Lecture 4
Label Aggregation: EM-Based Methods

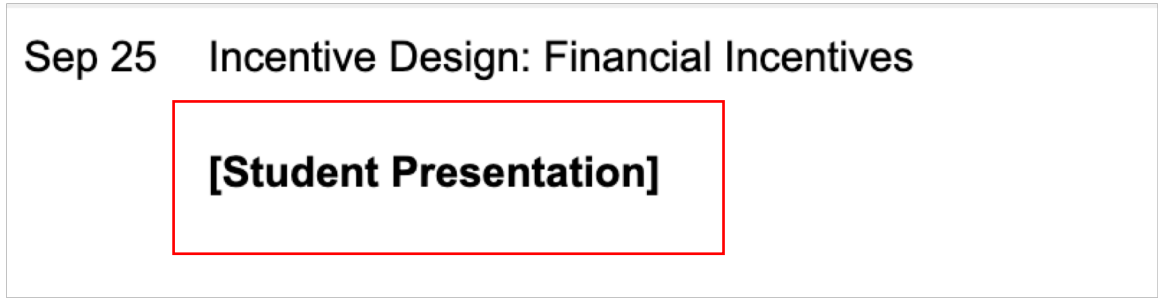
Instructor: Chien-Ju (CJ) Ho

Logistics: Reviews / Assignment 1

- Reviews:
  - What counts as Pass / Not Pass
- The next required reading is more mathematically dense. Try to at least understand the formulation and key results.
- Assignment 1 due this Friday.
  - Unless specified, all reviews/assignments should be done individually, not in groups.

Logistics: Bidding for Presentations

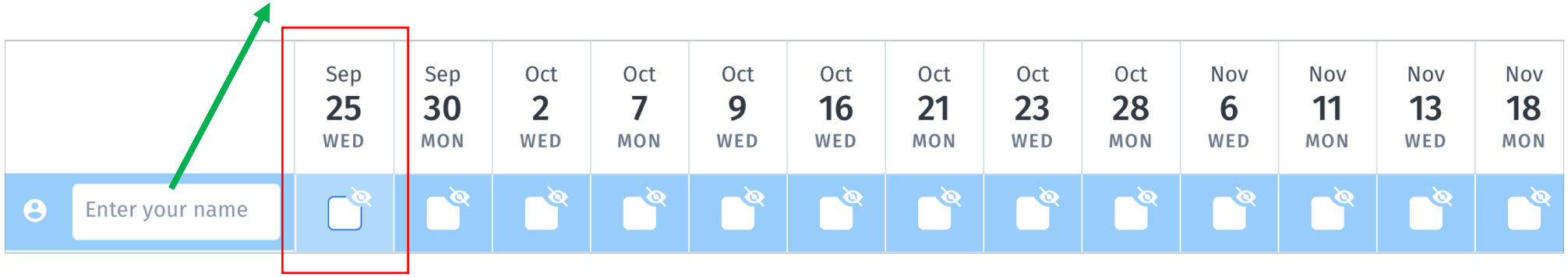
- Check out the course schedule for the presentation slots:



- Provide around 3~5 bids by the end of this Wednesday (hard deadline).
  - https://doodle.com/poll/yycvun8fx8z8bde2
  - You might want to glance over the papers of your bidding.

Logistics: Bidding for Presentations

- Bidding interface
  - Enter the **names of all members** in your group



Make sure you can make it before bidding this one.

There might be small changes on the exact list of papers for later topics.

- I'll announce the assignment by this Thursday.
  - Manually solve the max-cover problem.
  - I'll try to accommodate your interests, but no guarantee on that.
  - Random assignments will be used if there is no feasible solutions.
  - I'll fill in the slots if there are fewer groups than slots.

What's the best bidding strategy?

Logistics: Presentation

- Read the required paper and at least 1~2 optional papers.
  - You are encouraged to find additional materials to present. Given the nature of the course, you should be able to find interesting/relevant things yourself.

- **Talk to me one week before your presentation.**
  - Default: talk to me after class the week before your presentation.
  - You should have finished reading and have a presentation plan by then.
  - Come up with two discussion questions.
    - Ideal discussion questions:
      - Open-ended questions that do not required deep technical background.
  - If you are bidding on the Sep 25th presentation, you need to do these next Wednesday!

