAR 297.
- Final project reports due Tuesday, Dec 9th, at 11:45pm.
- All final project relate assignment must be turned in electronically via our class web page.

Final Project: Alternate

- There is a chance we will do a graphical model inference contest as the final project. More on this as the class progresses.

Our texts

- There will be three sources of reading material we'll use this term.

Our texts

- There will be three sources of reading material we'll use this term.
  - Handouts written by me (these are being prepared/updated now, and are not entirely finished). Material here will be mostly on GM semantics and exact inference methods.

Our texts

- There will be three sources of reading material we'll use this term.
  - Handouts written by me (these are being prepared/updated now, and are not entirely finished). Material here will be mostly on GM semantics and exact inference methods.
  - Two text books (next page). One is available for free electronically.

Our texts

- There will be three sources of reading material we'll use this term.
  - Handouts written by me (these are being prepared/updated now, and are not entirely finished). Material here will be mostly on GM semantics and exact inference methods.
  - Two text books (next page). One is available for free electronically.
  - Research papers (links will be given in the class slides and on the web).

Our texts

- There will be three sources of reading material we'll use this term.
  - Handouts written by me (these are being prepared/updated now, and are not entirely finished). Material here will be mostly on GM semantics and exact inference methods.
  - Two text books (next page). One is available for free electronically.
  - Research papers (links will be given in the class slides and on the web).
- Also might pick up a copy of the recent book by Koller and Friedman.

Our texts

- There will be three sources of reading material we'll use this term.
  - Handouts written by me (these are being prepared/updated now, and are not entirely finished). Material here will be mostly on GM semantics and exact inference methods.
  - Two text books (next page). One is available for free electronically.
  - Research papers (links will be given in the class slides and on the web).
- Also might pick up a copy of the recent book by Koller and Friedman.
- Lauritzen 1996 is a classic book on GMs.

Our texts

- There will be three sources of reading material we'll use this term.
  - Handouts written by me (these are being prepared/updated now, and are not entirely finished). Material here will be mostly on GM semantics and exact inference methods.
  - Two text books (next page). One is available for free electronically.
  - Research papers (links will be given in the class slides and on the web).
- Also might pick up a copy of the recent book by Koller and Friedman.
- Lauritzen 1996 is a classic book on GMs.
- Two other books on Bayesian networks include Jensen 1996 and 2001.

Our texts

- There will be three sources of reading material we'll use this term.
  - Handouts written by me (these are being prepared/updated now, and are not entirely finished). Material here will be mostly on GM semantics and exact inference methods.
  - Two text books (next page). One is available for free electronically.
  - Research papers (links will be given in the class slides and on the web).
- Also might pick up a copy of the recent book by Koller and Friedman.
- Lauritzen 1996 is a classic book on GMs.
- Two other books on Bayesian networks include Jensen 1996 and 2001.
- Two nice books on machine learning and pattern recognition are Bishop 2006 and Murphy 2012.

Our two main texts

- Wainwright and Jordan *Graphical Models, Exponential Families, and Variational Inference* `http://www.nowpublishers.com/product.aspx?product=MAL&doi=2200000001`
- *Markov Random Fields for Vision and Image Processing* `http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=12668` edited by Andrew Blake, Pushmeet Kohli and Carsten Rother

Announcements

- Reading assignment: Read the "trees.pdf" chapter soon to be posted on our canvas announcements page (https://canvas.uw.edu/courses/914697/announcements).
- Slides from previous time this course was offered are at our previous web page (http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2011/) and even earlier at http://melodi.ee.washington.edu/~bilmes/ee512fa09/.

Class Road Map - EE512a

- L1 (9/29): Introduction, Families, Semantics
- L2 (10/1):
- L3 (10/6):
- L4 (10/8):
- L5 (10/13):
- L6 (10/15):
- L7 (10/20):
- L8 (10/22):
- L9 (10/27):
- L10 (10/29):
- L11 (11/3):
- L12 (11/5):
- L13 (11/10):
- L14 (11/12):
- L15 (11/17):
- L16 (11/19):
- L17 (11/24):
- L18 (11/26):
- L19 (12/1):
- L20 (12/3):
- Final Presentations: (12/10):

Finals Week: Dec 8th-12th, 2014.

Review

- This is where we will review previous lectures.

Probabilistic Inference

- Probabilistic Inference involves computing quantities of interest based on probability distributions. E.g., marginalization $\sum_{x_1} p(x_1, x_2)$ or maximization $\max_{x_1} p(x_1, \bar{x}_2)$.

Probabilistic Inference

- Probabilistic Inference involves computing quantities of interest based on probability distributions. E.g., marginalization $\sum_{x_1} p(x_1, x_2)$ or maximization $\max_{x_1} p(x_1, \bar{x}_2)$.
- Probabilistic inference is hard, .

Probabilistic Inference

- Probabilistic Inference involves computing quantities of interest based on probability distributions. E.g., marginalization $\sum_{x_1} p(x_1, x_2)$ or maximization $\max_{x_1} p(x_1, \bar{x}_2)$.
- Probabilistic inference is hard, often NP-hard or even inapproximable.

Probabilistic Inference

- Probabilistic Inference involves computing quantities of interest based on probability distributions. E.g., marginalization $\sum_{x_1} p(x_1, x_2)$ or maximization $\max_{x_1} p(x_1, \bar{x}_2)$.
- Probabilistic inference is hard, often NP-hard or even inapproximable.
- Best of cases, exact inference is doable in polynomial time (trees).

Probabilistic Inference

- Probabilistic Inference involves computing quantities of interest based on probability distributions. E.g., marginalization $\sum_{x_1} p(x_1, x_2)$ or maximization $\max_{x_1} p(x_1, \bar{x}_2)$.
- Probabilistic inference is hard, often NP-hard or even inapproximable.
- Best of cases, exact inference is doable in polynomial time (trees).
- Worst of cases, exact inference is infeasible, so approximation is necessary

Probabilistic Inference

- Probabilistic Inference involves computing quantities of interest based on probability distributions. E.g., marginalization $\sum_{x_1} p(x_1, x_2)$ or maximization $\max_{x_1} p(x_1, \bar{x}_2)$.
- Probabilistic inference is hard, often NP-hard or even inapproximable.
- Best of cases, exact inference is doable in polynomial time (trees).
- Worst of cases, exact inference is infeasible, so approximation is necessary
- Plethora of approximation methods are possible.

Probabilistic Inference

- Probabilistic Inference involves computing quantities of interest based on probability distributions. E.g., marginalization $\sum_{x_1} p(x_1, x_2)$ or maximization $\max_{x_1} p(x_1, \bar{x}_2)$.
- Probabilistic inference is hard, often NP-hard or even inapproximable.
- Best of cases, exact inference is doable in polynomial time (trees).
- Worst of cases, exact inference is infeasible, so approximation is necessary
- Plethora of approximation methods are possible.
- The course this term will mostly concentrate on graphical model semantics, exact inference methods, and two broad approximation inference methods based on graphical models.

