2. Crowdsourcing for Datafication

Tutorial slides are found at

http://goo.gl/Amif93

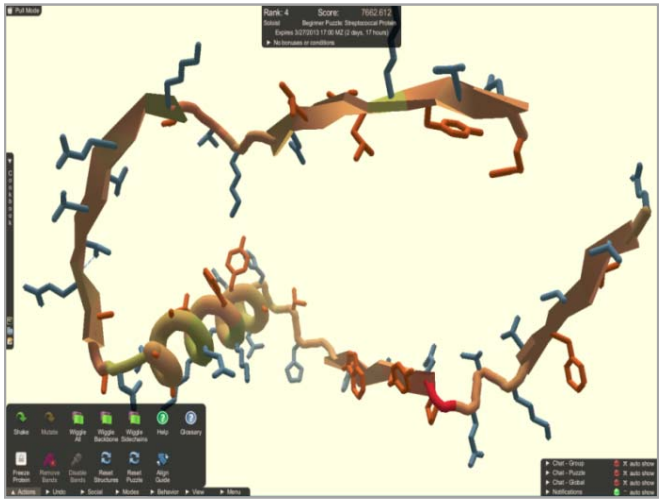
Obtaining "Human Judgment": Crowdsourcing Markets or Embedded Tasks

- Two approaches to asking people to make judgments:
  1. Direct Approach: Using a crowdsourcing market
     - Asking workers to answer questions with monetary rewards
  2. Indirect Approach: Task embedding
     - Task enforcement：embedding a target task into another task users cannot evade: ex) reCAPTCHA
     - Gamification：Game playing activities for fun are designed to become completing tasks for a different purpose

*Passive Approach: Extracting human judgments from existing resources such as social media

Gamification: A Useful Design Pattern for Crowdsourcing

- Players activities for enjoying a game are actually completing tasks for a different purpose
  - Game With A Purpose; GWAP
  - If successful, cost to complete tasks will become almost zero



fold.it

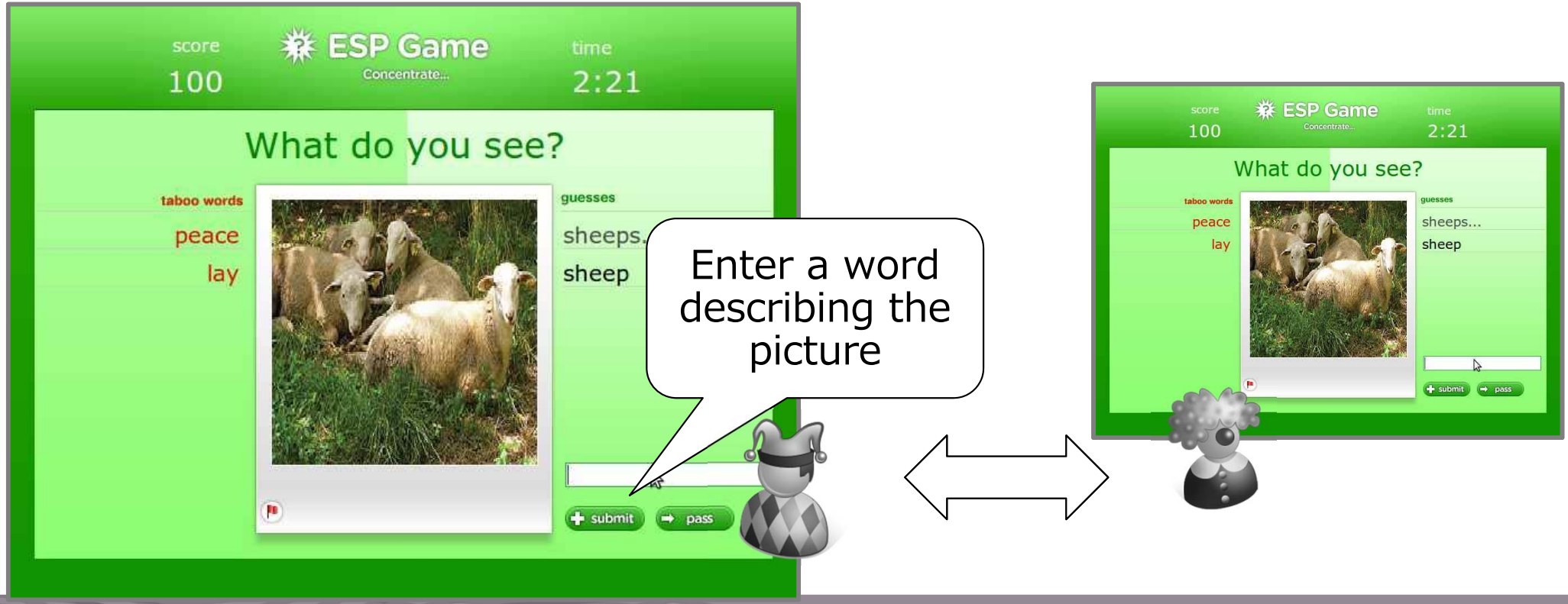
- Ex: Foldit: a game to find folding configurations of proteins
  - Gamification of a hard optimization problem
  - Hundreds of thousands of registered workers
  - Outperforming automatic prediction algorithms
  - A successful example of human computation leading to important scientific discoveries

Framework for Gamification: Output Agreement and Input Agreement

- The key to successful gamification is how cleverly we can embed the task into a game
  - It is not obvious how to make a game fun
  - There is no established methodology for successful gamification
- Several patterns used in gamification:
  - Output agreement
  - Input agreement

Example of Output Agreement
ESP Game: A Game for Image Tagging

- A “cooperative” game by two players on the Web
- Two players see the same image and independently give words to describe the image. If a common word is entered by the both players, points are given to them



Output Agreement:
Encouraging workers to agree on their outputs

- The hidden purpose behind the ESP game:
  - Collecting ``typical'' keywords for describing images
  - Avoiding too general or too specific keywords
- Output agreement
  - Rewards are given to the players if they can reach agreement on their outputs
  - Each player guesses keywords her partner will come up with
  - This leads to find a typical keyword for representing the image

