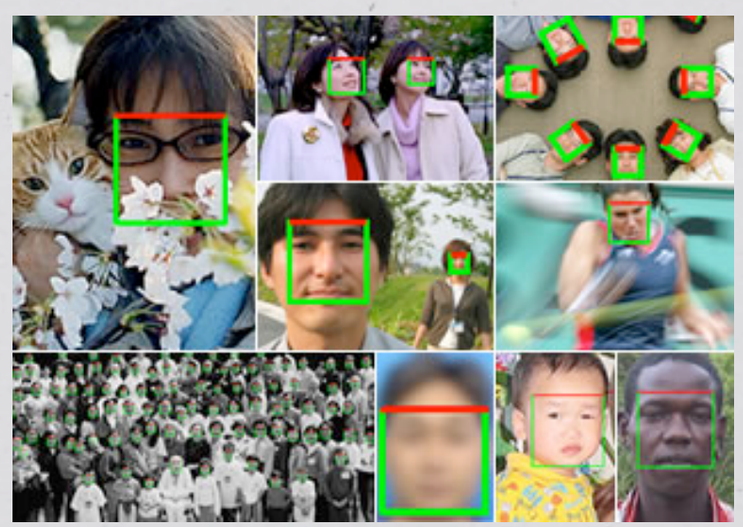
DISTRIBUTED LABELING OF MASSIVE DATASETS

Jacob Whitehill
Paul Ruvolo
Tingfan Wu
Jacob Bergsma
Javier Movellan

AI Seminar 1/26/9

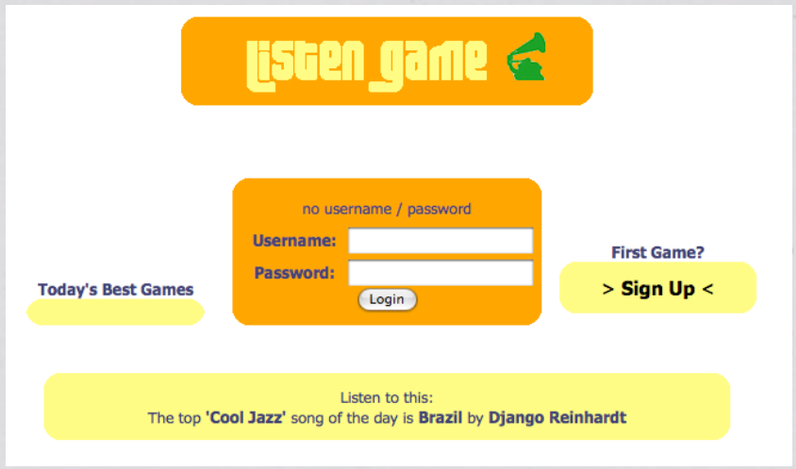
The Data Explosion

* Huge online repositories of data
  * Google images (somewhere in the billions of images)
  * Youtube (13 minutes of new video / second)
* Machine learning algorithms have an immense thirst for data
  * Viola and Jones Face Detector (2001) 4916 training images
  * Omron Face Detector (200?) 5 million training images



Some Hope…

* Games for collecting labels



* Markets for Getting Labels

amazon mechanical turk
beta
Artificial Artificial Intelligence

Labeling Using Mechanical Turk

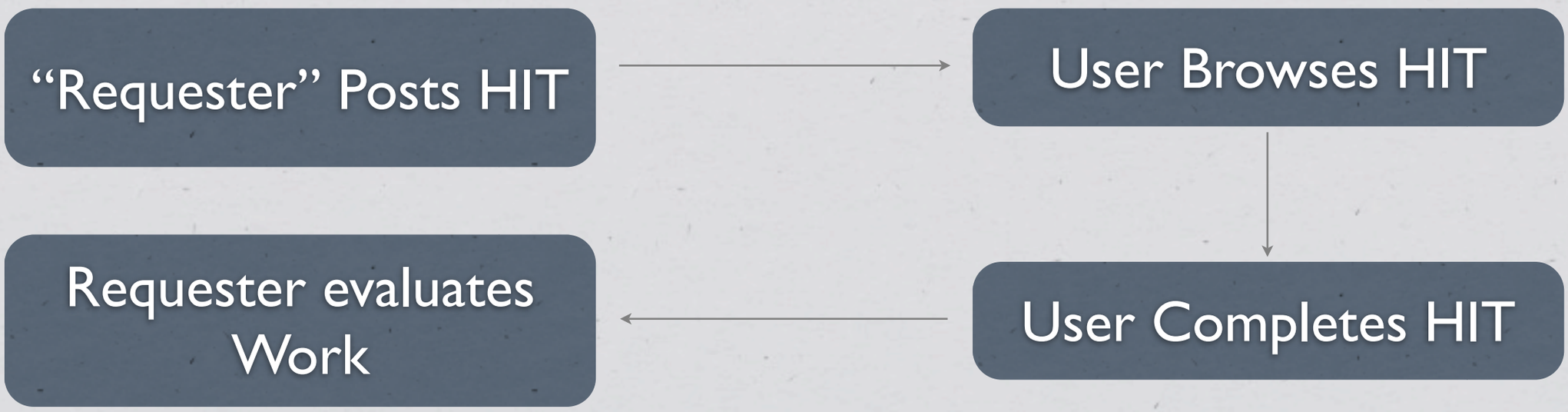
Translate this word or phrase into ${language}