Approximation Method: Variational

The general variational approach encompasses many standard approximate inference methods, including:

Approximation Method: Variational

The general variational approach encompasses many standard approximate inference methods, including:

- sum-product
- cluster variational methods
- expectation-propagation
- mean field methods
- max-product
- linear programming relaxations
- conic programming relaxations

and is therefore worthy of study. Of particular interest is for the class of exponential models (which have strong relationships to convexity).

Approximation Method: Move making

- Many inference methods from computer vision have appeared recently.

Approximation Method: Move making

- Many inference methods from computer vision have appeared recently.
- Simplest of the ideas: use an efficient graph-cut approach to find the minimum energy configuration in a pairwise binary Markov random field.

Approximation Method: Move making

- Many inference methods from computer vision have appeared recently.
- Simplest of the ideas: use an efficient graph-cut approach to find the minimum energy configuration in a pairwise binary Markov random field.
- When is this optimal? When is this an approximation? How can we generalize this to non-binary variables, non-pairwise potentials, and richer potentials?

Approximation Method: Move making

- Many inference methods from computer vision have appeared recently.
- Simplest of the ideas: use an efficient graph-cut approach to find the minimum energy configuration in a pairwise binary Markov random field.
- When is this optimal? When is this an approximation? How can we generalize this to non-binary variables, non-pairwise potentials, and richer potentials?
- Many generalizations, including move making algorithms such as alpha-beta swaps, alpha expansions, fusion moves, and other recent more sophisticated and energy aware "move making" algorithms.

Approximation Method: Move making

- Many inference methods from computer vision have appeared recently.
- Simplest of the ideas: use an efficient graph-cut approach to find the minimum energy configuration in a pairwise binary Markov random field.
- When is this optimal? When is this an approximation? How can we generalize this to non-binary variables, non-pairwise potentials, and richer potentials?
- Many generalizations, including move making algorithms such as alpha-beta swaps, alpha expansions, fusion moves, and other recent more sophisticated and energy aware "move making" algorithms.
- Computer vision and beyond.

Other inference methods

- Sampling, Monte Carlo, MCMC methods, importance sampling

Other inference methods

- Sampling, Monte Carlo, MCMC methods, importance sampling
- Search based methods, cut condition, value elimination, as done in CSP/SAT communities. This includes AND/OR search trees, sum/product networks, where the network represents the operations necessary to do inference.

Other inference methods

- Sampling, Monte Carlo, MCMC methods, importance sampling
- Search based methods, cut condition, value elimination, as done in CSP/SAT communities. This includes AND/OR search trees, sum/product networks, where the network represents the operations necessary to do inference.
- Also, other modern search based methods.

Other inference methods

- Sampling, Monte Carlo, MCMC methods, importance sampling
- Search based methods, cut condition, value elimination, as done in CSP/SAT communities. This includes AND/OR search trees, sum/product networks, where the network represents the operations necessary to do inference.
- Also, other modern search based methods.
- Beam pruning methods often go hand-in-hand with search based methods.

Some notation

- Distributions

$$p(x) \equiv p(x_{1:N}) \equiv p(x_1, \dots, x_N) \equiv P_{X_1,\dots,X_N}(X_1 = x_1, \dots, X_N = x_N)$$

$X_{1:N} = x_{1:N}$

- Subsets

$$V \triangleq \{1, 2, \dots, N\} \quad A, B \subseteq V \quad A = \{a_1, \dots, a_{|A|}\} \tag{1.1}$$

$$X_A \triangleq \{X_{a_1}, X_{a_2}, \dots, X_{a_{|A|}}\} \tag{1.2}$$

- Example: If $A = \{1, 3, 7\}$ then $X_A = \{X_1, X_3, X_7\}$ and

$$p(X_A = x_A | X_B = x_B) \equiv p(x_1, x_2 | x_3, x_4)$$
$$\text{if } A = \{1, 2\}, B = \{3, 4\}$$

- $p(x_A)$ requires table of size $r^{|A|}$, $r = |\mathsf{D}_X|$ where $\forall i, x_i \in \mathsf{D}_X$
- $\bar{x}^{(i)}$ and $\bar{x}^{(j)}$ are different vector samples for $i \neq j$.

What might we want to do with $p(x)$?

- Marginal quantities

What might we want to do with $p(x)$?

- Marginal quantities
  - Given $\bar{x}$ compute $p(\bar{x})$

What might we want to do with $p(x)$?

- Marginal quantities
  - Given $\bar{x}$ compute $p(\bar{x})$
  - Given $E \subseteq V$, and $F \cup H = V \setminus E$ with F and H disjoint, then compute

$$p(x_F, \bar{x}_E) = \sum_{x_H} p(x_F, x_H, \bar{x}_E). \tag{1.3}$$

$$p(x_F | \bar{x}_E) \propto p(x_F, \bar{x}_E)$$

What might we want to do with $p(x)$?

- Marginal quantities
  - Given $\bar{x}$ compute $p(\bar{x})$
  - Given $E \subseteq V$, and $F \cup H = V \setminus E$ with F and H disjoint, then compute

$$p(x_F, \bar{x}_E) = \sum_{x_H} p(x_F, x_H, \bar{x}_E). \tag{1.3}$$

- Model relationship between two signals x_1 and x_2 (e.g., x_1 a feature vector, x_2 is a class or regression variable).

What might we want to do with $p(x)$?

- Marginal quantities
  - Given $\bar{x}$ compute $p(\bar{x})$
  - Given $E \subseteq V$, and $F \cup H = V \setminus E$ with F and H disjoint, then compute

$$p(x_F, \bar{x}_E) = \sum_{x_H} p(x_F, x_H, \bar{x}_E). \tag{1.3}$$

- Model relationship between two signals x_1 and x_2 (e.g., x_1 a feature vector, x_2 is a class or regression variable).
  - compute $p(\bar{x}_1, \bar{x}_2)$.

What might we want to do with $p(x)$?

- Marginal quantities
  - Given $\bar{x}$ compute $p(\bar{x})$
  - Given $E \subseteq V$, and $F \cup H = V \setminus E$ with F and H disjoint, then compute

$$p(x_F, \bar{x}_E) = \sum_{x_H} p(x_F, x_H, \bar{x}_E). \tag{1.3}$$

- Model relationship between two signals x_1 and x_2 (e.g., x_1 a feature vector, x_2 is a class or regression variable).
  - compute $p(\bar{x}_1, \bar{x}_2)$.
  - Given $\bar{x}_1$ compute

$$x_2^* \in \operatorname*{argmax}_{x_2} p(\bar{x}_1, x_2) \text{ or equivalently } x_2^* \in \operatorname*{argmax}_{x_2} p(x_2|\bar{x}_1) \tag{1.4}$$

What might we want to do with $p(x)$?

- Machine Learning is adjusting a model based on data.

What might we want to do with $p(x)$?