Input Agreement :
Overcoming the Difficulty in Output Agreement

- Difficulty in output agreement: Workers' outputs rarely coincide when the set of possible outputs is large (It assumes that workers outputs coincide with a certain probability)
- Input agreement:
  - Each of two players receives an input (The two inputs can be the same or different)
  - The goal of the players is guess if they have received the same input after communicating to each other
  - If the communication is successful, they will guess correctly (and we can extract useful information from the communication)

Example of Input Agreement:
Tag-A-Tune: Game for Tagging Music

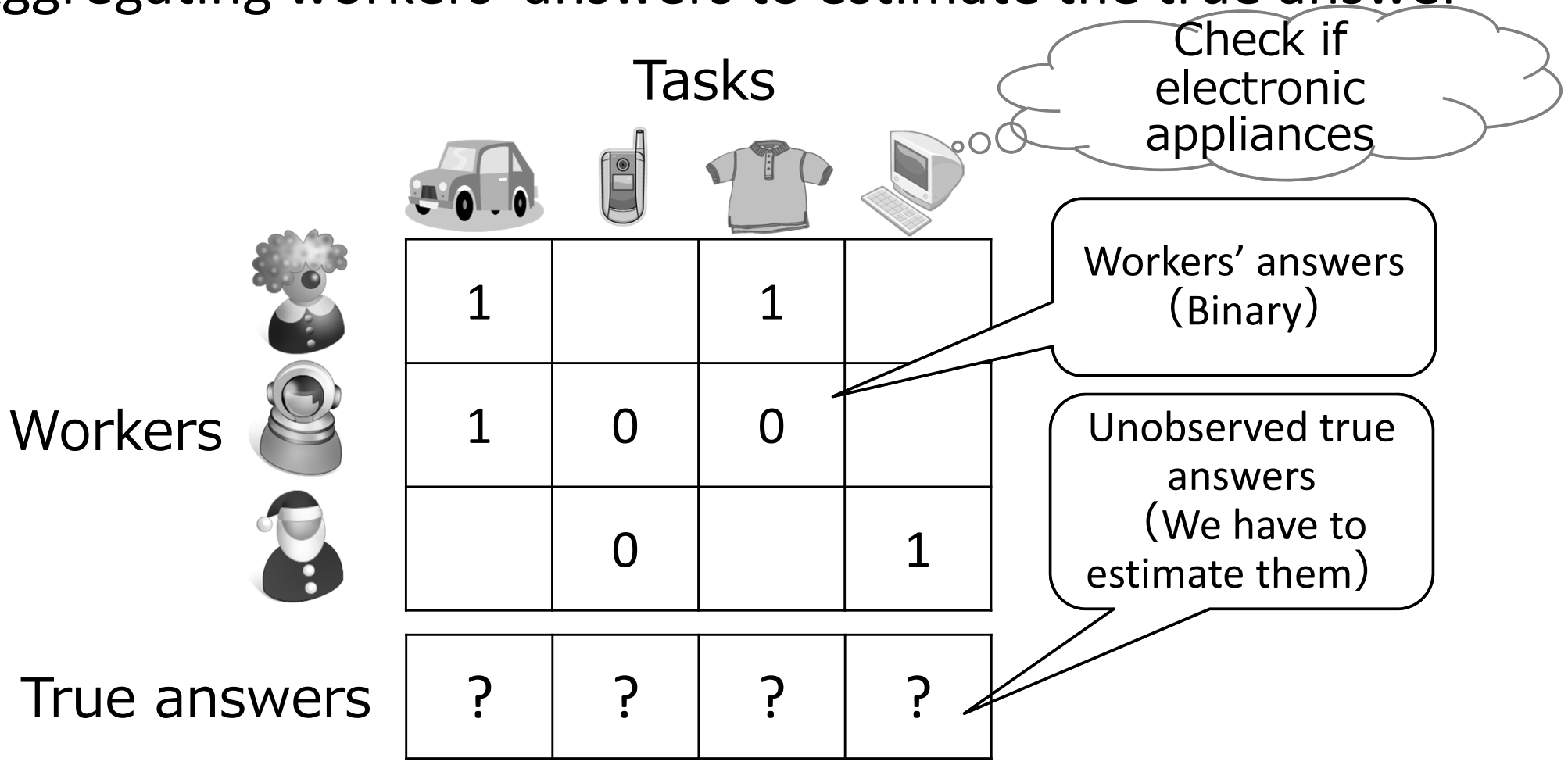
- Each of two players receives an audio clip
  - Audio clips they receive can be the same or different
- The goal of the players: Guessing whether the audio clips they have been given are the same or not
  - Before answering the question, players exchange keywords that represent the music, such "dynamic" or "peaceful"
  - Players receive points if their answers are correct
  - Note the system knows the true answer (whether the two audio clips are the same) but the goal is collecting keywords entered by the players

Problem in Crowdsourcing Markets
Variance in Quality among Workers

- In a crowdsourcing market, not all workers do not have ability and dedication to complete the task
  - Spam workers: work insincerely only to obtain a reward
- Quality control mechanisms in using a crowdsourcing market
  - Filtering workers based on the number of tasks they have completed or approval rates of their results
  - Qualification tests (hard to prepare for each kind of tasks)
  - Gold standard (not always available)

Quality Control: Introducing Redundancy and Using Statistical Inference

- Assigning the same task to multiple workers
- Aggregating workers' answers to estimate the true answer

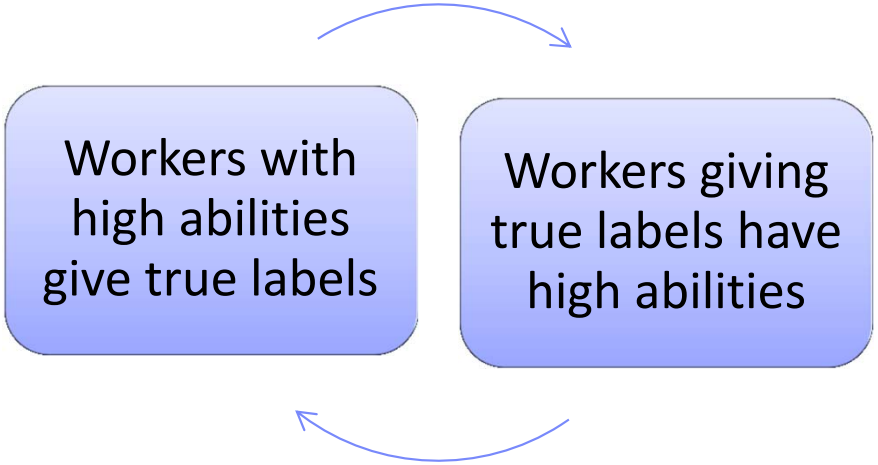


Obtaining High Quality Answers from Redundant Answers

- Aggregating answers using statistical inference
  - Majority voting, averaging
- Assuming different workers have the same ability to give correct answers
  - Every worker has the same power to vote
  - However, their abilities are different actually
- If we know workers' abilities (probabilities of giving true labels), we can use them to weight their votes

