A Graphical Model Approach for Crowdsourcing

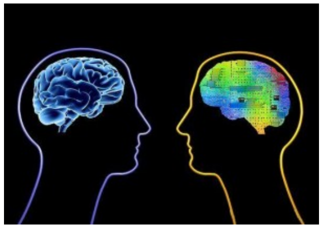
Qiang Liu (UC Irvine)

With Jian Peng, (MIT)   Prof. Alexander Ihler (UC Irvine)

Presented in NIPS 2012

Crowdsourcing

- Outsourcing tasks to crowds of of people
  - very powerful
    - human v.s. artificial intelligence
    - wisdom of crowds
    - cheap, fast, convenient



CrowdFlower

amazon mechanical turk™
Artificial Artificial Intelligence

Crowdsourcing

- Harvesting Human Intelligence / Judgments

Search Relevance Evaluation

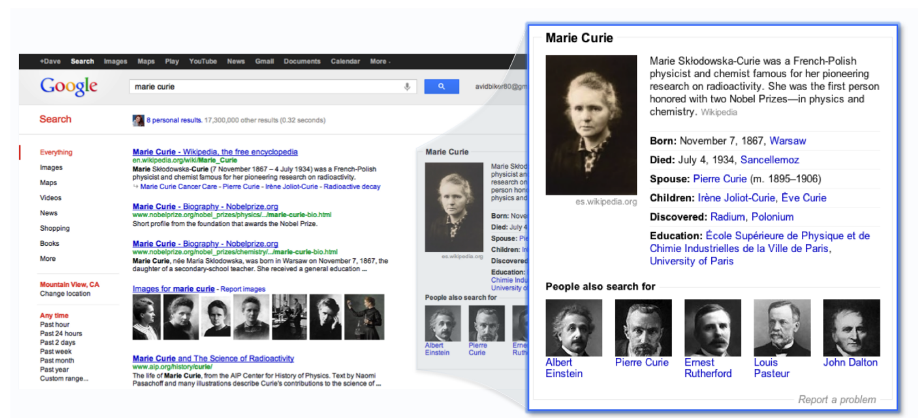
Translation

Predication Market

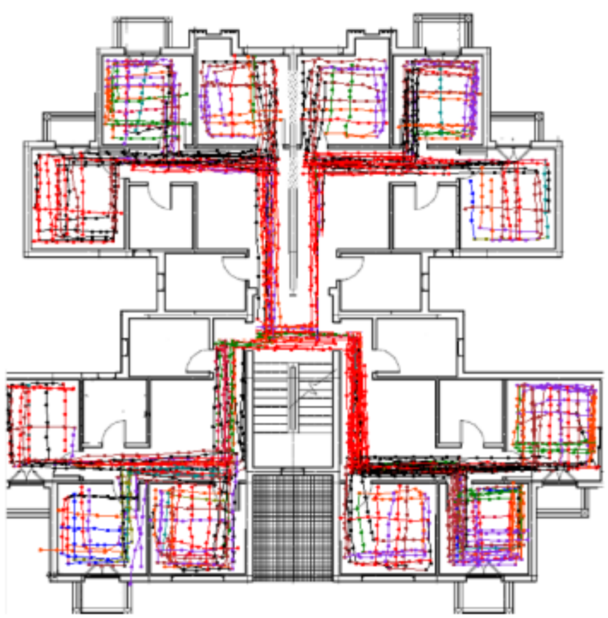
Crowdsourcing

- Collecting Information Distributed in Crowds

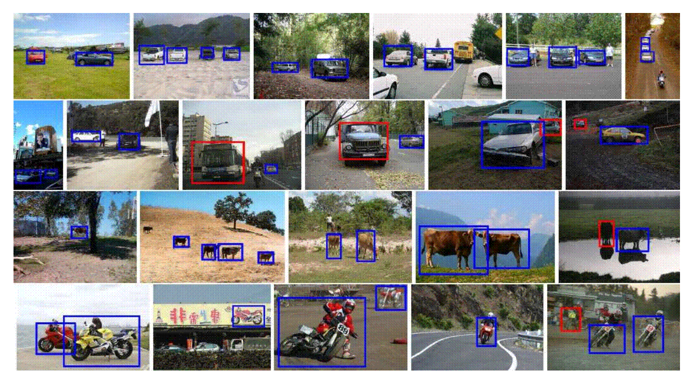
Knowledge graph



Indoor maps



Research datasets



Crowdsourcing

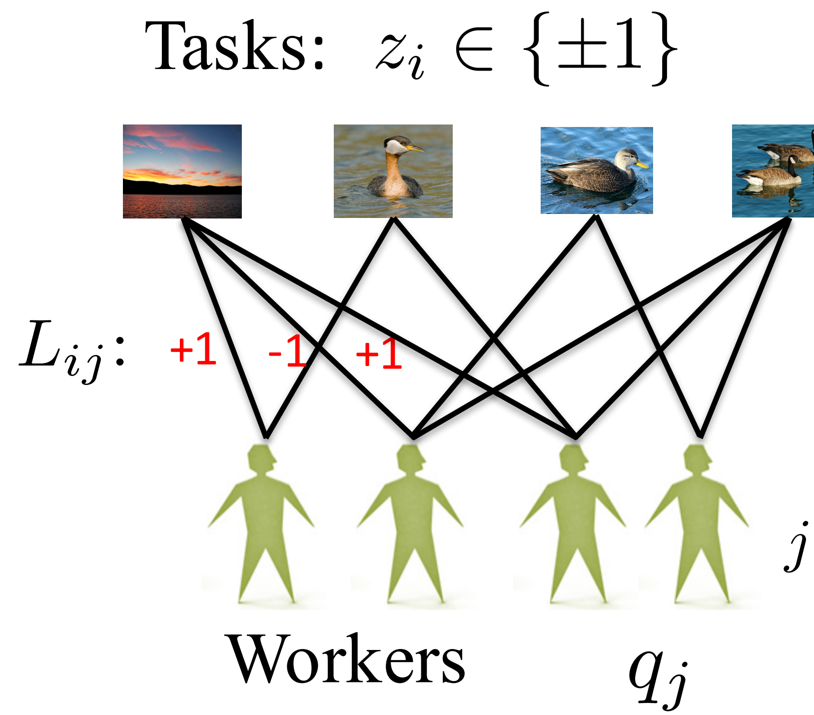
- Outsourcing tasks to crowds of people
  - but be careful!
    - humans are unreliable and diverse
    - Need to ask different people and aggregate their opinions

- This talk:
  - Aggregation algorithms

Crowdsourcing for Labeling

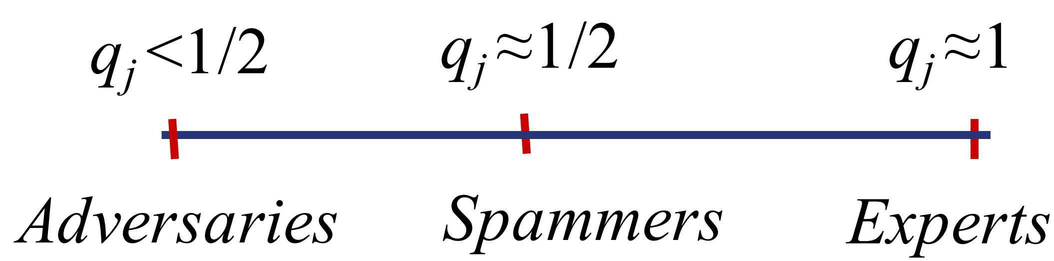
- Goal: estimate true z_i from noisy labels $\{L_{ij}\}$.
  - minimize bit-wise error:

$$\min_{\hat{z}_i} \mathbb{E}[\sum_i 1(\hat{z}_i \neq z_i)]$$



- Workers are diverse:
  - reliability:

$$q_j = \text{prob}[L_{ij} = z_i]$$



Previous Works

- Majority Voting (MV):

$$\hat{z}_i = \mathrm{sign}[\sum_{j \in \partial_i} L_{ij}]$$

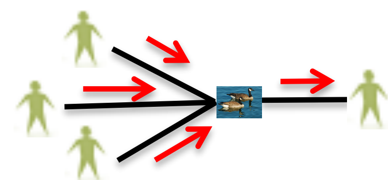
Previous Works

- Majority Voting (MV):

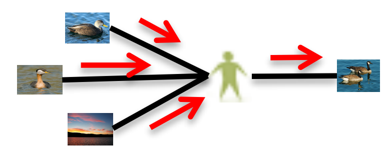
$$\hat{z}_i = \text{sign}[\sum_{j \in \partial_i} L_{ij}]$$

- Iterative algorithm by Karger, Oh, Shah 2011 (KOS):

$$\hat{z}_i = \text{sign}[\sum_{j \in \partial_i} L_{ij} y_{j \to i}]$$



Task-> workers: $$x_{i \to j}^{t+1} = \sum_{j' \neq j} L_{ij'} y_{j' \to i}^{t}$$



Workers->Tasks: $$y_{j \to i}^{t+1} = \sum_{i' \neq i} L_{i'j} x_{i' \to j}^{t+1},$$

