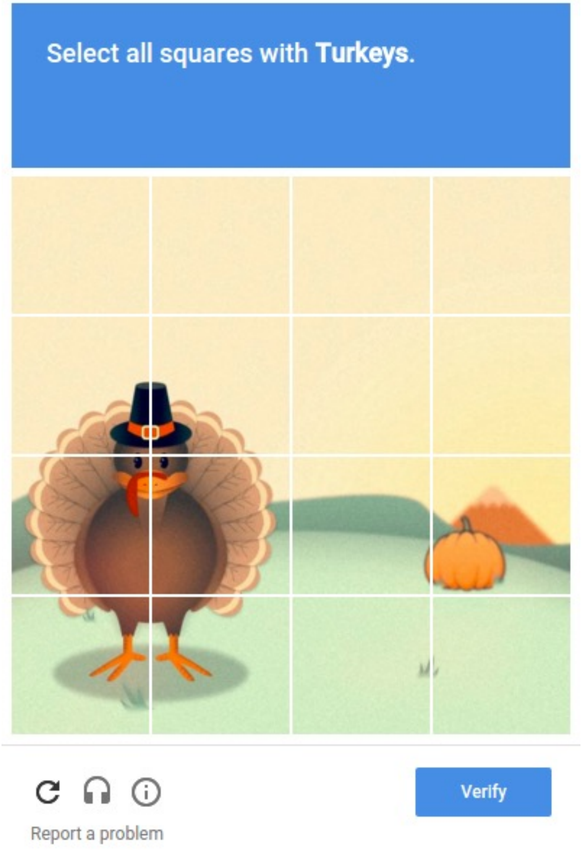
Lecture 4
Label Aggregation: EM-Based Methods

Instructor: Chien-Ju (CJ) Ho

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?)

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

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

Majority Voting for ”Homogeneous” Workers

- Model: Every worker gives correct label with probability $p > 0.5$
- Majority voting leads to maximum likelihood estimation (MLE)

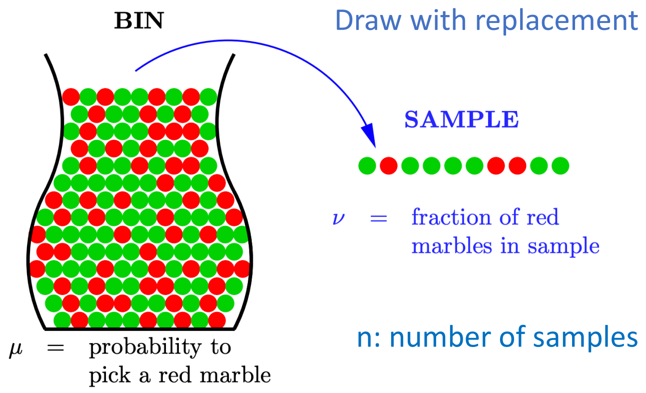
MLE (roughly speaking):
Find $\theta^* = argmax_\theta \Pr[D|\theta]$

- Theoretical guarantees of majority voting
  - Hoeffding’s Inequality

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

  - Plug it into label aggregation binary classification

    $Prob\ of\ error \leq e^{-\frac{1}{2}n\,(2p-1)^2}$, where p is the prob of correct label



What if Workers are Heterogeneous

- Worker $i \in \{1, \ldots, n\}$
  - provides label $l_i \in \{-1, +1\}$
  - assumption: each label l_i is correct with probability p_i
  - assume p_i is known

Remember why we can write it in this way?
Hint: it's due to the choice of the label presentation $\{+1, -1\}$

- Weighted majority voting

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

  - Weights that lead to MLE: $w_i = \ln \frac{p_i}{1-p_i}$ for label l_i
    - You can prove this yourself following the proof of simple majority voting
  - Weights that minimizes error bound: $w_i = 2p_i - 1$ for label l_i
    - (Lemma 1 in [Ho et al. ICML 2013])

