



Probabilistic Graphical Models & Probabilistic AI

Ben Lengerich

Lecture 7: Project Ideas & Learning Generalized Linear Models

February 18, 2025

Reading: See course homepage



Logistics Reminders

- Project Proposal due **Friday, Feb 21** on Canvas
 - We will discuss this today.
- Congratulations, we've **completed the first module:**

Weeks	Lecture Dates	Topic	Assignments
Module 1: Foundations of PGMs, Exact Inference			
1-4	Jan 21- Feb 13	Course Introduction, Foundations of PGMs, Exact Inference	HWs 1, 2
4	Feb 13	Quiz	
Module 2: Learning			
5-9	Feb 18 - Mar 18	Parameter Learning, Structure Learning, Approximate Inference	HWs 3,4,5
9	Mar 20	Midterm Exam	
10	Mar 21 - Mar 30	Spring Recess	
Module 3: Modern Probabilistic AI			
11-14	Apr 1 - Apr 24	Deep Learning, LLMs from a GM perspective	Project Midway Report
15	Apr 29 - May 1	Project Presentations	Project Final Report



Today

- Project Ideas
- Learning Generalized Linear Models
 - Exponential Family: A Basic Building Block
 - Sufficient Statistics
 - MLE for GLMs

Project





Project Proposal

- **Updated due date: March 7, 2025**
- Recommend: Form teams of 2-4 students.
 - **Canvas discussion board for help forming teams.**
- Write a proposal (≤ 2 pages). Use [ICML Latex format](#). Include:
 - Project title + team members
 - Problem statement and motivation (1/2 page)
 - Literature review of at least 4 relevant papers
 - Description of dataset(s) and planned activities.
- Grading:
 - 40%: Clear and concise description of the project.
 - 40%: Quality of literature survey.
 - 10%: Feasibility and detail of activity plan.
 - 10%: Writing quality.



Project Expectations

- Do something **fun**.
- Do something **amazing**.
- Do something **novel**.

Retrofitting Word Vectors to Semantic Lexicons

Manaal Faruqui Jesse Dodge Sujay K. Jauhar
Chris Dyer Eduard Hovy Noah A. Smith

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

{mfaruqui, jessed, sjauhar, cdyer, ehovy, nasmith}@cs.cmu.edu

Won Best Student Paper @ NAACL 2015



3 Types of Projects

- Experiment
 - Can be a new model, a new application, or a reproduction.
 - Something needs to be **new (even for reproduction)**. Code, Dataset, Analysis, etc.
 - Everyone in class has Python experience.
- Theory
 - Analyze a model in a new way.
- Review
 - You may write a review paper of recent progress in a particular area of graphical models / probabilistic AI.
 - **Something needs to be new!** Insight, Synthesis, Opinions, Illustrative Figures, Animations, etc.



A few project ideas

Benchmarking Properties of PGMs

- What's the actual tradeoff of **exact vs approximate inference** algorithms? Compute time vs accuracy for different model complexity. Is this outdated with modern GPUs?
- Empirical complexity of **Bayesian Network structure learning**
 - In theory, structure learning is NP-hard, but how does runtime scale in real cases?
- When do **different model classes** perform better / worse? How does imposing structure affect our estimation and inference?
- Trade-off between **model complexity and inference time?** Sparse vs dense networks, etc.

Applications of Classical PGMs

- In biology, experiments are often performed with two cases -- Does this result in a **bias toward monotone effects** in biological knowledge bases? Do more expressive models fit the data but monotone models fit the knowledge bases better?
- Graphical models for **Stock Market Prediction** – Bayesian Networks, Time-varying networks, etc. Do model parameters change for industry sectors or over time?
- **Combining information** from multiple sources
 - **NLP**: Retrofitting embeddings from unstructured data to respect relationships in structured graphs
 - **Medicine**: How should we combine data from lab tests, patient history, and imaging results?
 - **Robotics**: How should we combine readings from multiple sensors?