Requester: James Boyle
HIT Expiration Date: Jan 28, 2009 (2 days 23 hours)
Reward: $0.01
Time Allotted: 5 minutes
HITs Available: 4729

Description: Translate a word or phrase from English into ${language}

Keywords: translate, tranlsation, enligsh, language, thai, portuguese, spanish, french, german, greek, arabic, hindi, italian, romanian, dutch, polish

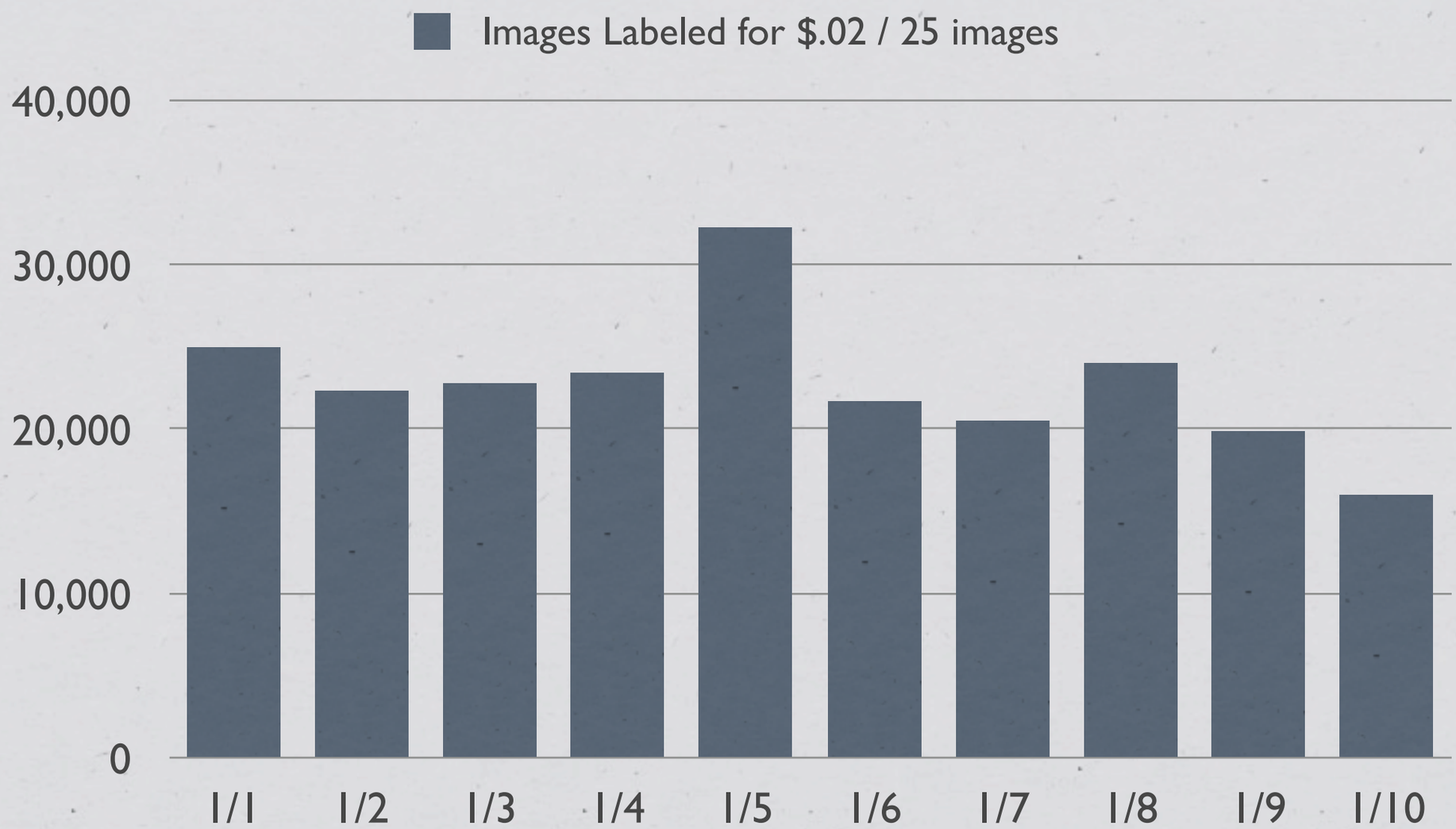
Qualifications Required:
HIT approval rate (%) is not less than 95