Previous Works

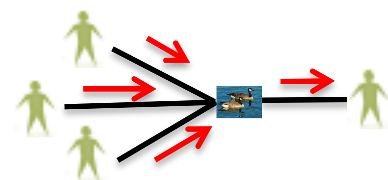
- Majority Voting (MV):

$$\hat{z}_i = \text{sign}[\sum_{j \in \partial_i} L_{ij}]$$

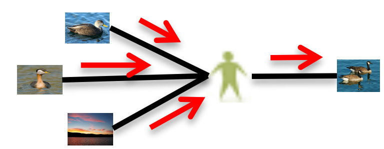
- Iterative algorithm by Karger, Oh, Shah 2011 (KOS):

$$\hat{z}_i = \text{sign}[\sum_{j \in \partial_i} L_{ij} y_{j \to i}]$$

Task-> workers: $$x_{i \to j}^{t+1} = \sum_{j' \neq j} L_{ij'} y_{j' \to i}^{t}$$



Workers->Tasks: $$y_{j \to i}^{t+1} = \sum_{i' \neq i} L_{i'j} x_{i' \to j}^{t+1},$$



- This talk:
  - generalizes both MV and KOS
  - Better performance, principled derivation

Previous Works

- Expectation Maximization (Dawid & Skene 79, Smyth et al 95, Raykar et al 10, Whitehill et al 09, Welinder et al 10, etc)
  - Build generative probabilistic models
  - Maximizing likelihood by EM
  - Estimate $\hat{z}_i$ with fixed parameters

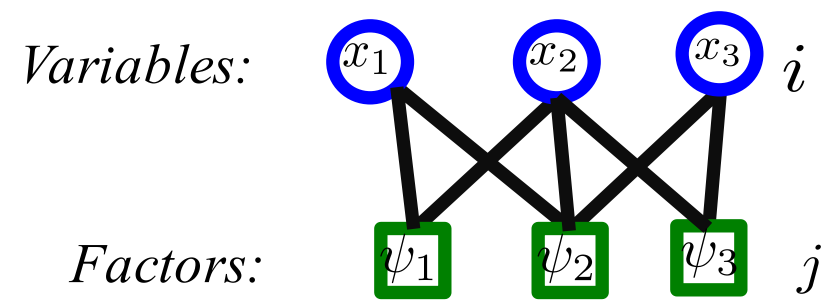
- This talk:
  - Model complexity v.s. inference methods
  - Crowdsourcing as probabilistic graphical models

Probabilistic Graphical Models

- Factorized probability:

$$p(z) = \frac{1}{Z} \prod_{j \in [N]} \psi_j(z_{\partial_j})$$

Factor graph



– *Example:* $$p(z) = \frac{1}{Z}\psi_1(x_1, x_2)\psi_2(x_1, x_2, x_3)\psi_3(x_2, x_3)$$

- Inference: calculating the marginal probabilities

$$p(z_i) = \sum_{z_{N \setminus i}} p(z)$$

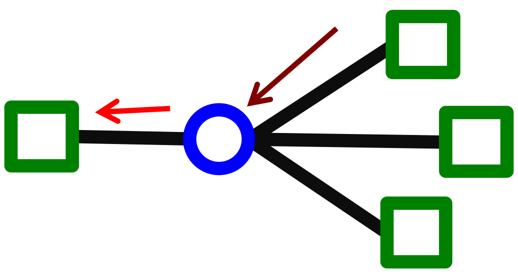
- approximation: belief propagation, mean field, etc

Probabilistic Graphical Models

- Belief propagation: $p(z_i) \approx \prod_{j \in \partial_i} m_{j \to i}(z_i)$

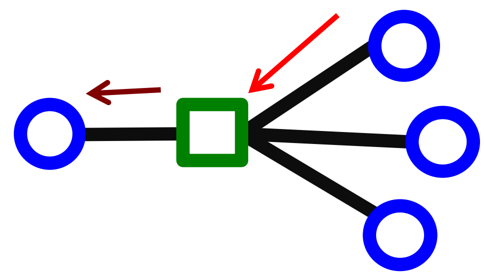
Variables -> Factors:

$$m_{i \to j}(z_i) \propto \prod_{j' \neq j} m_{j' \to i}(z_i)$$



Factors -> Variables:

$$m_{j \to i}(z_i) \propto \sum_{z_{\partial_j \setminus \{i\}}} \psi_j \prod_{i' \neq i} m_{i' \to j}(z_{i'})$$



- Mean field method:
  - approximate $p(z)$ by fully independent models.

$$\min_{\{\tilde{p}_i\}} \mathrm{KL}\Big[\prod_i \tilde{p}_i(z_i) || p(z)\Big]$$

See Wainwright & Jordan 08, Koller & Friedman's book, Murphy's book.

Assumptions

- Workers' model: $$p(L_{ij}|z_i, q_j) = \begin{cases} q_j, & \text{if } L_{ij} = z_i \text{ (correct)} \\ 1 - q_j, & \text{if } L_{ij} \neq z_i \text{ (wrong)} \end{cases}$$
- Prior of workers' reliability: $q_j \sim p(q_j)$
- Joint posterior distribution:

$$p(z, q|L) \propto \prod_j p(q_j) q_j^{c_j} (1 - q_j)^{d_j - c_j}$$

d_j: # of images by j in total

c_j: # of correct images by j:

$$c_j = \sum_i 1[L_{ij} = z_i]$$

- Estimator (Minimizes the bit-wise error):

$$\hat{z}_i = \arg\max_{z_i} p(z_i|L)$$

Assumptions

- Workers' model: $p(L_{ij}|z_i, q_j) = \begin{cases} q_j, & \text{if } L_{ij} = z_i \text{ (correct)} \\ 1 - q_j, & \text{if } L_{ij} \neq z_i \text{ (wrong)} \end{cases}$

- Prior of workers' reliability: $q_j \sim p(q_j)$

- Joint posterior distribution:

$$p(z, q|L) \propto \prod_j p(q_j) q_j^{c_j} (1 - q_j)^{d_j - c_j}$$

d_j: # of images by j in total

c_j: # of correct images by j:

$$c_j = \sum_i 1[L_{ij} = z_i]$$

- Estimator (Minimizes the bit-wise error):

$$\hat{z}_i = \arg\max_{z_i} \{p(z_i|L) \equiv \sum_{z_{[N]\setminus i}} \int_q p(z, q|L) dq\}$$

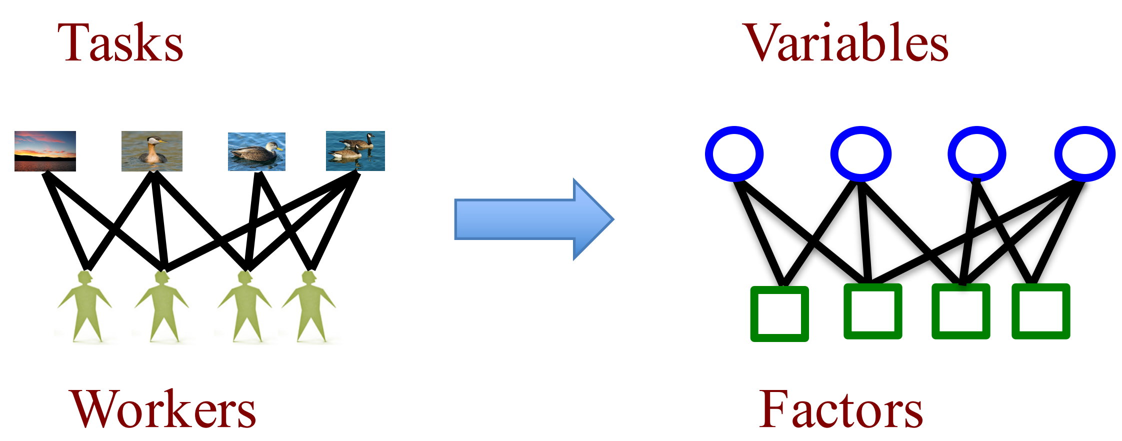
Graph Model for Crowdsourcing

- Integrate out q_j analytically:

$$p(z|L) = \int_q p(z, q|L) dq = \prod_j \int_0^1 p(q_j) q_j^{c_j} (1 - q_j)^{d_j - c_j} dq_j$$

$$\stackrel{def}{=} \prod_j \psi_j(c_j)$$

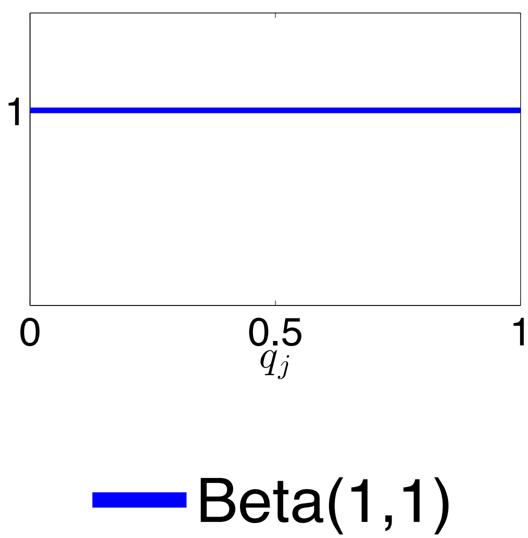
where $\psi_j(c_j) = \int_0^1 p(q_j) {q_j}^{c_j} (1 - q_j)^{d_j - c_j} dq_j$