Joint Estimation of Workers' Abilities and True Labels

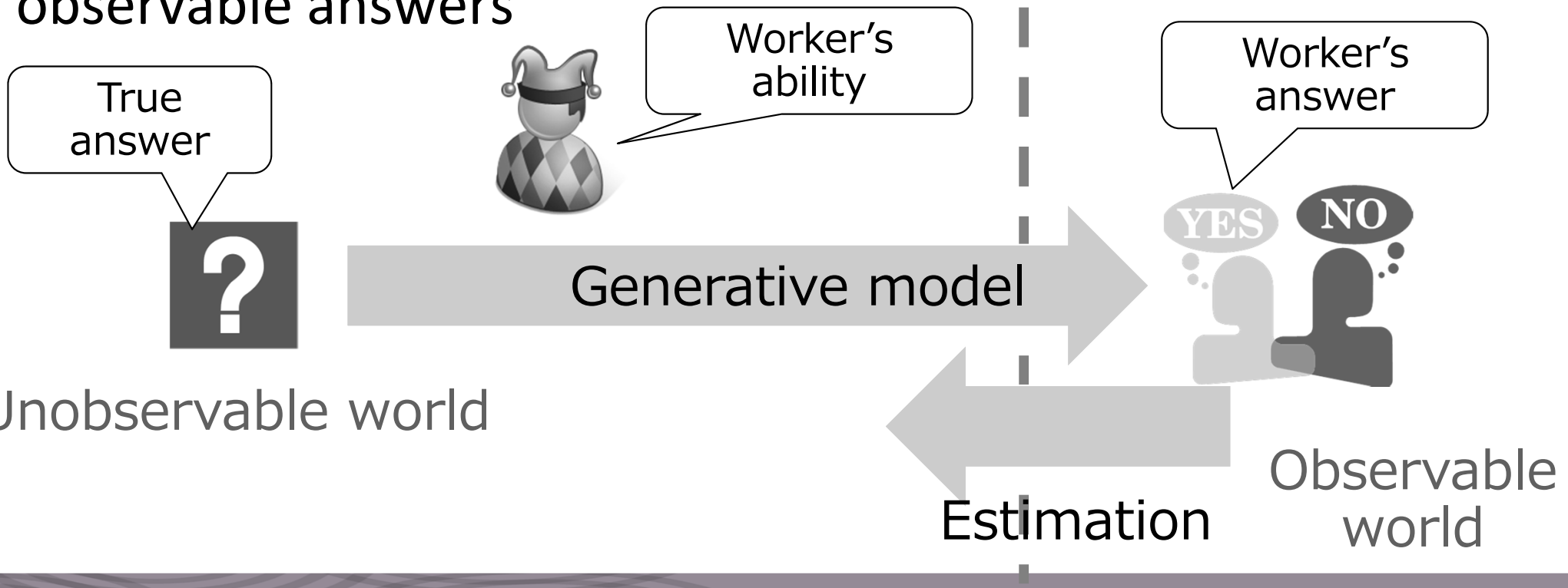
- Usually we know neither workers' abilities nor true labels
- Mutual dependency between abilities and true labels



- Dawid&Skene (1979) proposed the latent class model to alternatingly estimate true labels and workers' abilities using the EM algorithm

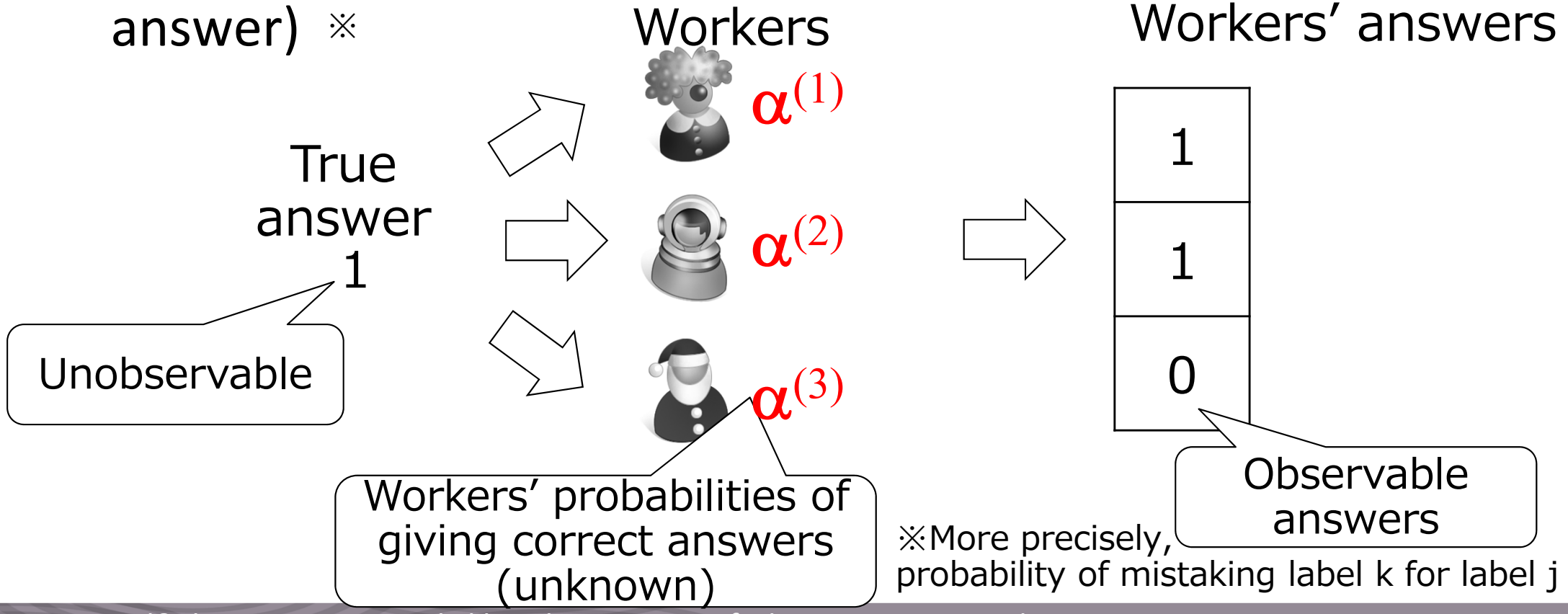
Statistical Modeling Approach: Generative Model with True Answers as Latent Variables