Labeling Images using Mechanical Turk

* demo

Economics of Mechanical Turk



Issues in Labeling Large Datasets

* Quality Control
* Confidence in labels
* Rewarding good Workers
* Intelligent sampling

Related Work

- Item Response Theory (e.g., Rasch, Birnbaum):
  - Model both labeler accuracy and image difficulty.
  - But true labels Z are assumed to be known.
- Dawid and Skene (1979):
  - Use EM, but do not model difficulty.
  - As shown in paper (not presented here), difficulty parameters can significantly improve accuracy.

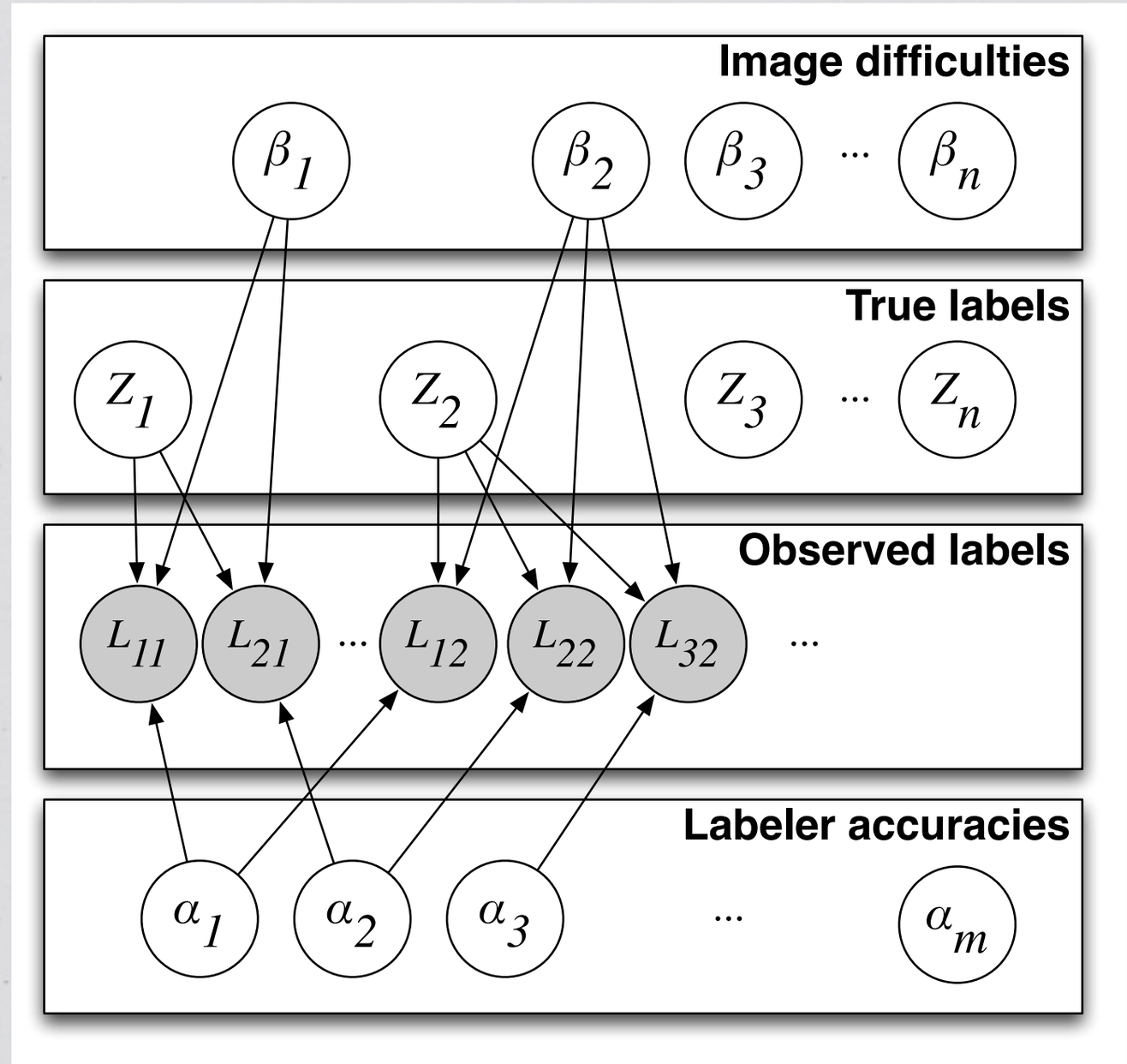
Issues in Labeling Large Datasets

* Quality Control
* Confidence in labels
* Rewarding good Workers

Problem Formulation

* Given image labels from a set of labelers
* Goal:
  * determine accuracy of each labeler (use: give bonus payments)
  * determine difficulty of each image (use: choose images for training)
  * determine belief about each label (use: weight predictions of various labelers differentially)
* Analogous to the problem of giving a bunch of people a test and simultaneously grading each person, estimating the true answers, and assigning difficulties to each item.