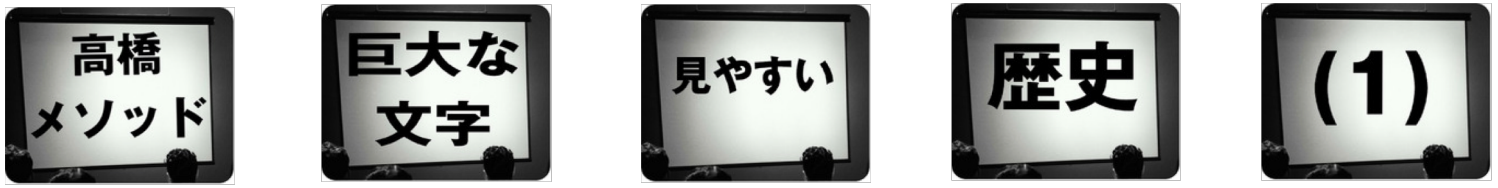
Logistics: Presentation

- Try to engage the class as much as possible.
- You need to present for the lecture time (70~80 min).
- You can present in ways you feel comfortable with. In case you have no clues on what to do, below is one potential format:
  - Short overview of the topic
  - Summarize the required reading (25 min)
  - Discussion (10 min)
  - Summarize another paper (25 min)
  - Discussion (10 min)
  - (Optional) Any materials you like to share.

Logistics: Presentation

- Notes
  - Spend time **slowly and carefully go through the model and assumptions** before jumping into the results.
  - You might want to **summarize the required reading** carefully.
    - Yes, everyone should have already read it, but everyone might be confused about some parts of the paper
- High-level suggestions
  - When explaining math, give intuitions
  - Be enthusiastic and confident (if you are not, pretend to be)
  - Avoid putting too many words in the slides

Takahashi method:



Logistics: Peer Reviews for Presentations

- I'll distribute peer review forms for student presentations.
- The review forms are not anonymized, but I'll remove the names and send the reviews to presenters.

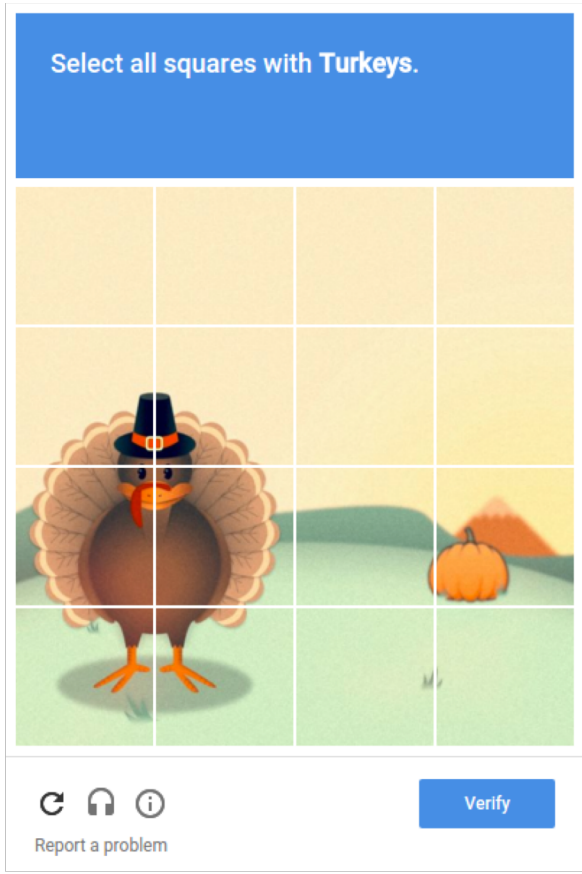
- Try to give concrete suggestions.

Logistics: Tentative Project Timeline

- Sep 20: Project proposal
  - Brief description of the proposed project (1~2 paragraph)
  - Citing at least one paper that's relevant to your proposal
- Oct 9: Milestone 1
  - A brief literature review and the description of your plan (one page)
  - Last chance to change the topic of the project
- Oct 30: Milestone 2
  - Summary of your current progress (up to 2 pages)
  - Last chance to convert the research project to (a more extensive) literature review
- Dec 2-4: In-class project presentations
- Dec 8: Project report due

A Short Recap of Last Lecture

Course Overview



Human as data sources: Label aggregation
Probabilistic reasoning to aggregate noisy human data

Practical challenges: Real-time and complex tasks
Studies on workflow and team designs from HCI perspective

Humans are "Humans": Incentive design
Game theoretical modeling of humans and incentive design

Selected recent topics:
Ethical issues of AI/ML, learning with strategic behavior, Human-AI collaborations.