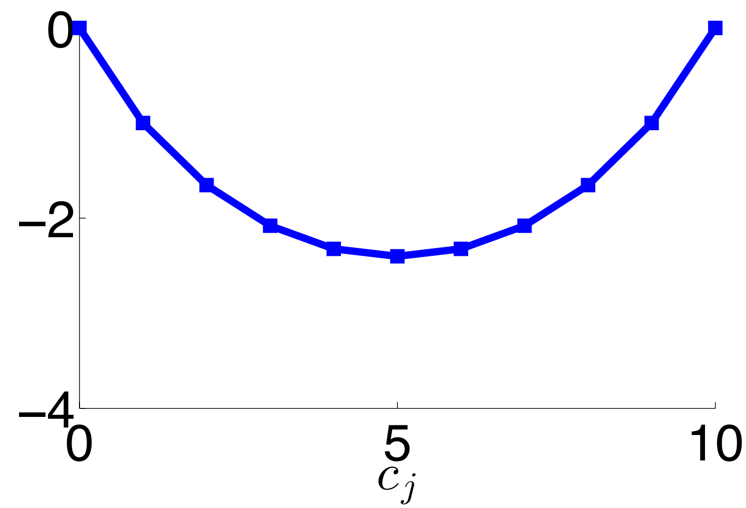
Shape of $\psi_j(c_j)$

$$\psi_j(c_j) = \int_0^1 q_j^{c_j}(1-q_j)^{d_j-c_j}p(q_j)dq_j$$

Prior of q_j



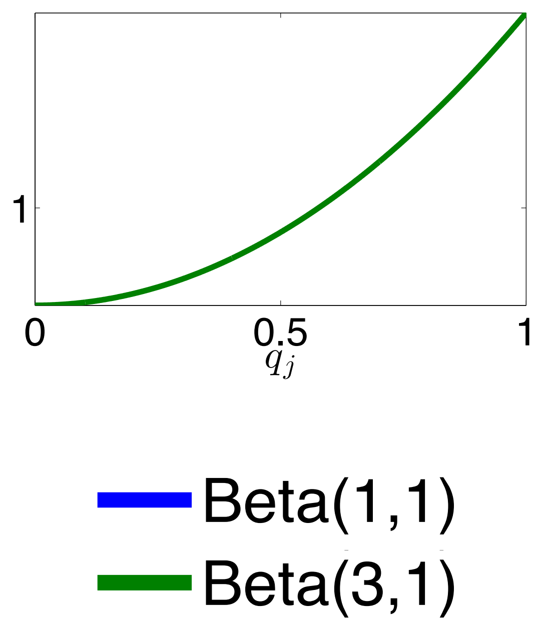
$\log \psi_j(c_j)$



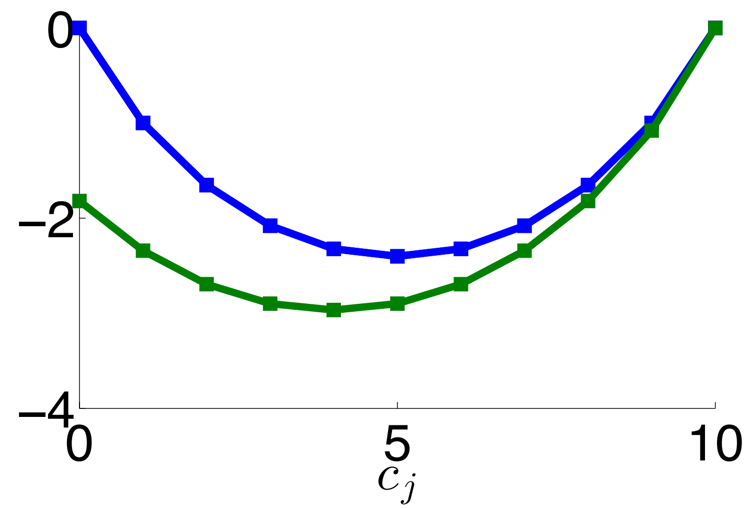
Shape of $\psi_j(c_j)$

$$\psi_j(c_j) = \int_0^1 q_j^{c_j}(1-q_j)^{d_j-c_j}p(q_j)dq_j$$

Prior of q_j



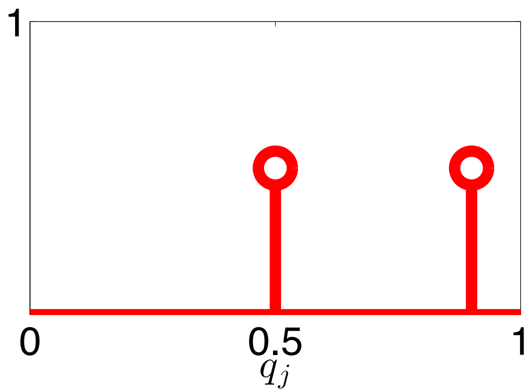
$\log \psi_j(c_j)$

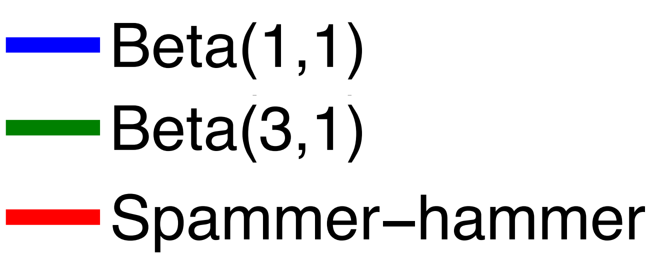


Shape of $\psi_j(c_j)$

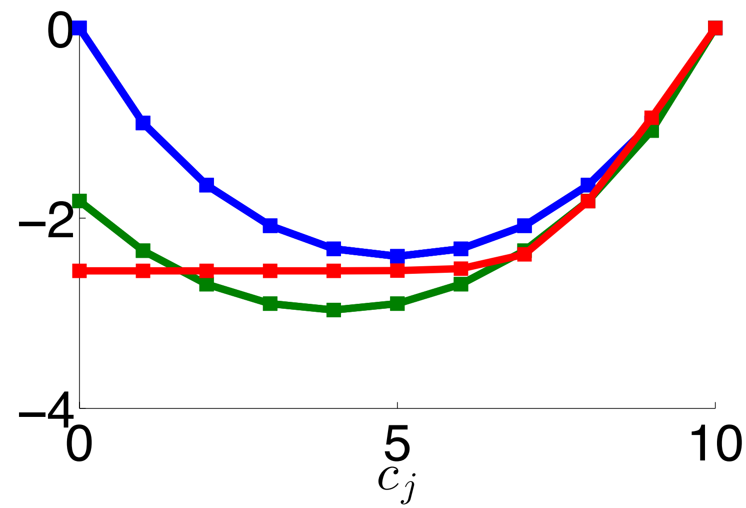
$$\psi_j(c_j) = \int_0^1 q_j^{c_j}(1-q_j)^{d_j-c_j}p(q_j)dq_j$$

Prior of q_j





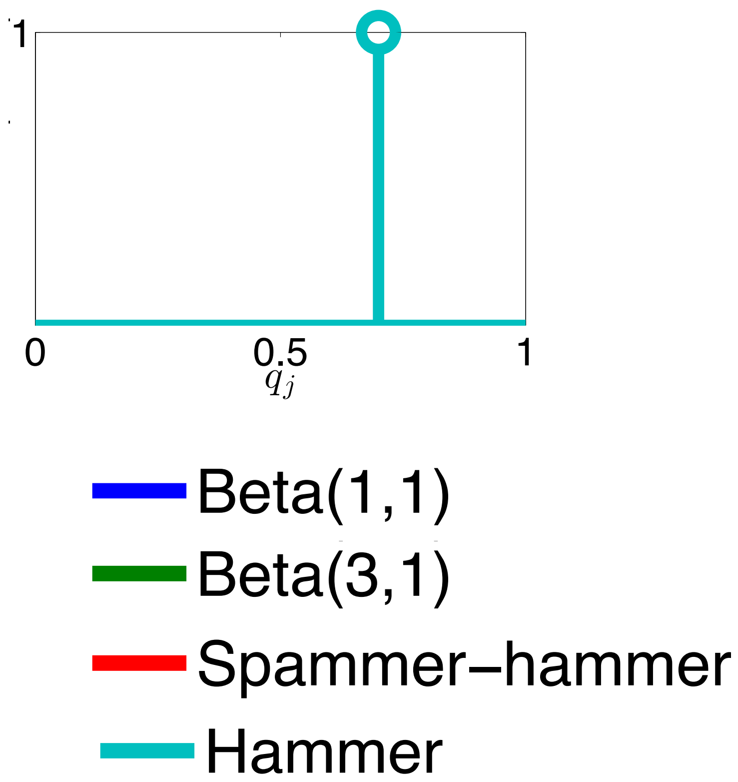
$\log \psi_j(c_j)$



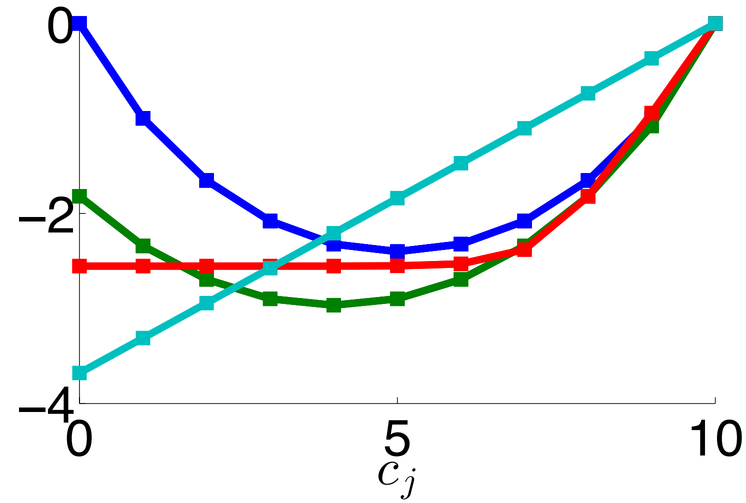
Shape of $\psi_j(c_j)$

$$\psi_j(c_j) = \int_0^1 q_j^{c_j}(1-q_j)^{d_j-c_j}p(q_j)dq_j$$

Prior of q_j



$\log \psi_j(c_j)$



BP for Crowdsourcing

- Standard belief propagation on $p(z) = \prod_j \psi_j(x_{\partial_j})$

Marginal probability: $p(z_i) \approx \prod_{j \in \partial_i} m_{j \to i}(z_i)$

Variables -> Factors: $m_{i \to j}(z_i) \propto \prod_{j' \neq j} m_{j' \to i}(z_i)$