- Machine Learning is adjusting a model based on data.
- Machine Learning almost always requires being able to do inference efficiently.

What might we want to do with $p(x)$?

- Machine Learning is adjusting a model based on data.
- Machine Learning almost always requires being able to do inference efficiently.
- We are given set of training samples $\mathbf{D} = \{x^{(1)}, x^{(2)}, \ldots\}$.

What might we want to do with $p(x)$?

- Machine Learning is adjusting a model based on data.
- Machine Learning almost always requires being able to do inference efficiently.
- We are given set of training samples $\mathbf{D} = \{x^{(1)}, x^{(2)}, \dots\}$.
- Then find $\theta^* \in \operatorname{argmin}_\theta R(\mathbf{D}, \theta)$ where $R(\mathbf{D}, \theta)$ is a risk.

What might we want to do with $p(x)$?

- Machine Learning is adjusting a model based on data.
- Machine Learning almost always requires being able to do inference efficiently.
- We are given set of training samples $\mathbf{D} = \{x^{(1)}, x^{(2)}, \dots\}$.
- Then find $\theta^* \in \operatorname{argmin}_\theta R(\mathbf{D}, \theta)$ where $R(\mathbf{D}, \theta)$ is a risk.
- Given θ^*, we may we interpret its values (generative learning)?

What might we want to do with $p(x)$?

- Machine Learning is adjusting a model based on data.
- Machine Learning almost always requires being able to do inference efficiently.
- We are given set of training samples $\mathbf{D} = \{x^{(1)}, x^{(2)}, \dots\}$.
- Then find $\theta^* \in \operatorname{argmin}_\theta R(\mathbf{D}, \theta)$ where $R(\mathbf{D}, \theta)$ is a risk.
- Given θ^*, we may we interpret its values (generative learning)?
- Given θ^*, could form distribution $p_{\theta^*}(x)$ or marginal $p_{\theta^*}(x_1, x_2)$, etc.

Learning depends on loss functions, but needs inference

- Generative learning if

$$R(\mathbf{D},\theta) = -\frac{1}{|\mathbf{D}|}\sum_{j=1}^{|\mathbf{D}|} \log p_\theta(x^{(j)}) + \lambda\|\theta\| \tag{1.5}$$

where $\|\cdot\|$ is some norm. This includes the case of maximum likelihood learning.

Learning depends on loss functions, but needs inference

- Generative learning if

$$R(\mathbf{D},\theta) = -\frac{1}{|\mathbf{D}|}\sum_{j=1}^{|\mathbf{D}|} \log p_\theta(x^{(j)}) + \lambda\|\theta\| \tag{1.5}$$

where $\|\cdot\|$ is some norm. This includes the case of maximum likelihood learning.

- Discriminative learning results when

$$R(\mathbf{D},\theta) = -\frac{1}{|\mathbf{D}|}\sum_{j=1}^{|\mathbf{D}|} \log p_\theta(x_2^{(j)}|x_1^{(j)}) + \lambda\|\theta\| \tag{1.6}$$

and includes the case of maximum conditional likelihood learning.

Learning depends on loss functions, but needs inference

- Another form of discriminative learning, max-margin learning, occurs when if

$$R(\mathbf{D},\theta) = \frac{1}{|\mathbf{D}|}\sum_{i=1}^{|\mathbf{D}|}\Big[\max_{x_2}\Big(\log p_\theta(x_2, x_1^{(j)}) + \Delta(x_2^{(j)}, x_2')\Big) - \log p_\theta(x_2^{(j)}, x_1^{(j)})\Big] + \lambda\|\theta\| \quad (1.7)$$

where $\Delta(x_2^{(j)}, x_2')$ is a normalizing labeling cost. Overall, this corresponds to a generalized hinge-loss.

Learning depends on loss functions, but needs inference

- Another form of discriminative learning, max-margin learning, occurs when if

$$R(\mathbf{D},\theta) = \frac{1}{|\mathbf{D}|}\sum_{i=1}^{|\mathbf{D}|}\Bigg[\max_{x_2}\Big(\log p_\theta(x_2, x_1^{(j)}) + \Delta(x_2^{(j)}, x_2')\Big) - \log p_\theta(x_2^{(j)}, x_1^{(j)})\Bigg] + \lambda\|\theta\| \quad (1.7)$$

where $\Delta(x_2^{(j)}, x_2')$ is a normalizing labeling cost. Overall, this corresponds to a generalized hinge-loss.

- Optimizing each risk is unique, but each invariably entails computing quantities over $p(x)$ like the aforementioned inference.

Learning depends on loss functions, but needs inference

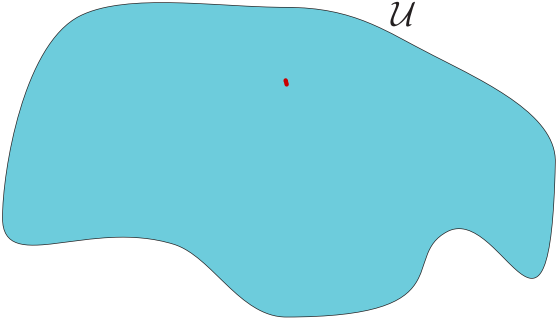
- Another form of discriminative learning, max-margin learning, occurs when if

$$R(\mathbf{D},\theta) = \frac{1}{|\mathbf{D}|}\sum_{i=1}^{|\mathbf{D}|}\Bigg[\max_{x_2}\Big(\log p_\theta(x_2, x_1^{(j)}) + \Delta(x_2^{(j)}, x_2')\Big) - \log p_\theta(x_2^{(j)}, x_1^{(j)})\Bigg] + \lambda\|\theta\| \quad (1.7)$$

  where $\Delta(x_2^{(j)}, x_2')$ is a normalizing labeling cost. Overall, this corresponds to a generalized hinge-loss.
- Optimizing each risk is unique, but each invariably entails computing quantities over $p(x)$ like the aforementioned inference.
- The need to efficiently compute with $p(x)$ is critical for most machine learning problems.

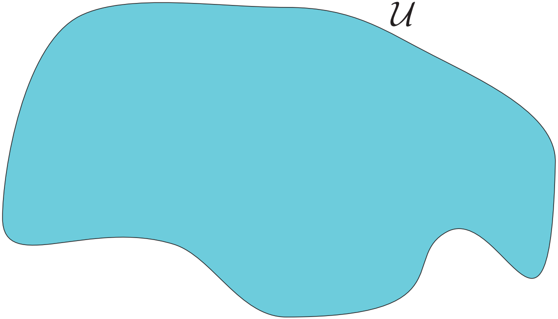
Machine learning within restricted families

- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.