- Probability model to represent the generative process of workers' answers from the true answer, depending on worker specific parameters
- Statistical estimation of the unobserved true answer from observable answers



Latent Class Model:
A Basic Model to Represent Workers' Abilities

- A basic model for quality control:
  - Proposed in 1979 for the purpose of medical diagnosis
  - Introducing worker's ability (probability of giving the correct answer) ※

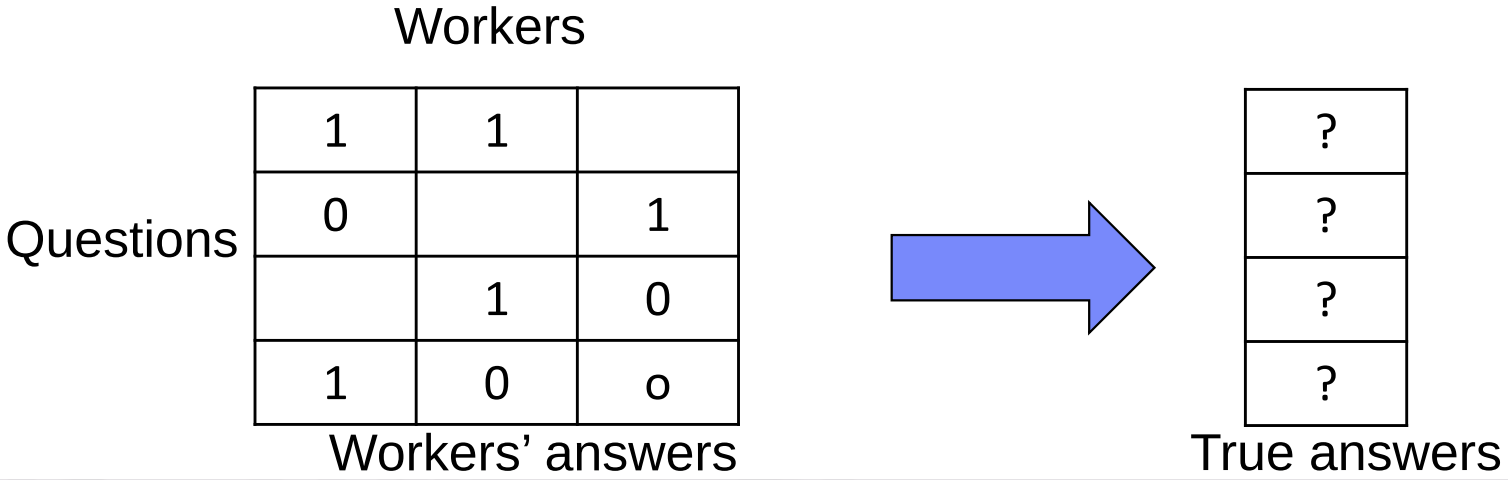


※More precisely, probability of mistaking label k for label j

Dawid&Skene. Maximum Likelihood Estimation of Observer Error-rated Using the EM Algorithm. *Applied Statistics*, 28(1), pp.20-28, 1979

Formal Problem Setting for the Latent Class Model

- N questions, J workers
  - A worker does not necessarily answer all questions
  - $J_i \subseteq \{1, \ldots, J\}$: Set of workers who answered question i
- $t_i \in \{0, 1\}$ $(i \in \{1, \ldots, N\})$: true answer for question i
- $y_{ij} \in \{0, 1\}$ $(j \in J_i)$: answer for question i given by worker j
- Our goal is to infer $\{t_i\}$ given $\{y_{ij}\}$



Modeling Worker's Ability

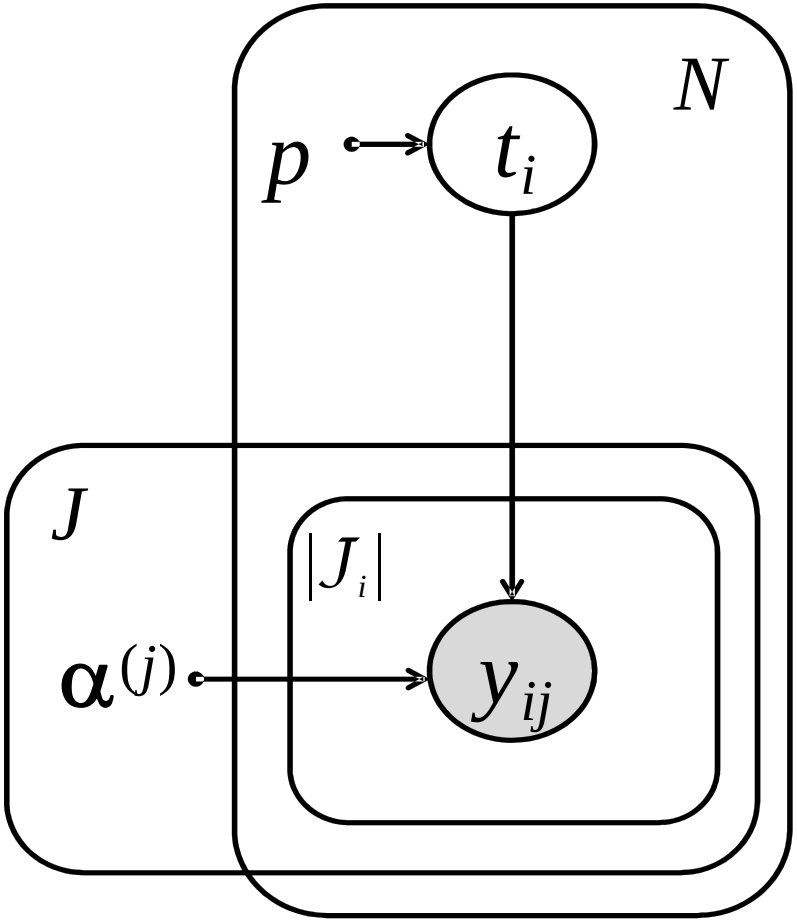
- Worker's confusion matrix

Worker's answer	True answer	
	1	0
1	$a_1^{(j)}$	$a_0^{(j)}$
0	$1\text{-}a_1^{(j)}$	$1\text{-}a_0^{(j)}$