PGMs + Modern Systems

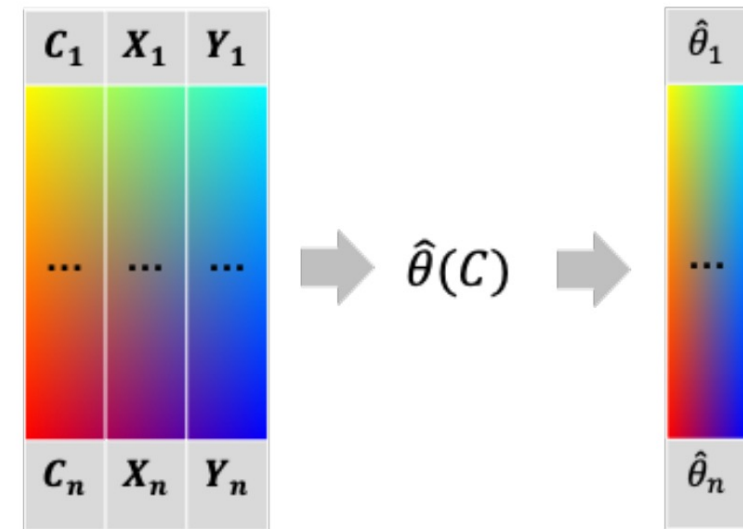
- Could we **store cliques** of graphs independently? Can we compress storage of cliques when we know we are storing many cliques? How do we efficiently pass messages between parts of graphs that may or may not be localized together?
- Can we **compress storage of Bayesian Networks**? What if we have 1000s of similar, but not quite identical, networks?
- Can we **parallelize exact inference** algorithms? Do modern GPUs built for NNs change anything about PGM inference?

Contextualized Graphical Models

- **Hierarchical Context:** Can the context have an internal hierarchy? Can this be learned as GM?
- Applications of Contextualized Learning to **Medical Problems:** Sarcoma is a rare form of cancer; can we learn model parameters in the context of other cancer types?
- **Analyzing Context-Specific Parameters:** Given a set of context-specific parameters, how do we understand and summarize the learned dynamics of the system?
- Contextualized **Differential Expression**
- Contextualized **Mendelian Randomization**

Contextualized Models:

- Parameters are functions of context
- Shares power between all samples



[\[Lengerich 2023\]](#)

Hybrid PGMs + DL Models

- Can we use **LLMs as probabilistic parameter generators**?
How could we combine estimation arising from data vs prior knowledge embedded in LLMs?
- Can we design a **PGM to optimize prompting** in soft-prompting of an LLM?
- Can **NNs approximate the behavior of a BN**? Can we extract BN parameters / structure from the a NN trained for a supervised task?
- **Graph Neural Networks** for Probabilistic Inference? Can we train a GNN to learn belief propagation test if it speeds up inference?

Benchmarking Probabilistic AI Properties

- **Uncertainty Calibration** in DL vs BNs: Are DL confidence scores poorly calibrated compared to PGMs?
- **Causal Inference:** Can PGMs give us causal interpretations of the internal activations of DL models?
- **Causal Discovery using LLMs:** Can LLMs help identify causal structures in a dataset by
- **LLMs as Data Scientists:** Can LLMs reliably interpret the meaning of PGM structures/parameters?
- **Probabilistic Reasoning in Diffusion Models:** Can PGMs help us interpret the uncertainty in diffusion-based generative models (e.g. Stable Diffusion)?

