CS 229 Lecture Fifteen
Weakly Supervised Learning: Graphical Models and Method of Moments

Chris Ré

May 21, 2023

Topics for Today

- We'll discuss weak supervision and method of moments.
  - Used in crowd workers, new programming models like Snorkel, used in Google, and set off the *"Data-centric AI"* movement.
  - Oddly, you've likely used a product that has it today!
- We'll introduce the basics of graphical models.
- We'll introduce covariance-like matrices for discrete models.

Topics for Today

- We'll discuss weak supervision and method of moments.
  - Used in crowd workers, new programming models like Snorkel, used in Google, and set off the *"Data-centric AI"* movement.
  - Oddly, you've likely used a product that has it today!
- We'll introduce the basics of graphical models.
- We'll introduce covariance-like matrices for discrete models.

I'm giving a minimal practical material. I have talks about that online.[a] Today, we are focused on technical material. It's not an accident this comes after the midterm...

[a] Here is one: `https://youtu.be/k20oLegpDW8?t=4966`

Simple Motivating Example

You have some data $x^{(1)}, \ldots, x^{(n)}$, and you ask for labels from the crowd $y^{(1)}, \ldots, y^{(n)}$. How do you do it?

- **Observation**. Labelers have different accuracies per task.
  - Some labelers are better, more familiar, or have expertise with the task.
  - Some labelers maybe are spammers–may be random or intentionally inaccurate.
  - Labelers don't look at all the items

Simple Motivating Example

You have some data $x^{(1)}, \dots, x^{(n)}$, and you ask for labels from the crowd $y^{(1)}, \dots, y^{(n)}$. How do you do it?

- **Observation**. Labelers have different accuracies per task.
  - Some labelers are better, more familiar, or have expertise with the task.
  - Some labelers maybe are spammers–may be random or intentionally inaccurate.
  - Labelers don't look at all the items *we'll come back to this later.*

Simple Motivating Example

You have some data $x^{(1)}, \ldots, x^{(n)}$, and you ask for labels from the crowd $y^{(1)}, \ldots, y^{(n)}$. How do you do it?

- **Observation**. Labelers have different accuracies per task.
  - Some labelers are better, more familiar, or have expertise with the task.
  - Some labelers maybe are spammers–may be random or intentionally inaccurate.
  - Labelers don't look at all the items *we'll come back to this later.*
- **Idea**: Identify the reliability of *each labeler* for the task, and use that to compute how likely their vote is correct.

This is a great framework with long history back to the (Dawid-Skene 1979). Originally done with EM!

More complex systems: Briefly

Transformer era: Majority of time was building training data–not the model: realized data preparation for AI was more like purifying a sewer. Used to clean data *to feed to* deep learning.

More complex systems: Briefly

Transformer era: Majority of time was building training data–not the model: realized data preparation for AI was more like purifying a sewer. Used to clean data *to feed to* deep learning.

`Snorkel.ai` pioneered (when they were students!) treating many different sources of information: crowd labels, programmatic labels, old models, etc.–and maximally purifying them.

More complex systems: Briefly

Transformer era: Majority of time was building training data–not the model: realized data preparation for AI was more like purifying a sewer. Used to clean data *to feed to* deep learning.

`Snorkel.ai` pioneered (when they were students!) treating many different sources of information: crowd labels, programmatic labels, old models, etc.–and maximally purifying them.

Technical Challenge: sources may have correlations!

Technical Overview in Two Stages

- Independent Labelers: A simple “method of moments” that captures the crowd.
- Correlations Labelers: We’ll see the basics of graphical models.

These are *latent variable models* of the type we’ve seen, but we’ll solve in a different way (not EM). Stronger guarantees!

Independent case, No Abstains

- **Given** Data points $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ and labeling functions $\lambda_1, \ldots, \lambda_m$ where $\lambda_j \in \mathbb{R}^d \to \{-1, 1\}$ for $j = 1, \ldots, m$.
- **Do** Find $y^{(1)}, \ldots, y^{(n)} \in \mathbb{R}$

$$P(y^{(i)} \mid \lambda_1, \ldots, \lambda_m, x^{(i)}) \text{ for } i = 1, \ldots, n$$

Independent case, No Abstains

- **Given** Data points $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ and labeling functions $\lambda_1, \ldots, \lambda_m$ where $\lambda_j \in \mathbb{R}^d \to \{-1, 1\}$ for $j = 1, \ldots, m$.
- **Do** Find $y^{(1)}, \ldots, y^{(n)} \in \mathbb{R}$

$$P(y^{(i)} \mid \lambda_1, \ldots, \lambda_m, x^{(i)}) \text{ for } i = 1, \ldots, n$$

- **Model** We imagine that the voter **agrees** with the ground truth, but we don't know *how often*

$$P\left(\lambda_j(x^{(i)}) = y^{(i)} \mid y^{(i)}\right) \sim \text{Bernoulli}(\alpha_j).$$

The challenge is that $y^{(i)}$ is latent: we do not observe its value: we only observe voters $\lambda_j(x^{(i)})$ on points.

.

Let's unpack the Model

$$P\left(\lambda_j(x^{(i)}) = y^{(i)} \mid y^{(i)}\right) \sim \text{Bernoulli}(\alpha_j).$$

- Each labeler has a hidden accuracy $\alpha_j \neq 0.5$. We say that a labeler is **informative** if $\alpha_j \neq 0.5$.
- This means, given a data point, the labeler λ_j returns the correct label with probability α_j and flips the label with probability $1 - \alpha_j$. So for a data point x and label y.

$$\lambda(x) = y \text{ with probability } \alpha_j$$
$$\lambda(x) = -y \text{ with probability } 1 - \alpha_j$$

A much bigger challenge is that we don't observe $y \in \{-1, 1\}$ but want to estimate α_j.