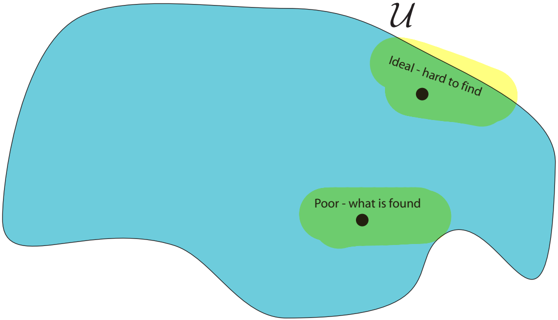
Machine learning within restricted families

- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.
- Sample data, along with domain knowledge, used to select resulting $p(x)$ from $\mathcal{U}$ that is "close enough" to $p_{\text{true}}(x_1, \ldots, x_N)$.



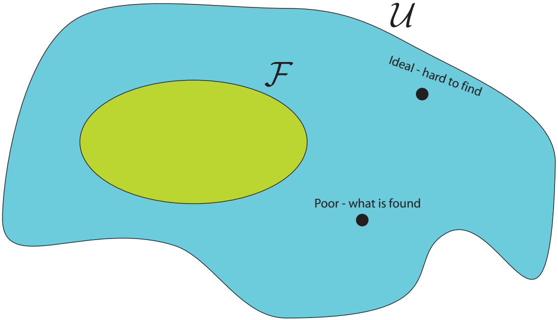
Machine learning within restricted families

- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.
- Sample data, along with domain knowledge, used to select resulting $p(x)$ from $\mathcal{U}$ that is "close enough" to $p_{\text{true}}(x_1, \ldots, x_N)$.
- Searching within $\mathcal{U}$ is infeasible/intractable/impossible.



Machine learning within restricted families

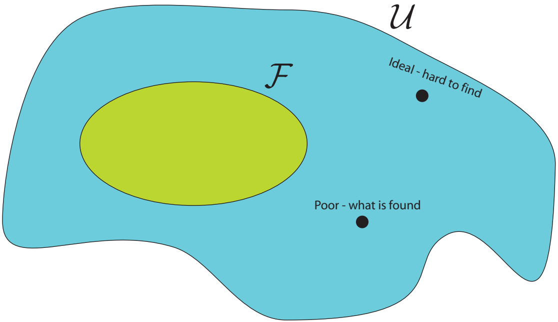
- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.
- Sample data, along with domain knowledge, used to select resulting $p(x)$ from $\mathcal{U}$ that is "close enough" to $p_{\text{true}}(x_1, \ldots, x_N)$.
- Searching within $\mathcal{U}$ is infeasible/intractable/impossible.
- Desire a restricted but useful family $\mathcal{F} \subset \mathcal{U}$.



Machine learning within restricted families

- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.
- Sample data, along with domain knowledge, used to select resulting $p(x)$ from $\mathcal{U}$ that is "close enough" to $p_{\text{true}}(x_1, \ldots, x_N)$.
- Searching within $\mathcal{U}$ is infeasible/intractable/impossible.
- Desire a restricted but useful family $\mathcal{F} \subset \mathcal{U}$.

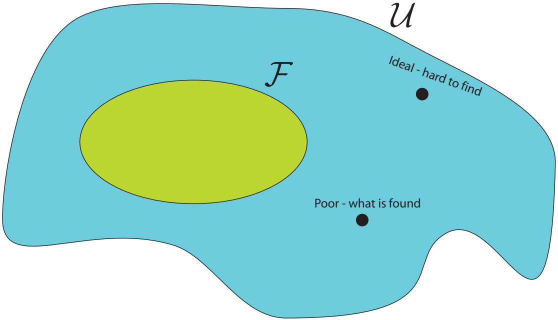
- Size of $\mathcal{U}$ too large



Machine learning within restricted families

- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.
- Sample data, along with domain knowledge, used to select resulting $p(x)$ from $\mathcal{U}$ that is "close enough" to $p_{\text{true}}(x_1, \ldots, x_N)$.
- Searching within $\mathcal{U}$ is infeasible/intractable/impossible.
- Desire a restricted but useful family $\mathcal{F} \subset \mathcal{U}$.

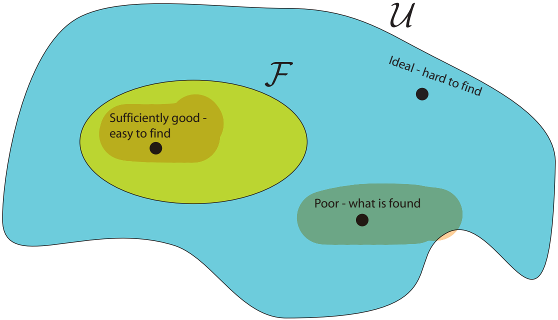
- Size of $\mathcal{U}$ too large
- $\mathcal{U}$ complex, local optima



Machine learning within restricted families

- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.
- Sample data, along with domain knowledge, used to select resulting $p(x)$ from $\mathcal{U}$ that is "close enough" to $p_{\text{true}}(x_1, \ldots, x_N)$.
- Searching within $\mathcal{U}$ is infeasible/intractable/impossible.
- Desire a restricted but useful family $\mathcal{F} \subset \mathcal{U}$.

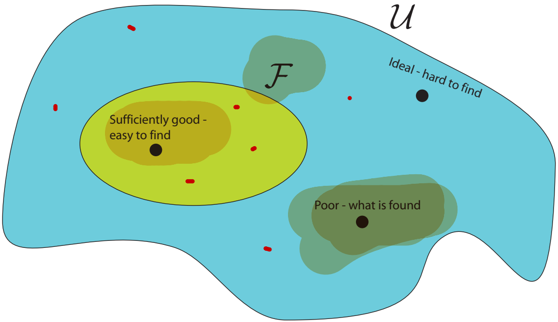
- Size of $\mathcal{U}$ too large
- $\mathcal{U}$ complex, local optima
- Actual solution in $\mathcal{F}$ better than possible solution in $\mathcal{U}$



Machine learning within restricted families

- Let $\mathcal{U}$ be the universe of all distributions over N r.v.s.
- Sample data, along with domain knowledge, used to select resulting $p(x)$ from $\mathcal{U}$ that is "close enough" to $p_{\text{true}}(x_1, \ldots, x_N)$.
- Searching within $\mathcal{U}$ is infeasible/intractable/impossible.
- Desire a restricted but useful family $\mathcal{F} \subset \mathcal{U}$.