BLOG: Bilinear Log Odds Generative



Some Notation

m labelers

n images

$L \in \{-1, 1\}^{m \times n}$ denotes the labelers' responses

L_{ij} denotes the label given by labeler i to image j

$\mathbf{l}_i$ denotes the collection of labels given to image i

$\alpha \in \mathbb{R}^m$ denotes the vector of accuracies of each labeler

$\beta \in \mathbb{R}^{+n}$ denotes the vector of difficulties of each image

$Z \in \{-1, 1\}^n$ denotes the vector of true labels of each image

Likelihood

$$p(L_{ij} = Z_i | \alpha_j, \beta_i) = \frac{1}{1 + e^{-\alpha_j \beta_i}}$$

$$p(L_{ij} | \alpha_j, \beta_i, Z_i) = p(L_{ij} = Z_i | \alpha_j, \beta_i)^{Z_i = Lij} + (1 - p(L_{ij} = Z_i | \alpha_j, \beta_i))^{Z_i \neq Lij}$$

* Model behavior
  * Infinitely hard image will give a 50% chance of correctness
  * Infinitely good labeler (alpha very large) has a 100% chance of correctness
  * Infinitely good adversarial labeler (alpha very negative) has a 0% chance of correctness
* This model is also utilized in item response theory

Marginal and Conditional Distributions of Interest

* Determine the distribution of true labels given the labelers' responses

$$p(Z|L) = p(Z) \int \dots \int p(L|Z,\alpha,\beta)p(\alpha)p(\beta)d\alpha d\beta$$

Intractable!

* Determine the distribution of accuracies and difficulties conditioned on the labelers' responses

$$p(\alpha,\beta|L) \quad \propto \quad p(\alpha)p(\beta)\sum_{Z} p(L|Z,\alpha,\beta)p(Z)$$

Intractable to compute full distribution, but we can maximize

$$= \quad p(\alpha)p(\beta)\Pi_j \left[\sum_{z' \in \{-1,+1\}} p(\mathbf{l}_j|\alpha,\beta_j,Z_j = z') \right]$$

Inference Using EM

* We can use Expectation-Maximization (EM) to maximize

$$p(\alpha, \beta | L)$$

* E-Step: Update $p(Z|\alpha, \beta, L)$

* M-Step: Maximize $E[\ln p(L, Z\alpha, \beta)]$ w.r.t. α, β.

Inference Using EM

* EM finds MAP estimates of α, β.

* For the Z, we take the probability estimates $p(Z|\alpha, \beta, L)$ from the last E-Step.

E-Step

* Calculate the distribution of the hidden variables (Z) given L and the estimates of α, β from the last M-Step:

$$
\begin{aligned}
p(z_j|\mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= p(z_j|\mathbf{l}_j, \boldsymbol{\alpha}, \beta_j) \\
&\propto p(z_j|\boldsymbol{\alpha}, \beta_j)p(\mathbf{l}_j|z_j, \boldsymbol{\alpha}, \beta_j) \\
&\propto p(z_j)\prod_i p(l_{ij}|z_j, \alpha_i, \beta_j)
\end{aligned}
$$

where $p(l_{ij}|z_j, \alpha_i, \beta_j)$ can be evaluated in terms of *probability of correctness* (discussed earlier).

M-Step

* The expression $p(\mathbf{l},\mathbf{z} \mid \alpha,\beta)$ may contain a huge number of variables.
* However, any particular given label l_{ij} depends only on α_i,β_j, and z_j.
* The given labels $\{ l_{ij} \}$ are conditionally independent given Z, α, β.

$$
\begin{aligned}
& Q(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\
& \quad = \quad E\left[\ln p(\mathbf{l}, \mathbf{z} | \boldsymbol{\alpha}, \boldsymbol{\beta})\right] \\
& \quad = \quad E\left[\ln \prod_j \left(p(z_j) \prod_i p(l_{ij} | z_j, \alpha_i, \beta_j) \right)\right] \\
& \qquad \text{since } l_{ij} \text{ are cond. indep. given } \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}
\end{aligned}
$$

M-Step (cont.)

* To maximize the auxiliary function Q, we use gradient ascent.
* The logistic probability of correctness readily lends itself to this operation.