Warmup: Observable y

Data	λ_1	λ_2	λ_3	$\dots$	λ_m	y
$x^{(1)}$	1	1	1	$\dots$	1	1
$x^{(2)}$	-1	-1	-1	$\dots$	1	-1
$x^{(3)}$	-1	1	-1	$\dots$	1	-1
$\vdots$						
$x^{(n)}$	-1	-1	-1	$\dots$	-1	-1

Suppose we saw $y^{(1)}, \dots, y^{(n)}$, how do we estimate α_j?

Warmup: Observable y

Data	λ_1	λ_2	λ_3	$\dots$	λ_m	y
$x^{(1)}$	1	1	1	$\dots$	1	1
$x^{(2)}$	-1	-1	-1	$\dots$	1	-1
$x^{(3)}$	-1	1	-1	$\dots$	1	-1
$\vdots$						
$x^{(n)}$	-1	-1	-1	$\dots$	-1	-1

Suppose we saw $y^{(1)}, \dots, y^{(n)}$, how do we estimate α_j? Define β_j

$$\beta_j = \mathbb{E}[\lambda_j(x)y] = (1)\alpha_j + (-1)(1-\alpha_j) = 2\alpha_j - 1$$

Note the expectation ranges over the choice of data point *and* the randomness in the model. That is, we can estimate β_j as

$$\mathbb{E}[\lambda_j(x)y] \approx \frac{1}{n}\sum_{i=1}^{n} \lambda_j(x^{(i)})y^{(i)} \text{ so } \alpha_j = \frac{1+\beta_j}{2}$$

Warmup: Observable y

Data	λ_1	λ_2	λ_3	$\ldots$	λ_m	y
$x^{(1)}$	1	1	1	$\ldots$	1	1
$x^{(2)}$	-1	-1	-1	$\ldots$	1	-1
$x^{(3)}$	-1	1	-1	$\ldots$	1	-1
$\vdots$						
$x^{(n)}$	-1	-1	-1	$\ldots$	-1	-1

Suppose we saw $y^{(1)}, \ldots, y^{(n)}$, how do we estimate α_j? Define β_j

$$\beta_j = \mathbb{E}[\lambda_j(x)y] = (1)\alpha_j + (-1)(1 - \alpha_j) = 2\alpha_j - 1$$

Note the expectation ranges over the choice of data point *and* the randomness in the model. That is, we can estimate β_j as

$$\mathbb{E}[\lambda_j(x)y] \approx \frac{1}{n}\sum_{i=1}^{n} \lambda_j(x^{(i)})y^{(i)} \text{ so } \alpha_j = \frac{1+\beta_j}{2}$$

But we do *not* observe y, so we cannot compute $\mathbb{E}[\lambda_i(x)y]$. What can we do?

What do we observe?

Data	λ_1	λ_2	λ_3	$\ldots$	λ_m	Y
$x^{(1)}$	1	1	1	$\ldots$	1	1
$x^{(2)}$	-1	-1	-1	$\ldots$	1	-1
$x^{(3)}$	-1	1	-1	$\ldots$	1	-1
$\vdots$						$\vdots$
$x^{(n)}$	-1	-1	-1	$\ldots$	-1	-1

- An **EM** idea: (1) We estimate the latent values $y^{(i)}$ for $i = 1, \ldots, n$, and (2) run the previous estimation procedure.

What do we observe?

Data	λ_1	λ_2	λ_3	$\dots$	λ_m	Y
$x^{(1)}$	1	1	1	$\dots$	1	1
$x^{(2)}$	-1	-1	-1	$\dots$	1	-1
$x^{(3)}$	-1	1	-1	$\dots$	1	-1
$\vdots$						$\vdots$
$x^{(n)}$	-1	-1	-1	$\dots$	-1	-1

- An **EM** idea: (1) We estimate the latent values $y^{(i)}$ for $i = 1, \dots, n$, and (2) run the previous estimation procedure.
- We'll use an alternate approach–closer to (Robust) PCA.
  - Stronger guarantees and more information about the solution!

What do we observe?

Data	λ_1	λ_2	λ_3	...	λ_m	Y
$x^{(1)}$	1	1	1	...	1	1
$x^{(2)}$	-1	-1	-1	...	1	-1
$x^{(3)}$	-1	1	-1	...	1	-1
$\vdots$						$\vdots$
$x^{(n)}$	-1	-1	-1	...	-1	-1

We do observe all the votes for $\lambda_j(x^{(i)})$. Since $\lambda_j(x) \in \{-1, 1\}$:

$$\mathbb{E}[\lambda_j(x)\lambda_j(x)] = \mathbb{E}[\lambda_j(x)^2] = 1 \text{ for } j = 1, \ldots, m$$

What do we observe?