Factors -> Variables: $m_{j \to i}(z_i) \propto \sum_{z_{\partial_j \setminus \{i\}}} \psi_j \prod_{i' \neq i} m_{i' \to j}(z_{i'})$

Decode solution: $\hat{z}_i = \arg\max_{z_i} p(z_i)$

- Log-odds transformation:

Marginals: $\hat{x}_i = \log \frac{p(z_i = +1)}{p(z_i = -1)}$

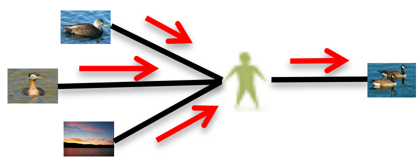
Messages: $x_{i \to j} = \log \frac{m_{i \to j}(+1)}{m_{i \to j}(-1)}$ $y_{j \to i} = \log \frac{m_{j \to i}(+L_{ij})}{m_{j \to i}(-L_{ij})}$

Belief Propagation (log-odds form)

Estimate labels:

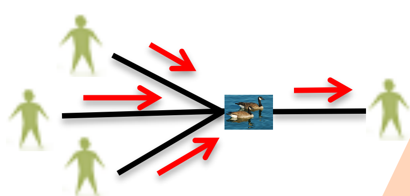
$$\hat{z}_i = \text{sign}\Big[\sum_{j \in \partial_i} L_{ij} y_{j \to i}\Big]$$

Tasks ➔ Workers:

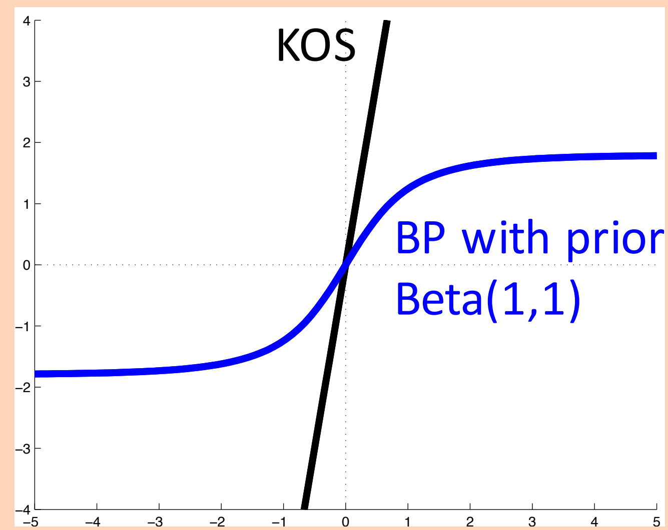


$$x_{i \to j}^{t+1} = \sum_{j' \neq j} L_{ij'} y_{j' \to i}^{t}$$

Workers ➔ Tasks:



$$y_{j \to i}^{t+1} = \sigma[\{L_{i'j} x_{i' \to j}^{t+1} : i' \neq i\}]$$



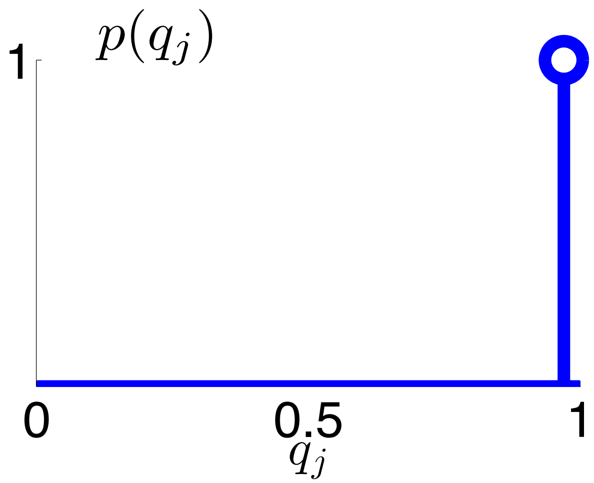
$$\sigma[\cdot] = \log \frac{\sum_{k=0}^{d_j - 1} \psi_j(k+1) e_k}{\sum_{k=0}^{d_j - 1} \psi_j(k) e_k}$$

e_k are the elementary symmetric polynomials in $\{\exp(L_{i'j} x_{i' \to j})\}$,

Complexity: $O(d_j \log(d_j)^2)$

Special Priors

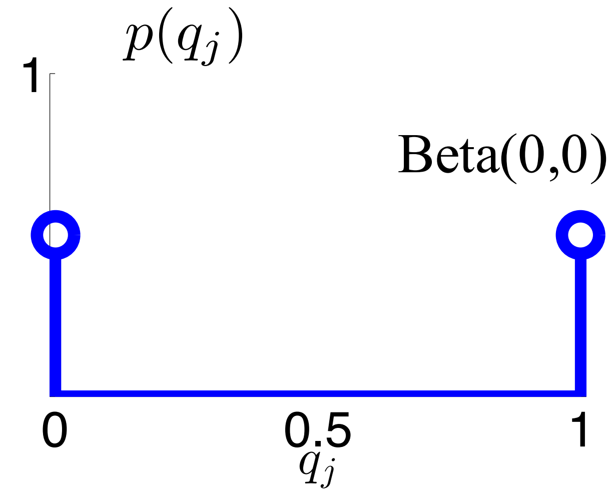
- Deterministic prior:



Our BP => majority voting

– Includes no diversity; sensitive to adversaries

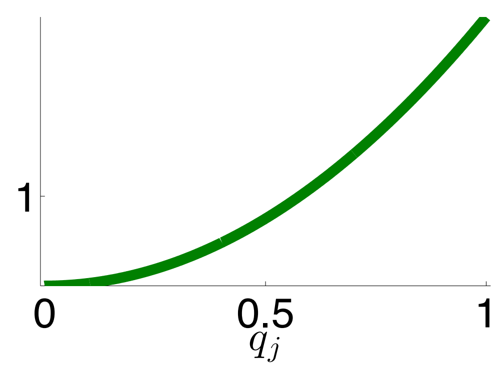
- Haldane prior:



Our BP => KOS

– High diversity (variance); emphasizes adversaries

Optimistic



Pessimistic

Frequentist Guarantees

- Density Evolution (e.g., Mezard & Montanari 09; Karger, Oh, Shah 11)

Frequentist Guarantees

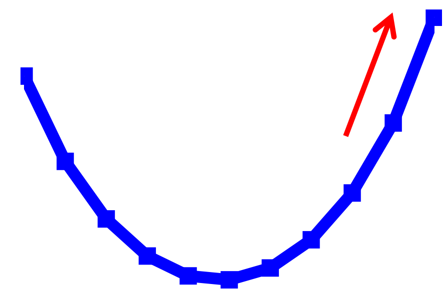
- Density Evolution (e.g., Mezard & Montanari 09; Karger, Oh, Shah 11)

Assumption:

1. Random regular bipartite graphs
2. True reliability prior: $\mathbb{E}[q_j] > 1/2$
3. Our algorithm:

$$\nabla^2 \log \psi(c_j) \geq 0, \quad \forall 0 \leq c_j \leq d_j$$

$$\nabla \log \psi(d_j) > 0$$



Result:

$$\limsup_{N \to +\infty} \frac{1}{N} \sum_{i \in [N]} \text{prob}[z_i \neq \hat{z}_i^t] \to 0, \quad \text{if } l \to +\infty$$

(l = # of workers per images)

Mean Field and EM

- Joint posterior distribution: $p(z, q|L)$
- Mean field approximation:

$$\min_{\tilde{p}} \mathrm{KL}[\tilde{p}(z,q)||p(z,q|L)] \qquad \text{where } \tilde{p}(z,q) = \prod_i \mu_i(z_i) \prod_j \nu_j(q_j)$$

- Coordinate descent (when prior is $\mathrm{Beta}(\alpha, \beta)$):

$$\textit{Updating } \mu_i\text{:} \quad \mu_i(z_i) \propto \prod_{j \in \mathcal{M}_i} a_j^{\delta_{ij}} b_j^{1-\delta_{ij}}, \qquad \text{where } \delta_{ij} = \mathbf{1}[z_i = L_{ij}]$$

$$\textit{Updating } \nu_j\text{:} \quad \nu_j(q_j) \sim \mathrm{Beta}(\sum_{i \in \mathcal{N}_j} \mu_i(L_{ij}) + \alpha, \sum_{i \in \mathcal{N}_j} \mu_i(-L_{ij}) + \beta),$$

$$\text{where } a_j = \exp(\mathbb{E}_{\nu_j}[\ln q_j]) \text{ and } b_j = \exp(\mathbb{E}_{\nu_j}[\ln(1-q_j)])$$

- Taylor expansion:

$$a_j \approx \bar{q}_j \text{ and } b_j \approx 1 - \bar{q}_j \qquad \text{where } \bar{q}_j = \mathbb{E}_{\nu_j}[q_j]$$

Mean Field and EM

- Joint posterior distribution: $p(z, q|L)$
- Mean field approximation:

$$\min_{\tilde{p}} \mathrm{KL}[\tilde{p}(z,q)||p(z,q|L)] \qquad \text{where } \tilde{p}(z,q) = \prod_i \mu_i(z_i) \prod_j \nu_j(q_j)$$

- Mean field (with first order approximation):

$$\mu_i(z_i) \propto \prod_j \bar{q}_j^{\mathbf{1}[L_{ij}=z_i]}(1-\bar{q}_j)^{\mathbf{1}[L_{ij}\neq z_i]} \qquad \bar{q}_j = \frac{\sum_i \mu_i(L_{ij}) + \alpha}{d_j + \alpha + \beta}$$

- EM:

$$\mu_i(z_i) \propto \prod_j \bar{q}_j^{\mathbf{1}[L_{ij}=z_i]}(1-\bar{q}_j)^{\mathbf{1}[L_{ij}\neq z_i]} \qquad \bar{q}_j = \frac{\sum_i \mu_i(L_{ij}) + \alpha - 1}{d_j + \alpha + \beta - 2}$$

- Different from EM only on replacing α-1 and β-1 with α and β.
- Add-one smoothing.