Using Prior Information

* If true labels (Z) are (somehow) known for certain images, then these labels can be "clamped" to their correct values.
  * Set $p(Z_j=z_j)$ for these images to be very high for the appropriate class.
* Priors over α and β can also be easily set.

Runtime Performance

* N images, M labelers, T total labels
* Each E-Step is linear N+T
* The M-Step requires repeated calculation of Q and ∇Q
  * Estimating Q and ∇Q is linear in N+M+T
  * Number of iterations for convergence will vary.

Runtime Performance

* On a set of 1,000,000 labels, BLOG converged in about 8 minutes on a single-core.
* Algorithm is parallelizable.
* When appending new data to L, it is possible that convergence will be faster when good starting values for α, β are known.

Simulation

* We demonstrate the utility of BLOG using simulation.
    * The data are drawn according to the generative model on which BLOG is based.

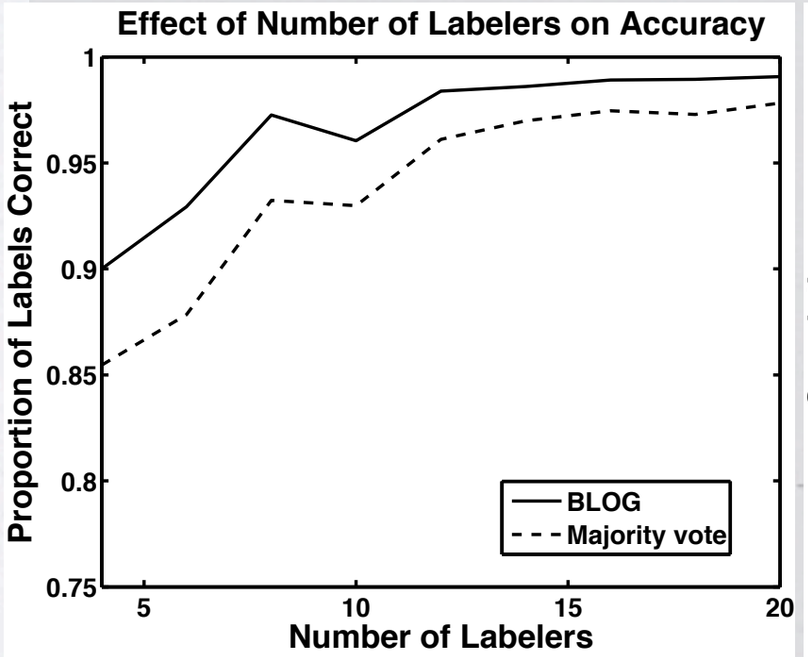
Simulation

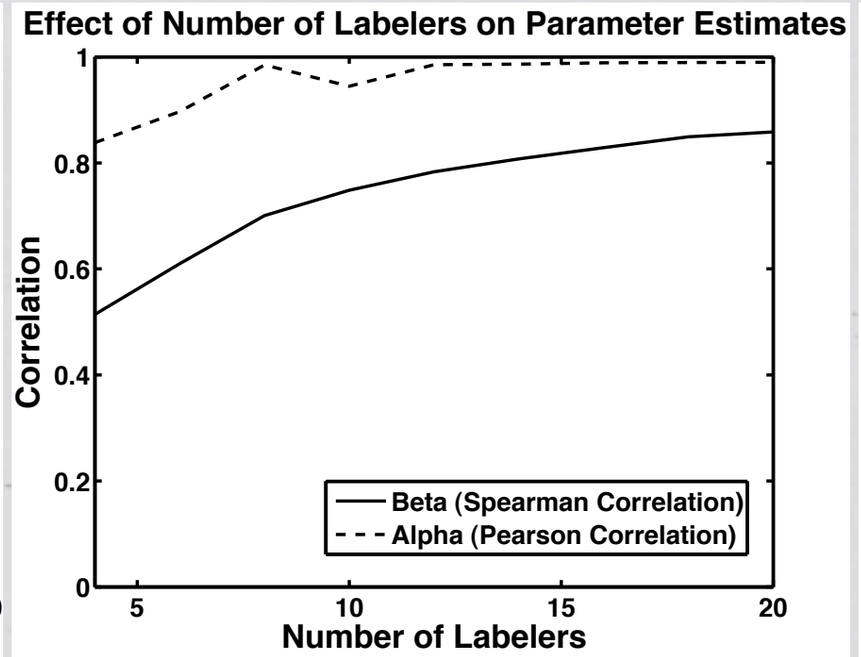
* 2000 images (N = 2000)
* Up to 20 labelers (4 <= M <= 20)
* Model:
  * Ability $\alpha \sim$ Gaussian(1,1)
  * Difficulty $\beta \sim$ LogGaussian(1,1)
  * True labels $Z \sim$ Uniform({0,1})
  * L ~ BLOG(α, β,Z)

Simulation

* On each simulation run, MAP estimates α, β (and Z) were calculated.
* Correlations with true α, β, and Z values were calculated as a function of M (number of labelers).
* Correlations were averaged over 40 simulation runs.