Label aggregation

	Worker 1	Worker 2	Worker 3	Worker 4	…
Task 1	+1	-1		-1	
Task 2		-1	+1		
Task 3	-1			+1	
Task 4		+1	+1		
…					

- Goal: infer true labels
- Challenges
  - Unknown worker skills
  - Different task difficulties
  - More factors to consider (some structures of tasks/workers?)

Typical label aggregation approach

- Propose a model to describe the label generation process
- True labels are the “latent variables” of the process
- Using inference algorithms (e.g., EM) to learn the latent variables

Sep 9	Label Aggregation: EM-based Algorithms	**Required** Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. Whitehill et al. NIPS 2009. **Optional** Learning from Crowds. Raykar et al. JMLR 2010. Maximum Likeihood Estimation of Observer Error-Rates Using the EM Algorithm. Dawid and Skene. Applied Statistics. 1979.
Sep 11	Label Aggregation: Matrix-based Methods	**Required** Who Moderates the Moderators? Crowdsourcing Abuse Detection in User-Generated Content. Ghosh, Kale, and McAfee. EC 2011. - If you want to refresh your memory on matrix algebra, Matrix Cookbook is a good resource. Section 5 contains the matrix decomposition part. **Optional** Budget-Optimal Crowdsourcing using Low-rank Matrix Approximations. Karger, Oh, and Shah. Allerton 2011. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. Zhang et al. JMLR 2016.
Sep 16	Label Aggregation: Others and Discussion	**Required** Iterative Learning for Reliable Crowdsourcing Systems. Karger, Oh, and Shah. NIPS 2011. **Optional** Variational Inference for Crowdsourcing. Liu, Peng, and Ihler. NIPS 2012. Learning from the Wisdom of Crowds by Minimax Entropy. Zhou et al. NIPS 2012.

Write down likelihood/posterior function
Using EM algorithms to find the parameters that maximize likelihood/posterior

Write labels as a matrix (worker by task)
Using low rank matrix approximation

A bunch of other methods

Probabilistic Approach for Label Aggregation

- High-level ideas:
  - Let D be the set of observations
    (e.g., training dataset, the set of labels we got from workers)
  - Let θ be the set of latent parameters we care about
    (e.g., ML hypothesis, true labels)

  - Two important concepts
    - Posterior: $\Pr(\theta|D)$ [More discussion in CSE515T]
    - Likelihood: $\Pr(D|\theta)$ [More discussion in CSE417T]

    - Connection: $\Pr(\theta|D) = \frac{\Pr(\theta)\Pr(D|\theta)}{\Pr(D)}$

Why Majority Voting:
Majority Voting Gives the Maximum-Likelihood Estimation

- Consider a task with true label l^*
- We collect labels $L = \{l_1, l_2, \ldots, l_n\}$ from n workers for this task.
- Each worker gives the correct label with probability $p > 0.5$.