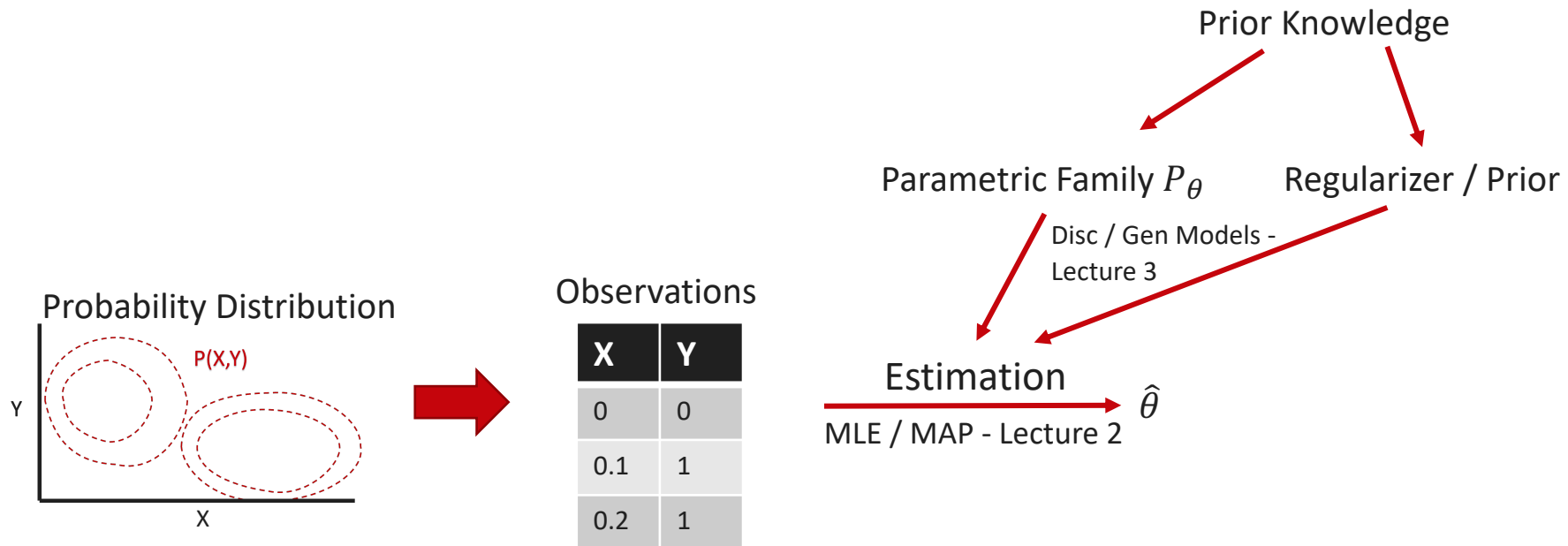
Apply Modern AI Techniques to PGMs

- **Adaptive Test-Time Compute:** Can we dynamically decide how much computation is needed for an inference query?
- **Post-training Pruning:** Can we prune unnecessary nodes or edges in a PGM after estimating parameters? Or can we even do this dynamically at time-time for an individual sample?
- **Mixture of experts:** Can we mix inference from PGMs of different classes?
- **Federated PGMs:** Suppose each “federated” PGM gets trained on a separate slice of the data. Can we combine estimation of parameters? Can we perform valid inference even if we don’t share any parameter information between model?