Today’s Lecture

Framework for Probabilistic Inference

Each d_i is often assumed to be independently drawn

- 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)
    $= argmax_\theta \log \prod_{i=1}^{n} \Pr(d_i|\theta)$
    $= argmax_\theta \sum_{i=1}^{n} \boxed{\log \Pr(d_i|\theta)}$

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

Framework for Probabilistic Inference

Each d_i is often assumed to be independently drawn

- Notations:
  - $\mathrm{D} = \{\mathrm{d}_1, \dots, \mathrm{d}_\mathrm{n}\}$: 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 $\Pr(d_i|\theta)$ (define loss functions in ML)
    - 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 the last lecture, there are only two possible values for θ. So we find it in a brute-force way

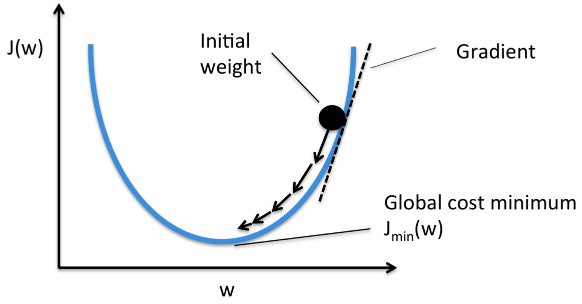
      - Maximum likelihood estimation (MLE):
        - Predict +1 if $\Pr[\,L|l^* = +1] \geq \Pr[\,L|l^* = -1]$
        - Predict -1 otherwise

    - 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
  - The reason 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 commonly-used algorithm



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

1. Requires gradient to exist everywhere
2. 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

Expectation-Maximization (EM)

- What if gradient doesn't always exist
- Consider the function we want to minimize: $L(\theta_1, \theta_2)$
  - $\partial L / \partial \theta_1$ can be obtained (e.g., θ_1 are the unknown worker skills)
  - $\partial L / \partial \theta_2$ 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 (e.g., find MLE)
    - 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 diagnosis based on a patient’s information
- Doctors might make mistakes (with unknown probability)
- Given diagnosis from multiple doctors, how to infer the patients’ true condition

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

Reminder: If Worker Skills are Known

- Worker $i \in \{1, \dots, n\}$
  - provides label $l_i \in \{-1, +1\}$
  - assumption: each label l_i is correct with probability p_i
  - assume we know p_i

Think about why we can write it in this way?
Hint: it's due to the choice of the label presentation $\{+1, -1\}$

- Weighted majority voting

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

  - Weights that lead to MLE: $\mathrm{w_i} = \ln \frac{p_i}{1-p_i}$ for label l_i
    - You can prove this yourself following the proof of simple majority voting
  - Weights that minimizes error bound: $w_i = 2p_i - 1$ for label l_i
    - (Lemma 1 in Ho et al. ICML 2013)

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

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 (e.g., find MLE)
Fix θ_2, estimate θ_1
Stopping condition: converged, # iterations >= pre-determined threshold, etc

What if Workers' Skills are Unknown

	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.