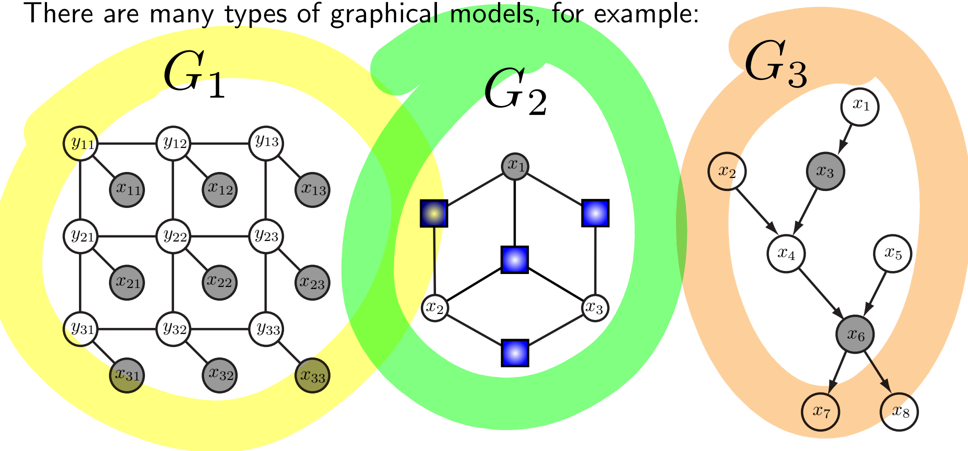
- Size of $\mathcal{U}$ too large
- $\mathcal{U}$ complex, local optima
- Actual solution in $\mathcal{F}$ better than possible solution in $\mathcal{U}$
- Graphical models provide a framework for specifying $\mathcal{F} \subseteq \mathcal{U}$



Graphical Models

- A graphical model is a visual, abstract, and mathematically formal description of properties of families of probability distributions (densities, mass functions)

There are many types of graphical models, for example:



Graphical Models

- Graphical models are encodings of families of probability distributions. For the most part, the encodings are done via a graph that formally specifies either a set (conditional) independence properties, or more fundamentally, a set of factorization properties.

$$p(a,b|c) = p(a|c)p(b|c)$$

Graphical Models

- Graphical models are encodings of families of probability distributions. For the most part, the encodings are done via a graph that formally specifies either a set (conditional) independence properties, or more fundamentally, a set of factorization properties.
- This is a crucial idea to understand: a graphical model is a set of constraints that all family members must obey.

Graphical Models

- Graphical models are encodings of families of probability distributions. For the most part, the encodings are done via a graph that formally specifies either a set (conditional) independence properties, or more fundamentally, a set of factorization properties.
- This is a crucial idea to understand: a graphical model is a set of constraints that all family members must obey.
- Graphical Models encode constraints by factorization requirements that all members of the family must obey.

Graphical Models

- Graphical models are encodings of families of probability distributions. For the most part, the encodings are done via a graph that formally specifies either a set (conditional) independence properties, or more fundamentally, a set of factorization properties.
- This is a crucial idea to understand: a graphical model is a set of constraints that all family members must obey.
- Graphical Models encode constraints by factorization requirements that all members of the family must obey.
- Factorization requirements are often identical to conditional independence requirements.

Graphical Models

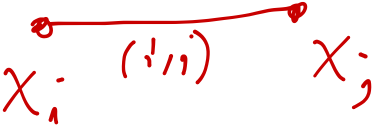
- Graphical models are encodings of families of probability distributions. For the most part, the encodings are done via a graph that formally specifies either a set (conditional) independence properties, or more fundamentally, a set of factorization properties.
- This is a crucial idea to understand: a graphical model is a set of constraints that all family members must obey.
- Graphical Models encode constraints by factorization requirements that all members of the family must obey.
- Factorization requirements are often identical to conditional independence requirements.
- Factorization, in general, allows sums to be distributed into products thereby making (exact) inference quantities more efficient than if factorization properties did not exist.

Graph Theory

- We'll define what we need as we go along.

Graph Theory

- We'll define what we need as we go along.
- Graph $G = (V, E)$ where V is set of nodes (or vertices) and $E \subseteq V \times V$ is a set of edges. If $i, j \in V$ then $(i, j) \in E$ means that nodes i and j are connected.



Graph Theory

- We'll define what we need as we go along.
- Graph $G = (V, E)$ where V is set of nodes (or vertices) and $E \subseteq V \times V$ is a set of edges. If $i, j \in V$ then $(i, j) \in E$ means that nodes i and j are connected.
- Nodes are in one-to-one correspondence to a set of random variables. For each $v \in V$ we have that X_v is a random variable (r.v.). X_V is the complete set of r.v.'s.

Graph Theory

- We'll define what we need as we go along.
- Graph $G = (V, E)$ where V is set of nodes (or vertices) and $E \subseteq V \times V$ is a set of edges. If $i, j \in V$ then $(i, j) \in E$ means that nodes i and j are connected.
- Nodes are in one-to-one correspondence to a set of random variables. For each $v \in V$ we have that X_v is a random variable (r.v.). X_V is the complete set of r.v.'s.
- A graphical model describes a family of distributions $p(x_V)$ over X_V.

Graphical Models

- A graphical model consists of a graph and a set of rules or properties $\mathcal{M}$ (often called *Markov properties*).

Graphical Models

- A graphical model consists of a graph and a set of rules or properties $\mathcal{M}$ (often called *Markov properties*).
- Unlike $\mathcal{U}$, which is the family of all distributions over n r.v.s, $\mathcal{F}(G, \mathcal{M}) \subseteq \mathcal{U}$ is a subset of distributions.

Graphical Models

- A graphical model consists of a graph and a set of rules or properties $\mathcal{M}$ (often called *Markov properties*).
- Unlike $\mathcal{U}$, which is the family of all distributions over n r.v.s, $\mathcal{F}(G, \mathcal{M}) \subseteq \mathcal{U}$ is a subset of distributions.
- Any member of $\mathcal{F}(G, \mathcal{M})$ must respect the constraints that are specified by the GM.

Graphical Models

- A graphical model consists of a graph and a set of rules or properties $\mathcal{M}$ (often called *Markov properties*).
- Unlike $\mathcal{U}$, which is the family of all distributions over n r.v.s, $\mathcal{F}(G, \mathcal{M}) \subseteq \mathcal{U}$ is a subset of distributions.
- Any member of $\mathcal{F}(G, \mathcal{M})$ must respect the constraints that are specified by the GM.
- Any distribution that does not respect even one of the GM's constraints is not a member of the family.

Graphical Models

- A graphical model consists of a graph and a set of rules or properties $\mathcal{M}$ (often called *Markov properties*).
- Unlike $\mathcal{U}$, which is the family of all distributions over n r.v.s, $\mathcal{F}(G, \mathcal{M}) \subseteq \mathcal{U}$ is a subset of distributions.
- Any member of $\mathcal{F}(G, \mathcal{M})$ must respect the constraints that are specified by the GM.
- Any distribution that does not respect even one of the GM's constraints is not a member of the family.
- In a GM, the constraints take the form of factorization (which are most often, conditional independence constraints).

Graphical Models

- A graphical model consists of a graph and a set of rules or properties $\mathcal{M}$ (often called *Markov properties*).
- Unlike $\mathcal{U}$, which is the family of all distributions over n r.v.s, $\mathcal{F}(G, \mathcal{M}) \subseteq \mathcal{U}$ is a subset of distributions.
- Any member of $\mathcal{F}(G, \mathcal{M})$ must respect the constraints that are specified by the GM.
- Any distribution that does not respect even one of the GM's constraints is not a member of the family.
- In a GM, the constraints take the form of factorization (which are most often, conditional independence constraints).
- Factorization is useful since it allows for the distributive law to enable the use of dynamic programming for much faster exact inference than naive.