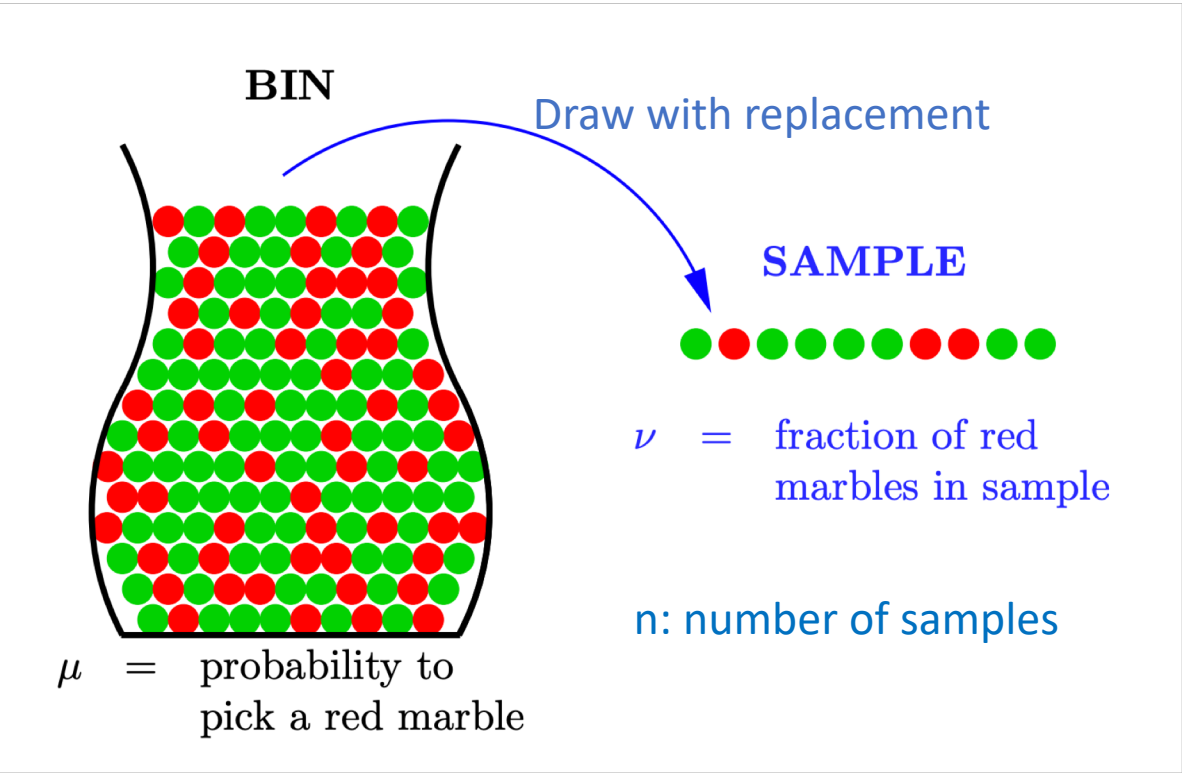
- l^* is the latent variable and L is our observation.

- Maximum likelihood estimation (MLE):
  - Predict +1 if $\Pr[\, L|l^* = +1] \geq \Pr[\, L|l^* = -1]$
  - Predict -1 otherwise

Likelihood: $\Pr[D|\theta]$
D: Observations
θ: latent variables

MLE approach (roughly speaking):
Find $\theta^* = argmax_\theta \Pr[D|\theta]$

Proving Theoretical Bounds: Utilizing some form of law of large numbers



What can we say about μ from ν?

Law of large numbers

- When $n \to \infty$, $\nu \to \mu$

Hoeffding's Inequality

- $\Pr[|\mu - \nu| > \epsilon] \leq 2e^{-2\epsilon^2 n}$ for any $\epsilon > 0$

Today’s Lecture

Framework for Probabilistic Inference

- Notations:
  - $\mathrm{D} = \{\mathrm{d}_1, \ldots, \mathrm{d}_\mathrm{n}\}$: observations (e.g., training data, labels we got from workers)
  - θ: be the set of latent parameters we care about (e.g., ML hypothesis, true labels)

- MLE approach
  - $\theta^* = argmax_\theta \Pr(D|\theta)$
    $= argmax_\theta \prod_{i=1}^{n} \Pr(d_i|\theta)$ (from the common "independence" assumption)
    $= argmax_\theta \log \prod_{i=1}^{n} \Pr(d_i|\theta)$
    $= argmax_\theta \sum_{i=1}^{n} \log \Pr(d_i|\theta)$

In machine learning, we often replace this as a (negative) point-wise "loss function"

Framework for Probabilistic Inference

- Notations:
  - $\mathrm{D} = \{\mathrm{d}_1, \ldots, \mathrm{d}_\mathrm{n}\}$: observations (e.g., training data, labels we got from workers)
  - θ: be the set of latent parameters we care about (e.g., ML hypothesis, true labels)

- MLE approach
  - $\theta^* = argmax_\theta \Pr(D|\theta)$
    $= argmax_\theta \prod_{i=1}^{n} \Pr(d_i|\theta)$ (from the common "independence" assumption)
    $= argmax_\theta \log \prod_{i=1}^{n} \Pr(d_i|\theta)$
    $= argmax_\theta \sum_{i=1}^{n} \log \Pr(d_i|\theta)$
- Another interpretation
  - Define point-wise loss function $\ell(d, \theta)$
  - Solving $\theta^* = argmin_\theta \sum_{i=1}^{n} \ell(d_i, \theta)$

Solving this optimization problem is one of the "key" steps in machine learning.

Get Back to Label Aggregation

- Steps for MLE approach
  - Define label generation model $\Pr(d_i|\theta)$
    - θ contains the true labels and other latent factors in your models
  - Optimization: Find $\theta^* = argmax_\theta \sum_{i=1}^{n} \log \Pr(d_i|\theta)$
    - In last lecture, there are only two possible values for θ. So we brute-force find it.