Data	λ_1	λ_2	λ_3	$\dots$	λ_m	Y
$x^{(1)}$	1	1	1	$\dots$	1	1
$x^{(2)}$	-1	-1	-1	$\dots$	1	-1
$x^{(3)}$	-1	1	-1	$\dots$	1	-1
$\vdots$						$\vdots$
$x^{(n)}$	-1	-1	-1	$\dots$	-1	-1

We do observe all the votes for $\lambda_j(x^{(i)})$. Since $\lambda_j(x) \in \{-1, 1\}$:

$$\mathbb{E}[\lambda_j(x)\lambda_j(x)] = \mathbb{E}[\lambda_j(x)^2] = 1 \text{ for } j = 1, \dots, m$$

In the case $j \neq k$, we have

$$\begin{aligned}\mathbb{E}[\lambda_j(x)\lambda_k(x)] =& (1)\left(\alpha_j\alpha_k + (1-\alpha_j)(1-\alpha_k)\right) && \text{agree on the label}\\ &+(-1)\left(\alpha_j(1-\alpha_k) + (1-\alpha_j)\alpha_k\right) && \text{disagree on the label}\end{aligned}$$

What do we observe?

Data	λ_1	λ_2	λ_3	...	λ_m	Y
$x^{(1)}$	1	1	1	...	1	1
$x^{(2)}$	-1	-1	-1	...	1	-1
$x^{(3)}$	-1	1	-1	...	1	-1
$\vdots$						$\vdots$
$x^{(n)}$	-1	-1	-1	...	-1	-1

We do observe all the votes for $\lambda_j(x^{(i)})$. Since $\lambda_j(x) \in \{-1, 1\}$:

$$\mathbb{E}[\lambda_j(x)\lambda_j(x)] = \mathbb{E}[\lambda_j(x)^2] = 1 \text{ for } j = 1, \ldots, m$$

In the case $j \neq k$, we have

$$\begin{aligned}
\mathbb{E}[\lambda_j(x)\lambda_k(x)] =& (1)\left(\alpha_j\alpha_k + (1-\alpha_j)(1-\alpha_k)\right) && \text{agree on the label} \\
& + (-1)\left(\alpha_j(1-\alpha_k) + (1-\alpha_j)\alpha_k\right) && \text{disagree on the label} \\
=& (2\alpha_j - 1)(2\alpha_k - 1) = \beta_j\beta_k
\end{aligned}$$

Make an observed matrix

Summarizing what we just derived more succinctly

$$\mathbb{E}[\lambda_j(x)\lambda_k(x)] = \begin{cases} 1 & j = k \\ \beta_j\beta_k & \text{o.w.} \end{cases}$$

Make an observed matrix

Summarizing what we just derived more succinctly

$$\mathbb{E}[\lambda_j(x)\lambda_k(x)] = \begin{cases} 1 & j = k \\ \beta_j \beta_k & \text{o.w.} \end{cases}$$

Form the observed matrix O

$$O_{j,k} = \mathbb{E}\left[\lambda_j(x)\lambda_k(x)\right].$$

This matrix intuitively tracks the disagreements and agreements, and importantly we *can estimate O from data.* Recall we have:

$$O_{j,k} \approx \frac{1}{n}\sum_{i=1}^{n} \lambda_j(x^{(i)})\lambda_k(x^{(i)}).$$

An Extremely Simple Algorithm

We make a very simple observation for distinct indexes j, k, m, i.e., $j \neq k, j \neq m,$ and $k \neq m$:

$$O_{j,k} O_{k,m} = (\beta_j \beta_k)(\beta_k \beta_m) = \beta_j \beta_k^2 \beta_m.$$

And so we can form the estimate:

$$\frac{O_{j,k} O_{k,m}}{O_{j,m}} = \beta_k^2.$$

- For any k, any distinct (j, m) form an estimate for β_k^2.
  - They are consistent in moments (in practice, take median)

An Extremely Simple Algorithm

We make a very simple observation for distinct indexes j, k, m, i.e., $j \neq k, j \neq m$, and $k \neq m$:

$$O_{j,k} O_{k,m} = (\beta_j \beta_k)(\beta_k \beta_m) = \beta_j \beta_k^2 \beta_m.$$

And so we can form the estimate:

$$\frac{O_{j,k} O_{k,m}}{O_{j,m}} = \beta_k^2.$$

- For any k, any distinct (j, m) form an estimate for β_k^2.
  - They are consistent in moments (in practice, take median)
- **Note** We recovered the *magnitude* but not the *sign* of β_k.

An Extremely Simple Algorithm: More about Signs

Using our observation, we have formed estimates:

$$\frac{O_{j,k} O_{k,m}}{O_{j,m}} = \beta_k^2.$$

We have recovered the *magnitude* but not the *sign* of β_k. But there is structure in the signs...

An Extremely Simple Algorithm: More about Signs

Using our observation, we have formed estimates:

$$\frac{O_{j,k}O_{k,m}}{O_{j,m}} = \beta_k^2.$$

We have recovered the *magnitude* but not the *sign* of β_k. But there is structure in the signs...

- Suppose we knew β_j for *even a single j*

$$\text{sign}(O_{j,k}) = \text{sign}(\beta_j)\text{sign}(\beta_k).$$

We know $\text{sign}(\beta_k)$ for $k = 1, \ldots, m$.

An Extremely Simple Algorithm: More about Signs

Using our observation, we have formed estimates:

$$\frac{O_{j,k}O_{k,m}}{O_{j,m}} = \beta_k^2.$$

We have recovered the *magnitude* but not the *sign* of β_k. But there is structure in the signs...

- Suppose we knew β_j for *even a single j*

$$\text{sign}(O_{j,k}) = \text{sign}(\beta_j)\text{sign}(\beta_k).$$

  We know $\text{sign}(\beta_k)$ for $k = 1, \ldots, m$.
- If β is a solution, then $-\beta$ is too–and this is a *real symmetry*.
  - Could assume $\beta_j > 0$ for $j = 1, \ldots, m$ – but this would mean no scammers!