$\alpha_t^{(j)}$:probability that worker j gives answer 1 when the true answer is t

- Each element represents the probability of workers' answer given the true answer
- A worker who has large diagonal elements is considered to have a high ability

The Graphical Representation of the Latent Class Model



- Prior probability of true answer t_i

$$p(t_i) = p^{t_i}(1-p)^{(1-t_i)}$$

- Conditional probability of worker's answer given the true answer

$$p(y_{ij}|t_i = 1) = (\alpha_1^{(j)})^{y_{ij}}(1-\alpha_1^{(j)})^{(1-y_{ij})}$$
$$p(y_{ij}|t_i = 0) = (\alpha_0^{(j)})^{y_{ij}}(1-\alpha_0^{(j)})^{(1-y_{ij})}$$

Using the EM algorithm to Alternatingly Estimate Workers' Abilities and True Answers

- The model contains hidden variables (true answers)
  ⇒Use the EM algorithm to alternatingly infer hidden variables and model parameters

1. Initialization：Estimate the expected values of $\{t_i\}$ using majority votes
2. M-step： Estimate parameters $\{\alpha^{(j)}\}$ by using the current expectations of $\{t_i\}$
3. E-step：Estimate the expected values of hidden variables $\{t_i\}$ by using the current estimation of $\{\alpha^{(j)}\}$
4. Repeat steps 2 and 3 until convergence

Filtering Workers based on their Abilities

- Workers' confusion matrices contain information on their abilities
- Workers who always give wrong answers are actually "good" workers
- Using worker's expected label to calculate the ability of the worker [Ipeirotis 2010]

$$p(t_i = 1 \mid y_{ij} = 1) = \frac{p(y_{ij} = 1 \mid y_i = 1)\, p(y_i = 1)}{p(y_{ij} = 1)}$$

$$= \frac{\alpha_1^{(j)} p}{\alpha_1^{(j)} p + \alpha_0^{(j)} (1 - p)}$$

- Spam workers give their answers independent of true answers, that is

$$a_1^{(j)} = a_0^{(j)}$$

- Spammer score [Raykar 2011]

$$S^{(j)} = (a_1^{(j)} - a_0^{(j)})^2$$

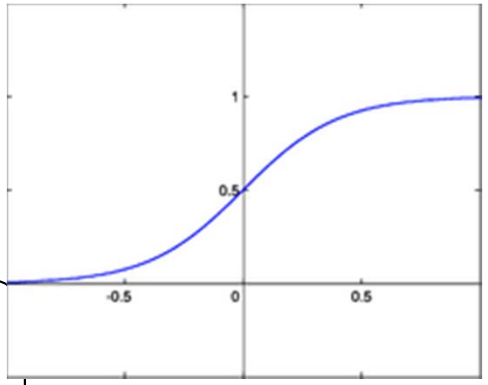
Extension of the Latent Class Model: Introducing Task Difficulties

- Introducing task difficulties as well as worker abilities
- Modelling the probability of worker j correctly answering question j

$$Pr[y_{ij} = t_i] = \frac{1}{1 + \exp(-w_j d_i)}$$

Worker ability $w_j \in (-\infty, +\infty)$

Task easiness $d_i \in [0, +\infty)$



  - When ability=0, probability of correct answer=0.5. When ability and easiness are large, the probability of correct answer is close to 1
  - Estimation is made by the EM algorithm

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

Further Extension of the Latent Class Model: Modeling Annotators' Different "Areas of Strength"

- In the models discussed so far:
  - Higher ability leads to higher probability of correctness
  - A harder task leads to lower probability of correctness
- Modeling "affinity" between a worker and a task