Mean Field and EM

- Approximation mean field (AMF):

$$\mu_i(z_i) \propto \prod_j \bar{q}_j^{\mathbf{1}[L_{ij}=z_i]}(1-\bar{q}_j)^{\mathbf{1}[L_{ij}\neq z_i]} \qquad \bar{q}_j = \frac{\sum_i \mu_i(L_{ij}) + \alpha}{d_j + \alpha + \beta}$$

- EM:

$$\mu_i(z_i) \propto \prod_j \bar{q}_j^{\mathbf{1}[L_{ij}=z_i]}(1-\bar{q}_j)^{\mathbf{1}[L_{ij}\neq z_i]} \qquad \bar{q}_j = \frac{\sum_i \mu_i(L_{ij}) + \alpha - 1}{d_j + \alpha + \beta - 2}$$

(E-step) (M-step)

- Different from EM only on replacing α-1 and β-1 with α and β.
- Add-one smoothing.

Extensions

- Sensitivity & Specificity models of workers:

One-coin Model:

$$q_j = \mathrm{prob}[L_{ij} = z_i]$$
$$= 1 - \mathrm{prob}[L_{ij} = -z_i]$$

Two-coin Model (Dawid & Skene 79)**:**

$$s_j = \mathrm{prob}[L_{ij} = +1 | z_i = +1], \quad \text{(sensitivity)}$$
$$t_j = \mathrm{prob}[L_{ij} = -1 | z_i = -1]. \quad \text{(specificity)}$$

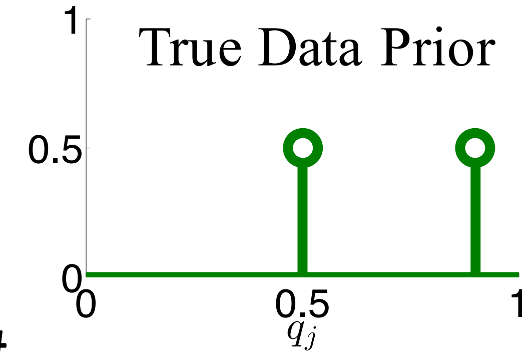
- Model selection by marginal likelihood:

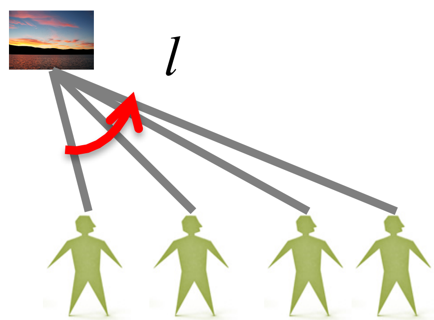
$$K = \frac{p(L|M_1)}{p(L|M_2)} = \frac{\sum_z \int_q p(z, q, L|M_1) dq}{\sum_z \int_q p(z, q, L|M_2) dq}$$

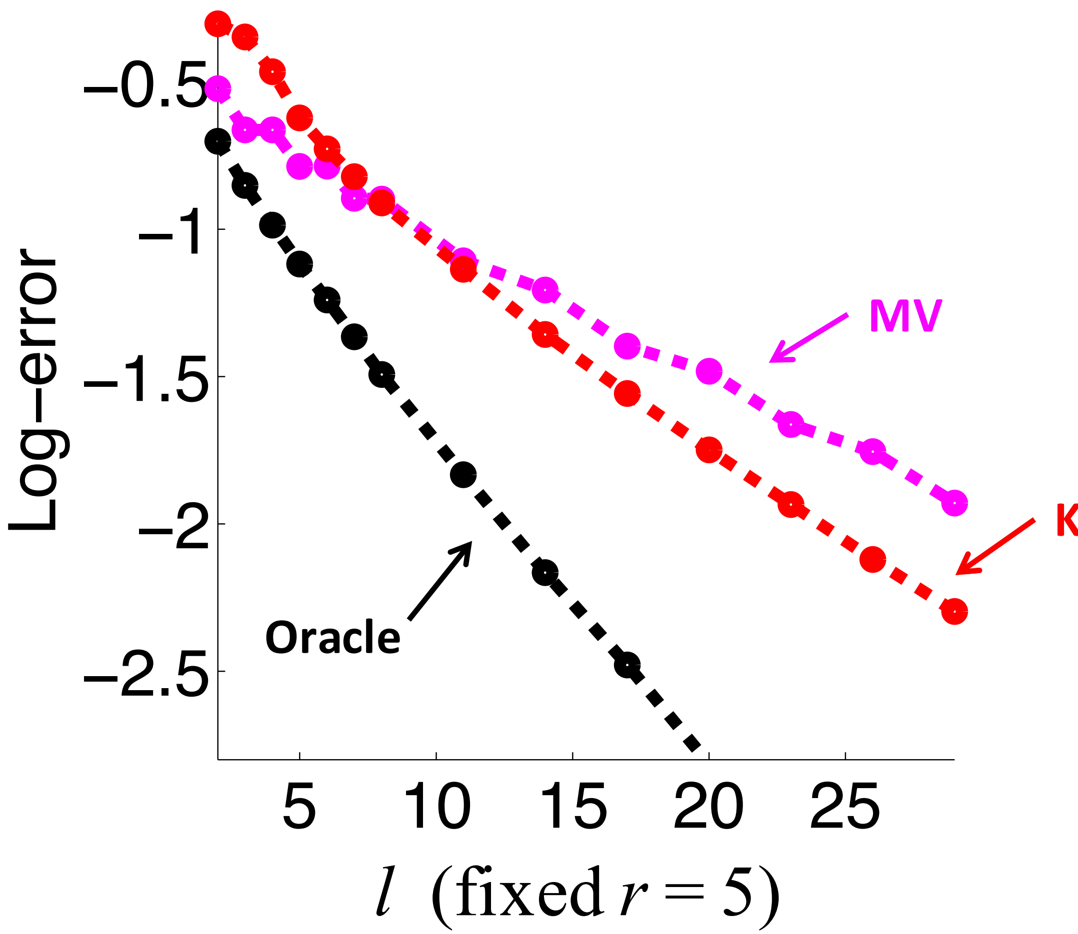
- Incorporating item features, and expect labels
- …

Random (l, r)-regular bipartite graph

- 1000 tasks
- True data prior: spammer-hammer prior
- Varying # of workers per images (l), fixing # of images per worker (r)