An Extremely Simple Algorithm: More about Signs

Using our observation, we have formed estimates:

$$\frac{O_{j,k}O_{k,m}}{O_{j,m}} = \beta_k^2.$$

We have recovered the *magnitude* but not the *sign* of β_k. But there is structure in the signs...

- Suppose we knew β_j for *even a single j*

$$\text{sign}(O_{j,k}) = \text{sign}(\beta_j)\text{sign}(\beta_k).$$

  We know $\text{sign}(\beta_k)$ for $k = 1, \ldots, m$.
- If β is a solution, then $-\beta$ is too–and this is a *real symmetry*.
  - Could assume $\beta_j > 0$ for $j = 1, \ldots, m$ – but this would mean no scammers!
  - Could assume we *knew* one labeler was good, e.g., $\beta_1 > 0$.

An Extremely Simple Algorithm: More about Signs

Using our observation, we have formed estimates:

$$\frac{O_{j,k}O_{k,m}}{O_{j,m}} = \beta_k^2.$$

We have recovered the *magnitude* but not the *sign* of β_k. But there is structure in the signs...

- Suppose we knew β_j for *even a single j*

$$\text{sign}(O_{j,k}) = \text{sign}(\beta_j)\text{sign}(\beta_k).$$

  We know $\text{sign}(\beta_k)$ for $k = 1, \ldots, m$.
- If β is a solution, then $-\beta$ is too–and this is a *real symmetry*.
  - Could assume $\beta_j > 0$ for $j = 1, \ldots, m$ – but this would mean no scammers!
  - Could assume we *knew* one labeler was good, e.g., $\beta_1 > 0$.
  - Could assume $\sum_j \beta_j > 0$, "not adversarial–on average–with the ground truth."

Is it Unique?

We have a β, is it unique up to this sign?

$$\frac{O_{j,k} O_{k,m}}{O_{j,m}} = \beta_k^2$$

Assume for the moment $O_{j,k} \neq 0$. Taking log absolute value for each (j, k, m) such that $j < k < m$ we have a non-redundant constraint:

$$\frac{1}{2} \left(\log |O_{i,k}| + \log |O_{k,m}| - \log |O_{j,m}| \right) = \log |\beta_k|$$

Is it Unique?

We have a β, is it unique up to this sign?

$$\frac{O_{j,k}O_{k,m}}{O_{j,m}} = \beta_k^2$$

Assume for the moment $O_{j,k} \neq 0$. Taking log absolute value for each (j,k,m) such that $j < k < m$ we have a non-redundant constraint:

$$\frac{1}{2}\left(\log|O_{i,k}| + \log|O_{k,m}| - \log|O_{j,m}|\right) = \log|\beta_k|$$

We can write this as a linear system. Let's look at $m = 3$:

$$\frac{1}{2}\underbrace{\begin{pmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix}}_{A}\begin{pmatrix} \log|O_{1,2}| \\ \log|O_{1,3}| \\ \log|O_{2,3}| \end{pmatrix} = \begin{pmatrix} \log|\beta_1| \\ \log|\beta_2| \\ \log|\beta_3| \end{pmatrix}$$

Note that A has full rank–and we can check this in higher dimensions. We need at least *three* voters–but then unique!

What does it mean if $O_{j,k} = 0$?

If $O_{j,k} = 0$ then either $\beta_j = 0$ or $\beta_k = 0$. Say $\beta_j = 0$,

$$\mathbb{E}[\lambda_j(x)Y] = 0 \implies \alpha_j = 0.5$$

That is, labeler j is not informative. That is, λ_i's labels are noise; they are indistinguishable from flipping a coin without looking at the data.

This is sensible: Adding labelers that just flip coins without looking at the data *shouldn't* give us any information!

Summary of Independent Case

We can estimate the *accuracy* of labelers without access to *any* ground truth. Instead, we examine the agreement rates.

- We require at least 3 informative labelers (amazing to me that we didn't need more!)
- There was a fundamental symmetry: β and $-\beta$ are solutions.
- Nevertheless, we were able to say *"this converges to the global optimal"* solution (upto symmetries) cf. EM guarantees.

Correlations and Graphical Models

What if Labelers are Correlated?

- In more advanced applications, we may label our data using previous models to label the data, rule-based systems, or experts with the same or dissimilar expertise.
- The labels produced my have *correlated* errors.
- Let's start to get a simple example of correlations and introduce graphical models.

Nugget: Covariance Matrices and Graphs

Let's consider the a very simple model:

$$
\begin{aligned}
x_1 \sim& \mathcal{N}(0,1) \\
x_2 =& x_1 + \epsilon_2 \text{ with } \epsilon_2 \sim \mathcal{N}(0,1) \quad \text{or } x_2 \sim \mathcal{N}(x_1,1) \\
x_3 =& x_2 + \epsilon_3 \text{ with } \epsilon_3 \sim \mathcal{N}(0,1) \quad \text{or } x_3 \sim \mathcal{N}(x_1,1)
\end{aligned}
$$

Notice x_2 and x_3 are correlated–not independent–but they are *conditionally* independent based on the value of x_1.