Simulation 1: Results





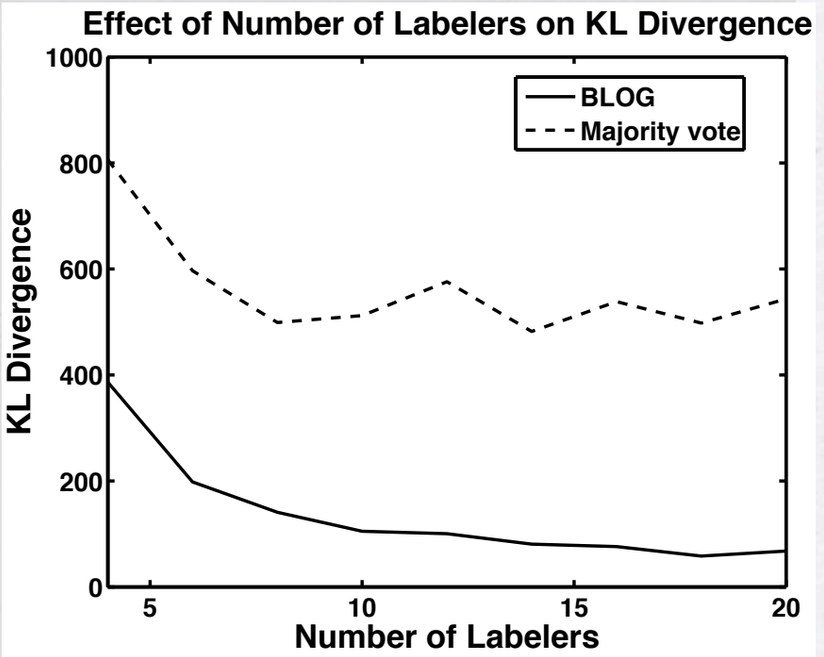


Figure 2. **Left:** The accuracies of the BLOG model versus simple voting for inferring the underlying class labels on simulation data. **Middle:** The ability of BLOG to recover the true alpha and beta parameters on simulation data. **Right:** The ability of BLOG versus a simple voting heuristic to predict accurate confidence estimates of the image labels on simulated data.

The fact that BLOG outperforms Majority Vote in the left graph means that BLOG inferred the correct label even when the true image was the *minority* opinion of the given labels.

Empirical Results: MTurk Data

* We collected labels of face gender:
  * 10,000 images
  * 10 labels per image
* Using the face patches and associated labels, we train an automated gender classifier using a single-cascade Viola-Jones architecture.

Gender Classification

* Question: Does BLOG help us create a better automatic gender classifier than the Majority Vote heuristic?
* Performance metric:
  * Area under ROC curve measured on an independent validation set.

Three Scenarios

* From the 100,000 given labels we collected, we studied two conditions:

1. Adversarial labelers: a fraction of labelers purposely labeled images incorrectly (flip all bits).

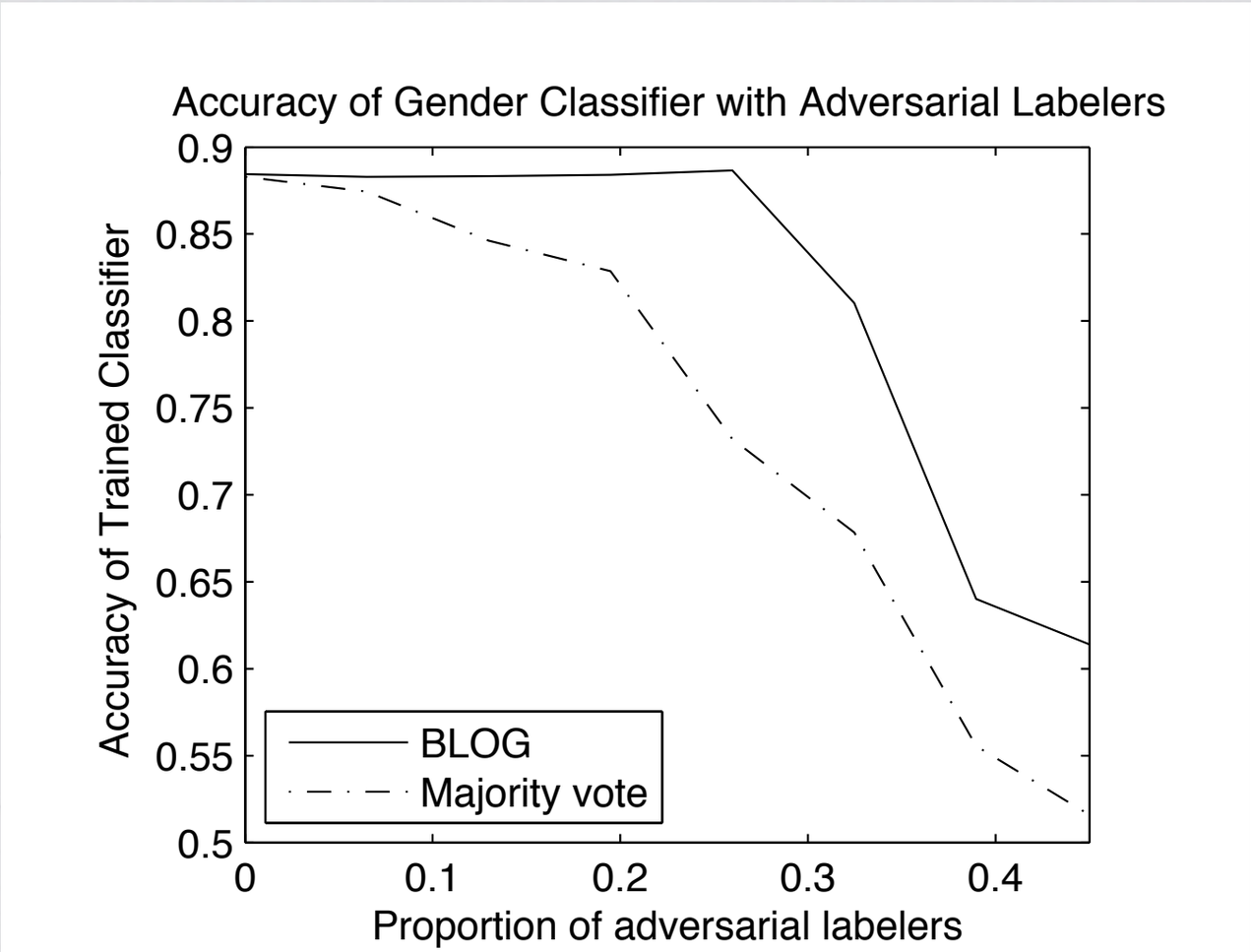
2. Noisy labelers: a fraction of labelers gave random or near-random labels (flip some bits).

3. Unmodified labelers: the raw labels.

Experimental Setup

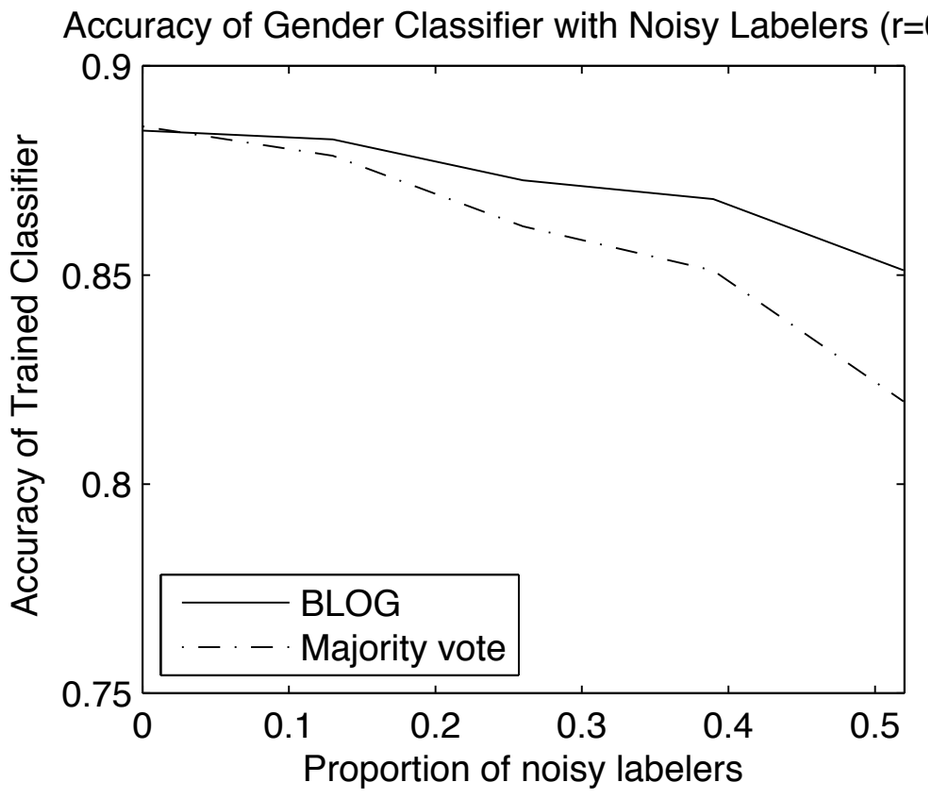
- Infer training labels:
  - $Z^1 = \text{BLOG}(L)$
  - $Z^2 = \text{MajorityVote}(L)$
- Train gender classifiers:
  - $C^1 = \text{ViolaJones}(\text{Faces}, Z^1)$
  - $C^2 = \text{ViolaJones}(\text{Faces}, Z^2)$
- Compute accuracies A^1 and A^2 and compare.