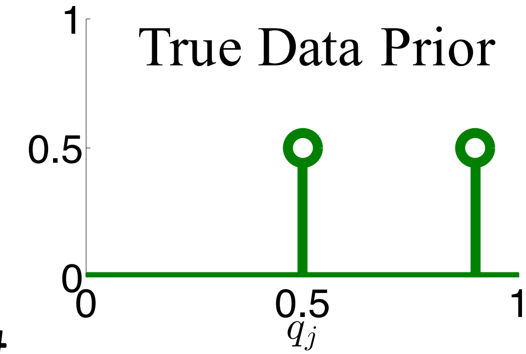


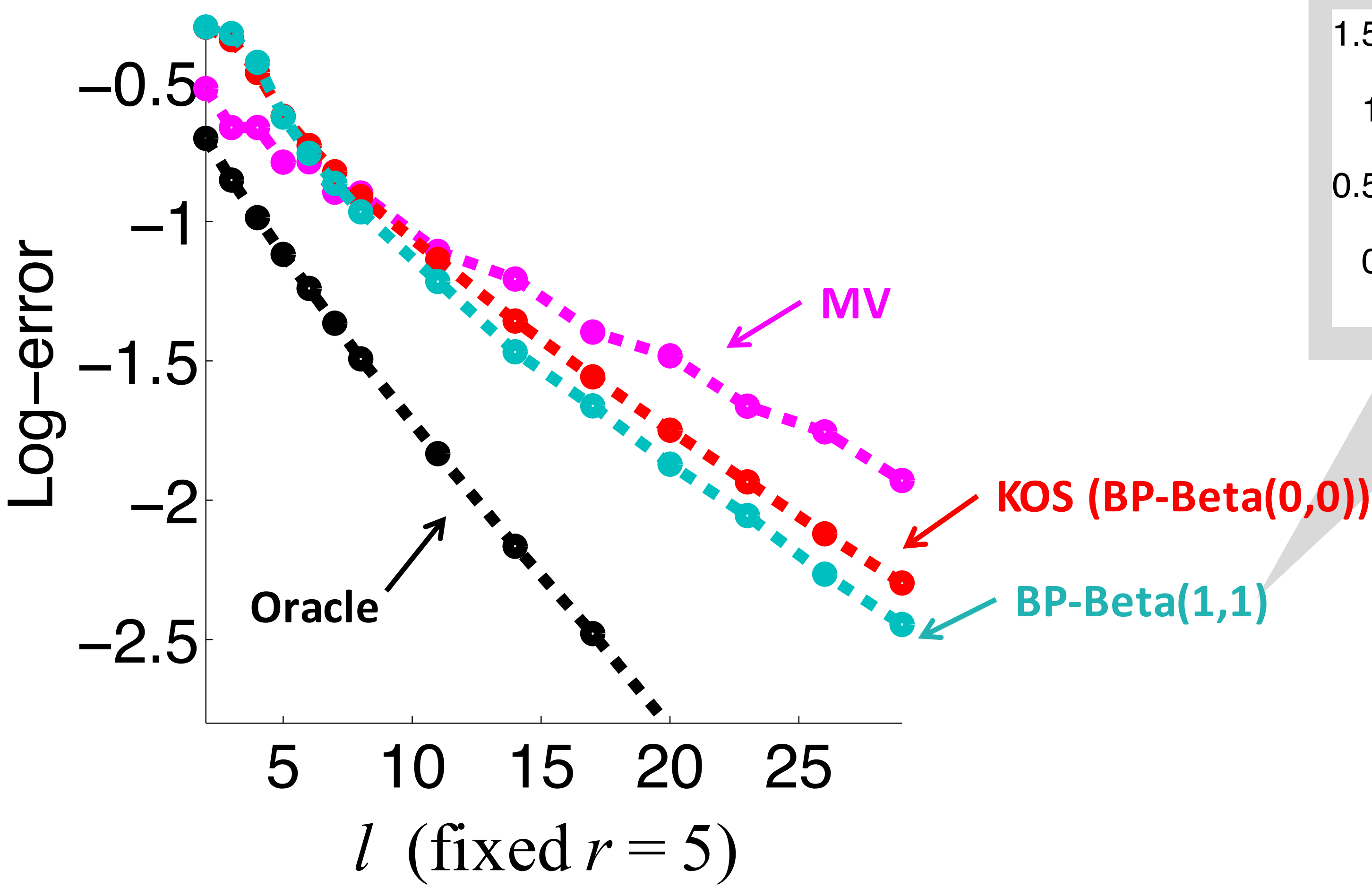


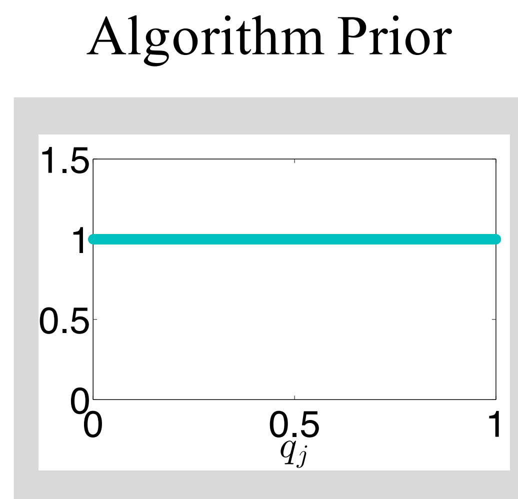


Random (l, r)-regular bipartite graph

- 1000 tasks
- True data prior: spammer-hammer prior
- Varying # of workers per images (l), fixing # of images per worker (r)

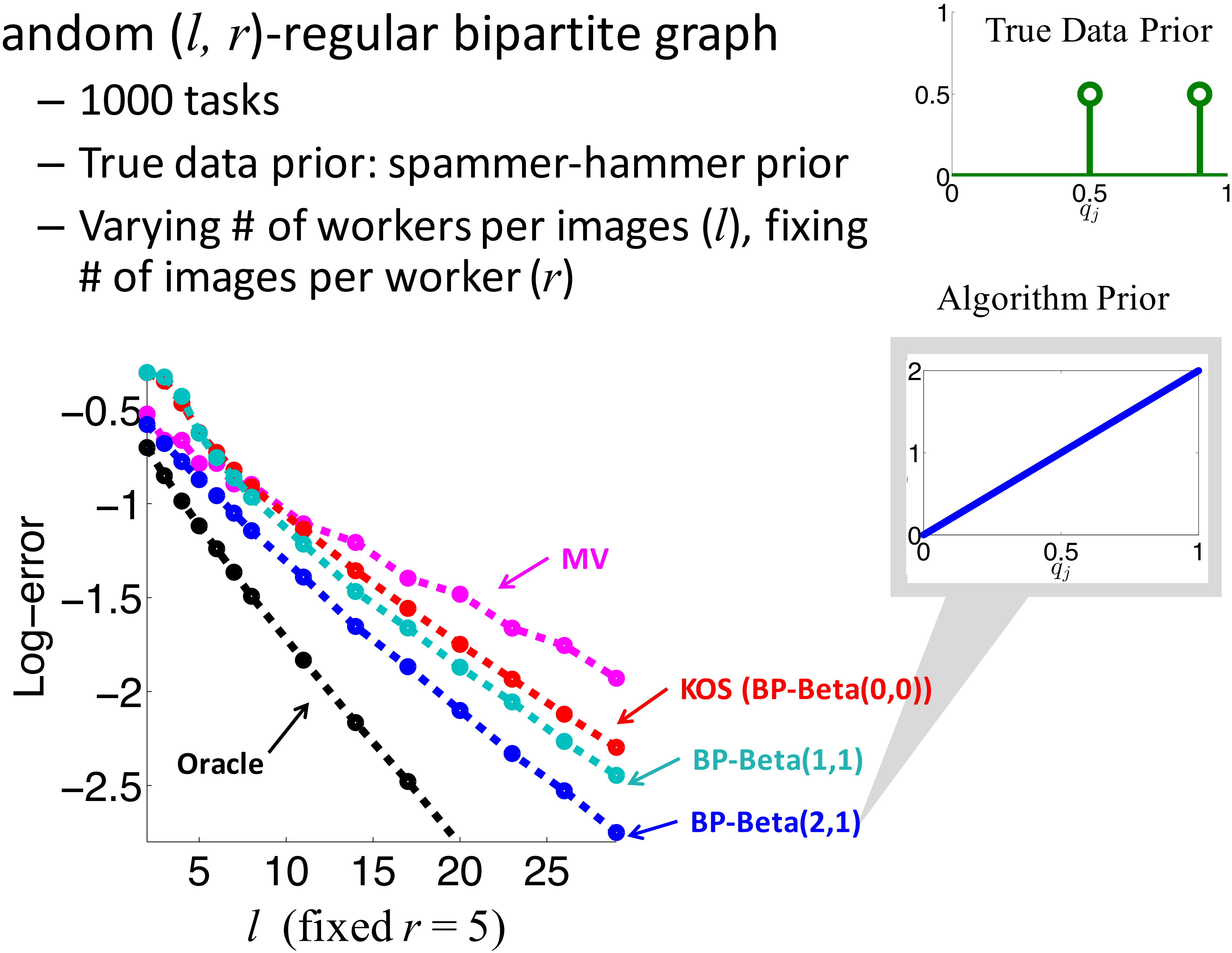






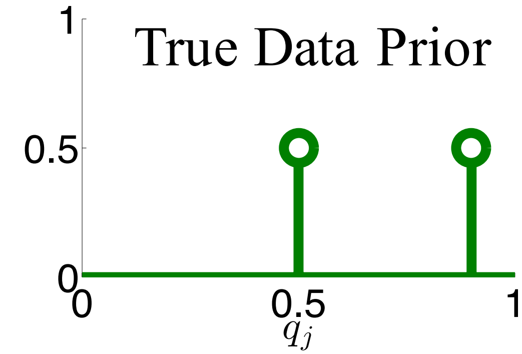
Random (*l*, *r*)-regular bipartite graph

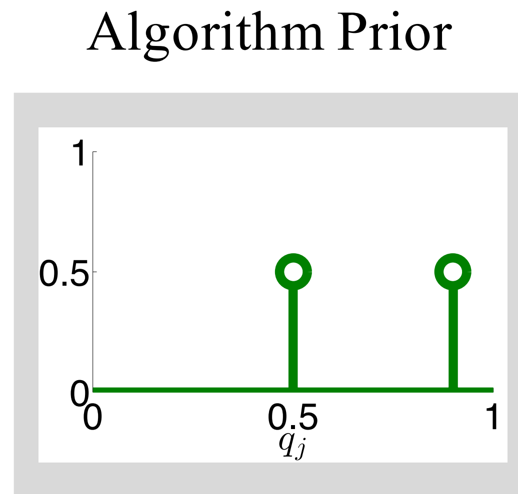
- 1000 tasks
- True data prior: spammer-hammer prior
- Varying # of workers per images (*l*), fixing # of images per worker (*r*)