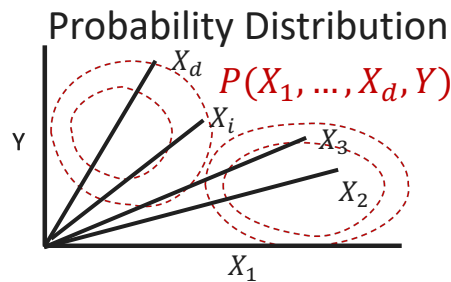


Learning Generalized Linear Models

A Brief Recap of our Roadmap

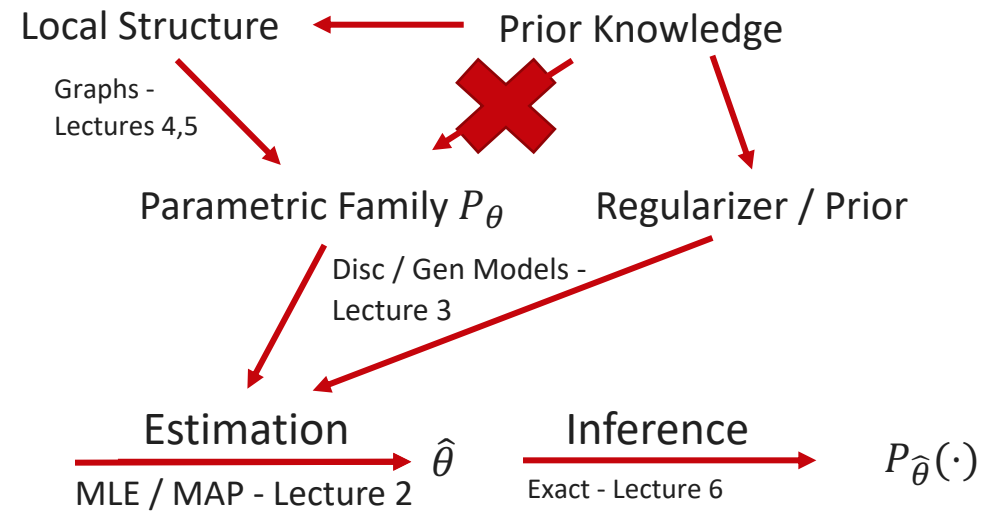


A Brief Recap of our Roadmap

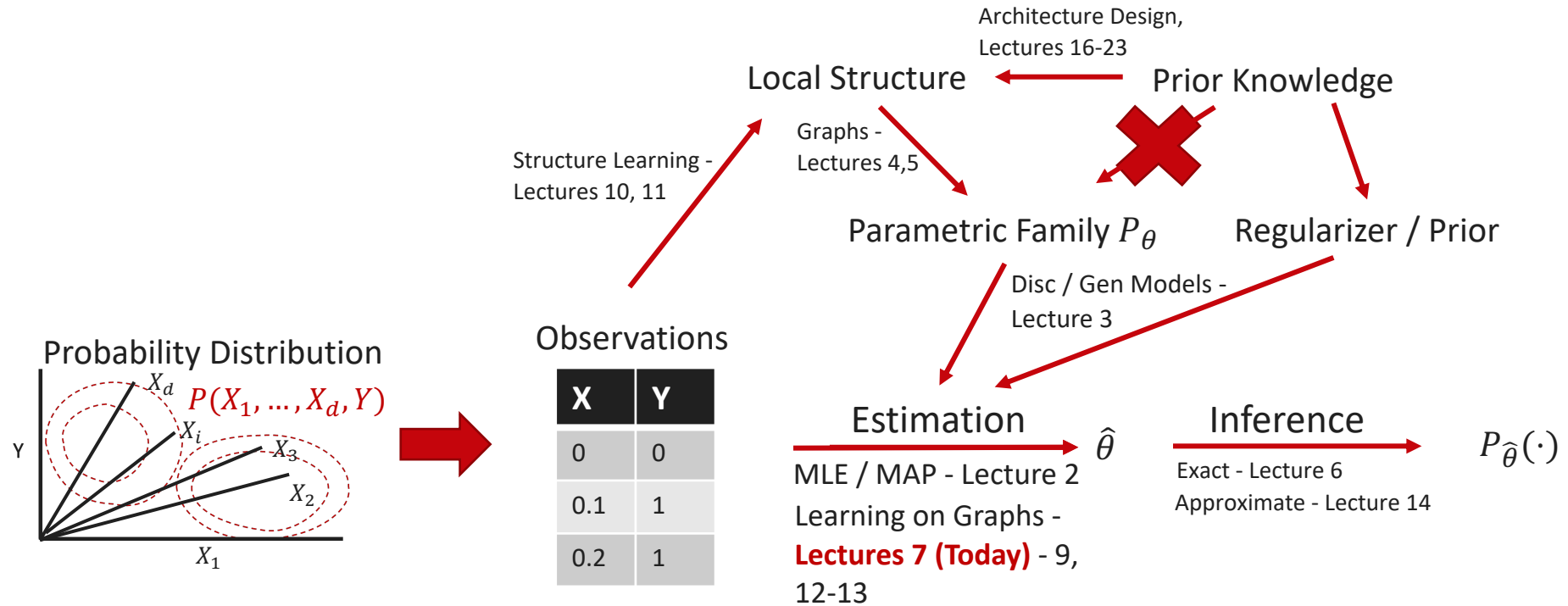


Observations

X	Y
0	0
0.1	1
0.2	1



A Brief Recap of our Roadmap



Two Regressions that we've seen so far

- Linear regression:

$$E[Y | X] = \theta^T X$$

- Logistic regression:

$$\log \frac{P(Y = 1 | X)}{1 - P(Y = 1 | X)} = \theta^T X$$

- Similarity here: We are assuming that a transformation of the conditional expectation is a linear function of X . i.e.

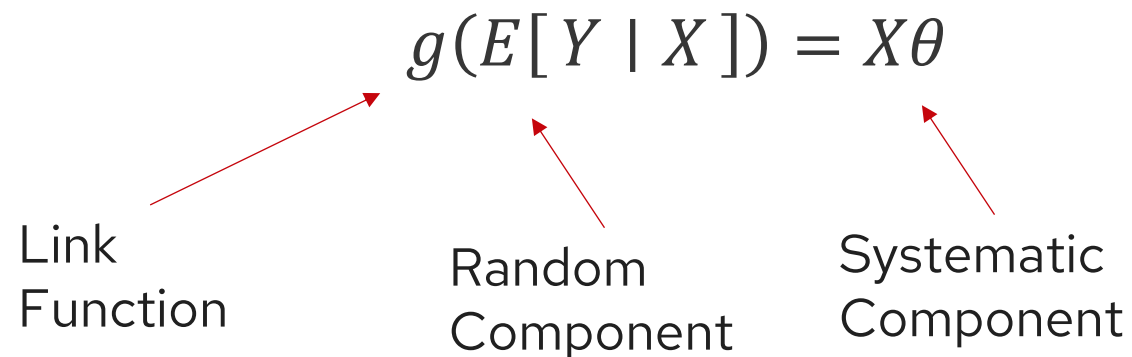
$$g(E(Y | X)) = \theta^T X$$

for some function g :

- Linear regression: $g(u) = u$
- Logistic regression: $g(u) = \log\left(\frac{u}{1-u}\right)$

Generalized Linear Models (GLMs)

- Defined by three components:
 - **Random Component:** Specifies distribution for $Y | X$
 - **Systematic Component:** Relates a parameter η to X
 - **Link Function:** Connects random and systematic components

$$g(E[Y | X]) = X\theta$$


The diagram illustrates the components of the GLM equation $g(E[Y | X]) = X\theta$. Three red arrows point from the labels below to the corresponding parts of the equation: 'Link Function' points to g , 'Random Component' points to $E[Y | X]$, and 'Systematic Component' points to $X\theta$.

Exponential Family: A Building Block

- For a numeric random variable X

$$p(x|\eta) = h(x) \exp(\eta^T T(x) - A(\eta)) = \frac{1}{Z(\eta)} h(x) \exp(\eta^T T(x))$$

is an **exponential family distribution** with natural (canonical) parameter η

- Function $T(x)$ is a *sufficient statistic*.
- Function $A(\eta) = \log Z(\eta)$ is the log normalizer
- Examples: Bernoulli, multinomial, Gaussian, Poisson, Gamma, Categorical

Why exponential family?