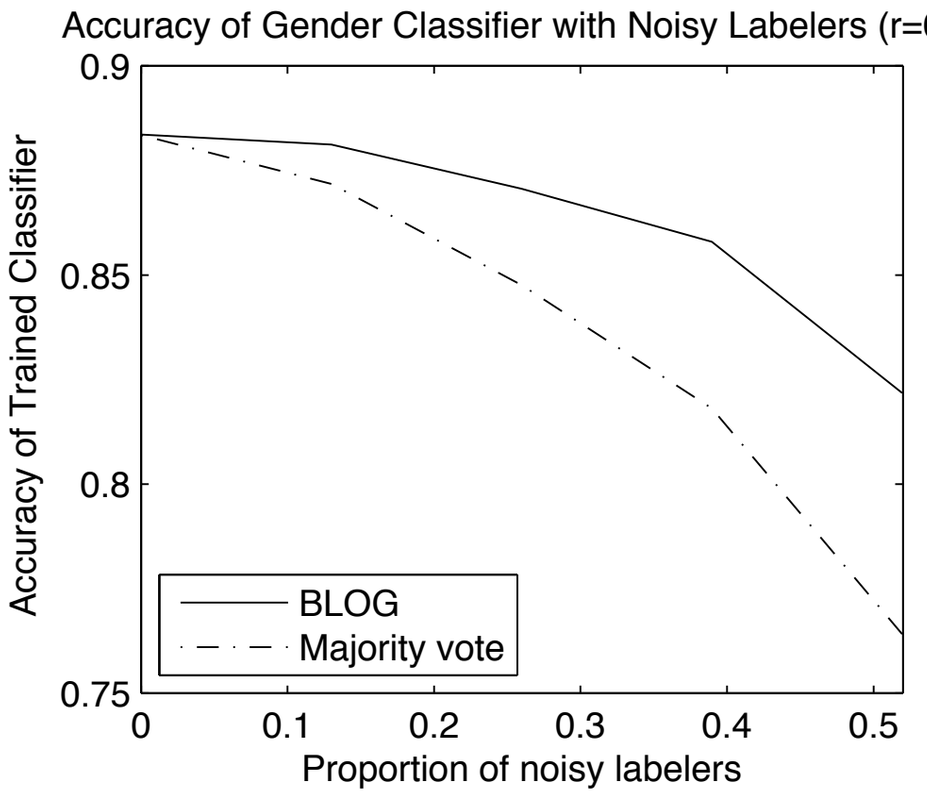
Results: Adversarial Labelers



* As proportion of adversarial labelers increases, BLOG maintains substantial advantage in accuracy.

Results: Noisy Labelers





Results: Unmodified Labelers

* Using the raw (unmodified) MTurk labels of gender, BLOG and Majority Vote delivered comparable performance.
  * Area under ROC: ~88%
* The gender labeling task may have been too easy.
* To show the benefit of BLOG, we may need a task where some people are good labelers and some people are bad.

Future Work

* Intelligent sampling (active learning problem with multiple noisy oracles)
* Dynamic pricing
* Continuous response variables
* Modeling “tricky” questions. (e.g. What is the plural of octopus?)

Questions

Heuristics

* Voting
  * Establishing confidence in labels (strength of agreement?)
  * Evaluating individual labelers (agreement with majority?)
  * Evaluating difficult of instances (labeler disagreement?)
* Unclear how to justify these heuristics in a unified fashion

M-Step (cont.)

* The probability $p(l_{ij} | z_j, \alpha_i, \beta_j)$ can be derived from the probability of correct response:
  * If z_j=1, then l_{ij}=1 iff Correct.
    then l_{ij}=0 iff Incorrect.
  * If z_j=0, then l_{ij}=1 iff Incorrect.
    then l_{ij}=0 iff Correct.

M-Step (cont.)

* Derivative w.r.t α (Ability): $dQ/d\alpha =$

$$\sum_j \left(p^1 l_{ij} + p^0(1 - l_{ij}) - \sigma\right) \beta_j$$

* Derivative w.r.t. β (Difficulty): $dQ/d\beta =$

$$\sum_i \left(p^1 l_{ij} + p^0(1 - l_{ij}) - \sigma\right) \alpha_i$$