Nugget: Covariance Matrices and Graphs

Let's consider the a very simple model:

$$\begin{aligned} x_1 &\sim \mathcal{N}(0,1) \\ x_2 &= x_1 + \epsilon_2 \text{ with } \epsilon_2 \sim \mathcal{N}(0,1) \quad \text{or } x_2 \sim \mathcal{N}(x_1, 1) \\ x_3 &= x_2 + \epsilon_3 \text{ with } \epsilon_3 \sim \mathcal{N}(0,1) \quad \text{or } x_3 \sim \mathcal{N}(x_1, 1) \end{aligned}$$

Notice x_2 and x_3 are correlated–not independent–but they are *conditionally* independent based on the value of x_1.

Let's compute some statistics!

$$\mathbb{E}[x_1] = 0 \text{ and } \mathbb{E}[x_j] = \mathbb{E}[x_1] + \mathbb{E}[\epsilon_j] = 0 \text{ for } j \in \{2, 3\}$$

The interesting one is the covariance...

For the model above, $x_1 \sim \mathcal{N}(0,1), x_j \sim \mathcal{N}(x_1, 1)$ for $j \in \{2,3\}$. The covariance, Σ has the following form:

$$\mathbb{E}[xx^T] = \Sigma = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

We record the computation:

$$\begin{aligned} \mathbb{E}[x_1^2] =& 0 \\ \mathbb{E}[x_2^2] =& \mathbb{E}[(x_1 + \epsilon_2)^2] = \mathbb{E}[x_1^2] + \mathbb{E}[\epsilon_2^2] + 2\mathbb{E}[x_1\epsilon_2] = 2 \\ \mathbb{E}[x_1 x_2] =& \mathbb{E}[x_1^2] + \mathbb{E}[x_1\epsilon_2] = 1 \\ \mathbb{E}[x_2 x_3] =& \mathbb{E}[x_1^2] + \mathbb{E}[x_1(\epsilon_2 + \epsilon_3)] + \mathbb{E}[\epsilon_2\epsilon_3] = 1 \end{aligned}$$

For the model above, $x_1 \sim \mathcal{N}(0,1), x_j \sim \mathcal{N}(x_1, 1)$ for $j \in \{2,3\}$. The covariance, Σ has the following form:

$$\mathbb{E}[xx^T] = \Sigma = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

We record the computation:

$$\begin{aligned} \mathbb{E}[x_1^2] =& 0 \\ \mathbb{E}[x_2^2] =& \mathbb{E}[(x_1 + \epsilon_2)^2] = \mathbb{E}[x_1^2] + \mathbb{E}[\epsilon_2^2] + 2\mathbb{E}[x_1 \epsilon_2] = 2 \\ \mathbb{E}[x_1 x_2] =& \mathbb{E}[x_1^2] + \mathbb{E}[x_1 \epsilon_2] = 1 \\ \mathbb{E}[x_2 x_3] =& \mathbb{E}[x_1^2] + \mathbb{E}[x_1(\epsilon_2 + \epsilon_3)] + \mathbb{E}[\epsilon_2 \epsilon_3] = 1 \end{aligned}$$

Pretty disappointing! Σ doesn't seem to have structure that matches the graph... huh.

The inverse reveals some structure . . .

For the model above, $x_1 \sim \mathcal{N}(0, 1), x_j \sim \mathcal{N}(x_1, 1)$ for $j \in \{2, 3\}$.

$$\Sigma = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \text{ then } \Sigma^{-1} = \begin{pmatrix} 3 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Wow! If we draw the graph, then the lack of edges precisely matches where those 0's show up . . . we need some notation to see if this generalizes.

Probability Distributions and Graphs

A probability distribution $p : \mathbb{R}^d \to [0, 1]$ factors with respect to a graph $G = (V, E)$ if

$$p(x) = c_0 \prod_{e=(x_i,x_j)\in E} p_e(x_i, x_j) \prod_{x_i \in V} p_{x_i}(x_i).$$

Factoring and Gaussians

Suppose a Gaussian model factors via a graph $G = (V, E)$. Let $A = \Sigma^{-1}$ just for notation then:

$$\log p(x) = \log \left(\exp \left\{ x^T \Sigma^{-1} x \right\} c_1 \right) = \log c_1 + \sum_{i,j} A_{i,j} x_i x_j$$

On the other hand, since it factors wrt G, we have an alternate expression for $\log p(x)$

$$\log p(x) = \log c_0 + \sum_{e=(x_i,x_j)\in E} \log p_e(x_i, x_j) + \sum_{x_i \in V} \log p_{x_i}(x_i).$$

Factoring and Gaussians

Suppose a Gaussian model factors via a graph $G = (V, E)$. Let $A = \Sigma^{-1}$ just for notation then:

$$\log p(x) = \log \left(\exp \left\{ x^T \Sigma^{-1} x \right\} c_1 \right) = \log c_1 + \sum_{i,j} A_{i,j} x_i x_j$$

On the other hand, since it factors wrt G, we have an alternate expression for $\log p(x)$

$$\log p(x) = \log c_0 + \sum_{e=(x_i,x_j)\in E} \log p_e(x_i, x_j) + \sum_{x_i \in V} \log p_{x_i}(x_i).$$

We differentiate each expression with respect to $\frac{\partial}{\partial x_i x_j}$ for $(i, j) \notin E$

- From the Gaussian form, we get $A_{i,j} + A_{j,i} = 2A_{i,j}$ since Σ^{-1} is symmetric.