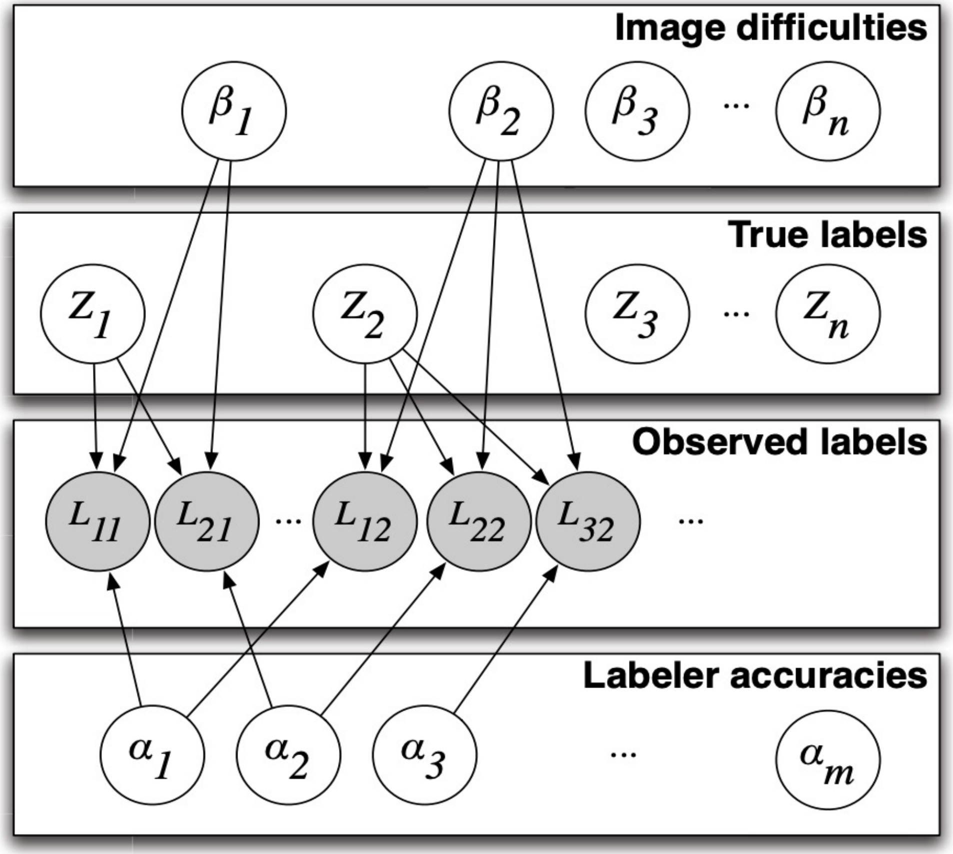
Reminder on the Framework

- 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 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 mean?