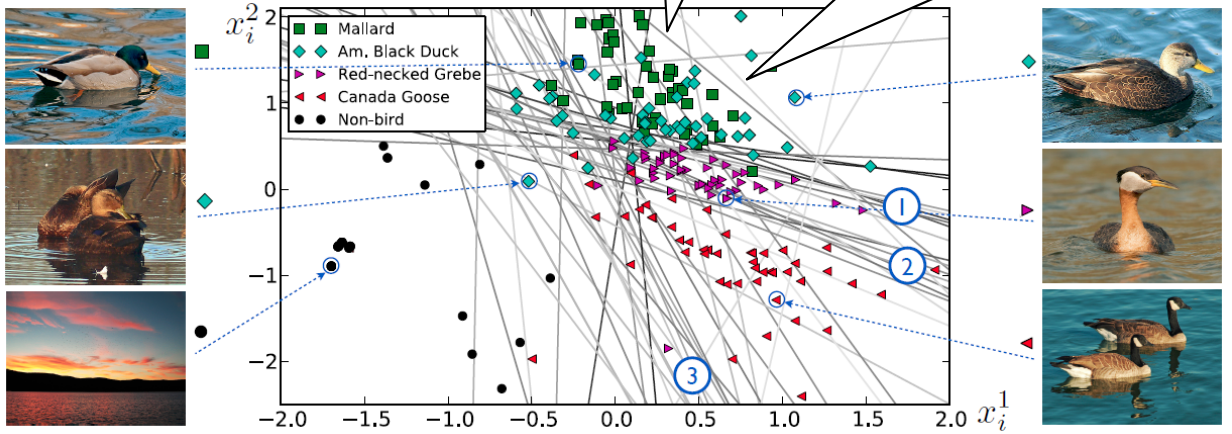
Worker's characteristics

Task characteristics

When $\mathbf{w}_j \cdot \mathbf{x}_i > \theta$

Threshold

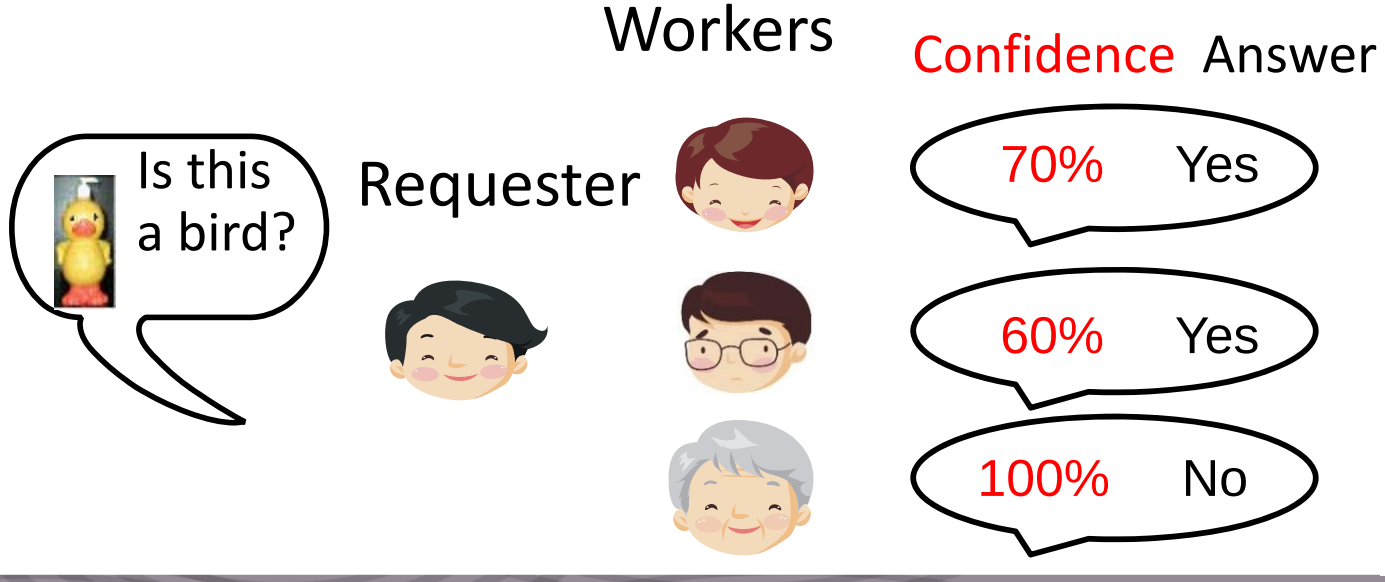
the worker answers 1

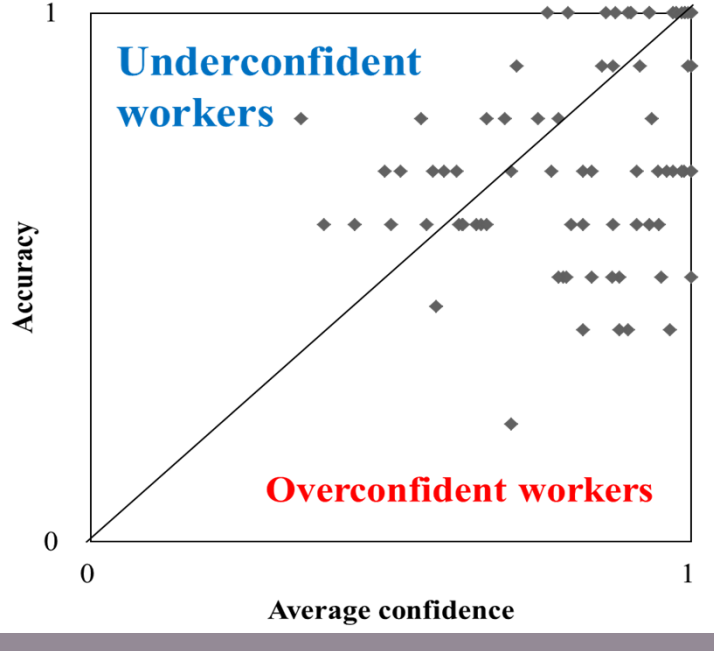


Welinder *et al*. The Multidimensional Wisdom of Crowds. *NIPS* 2010.

Further Extension of the Latent Class Model: Introducing Workers' Confidence

- Asking workers about ***confidence*** in their answers as well
  - In general, if a worker is more confidence about her answer, it is more probably correct
- Overconfident/underconfident workers
- A probabilistic model that considers the differences among workers in their accuracy of confidence judgments