- Moment generating property

$$\begin{aligned}\frac{dA}{d\eta} &= \frac{d}{d\eta} \log Z(\eta) = \frac{1}{Z(\eta)} \frac{d}{d\eta} Z(\eta) \\ &= \frac{1}{Z(\eta)} \frac{d}{d\eta} \int h(x) \exp\{\eta^T T(x)\} dx \\ &= \int T(x) \frac{h(x) \exp\{\eta^T T(x)\}}{Z(\eta)} dx \\ &= E[T(x)]\end{aligned}$$
$$\begin{aligned}\frac{d^2 A}{d\eta^2} &= \int T^2(x) \frac{h(x) \exp\{\eta^T T(x)\}}{Z(\eta)} dx - \int T(x) \frac{h(x) \exp\{\eta^T T(x)\}}{Z(\eta)} dx \frac{1}{Z(\eta)} \frac{d}{d\eta} Z(\eta) \\ &= E[T^2(x)] - E^2[T(x)] \\ &= \text{Var}[T(x)]\end{aligned}$$

We can easily compute moments of any exponential family distribution by taking the derivatives of the log normalizer $A(\eta)$

MLE for Exponential Family

- For iid data the log-likelihood is

$$\begin{aligned}\ell(\eta; D) &= \log \prod_n h(x_n) \exp\{\eta^T T(x_n) - A(\eta)\} \\ &= \sum_n \log h(x_n) + \left(\eta^T \sum_n T(x_n) \right) - NA(\eta)\end{aligned}$$

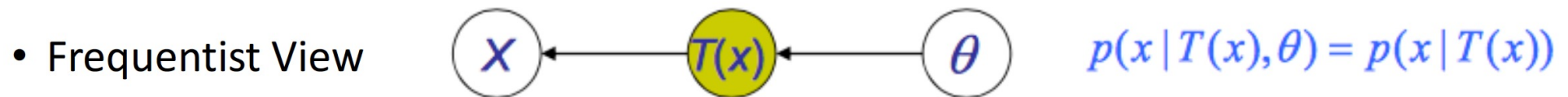
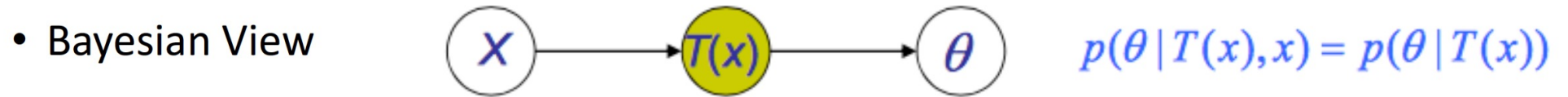
- We take the derivatives and set them to zero
- We perform **moment matching**

$$\begin{aligned}\frac{\partial \ell}{\partial \eta} &= \sum_n T(x_n) - N \frac{\partial A(\eta)}{\partial \eta} = 0 \\ \Rightarrow \frac{\partial A(\eta)}{\partial \eta} &= \frac{1}{N} \sum_n T(x_n) \\ \hat{\mu}_{MLE} &= \frac{1}{N} \sum_n T(x_n)\end{aligned}$$

- We can infer the canonical parameters using $\hat{\eta}_{MLE} = \psi(\hat{\mu}_{MLE})$

Sufficient Statistics

- For $p(x|\theta)$, $T(x)$ is **sufficient** for θ if there is no information in X regarding θ beyond that in $T(x)$
 - We can throw away X for the purpose of inference w.r.t. Θ



- Neyman factorization theorem
 - $T(x)$ is sufficient for θ if

$$p(x, T(x), \theta) = \psi_1(T(x), \theta) \psi_2(x, T(x))$$

$$\Rightarrow p(x | \theta) = g(T(x), \theta) h(x, T(x))$$

MLE for GLMs with natural response

- Log-likelihood $\ell = \sum_n \log h(y_n) + \sum_n (\theta^T x_n y_n - A(\eta_n))$

- Derivative of log-likelihood

$$\begin{aligned} \frac{d\ell}{d\theta} &= \sum_n \left(x_n y_n - \frac{dA(\eta_n)}{d\eta_n} \frac{d\eta_n}{d\theta} \right) \\ &= \sum_n (y_n - \mu_n) x_n \\ &= X^T (y - \mu) \end{aligned}$$