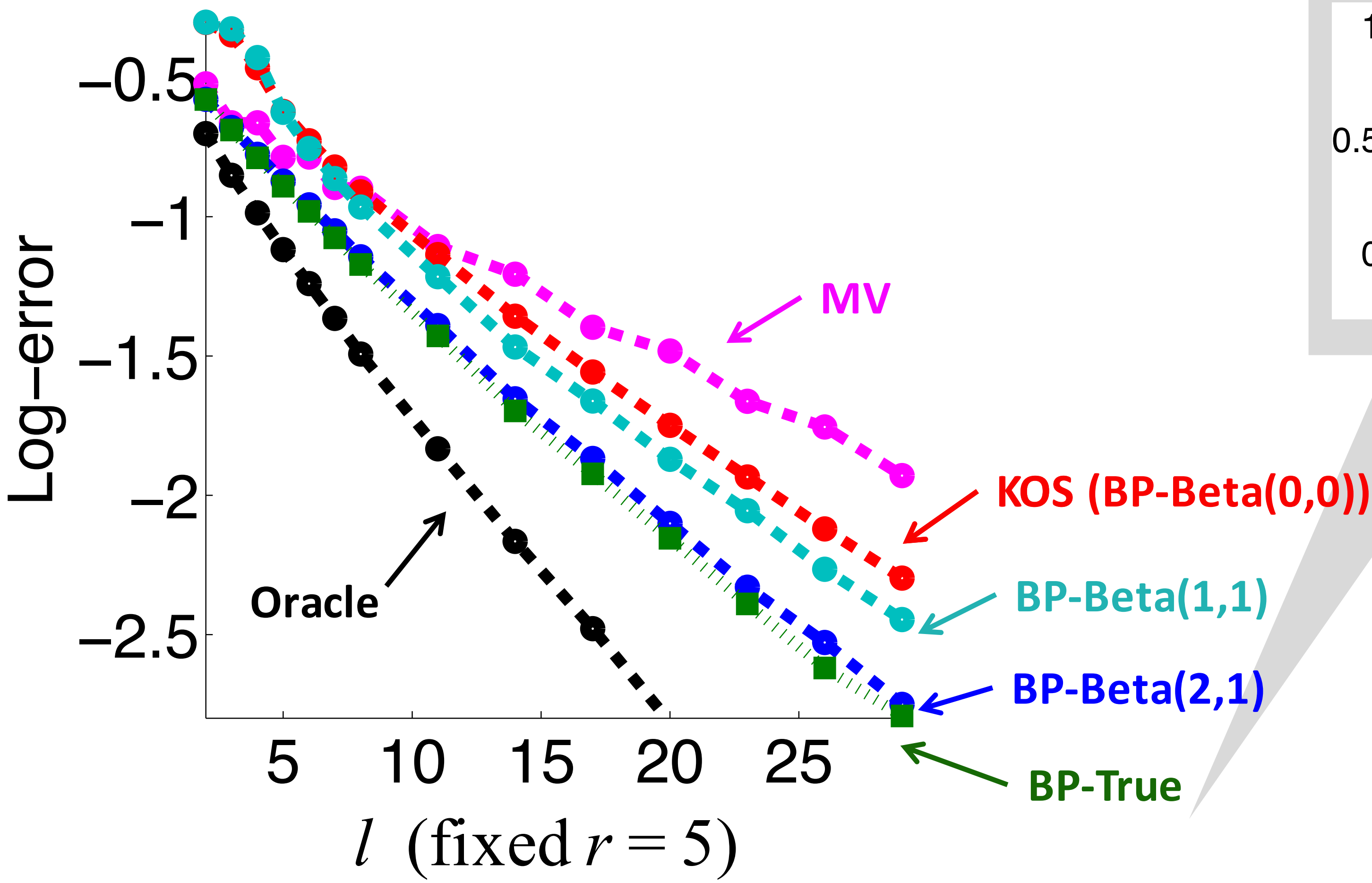


Random (l, r)-regular bipartite graph

- 1000 tasks
- True data prior: spammer-hammer prior
- Varying # of workers per images (l), fixing # of images per worker (r)

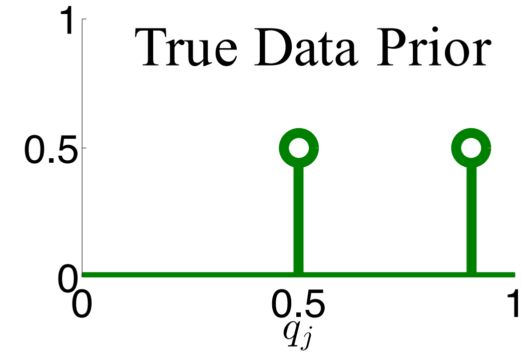


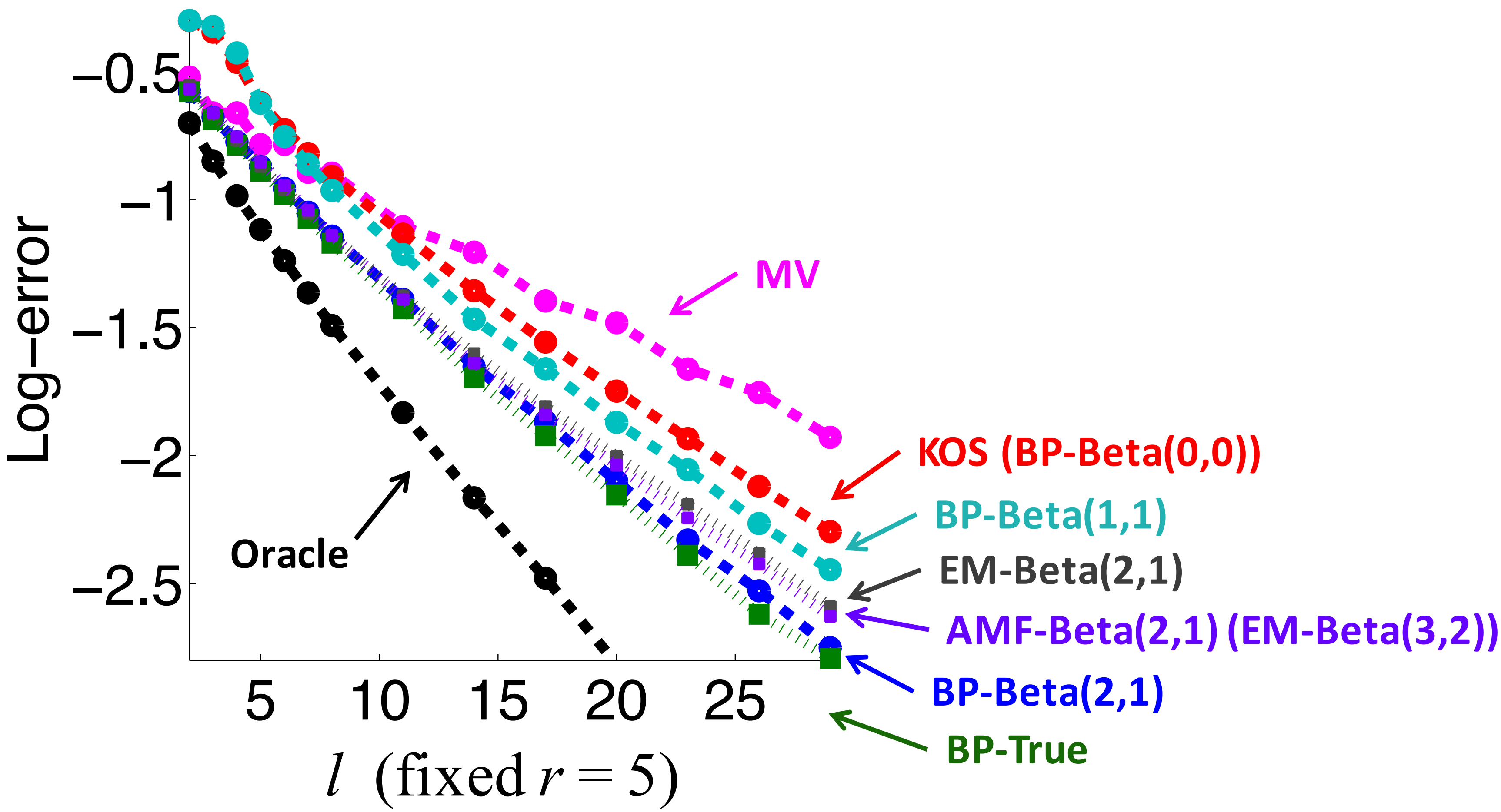




Random (l, r)-regular bipartite graph

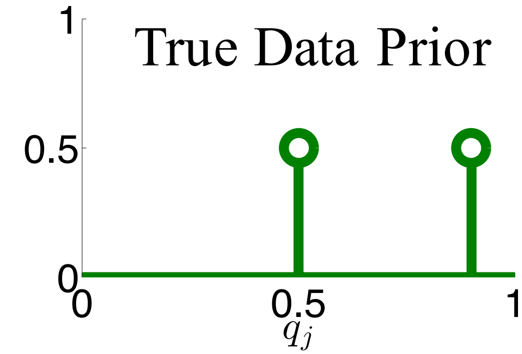
- 1000 tasks
- True data prior: spammer-hammer prior
- Varying # of workers per images (l), fixing # of images per worker (r)