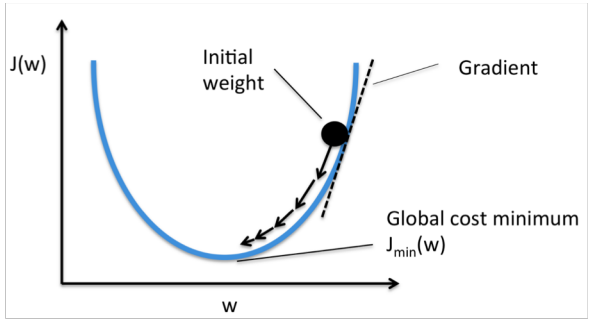
> - Maximum likelihood estimation (MLE):
>   - Predict +1 if $\Pr[L|l^* = +1] \geq \Pr[L|l^* = -1]$
>   - Predict -1 otherwise

  - What if there are infinitely many possible values of θ?
    - Need to perform "optimization" algorithms to find it θ^*.

Optimization

- One of the key elements in modern machine learning.
  - Therefore, most ML courses require probability, calculus, and linear algebra.

- Assume the function we want to minimize (maximize) is convex (concave).
  - Gradient descent is one of the most common-used algorithm.



$$w_{t+1} = w_t - \gamma_t \nabla J(w)$$

Only guarantees to find local optimum
In convex functions, local optimum == global optimum

  - What if the function is not convex
    - Start at a random point, do many times, report the best one.
    - (This is essentially what deep learning is doing for minimizing loss)

Expectation-Maximization (EM)

- Gradient descent algorithms require “gradient” information.
- Consider the function we want to minimize: $L(\theta_1, \theta_2)$
  - $\partial L/\partial \theta_1$ can be obtained
  - $\partial L/\partial \theta_2$ are hard to obtain (e.g., θ_2 are the ”true” labels)

Only guarantee to converge to local optimum.

- EM: an iterative approach
  - Start with some initial estimates of θ_1, θ_2
  - Iteratively perform the following until the stop conditions are met:
    - Fix θ_1, estimate θ_2
    - Fix θ_2, estimate θ_1
    - Stopping condition: converged, # iterations >= pre-determined threshold, etc

Consider a simpler case: Optional Reading

Maximum Likeihood Estimation of Observer Error-Rates Using the EM Algorithm. Dawid and Skene. Applied Statistics. 1979.

Motivating Scenario

- Multiple doctors give judgements of patients' information
- Doctors might make mistakes (with unknown probability)
- Given judgements from multiple doctors, how to infer patients' true information.

- In the context of label aggregation
  - Doctors -> workers
  - Judgements -> labels
  - They consider the setting all tasks are the same

Review of Last Lecture

- Homogeneous workers (all workers are the same)
  - Majority voting leads to MLE

- If each worker i gives correct labels with probability p_i (p_i: worker skill)
  - Weighted majority voting leads to MLE
    - With weight $w_i = \ln \frac{p_i}{1-p_i}$ for label l_i

Predict $sign(\sum_{i=1}^{n} w_i l_i)$

  - Given worker skills, we have an easy way to

What if Workers' Skills are Unknown

- Short Discussion: What can we do?
  - Think about the EM idea we just discussed

	Worker 1	Worker 2	Worker 3	Worker 4	Worker 5
Task 1	-1	+1	-1	+1	-1
Task 2	+1	+1	-1	+1	-1
Task 3	+1	-1	+1	-1	+1
Task 4	-1	-1	+1	+1	+1

High-Level Description of EM

Algorithm 1 The basic EM framework of Dawid and Skene (1979).

Input: Sets of worker-generated labels for each instance
Initialize each instance's label based on a simple majority vote
repeat
  for all Workers w **do**
    Calculate w's quality parameter(s), treating each instance's current label as ground truth
  end for
  for all Instances i **do**
    Calculate the most likely label for i, treating each worker's approximated quality parameter(s) as ground truth
  end for
until Label assignments have converged
Output: The current label assignments for each instance

Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. Vaughan. JMLR 2018.

Required Reading

Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. Whitehill et al. NIPS 2009.