Factoring and Gaussians

Suppose a Gaussian model factors via a graph $G = (V, E)$. Let $A = \Sigma^{-1}$ just for notation then:

$$\log p(x) = \log \left(\exp \left\{ x^T \Sigma^{-1} x \right\} c_1 \right) = \log c_1 + \sum_{i,j} A_{i,j} x_i x_j$$

On the other hand, since it factors wrt G, we have an alternate expression for $\log p(x)$

$$\log p(x) = \log c_0 + \sum_{e=(x_i,x_j)\in E} \log p_e(x_i, x_j) + \sum_{x_i \in V} \log p_{x_i}(x_i).$$

We differentiate each expression with respect to $\frac{\partial}{\partial x_i x_j}$ for $(i, j) \notin E$

- From the Gaussian form, we get $A_{i,j} + A_{j,i} = 2A_{i,j}$ since Σ^{-1} is symmetric.
- From the factorized form, if $(i, j) \notin E$ then the derivative is 0 since no term contains both x_i and x_j.

Factoring and Gaussians

Suppose a Gaussian model factors via a graph $G = (V, E)$. Let $A = \Sigma^{-1}$ just for notation then:

$$\log p(x) = \log\left(\exp\left\{x^T \Sigma^{-1} x\right\} c_1\right) = \log c_1 + \sum_{i,j} A_{i,j} x_i x_j$$

On the other hand, since it factors wrt G, we have an alternate expression for $\log p(x)$

$$\log p(x) = \log c_0 + \sum_{e=(x_i,x_j)\in E} \log p_e(x_i, x_j) + \sum_{x_i \in V} \log p_{x_i}(x_i).$$

We differentiate each expression with respect to $\frac{\partial}{\partial x_i x_j}$ for $(i, j) \notin E$

- From the Gaussian form, we get $A_{i,j} + A_{j,i} = 2A_{i,j}$ since Σ^{-1} is symmetric.
- From the factorized form, if $(i, j) \notin E$ then the derivative is 0 since no term contains both x_i and x_j.

Factoring and Gaussians

Suppose a Gaussian model factors via a graph $G = (V, E)$. Let $A = \Sigma^{-1}$ just for notation then:

$$\log p(x) = \log \left(\exp \left\{ x^T \Sigma^{-1} x \right\} c_1 \right) = \log c_1 + \sum_{i,j} A_{i,j} x_i x_j$$

On the other hand, since it factors wrt G, we have an alternate expression for $\log p(x)$

$$\log p(x) = \log c_0 + \sum_{e=(x_i,x_j)\in E} \log p_e(x_i, x_j) + \sum_{x_i \in V} \log p_{x_i}(x_i).$$

We differentiate each expression with respect to $\frac{\partial}{\partial x_i x_j}$ for $(i,j) \notin E$

- From the Gaussian form, we get $A_{i,j} + A_{j,i} = 2A_{i,j}$ since Σ^{-1} is symmetric.
- From the factorized form, if $(i,j) \notin E$ then the derivative is 0 since no term contains both x_i and x_j.

We conclude that our earlier observation is, in fact, general:

$$(i,j) \notin E \text{ then } \Sigma^{-1}_{i,j} = \Sigma^{-1}_{j,i} = 0.$$

Minor twist: Discrete Graphical Models

- We want to apply to case of discrete random variables.
  - **Good news**. We can write discrete distribution (exponential family!) in more or less the same form.
  - **Bad news.** Our argument uses a derivative in the variables, so depends on the *variables* being continuous.

Minor twist: Discrete Graphical Models

- We want to apply to case of discrete random variables.
  - **Good news**. We can write discrete distribution (exponential family!) in more or less the same form.
  - **Bad news.** Our argument uses a derivative in the variables, so depends on the *variables* being continuous.
  - **Happy Ending?** Indeed, we need a few more technical conditions–figured out in papers Loh and Wainwright 2014 and Ratner et al. 2018–but "morally" the same story.

Back to our Problem ...

- **Informative labelers**: λ_j correlated with y for $j = 1, \ldots, m$
- Only some labelers are correlated with each other.
- We assume we are given edges of graph (but not entries in covariance matrix).

Let's examine our analog of a covariance matrix

$$u = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \\ y \end{pmatrix} \text{ then } \mathbb{E}[uu^T] = \Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix}.$$

Here, the blocking is: $O \in \mathbb{R}^{m \times m}$ and $\beta \in \mathbb{R}^m$

Let's examine our analog of a covariance matrix

$$u = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \\ y \end{pmatrix} \text{ then } \mathbb{E}[uu^T] = \Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix}.$$

Here, the blocking is: $O \in \mathbb{R}^{m \times m}$ and $\beta \in \mathbb{R}^m$

- If we compute $\beta_i = \mathbb{E}[\lambda_i y] \in \mathbb{R}^m$ this is the accuracy–but we can't observe y, so β isn't measurable directly.
- We can observe O, since as before these are observed votes on every data point for $j, k \in 1, \ldots, m$

$$O_{j,k} = \mathbb{E}[\lambda_j(x)\lambda_k(x)] \approx \frac{1}{n}\sum_{i=1}^{n} \lambda_j(x^{(i)})\lambda_k(x^{(i)})$$

Let's assume we know the *graph structure*–i.e., the indexes for which *zeros* in Σ^{-1}. We can use this to recover β.