Graphical Models

- A graphical model consists of a graph and a set of rules or properties $\mathcal{M}$ (often called *Markov properties*).
- Unlike $\mathcal{U}$, which is the family of all distributions over n r.v.s, $\mathcal{F}(G, \mathcal{M}) \subseteq \mathcal{U}$ is a subset of distributions.
- Any member of $\mathcal{F}(G, \mathcal{M})$ must respect the constraints that are specified by the GM.
- Any distribution that does not respect even one of the GM's constraints is not a member of the family.
- In a GM, the constraints take the form of factorization (which are most often, conditional independence constraints).
- Factorization is useful since it allows for the distributive law to enable the use of dynamic programming for much faster exact inference than naive.
- Finding best way of doing inference is entirely graph theoretic operation.

Graphical Model

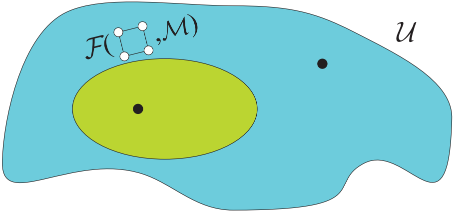
- Each type of graphical model requires a certain type of graph (e.g., undirected, or DAG) and a set of rules (or "Markov properties") to define the GM.
- A graphical model is a pair $(G, \mathcal{M}) = ((V, E), \mathcal{M})$, a graph G and a set of properties $\mathcal{M}$ that define what the graphical model means.
- Conceptually, one can think of a property $r \in \mathcal{M}$ is a predicate on a graph and a distribution, so $r(p, G) \in \{\text{true}, \text{false}\}$.
- $(G, \mathcal{M})$ consists of a family of distributions over x_V where all predicates hold. That is

$$\mathcal{F}(G, \mathcal{M}) = \{p : p \text{ is a distribution over } X_V \text{ and },\\ r(p, G) = \text{true}, \forall r \in \mathcal{M}\} \tag{1.8}$$

- $\mathcal{F}(G, \mathcal{M}) \subseteq \mathcal{U}$

Markov Properties

- Markov properties are rules that specify what are required of every family member. Any $p \in \mathcal{F}(G, \mathcal{M})$ satisfies all properties/rules $r \in \mathcal{M}$ for G. Any $p \in \mathcal{U} \setminus \mathcal{F}(G, \mathcal{M})$ violates at least one property for G.
- A $p \in \mathcal{U}$ might have more properties. $\mathcal{M}$ is like a filter, lets in those p that satisfy, but will let in those that satisfy more.



- Example $r \in \mathcal{M}$ might be "if there are two nodes $u, v \in V$ that are neither directly nor indirectly connected in G (i.e., there no path leading from u to v in G) then the corresponding random variables in p are independent"

Questions about Properties

Needing to be mathematically proven

- For a given type of graphical model, can the property set $\mathcal{M}$ be listed in finite space and computed efficiently? (answer, yes).
- For a given type of graphical model, are there more than one set of rules that define a family? In other words, are there rule sets $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\mathcal{F}(G, \mathcal{M}_1) = \mathcal{F}(G, \mathcal{M}_2)$ for all G? Answer, yes.
- Much of the Lauritzen 1996 book studies graphs, rules (or Markov properties) and proves when the corresponding families are either identical, or subsets of each other.

$$F(G, M) \supseteq F(G_2, M)$$

Questions about Properties (cont.)

Needing to be mathematically proven

- Is there a smallest rule set? In other words, are there rules sets $\mathcal{M}_1 \subset \mathcal{M}_2$ such that $\mathcal{F}(G, \mathcal{M}_1) = \mathcal{F}(G, \mathcal{M}_2)$, and can we compute the smallest set $\mathcal{M}'$ so that $\mathcal{F}(G, \mathcal{M}') = \mathcal{F}(G, \mathcal{M})$ where $|\mathcal{M}'|$ is minimal?

Questions about Properties (cont.)

Needing to be mathematically proven

- Is there a smallest rule set? In other words, are there rules sets $\mathcal{M}_1 \subset \mathcal{M}_2$ such that $\mathcal{F}(G, \mathcal{M}_1) = \mathcal{F}(G, \mathcal{M}_2)$, and can we compute the smallest set $\mathcal{M}'$ so that $\mathcal{F}(G, \mathcal{M}') = \mathcal{F}(G, \mathcal{M})$ where $|\mathcal{M}'|$ is minimal?
- Are there rule sets that are non-equivalent? I.e. $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\mathcal{F}(G, \mathcal{M}_1) \neq \mathcal{F}(G, \mathcal{M}_2)$ for some G? Answer, yes.

Questions about Properties (cont.)

Needing to be mathematically proven

- Is there a smallest rule set? In other words, are there rules sets $\mathcal{M}_1 \subset \mathcal{M}_2$ such that $\mathcal{F}(G, \mathcal{M}_1) = \mathcal{F}(G, \mathcal{M}_2)$, and can we compute the smallest set $\mathcal{M}'$ so that $\mathcal{F}(G, \mathcal{M}') = \mathcal{F}(G, \mathcal{M})$ where $|\mathcal{M}'|$ is minimal?
- Are there rule sets that are non-equivalent? I.e. $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\mathcal{F}(G, \mathcal{M}_1) \neq \mathcal{F}(G, \mathcal{M}_2)$ for some G? Answer, yes.
- In general, much of graphical model theory is regarding deducing properties of rules and corresponding properties of graphs and the distributions they represent. This allows us to reason about graphs as a means of reasoning about families of distributions.

A society of properties

- $\mathcal{G}_N$ = set of all undirected graphs over N nodes.
- Consider

$$\mathcal{F}_N(\mathcal{M}) = \cup_{G \in \mathcal{G}_N} \mathcal{F}(G, \mathcal{M}) \tag{1.9}$$

- and

$$\mathcal{F}(\mathcal{M}) = \cup_{N=1}^{\infty} \mathcal{F}_N(\mathcal{M}) \tag{1.10}$$

  family of all distributions over any number of random variables that obeys rules $\mathcal{M}$ for some undirected graph G.
- $\mathcal{M}$ determines the type of graphical model.
- $\mathcal{M}^{(\text{mrf})}$ rules for MRF, then $\mathcal{F}(\mathcal{M}^{(\text{mrf})})$ are the distributions representable by MRF.
- $\mathcal{M}^{(\text{bn})}$ rules for Bayesian network, then $\mathcal{F}(\mathcal{M}^{(\text{bn})})$ are the distributions representable by BN.