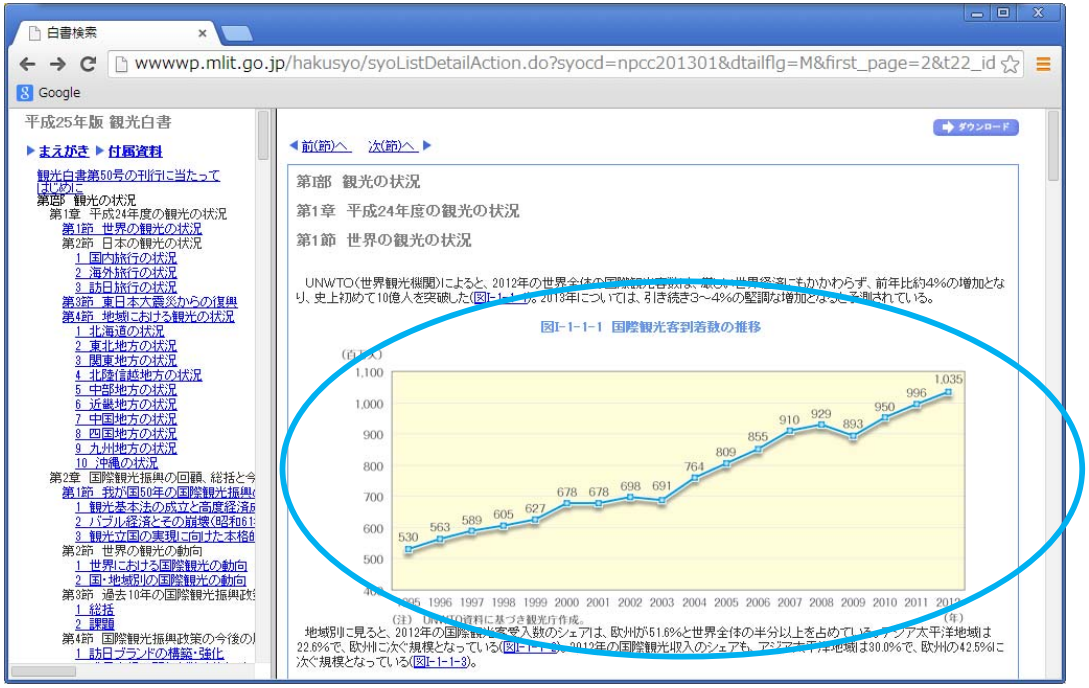


Sensing by Crowdsourcing:
A Human as a Sensor

- Crowd sensing/participatory sensing
  - Data collection framework considering a human as a sensor
  - Data collection involves human judgment
    - Simply collecting data from a sensor attached to a human without any human intervention is not called crowd sensing
- Example of crowd sensing:
  - CreekWatch (IBM)：Monitoring water pollution using cell phones
  - CarTel (MIT)：Monitoring traffic jams using data collected from cars
  - Noricell (MS)：Monitoring traffic noise and road surface using cell phones

Example of Crowdsourcing for Datafication:
Turning Legacy Open Data Machine-Readable

- Published data that are not always readily reusable
- Data charts are often designed to be understood by people, not by computers