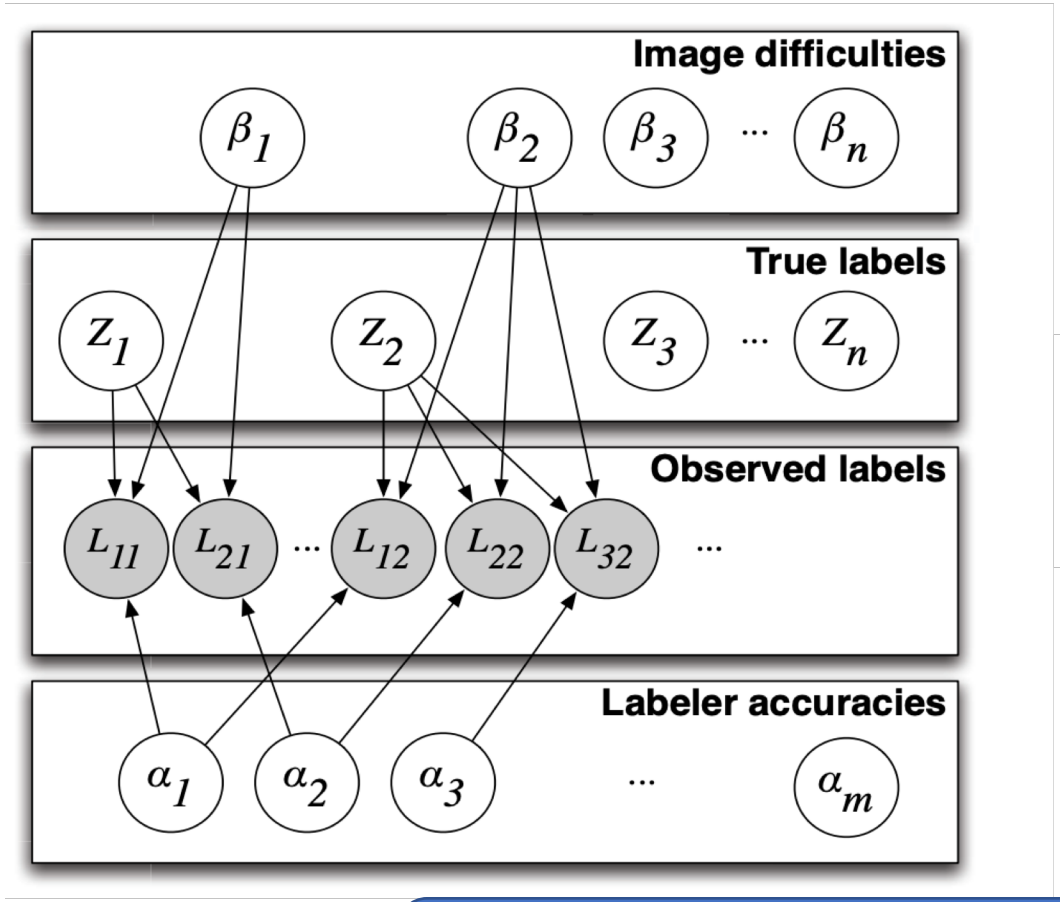
Reminder on the Framework

- Steps for MLE approach
  - Define label generation model $\Pr(d_i|\theta)$
    - θ contains the true labels and other latent factors in your models
  - Optimization: Find $\theta^* = argmax_\theta \sum_{i=1}^{n} \log \Pr(d_i|\theta)$
    - In last lecture, there are only two possible values for θ. So we brute-force find it.

> - Maximum likelihood estimation (MLE):
>   - Predict +1 if $\Pr[L|l^* = +1] \geq \Pr[L|l^* = -1]$
>   - Predict -1 otherwise

    - What if there are infinitely many possible values of θ?
      - Need to perform "optimization" algorithms to find it θ^*.