We need some heavier weight identities (very similar to the block Gaussians from earlier). Let's block decompose Σ^{-1}:

$$\Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix} \text{ and } \Sigma^{-1} = \begin{pmatrix} K & v \\ v^T & K_S \end{pmatrix}.$$

We need some heavier weight identities (very similar to the block Gaussians from earlier). Let's block decompose Σ^{-1}:

$$\Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix} \text{ and } \Sigma^{-1} = \begin{pmatrix} K & v \\ v^T & K_S \end{pmatrix}.$$

A tool that helps us is *the Matrix Inversion Lemma*, which says:

$$\Sigma^{-1} = \begin{pmatrix} O^{-1} + cO^{-1}\beta\beta^T O^{-1} & -cO^{-1}\beta \\ -c\beta^T O^{-1} & c \end{pmatrix} \text{ where } c^{-1} = 1 - \beta^T O^{-1}\beta.$$

We'll use:

$$K = O^{-1} + cO^{-1}\beta\beta^T O^{-1}$$

We show how to recover β:

$$\Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix} \text{ and } \Sigma^{-1} = \begin{pmatrix} K & v \\ v^T & K_S \end{pmatrix}.$$

The matrix inversion lemma tells us:

$$K = O^{-1} + cO^{-1}\beta\beta^T O^{-1} \text{ in which } c = \left(1 - \beta^T O^{-1} \beta\right)^{-1} \in \mathbb{R}^+$$

We show how to recover β:

$$\Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix} \text{ and } \Sigma^{-1} = \begin{pmatrix} K & v \\ v^T & K_S \end{pmatrix}.$$

The matrix inversion lemma tells us:

$$K = O^{-1} + cO^{-1}\beta\beta^T O^{-1} \text{ in which } c = \left(1 - \beta^T O^{-1}\beta\right)^{-1} \in \mathbb{R}^+$$

Let's define some notation:

$$z = \sqrt{c}O^{-1}\beta \text{ then } K = O^{-1} + zz^T$$

Assuming we can recover z, we show how recover β using *observable quantities*.

We show how to recover β:

$$\Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix} \text{ and } \Sigma^{-1} = \begin{pmatrix} K & v \\ v^T & K_S \end{pmatrix}.$$

The matrix inversion lemma tells us:

$$K = O^{-1} + cO^{-1}\beta\beta^T O^{-1} \text{ in which } c = \left(1 - \beta^T O^{-1}\beta\right)^{-1} \in \mathbb{R}^+$$

Let's define some notation:

$$z = \sqrt{c}O^{-1}\beta \text{ then } K = O^{-1} + zz^T$$

Assuming we can recover z, we show how recover β using *observable quantities*. First, we recover c. Specifically, $1 + z^T Oz = c$

$$1 + z^T Oz = 1 + c\beta^T O^{-1} O O^{-1}\beta = 1 + c\beta^T O^{-1}\beta$$

On the other hand, using the definition of c

$$\frac{1}{c} = 1 - \beta^T O^{-1}\beta \iff 1 + c\beta^T O^{-1}\beta = c$$

We show how to recover β:

$$\Sigma = \begin{pmatrix} O & \beta \\ \beta^T & 1 \end{pmatrix} \text{ and } \Sigma^{-1} = \begin{pmatrix} K & v \\ v^T & K_S \end{pmatrix}.$$

The matrix inversion lemma tells us:

$$K = O^{-1} + cO^{-1}\beta\beta^T O^{-1} \text{ in which } c = \left(1 - \beta^T O^{-1}\beta\right)^{-1} \in \mathbb{R}^+$$

Let's define some notation:

$$z = \sqrt{c}O^{-1}\beta \text{ then } K = O^{-1} + zz^T$$

Assuming we can recover z, we show how recover β using *observable quantities*. First, we recover c. Specifically, $1 + z^T Oz = c$

$$1 + z^T Oz = 1 + c\beta^T O^{-1} O O^{-1}\beta = 1 + c\beta^T O^{-1}\beta$$

On the other hand, using the definition of c

$$\frac{1}{c} = 1 - \beta^T O^{-1}\beta \iff 1 + c\beta^T O^{-1}\beta = c$$

Now set $\beta = \frac{Oz}{\sqrt{c}}$. So we just need to find z!

Our problem is to find z given this equation:

$$K = O^{-1} + zz^T$$

- We observe O (so can compute its inverse, O^{-1}).
- We don't know z, we have to compute it.

Our problem is to find z given this equation:

$$K = O^{-1} + zz^T$$

- We observe O (so can compute its inverse, O^{-1}).
- We don't know z, we have to compute it.
- We don't know the values of K, but we do know some entries where $K_{j,k} = 0$ (the graph assumption!). Define:

  $$\Omega = \{(j,k) : K_{j,k} = 0\} .$$

  For $(j,k) \in \Omega$ our equation reduces to:

  $$O^{-1}_{i,j} = -z_j z_k$$

When is $z_j z_k$ uniquely defined?

Our problem is to find z given this equation:

$$K = O^{-1} + zz^T$$

- We observe O (so can compute its inverse, O^{-1}).
- We don't know z, we have to compute it.
- We don't know the values of K, but we do know some entries where $K_{j,k} = 0$ (the graph assumption!). Define:

  $$\Omega = \{(j,k) : K_{j,k} = 0\}.$$

  For $(j,k) \in \Omega$ our equation reduces to:

  $$O^{-1}_{i,j} = -z_j z_k$$

When is $z_j z_k$ uniquely defined? Almost same as before–square and take logs let $a_{j,k} = O^{-1}_{j,k}$

$$\log a^2_{j,k} = \log z^2_j + \log z^2_k.$$

A linear program? only recover up to sign. . .