Different families

- Families may be different.
- For a given graph G, we might have neither $\mathcal{F}(G, \mathcal{M}^{(\text{mrf})}) \subset \mathcal{F}(G, \mathcal{M}^{(\text{bn})})$ nor $\mathcal{F}(G, \mathcal{M}^{(\text{bn})}) \subset \mathcal{F}(G, \mathcal{M}^{(\text{mrf})})$.
- The relationship for the family in its entirety might be different. I.e., when we compare the set of all MRFs vs. the set of all BNs, i.e., $\mathcal{F}(\mathcal{M}^{(\text{mrf})})$ vs. $\mathcal{F}(\mathcal{M}^{(\text{bn})})$.
- Large part of GM research is understanding these relationships.

What is graphical model inference?

- Inference, as we saw, is computing probabilistic queries such as:

What is graphical model inference?

- Inference, as we saw, is computing probabilistic queries such as:
  1. probability of evidence (marginalize the hidden variables)

$$p(\bar{x}_E) \tag{1.11}$$

What is graphical model inference?

- Inference, as we saw, is computing probabilistic queries such as:
  1. probability of evidence (marginalize the hidden variables)

$$p(\bar{x}_E) \tag{1.11}$$

  2. posterior probability, for $S \subseteq V \setminus E$ do

$$p(x_S|\bar{x}_E) \tag{1.12}$$

What is graphical model inference?

- Inference, as we saw, is computing probabilistic queries such as:
  1. probability of evidence (marginalize the hidden variables)
  $$p(\bar{x}_E) \tag{1.11}$$
  2. posterior probability, for $S \subseteq V \setminus E$ do
  $$p(x_S|\bar{x}_E) \tag{1.12}$$
  3. most probable assignment, for $S \subseteq V \setminus E$ do
  $$\operatorname*{argmax}_{x_S \in \mathcal{D}_{X_S}} p(x_S, \bar{x}_E). \tag{1.13}$$

What is graphical model inference?

- Inference, as we saw, is computing probabilistic queries such as:
  1. probability of evidence (marginalize the hidden variables)
  $$p(\bar{x}_E) \tag{1.11}$$
  2. posterior probability, for $S \subseteq V \setminus E$ do
  $$p(x_S|\bar{x}_E) \tag{1.12}$$
  3. most probable assignment, for $S \subseteq V \setminus E$ do
  $$\operatorname*{argmax}_{x_S \in \mathcal{D}_{X_S}} p(x_S, \bar{x}_E). \tag{1.13}$$
- Given a graph G, we want to derive this just based just on $(G, \mathcal{M})$ and derive this automatically.

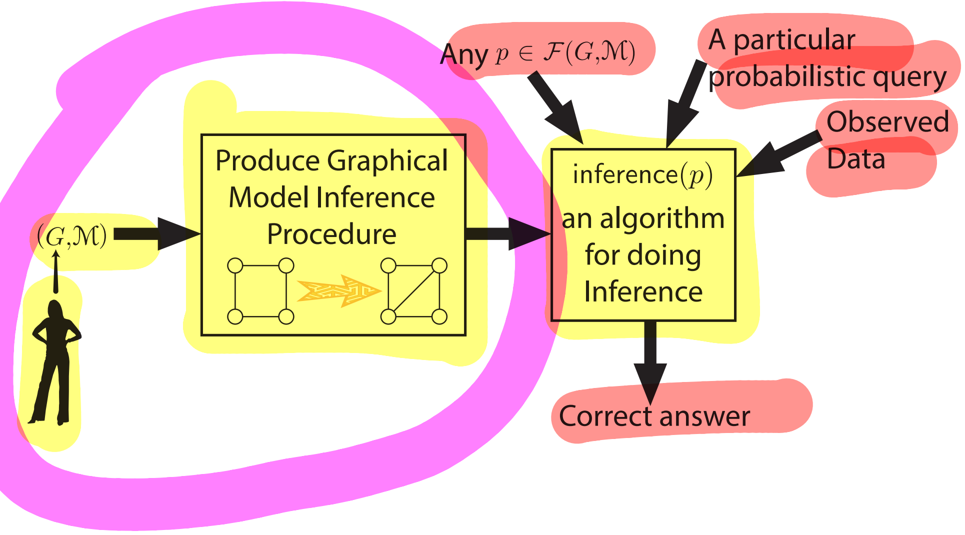
What is graphical model inference?

- Inference, as we saw, is computing probabilistic queries such as:
  1. probability of evidence (marginalize the hidden variables)
  $$p(\bar{x}_E) \tag{1.11}$$
  2. posterior probability, for $S \subseteq V \setminus E$ do
  $$p(x_S|\bar{x}_E) \tag{1.12}$$
  3. most probable assignment, for $S \subseteq V \setminus E$ do
  $$\operatorname*{argmax}_{x_S \in \mathcal{D}_{X_S}} p(x_S, \bar{x}_E). \tag{1.13}$$
- Given a graph G, we want to derive this just based just on $(G, \mathcal{M})$ and derive this automatically.
- We want to understand the computational complexity of the procedure based just on $(G, \mathcal{M})$.

What is graphical model inference?

- Inference, as we saw, is computing probabilistic queries such as:
  1. probability of evidence (marginalize the hidden variables)
  $$p(\bar{x}_E) \tag{1.11}$$
  2. posterior probability, for $S \subseteq V \setminus E$ do
  $$p(x_S|\bar{x}_E) \tag{1.12}$$
  3. most probable assignment, for $S \subseteq V \setminus E$ do
  $$\operatorname*{argmax}_{x_S \in \mathcal{D}_{X_S}} p(x_S, \bar{x}_E). \tag{1.13}$$
- Given a graph G, we want to derive this just based just on $(G, \mathcal{M})$ and derive this automatically.
- We want to understand the computational complexity of the procedure based just on $(G, \mathcal{M})$.
- amortization: we want to derive a procedure that works for any $p \in \mathcal{F}(G, R^{(\mathcal{M})})$ for a given rule set.

Graphical model inference diagrammatically



Markov random fields

- One type of graphical model (we'll study in this course).
- Has its origin in statistical physics (Boltzmann distributions, Ising models of atomic spin) and image processing (grid-based models).
- Example Ising model: Let $W = [w_{ij}]_{ij}$ be a matrix of weights. Note that many of these weights might be zero. Let $s = [s_i]_i = (s_1, s_2, \ldots, s_n)$ be a vector of binary random variables, $s_i \in \{-1, +1\}$. Define the "energy" as

$$E(s) = -\sum_{ij} s_i s_j w_{ij} \tag{1.14}$$

- Then define a distribution over s as

$$p(s) = \frac{1}{Z} \exp(-E(s)/T) \tag{1.15}$$

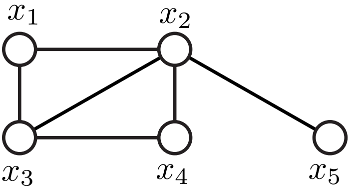
where T is the temperature of the model and $Z = \sum_s \exp(-E(s)/T)$ is a normalizing constant.