Model of Label Generation



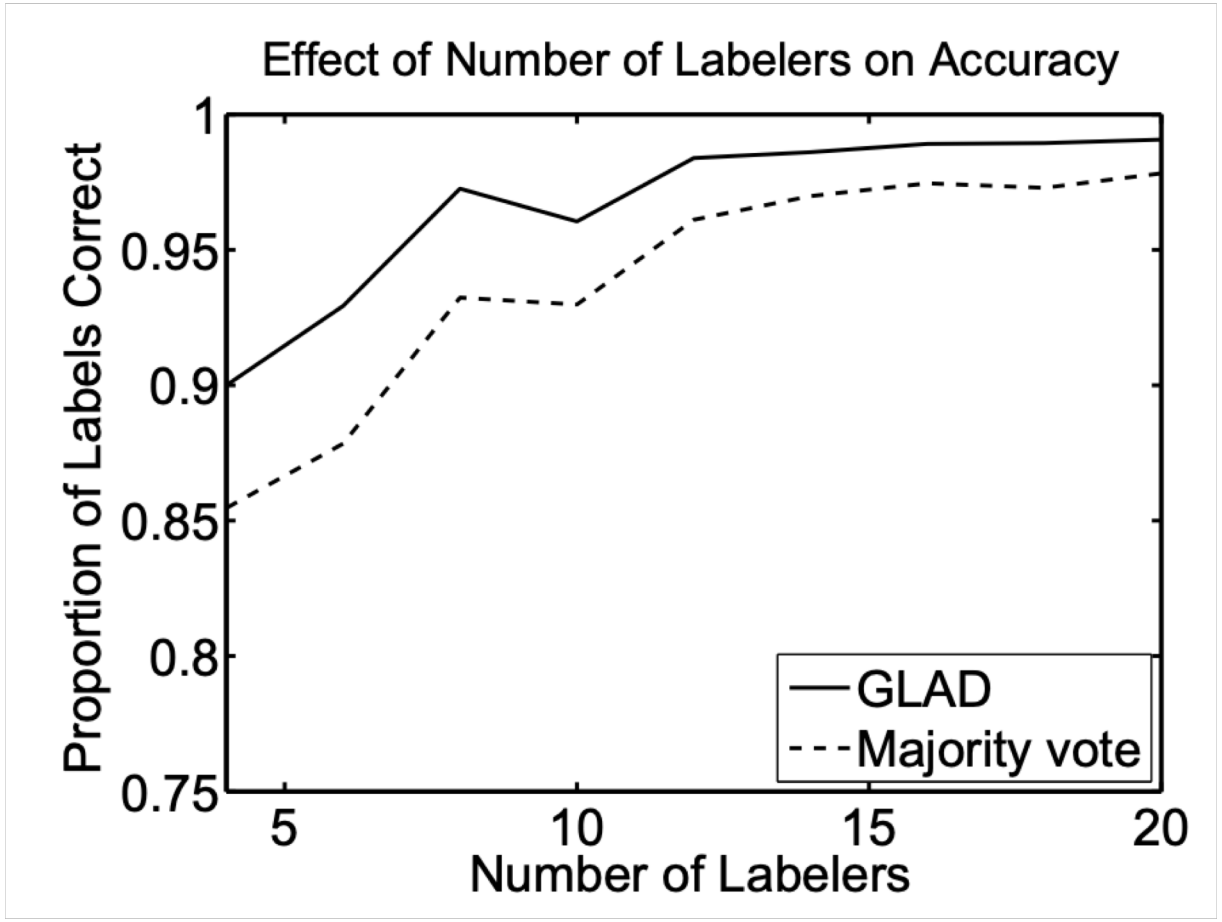
$$p(L_{ij} = Z_j | \alpha_i, \beta_j) = \frac{1}{1 + e^{-\alpha_i \beta_j}}$$

What do these parameters really mean?

Using EM to find the MLE

- E-Step:
  - Fix current estimate α and β, calculate the distribution of true labels
- M-Step
  - Fix current estimate of true labels, finding α and β that maximize likelihood
    - Using gradient descent

Simulation/Experiments



Discussion

- What are your general thoughts about the paper.
- When do you think majority voting would actually be a preferred method than GLAD or other more sophisticated method?
- What other aspects of label generation do you think can/should also be modeled (the application doesn't need to be restricted to image labeling)?

When Majority-Voting Might Be Preferred

- Not enough data: Occam's Razor
- Fairness considerations: When the outcome impacts people
  - Can we give different weights to voters in Presidential Elections?
- When the label is subjective
  - Aggregating preferences is a hard question
  - Arrow's impossibility theorem