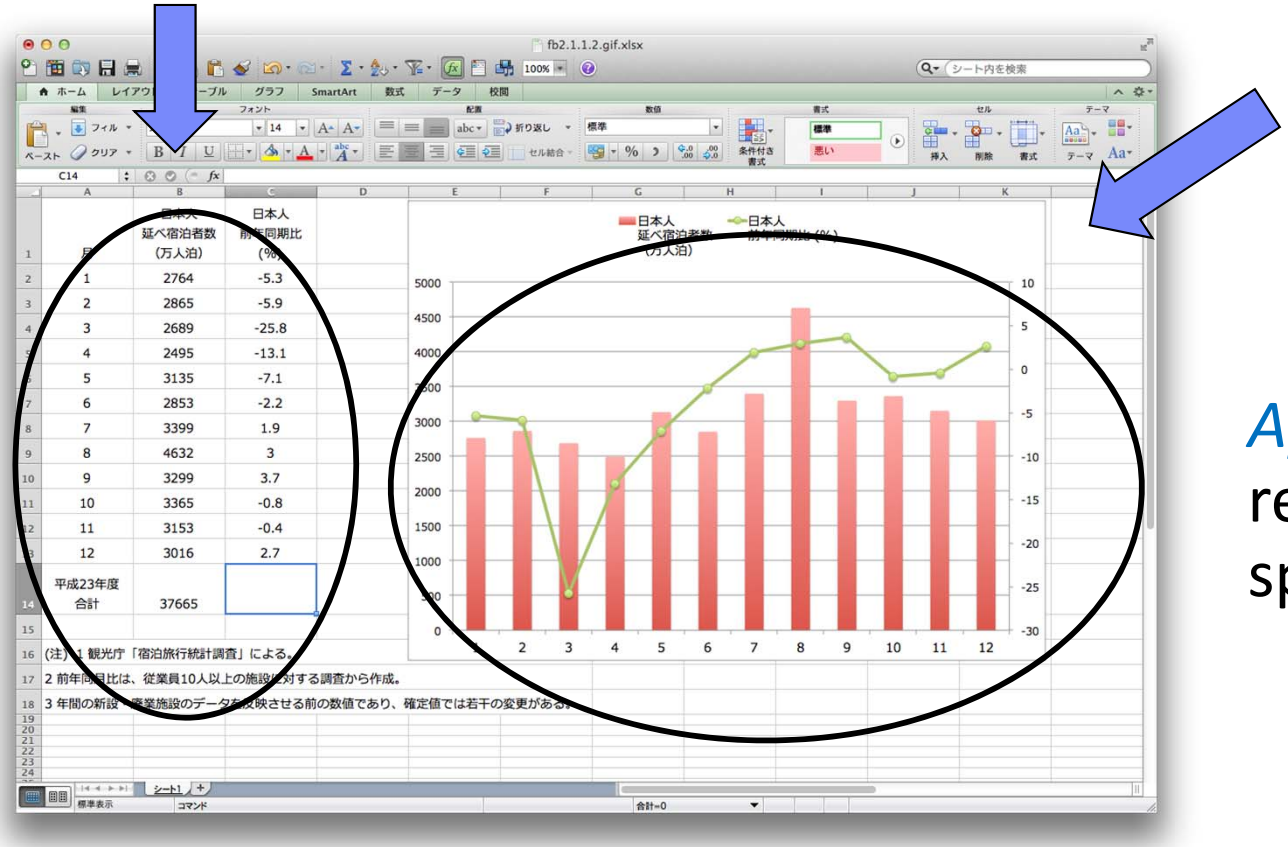


Embedded chart in an image format

Example of Crowdsourcing for Datafication: Extracting Data from Chart Image

- Ask workers to visually reproduce a chart image in a spreadsheet

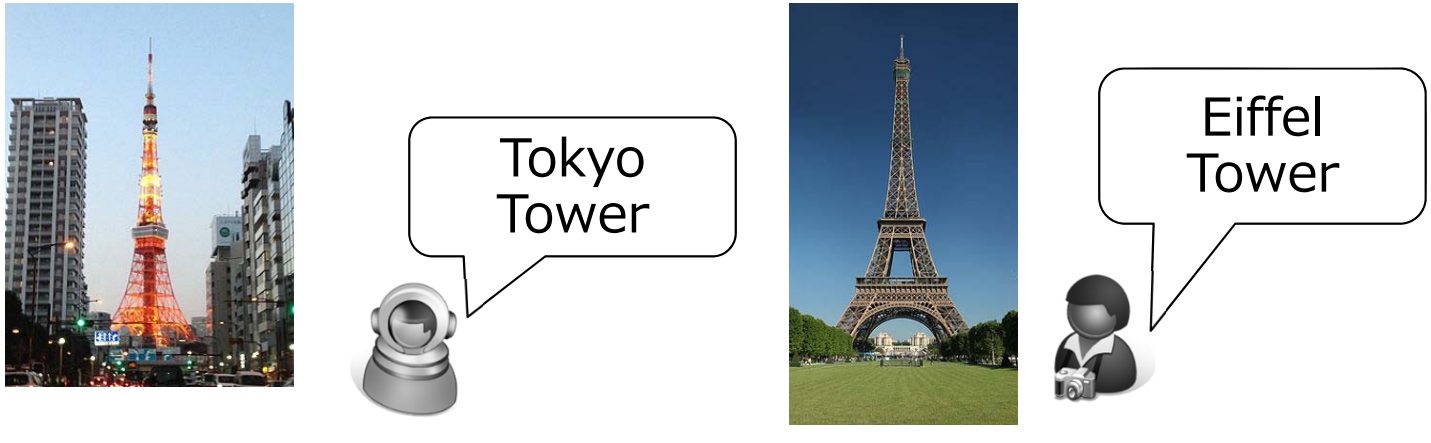
  Hidden purpose:
  extract data from image



Apparent purpose:
reproduce chart in a spreadsheet

Using Data Labeled by Crowdsourcing in Supervised Machine Learning

- Supervised learning requires training data
  - Input-output (gold answer) pairs
- Using crowdsourcing to collect training data
  - NLP: Web page categorization, information extraction
  - Image processing: Image tagging for retrieval, object recognition



http://en.wikipedia.org/wiki/Tokyo_Tower
http://en.wikipedia.org/wiki/Eiffel_Tower

Supervised Learning: Learning a Prediction Model from Input-Output Pairs

- Training data are given as input-output pairs
  $\{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,N}$
- Estimate the model
  $y_i \doteqdot f(\mathbf{x}_i)$
  that explains the input-output relation

Learning from Crowdsourced Data: Using Estimated True Answers

- Obtain (possibly incorrect) answers from multiple workers

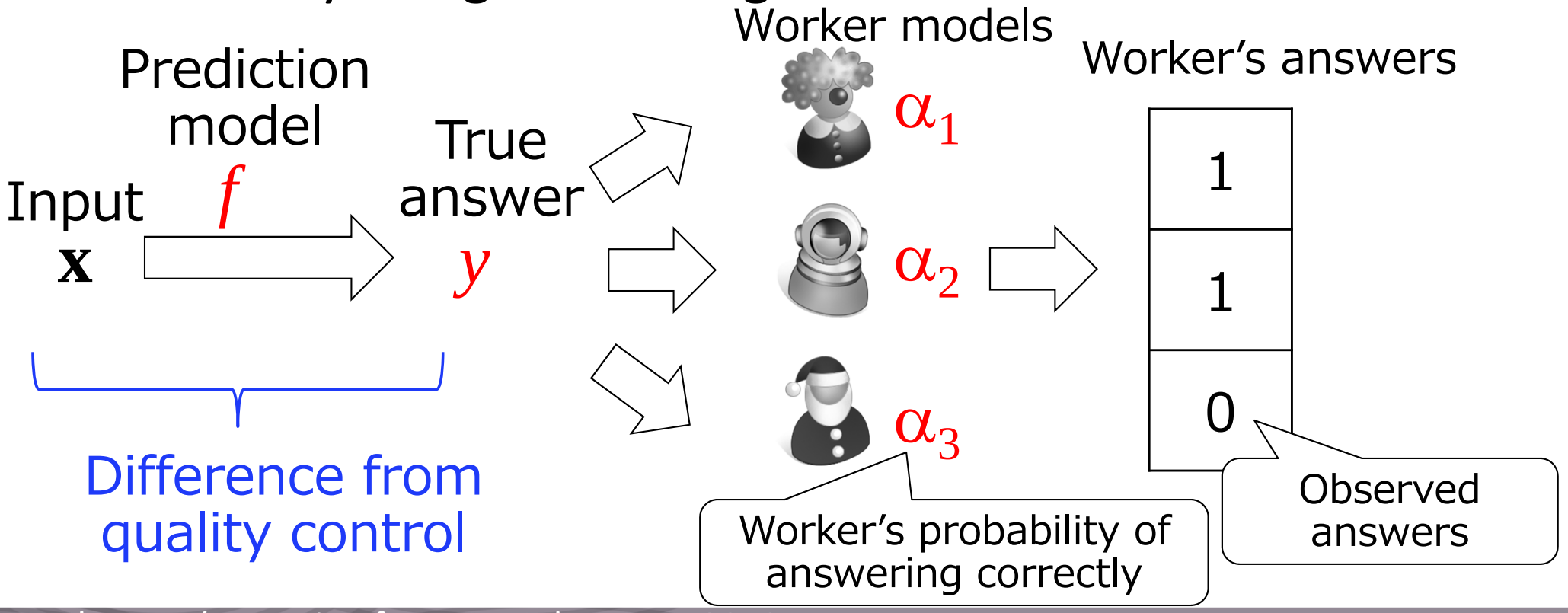
$$\{ (\mathbf{x}_i, y_{i,j}) \}_{i=1,\ldots,N \; j=1,\ldots,M^{(i)}}$$

  - j : worker id

1. Estimate the true answers from workers' answers by using a quality control method such as the Latent Class model
2. Learn a prediction model by using estimated answers as the true answers

Leaning from Crowdsourced Labels:Learning a Prediction Model Directly from Worker Labels

- Difference from quality control：Main objective is obtaining a prediction model rather than true answers
- Answers, worker models, and the prediction model are estimated by using the EM algorithm



Raykar *et al*. Learning from Crowds. *JMLR*, 2010
Kajino, Tsuboi, Kashima. Convex Formulations of Learning from Crowds. *AAAI*, 2012

Summary: Crowdsourcing for Datafication

- Gamification
  - Output agreement, input agreement
- Quality control by introducing redundancy
  - Latent class model
  - Extension to utilize problem difficulty, worker expertize, confidence
- Examples of datafication
  - Crowd sensing
  - Digitizing legacy data
- Learning a prediction model from crowds