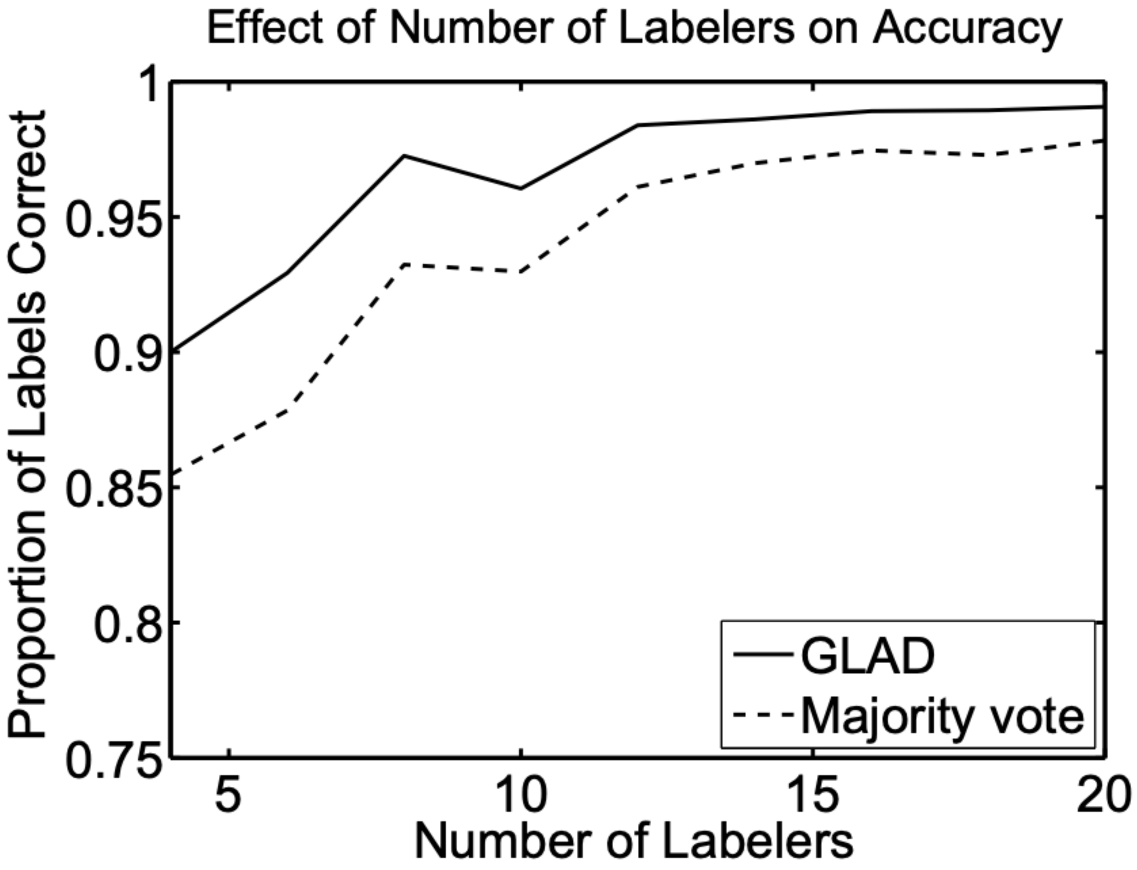
Using EM to find the MLE

- 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

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

Simulation/Experiments



Discussion

- What are your general thoughts about the paper.
- When do you think majority voting would 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

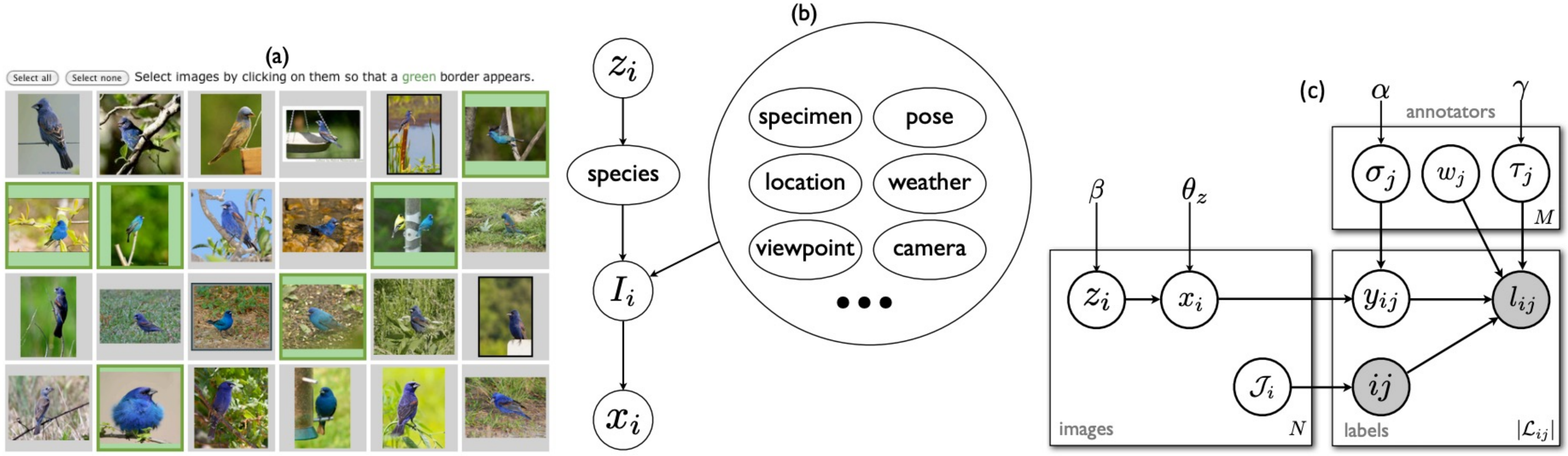
Ground Truth

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 in label aggregation follow this general 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 Lecture

- Read papers that give theoretical guarantees
  - Be prepared for the more math-heavy reading
  - Try to at least understand the formulation/models and the interpretations of the main results