What Other Aspects to Model

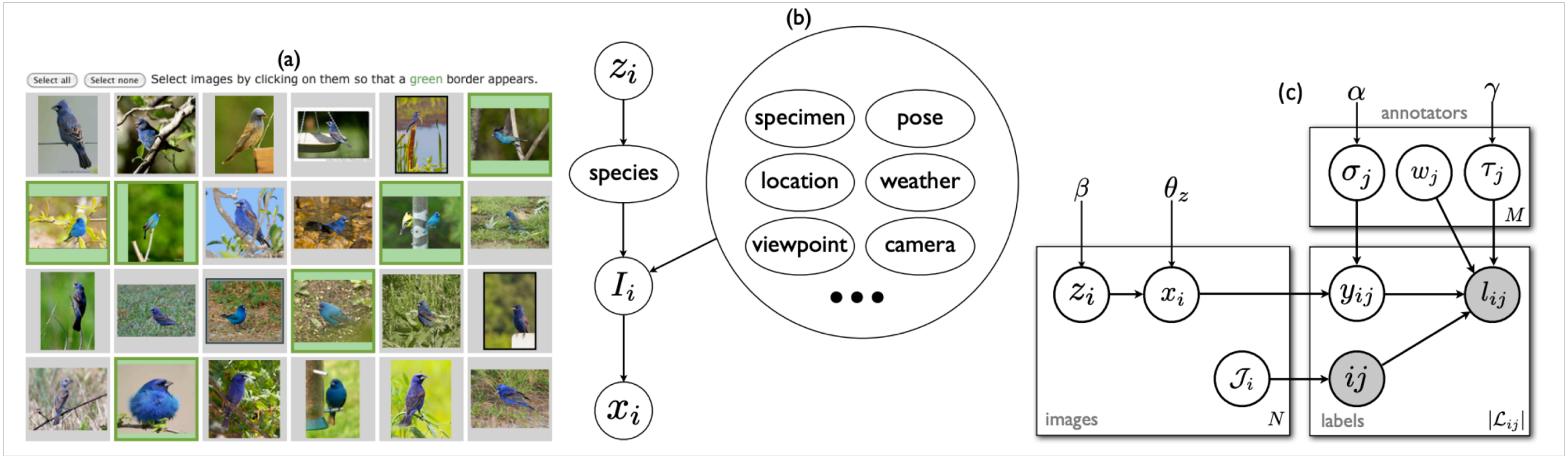
- Confusion matrix
  - Instead of using a single probability for modeling worker skills for tasks

Ground Truth

Worker Label	Label 1	Label 2	Label 3
Label 1	0.8	0.1	0.1
Label 2	0.1	0.9	0
Label 3	0.1	0.2	0.7

What Other Aspects to Model

- The Multidimensional Wisdom of Crowds. Welinder et al. NIPS 2010



What Other Aspects to Model

- Temporal Information
  - Workers get more experienced over time
    - [some recent relevant research topic: machine teaching]
  - Workers get tired over time
  - Most approaches are pretty ad-hoc

General Framework for Label Aggregation

- Most of the papers follow this idea.
- Steps:
  - Model label generation $\Pr(d_i|\theta)$
  - Optimization: Find $\theta^* = argmax_\theta \sum_{i=1}^{n} \log \Pr(d_i|\theta)$ [or other objective]

- With reasonable models, it works well in practice.
- However, no theoretical guarantees in general.

Next Two Lectures

- Read papers that give theoretical guarantees.
  - Be prepared for the more math-heavy readings (especially the next one)

Sep 9	Label Aggregation: EM-based Algorithms	**Required** Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. Whitehill et al. NIPS 2009. **Optional** Learning from Crowds. Raykar et al. JMLR 2010. Maximum Likeihood Estimation of Observer Error-Rates Using the EM Algorithm. Dawid and Skene. Applied Statistics. 1979.
Sep 11	Label Aggregation: Matrix-based Methods	**Required** Who Moderates the Moderators? Crowdsourcing Abuse Detection in User-Generated Content. Ghosh, Kale, and McAfee. EC 2011. - If you want to refresh your memory on matrix algebra, Matrix Cookbook is a good resource. Section 5 contains the matrix decomposition part. **Optional** Budget-Optimal Crowdsourcing using Low-rank Matrix Approximations. Karger, Oh, and Shah. Allerton 2011. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. Zhang et al. JMLR 2016.
Sep 16	Label Aggregation: Others and Discussion	**Required** Iterative Learning for Reliable Crowdsourcing Systems. Karger, Oh, and Shah. NIPS 2011. **Optional** Variational Inference for Crowdsourcing. Liu, Peng, and Ihler. NIPS 2012. Learning from the Wisdom of Crowds by Minimax Entropy. Zhou et al. NIPS 2012.

Write down likelihood/posterior function
Using EM algorithms to find the parameters that maximize likelihood/posterior

Write labels as a matrix (worker by task)
Using low rank matrix approximation

A bunch of other methods