Markov random fields
Ising model (cont.)

- Most often s corresponds to a grid (i.e., s is really a matrix or 3D-matrix).
- Ising model: w_{ij} determines the interaction style of variables: if $w_{ij} = 0$ the no interaction. If $w_{ij} > 0$ then more probable for $s_i = s_j = \pm 1$. If $w_{ij} < 0$ then more probable for $s_i \neq s_j$.
- We can think of matrix W and vector s as a graph, $G = (V, E)$ where s corresponds to V and W corresponds to E — that is, $(i, j) = e \in E$ only when $w_{ij} \neq 0$.
- We might expect that any Ising model $p \in \mathcal{F}(G, \mathcal{M}^{(\mathrm{mrf})})$ for appropriately defined MRF rules.

Clique Factorization

- The "Cliques" of a graph $G = (V, E)$, or $\mathcal{C}(G)$, in a graph are the set of fully connected nodes.
- If $C \in \mathcal{C}(G)$ and $u, v \in C$ then $(u, v) \in E(G)$
- In the following graph



cliques are $\mathcal{C} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}, \{3, 4\}, \{2, 4\}, \{2, 3, 4\}, \{2, 5\}\}$.

Clique Factorization

- Given graph G with cliques $\mathcal{C}(G)$, consider a probability distribution that can be represented as follows:

$$p(x_V) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \phi_C(x_C) \tag{1.16}$$

$$Z = \sum_{x_V} \prod_{C \in \mathcal{C}} \phi_C(x_C) \tag{1.17}$$

- Actually, we don't always need Z explicitly since it is a constant and can be distributed into the factors in a variety of ways, leading to:

$$p(x_V) = \prod_{C \in \mathcal{C}(G)} \phi_C(x_C) \tag{1.18}$$

where only the factorization is depicted.

Clique Factorization

- More formally, consider the following family:

$$\mathcal{F}(G, \mathcal{M}^{(\text{cf})}) = \Big\{ p : \forall C \in \mathcal{C}(G), \exists \psi_C(x_C) \geq 0 \\ \text{and } p(x_V) = \prod_{C \in \mathcal{C}(G)} \psi_C(x_C) \Big\} \tag{1.19}$$

Clique Factorization

- More formally, consider the following family:

$$\mathcal{F}(G, \mathcal{M}^{(\text{cf})}) = \Big\{ p : \forall C \in \mathcal{C}(G), \exists \psi_C(x_C) \geq 0 \text{ and } p(x_V) = \prod_{C \in \mathcal{C}(G)} \psi_C(x_C) \Big\} \tag{1.19}$$

- are the clique factors unique?

MaxClique Factorization

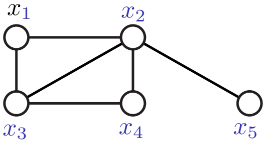
- The "MaxCliques" of a graph $G = (V, E)$, or $\mathcal{C}^{(mc)}(G)$, in a graph are the set of fully connected nodes that can't be enlarged

MaxClique Factorization

- The "MaxCliques" of a graph $G = (V, E)$, or $\mathcal{C}^{(mc)}(G)$, in a graph are the set of fully connected nodes that can't be enlarged — adding any node to a maxclique renders it no longer a clique.

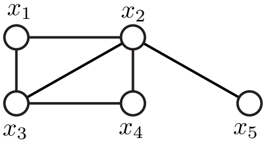
MaxClique Factorization

- The "MaxCliques" of a graph $G = (V, E)$, or $\mathcal{C}^{(mc)}(G)$, in a graph are the set of fully connected nodes that can't be enlarged — adding any node to a maxclique renders it no longer a clique.
- MaxCliques of previous graph (repeated below) are $\{\{1,2,3\},\{2,3,4\},\{2,5\}\}$



MaxClique Factorization

- The "MaxCliques" of a graph $G = (V, E)$, or $\mathcal{C}^{(mc)}(G)$, in a graph are the set of fully connected nodes that can't be enlarged — adding any node to a maxclique renders it no longer a clique.
- MaxCliques of previous graph (repeated below) are $\{\{1, 2, 3\}, \{2, 3, 4\}, \{2, 5\}\}$



- New properties $\mathcal{M}^{(\mathrm{mcf})}$ based on maxcliques define family

$$\mathcal{F}(G, \mathcal{M}^{(\mathrm{mcf})}) = \Big\{ p : \forall C \in \mathcal{C}^{(mc)}(G), \exists \psi_C(x_C) \geq 0 \\ \text{and } p(x_V) = \prod_{C \in \mathcal{C}^{(\mathrm{mc})}} \psi_C(x_C) \Big\} \tag{1.20}$$

Comparisons of families

- How do $\mathcal{F}(G, \mathcal{M}^{(\text{cf})})$ and $\mathcal{F}(G, \mathcal{M}^{(\text{mcf})})$ compare?

Comparisons of families

- How do $\mathcal{F}(G, \mathcal{M}^{(\mathsf{cf})})$ and $\mathcal{F}(G, \mathcal{M}^{(\mathsf{mcf})})$ compare?

Lemma 1.7.1

$\mathcal{F}(G, \mathcal{M}^{(cf)}) \subseteq \mathcal{F}(G, \mathcal{M}^{(mcf)})$

Comparisons of families

- How do $\mathcal{F}(G, \mathcal{M}^{(\mathsf{cf})})$ and $\mathcal{F}(G, \mathcal{M}^{(\mathsf{mcf})})$ compare?

Lemma 1.7.1

$\mathcal{F}(G, \mathcal{M}^{(cf)}) \subseteq \mathcal{F}(G, \mathcal{M}^{(mcf)})$

Lemma 1.7.2

$\mathcal{F}(G, \mathcal{M}^{(cf)}) \supseteq \mathcal{F}(G, \mathcal{M}^{(mcf)})$

Comparisons of families

- How do $\mathcal{F}(G, \mathcal{M}^{(\mathrm{cf})})$ and $\mathcal{F}(G, \mathcal{M}^{(\mathrm{mcf})})$ compare?

Lemma 1.7.1

$\mathcal{F}(G, \mathcal{M}^{(cf)}) \subseteq \mathcal{F}(G, \mathcal{M}^{(mcf)})$

Lemma 1.7.2

$\mathcal{F}(G, \mathcal{M}^{(cf)}) \supseteq \mathcal{F}(G, \mathcal{M}^{(mcf)})$

- Therefore

Corollary 1.7.3

$\mathcal{F}(G, \mathcal{M}^{(cf)}) = \mathcal{F}(G, \mathcal{M}^{(mcf)})$

- Since rules are identical, we use $\mathcal{M}^{(\mathrm{f})}$ for clique factorization, and family $\mathcal{F}(G, \mathcal{M}^{(\mathrm{f})})$.
- Often, it is not so obvious that different families are identical.
- Equally often, different families are indeed different.

Sources for Today's Lecture

- Most of this material comes from the reading handouts that will soon be made available.