This is a fixed point function because μ is a function of θ

- Learning for canonical GLIMs

- Stochastic gradient ascent = least mean squares (LMS)

$$\theta^{t+1} = \theta^t + \rho (y_n - \mu_n^t) x_n$$

where $\mu_n^t = (\theta^t)^T x_n$ and ρ is a step size

Second-order methods

- The Hessian matrix

$$\begin{aligned} H &= \frac{d^2 \ell}{d\theta d\theta^T} = \frac{d}{d\theta^T} \sum_n (y_n - \mu_n) x_n = \sum_n x_n \frac{d\mu_n}{d\theta^T} \\ &= -\sum_n x_n \frac{d\mu_n}{d\eta_n} \frac{d\eta_n}{d\theta^T} \\ &= -\sum_n x_n \frac{d\mu_n}{d\eta_n} x_n^T \quad \text{since } \eta_n = \theta^T x_n \\ &= -X^T W X \end{aligned}$$

- X is the design matrix and W is computed by calculating the 2-nd derivative of $A(\eta_n)$

$$W = \text{diag} \left(\frac{d\mu_1}{d\eta_1}, \dots, \frac{d\mu_N}{d\eta_N} \right)$$

Iteratively Reweighted Least Squares (IRLS)

- Newton-Raphson methods with objective J $\theta^{t+1} = \theta^t - H^{-1} \nabla_{\theta} J$

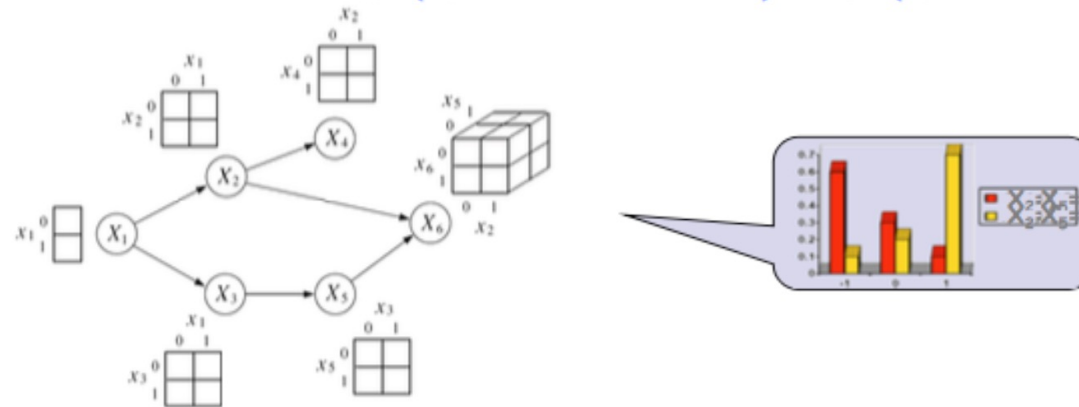
- We have $\nabla_{\theta} J = X^T (y - \mu)$
 $H = -X^T W X$

- Update $\theta^{t+1} = \theta^t + H^{-1} \nabla_{\theta} \ell$
 $= (X^T W^t X)^{-1} [X^T W^t X \theta^t + X^T (y - \mu^t)]$
 $= (X^T W^t X)^{-1} X^T W^t z^t$

MLE for General BNs

- If we assume the parameters for each CPD are globally independent, and all nodes are fully observed, then the log-likelihood function decomposes into a sum of local terms, one per node

$$\ell(\theta; D) = \log p(D | \theta) = \log \prod_n \left(\prod_i p(x_{n,i} | \mathbf{x}_{n,\pi_i}, \theta_i) \right) = \sum_i \left(\sum_n \log p(x_{n,i} | \mathbf{x}_{n,\pi_i}, \theta_i) \right)$$



- MLE-based parameter estimation of GM reduces to local est. of each GLIM.

Questions?