When do we have enough to complete?

$$K = O^{-1} + zz^T \text{ and } \Omega = \{(j,k) : K_{j,k} = 0.\}$$

So when can we find enough entries that this algorithm works?

$$\log a_{j,k}^2 = \log z_j^2 + \log z_k^2.$$

Define a set of matrices $M(\Omega) \in \{0,1\}^{m \times m}$ for the set Ω

$$M(\Omega)_{j,k} \in \begin{cases} \{0,1\} & \text{if } (j,k) \in \Omega \\ \{0\} & \text{o.w. } . \end{cases}$$

For a given Ω, if there is some $M \in M(\Omega)$ that is full rank, then we can compute z_j^2 for $j = 1, \ldots, m$.

Some examples of Mask Matrices

$$K = O^{-1} + zz^T \text{ and } \Omega = \{(j,k) : K_{j,k} = 0.\}$$

- **Sanity check**: if the labelers are independent, $\Omega = \{(j,k) : j \neq k\}$. If $m = 3$, the following choice of M:

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \leq \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Some examples of Mask Matrices

$$K = O^{-1} + zz^T \text{ and } \Omega = \{(j,k) : K_{j,k} = 0.\}$$

- **Sanity check**: if the labelers are independent, $\Omega = \{(j,k) : j \neq k\}$. If $m = 3$, the following choice of M:

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \leq \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

This choice of M corresponds to the following system of equations:

$$\log a_{2,1}^2 = \log z_2^2 + \log z_1^2$$
$$\log a_{3,2}^2 = \log z_3^2 + \log z_2^2$$
$$\log a_{1,3}^2 = \log z_1^2 + \log z_3^2$$

Some examples of Mask Matrices

$$K = O^{-1} + zz^T \text{ and } \Omega = \{(j,k) : K_{j,k} = 0.\}$$

- **Sanity check**: if the labelers are independent, $\Omega = \{(j,k) : j \neq k\}$. If $m = 3$, the following choice of M:

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \leq \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

This choice of M corresponds to the following system of equations:

$$\begin{aligned} \log a_{2,1}^2 &= \log z_2^2 + \log z_1^2 \\ \log a_{3,2}^2 &= \log z_3^2 + \log z_2^2 \\ \log a_{1,3}^2 &= \log z_1^2 + \log z_3^2 \end{aligned} \implies \begin{aligned} \log z_2^2 &= \log a_{2,1}^2 - \log z_1^2 \\ \log z_3^2 &= \log a_{3,2}^2 - \log z_1^2 \end{aligned}$$

Some examples of Mask Matrices

$$K = O^{-1} + zz^T \text{ and } \Omega = \{(j,k) : K_{j,k} = 0.\}$$

- **Sanity check**: if the labelers are independent, $\Omega = \{(j,k) : j \neq k\}$. If $m = 3$, the following choice of M:

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \leq \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

This choice of M corresponds to the following system of equations:

$$\begin{aligned} \log a_{2,1}^2 &= \log z_2^2 + \log z_1^2 \\ \log a_{3,2}^2 &= \log z_3^2 + \log z_2^2 \\ \log a_{1,3}^2 &= \log z_1^2 + \log z_3^2 \end{aligned} \implies \begin{aligned} \log z_2^2 &= \log a_{2,1}^2 - \log z_1^2 \\ \log z_3^2 &= \log a_{3,2}^2 - \log z_1^2 \\ \log a_{3,2}^2 &= \log a_{2,1}^2 + \log a_{3,2}^2 - 2\log z_1^2 \end{aligned}$$

An Example that Should Fail!

$$K = O^{-1} + zz^T \text{ and } \Omega = \{(j,k) : K_{j,k} = 0.\}$$

- **Should fail** What about the case of three voters: 1 and 2 are are correlated? $\Omega = \{(1,3),(2,3),(3,1),(3,2)\}$ Then, M is component-wise less than

$$M \leq \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Yikes, no way to make it full rank!

More examples

$$K = O^{-1} + zz^T \text{ and } \Omega = \{(j,k) : K_{j,k} = 0.\}$$

- What about with four labelers the 1 and 2 correlated:

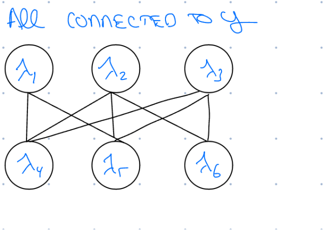
$$M \leq \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \text{ take } M = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

So in this case, we can recover it!

So we need *enough* independence to recover the voters. This exactly characterizes when recovery is possible.

Sign Recovery is More Complex for Correlations

- Recovering the signs is more complex with correlations. Let's illustrate the problem. Consider the following example:



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

- Observe that there are two connected components in $M(\Omega)$ and so *four* possible solutions

$$\begin{pmatrix} \pm\beta_1 \\ \pm\beta_2 \end{pmatrix} \text{ for } \beta_i \in \mathbb{R}^3 \text{ and } j \in \{1, 2\}.$$

To solve this, assumptions are typically made *per component* (each one applies!)

Wrap-up: Extensions

- It turns out you can learn Σ^{-1} as well–without EM. It has a really nice interpretation as Sparse PCA.
  - We didn't do sample efficiency in this course, but with SGD this and the MLE methods you know–it's sample optimal (Chen, Sala et al, 2020). Theory fun, algorithm simple.
- For simplicity, every voter voted on every instance. This is easy to relax (and hide) in the $\mathbb{E}$ notation.
- More sophisticated models use information from embeddings, voters that only vote one way (or have different accuracy), and more!

Summary

- Today was a lot! We saw methods to recover latent variable models that used linear algebra and information about the statistics to compute them–no EM.
- They had stronger guarantees–exact, provable recovery up to symmetry. (cf. with EM).
- The weak supervision methods are now used in many places. I'm as shocked as you are.