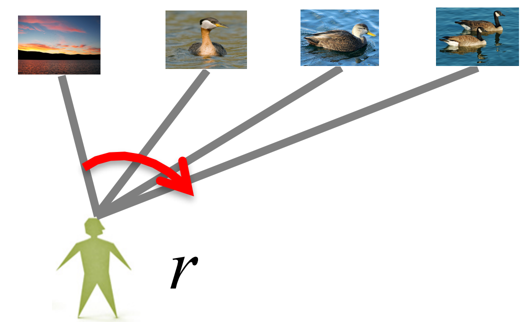


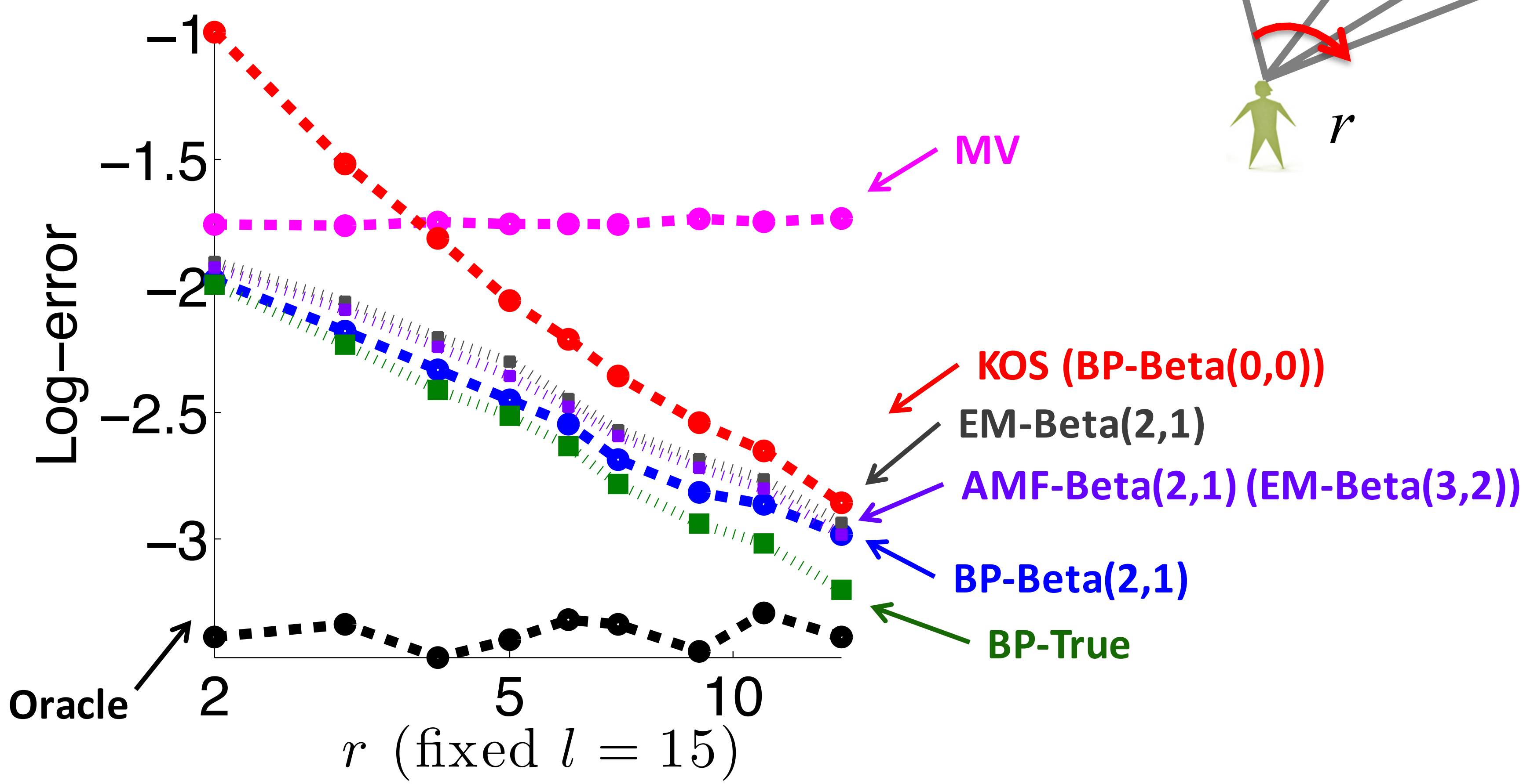


Random (l, r)-regular bipartite graph

- 1000 tasks
- True data prior: spammer-hammer prior
- Varying # of images per worker (r), fixing # of workers per image (l)

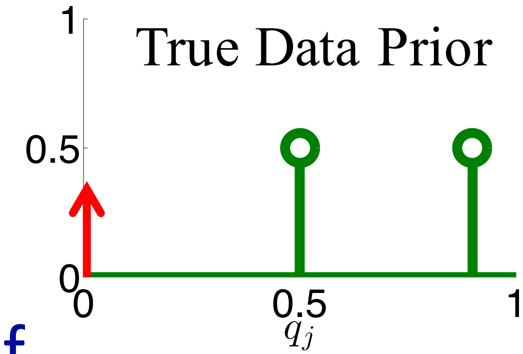


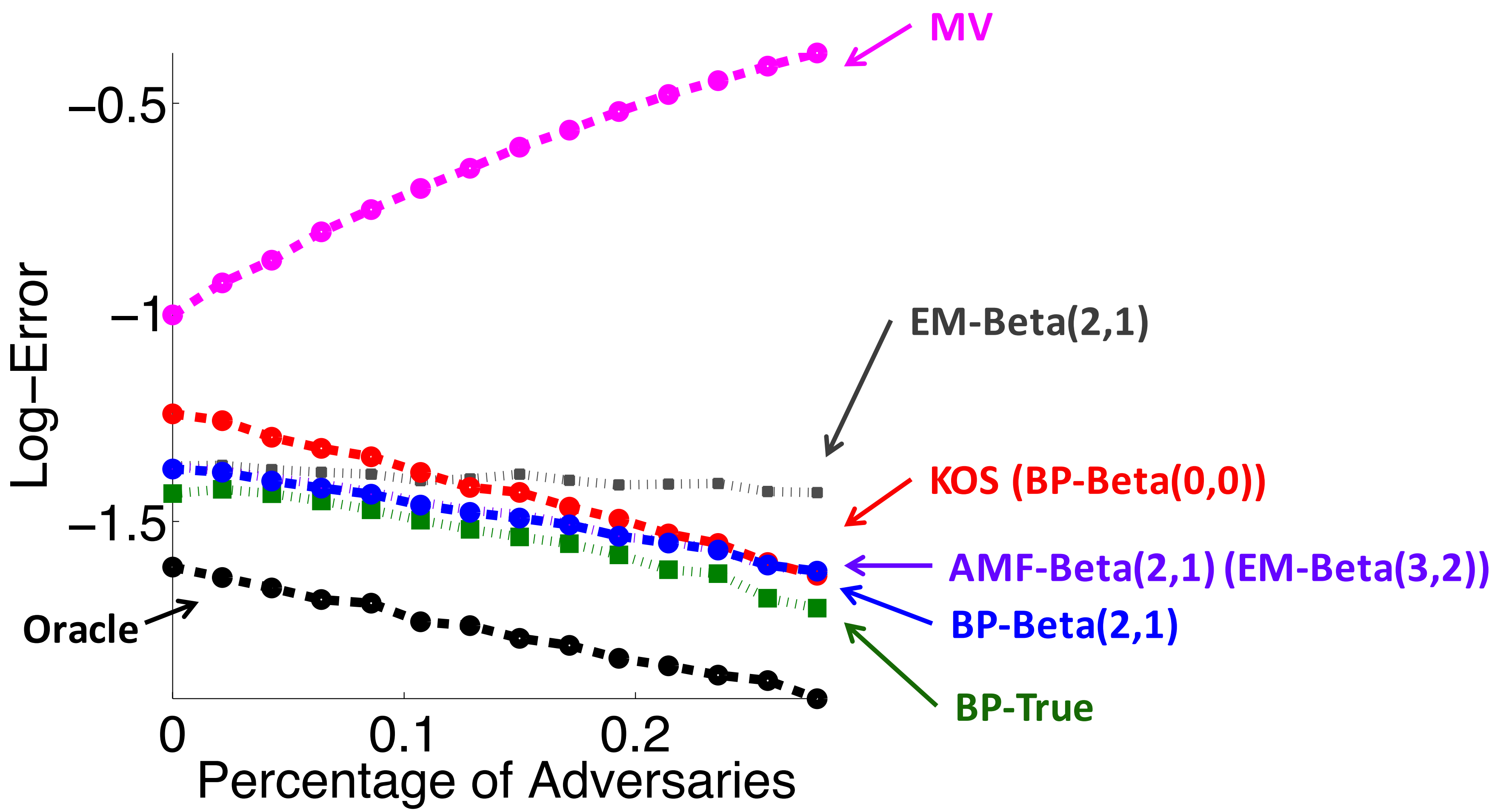




Random (l, r)-regular bipartite graph

- 1000 tasks
- Fixed degrees (l=9, r=9)
- Varying data prior: increasing percentage of adversaries



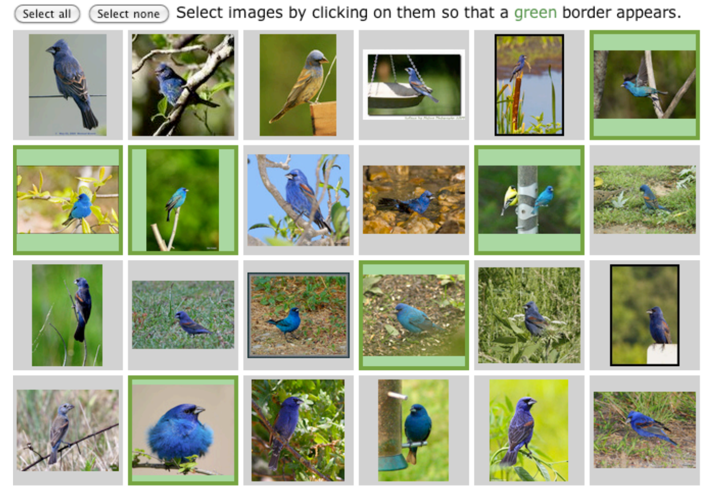


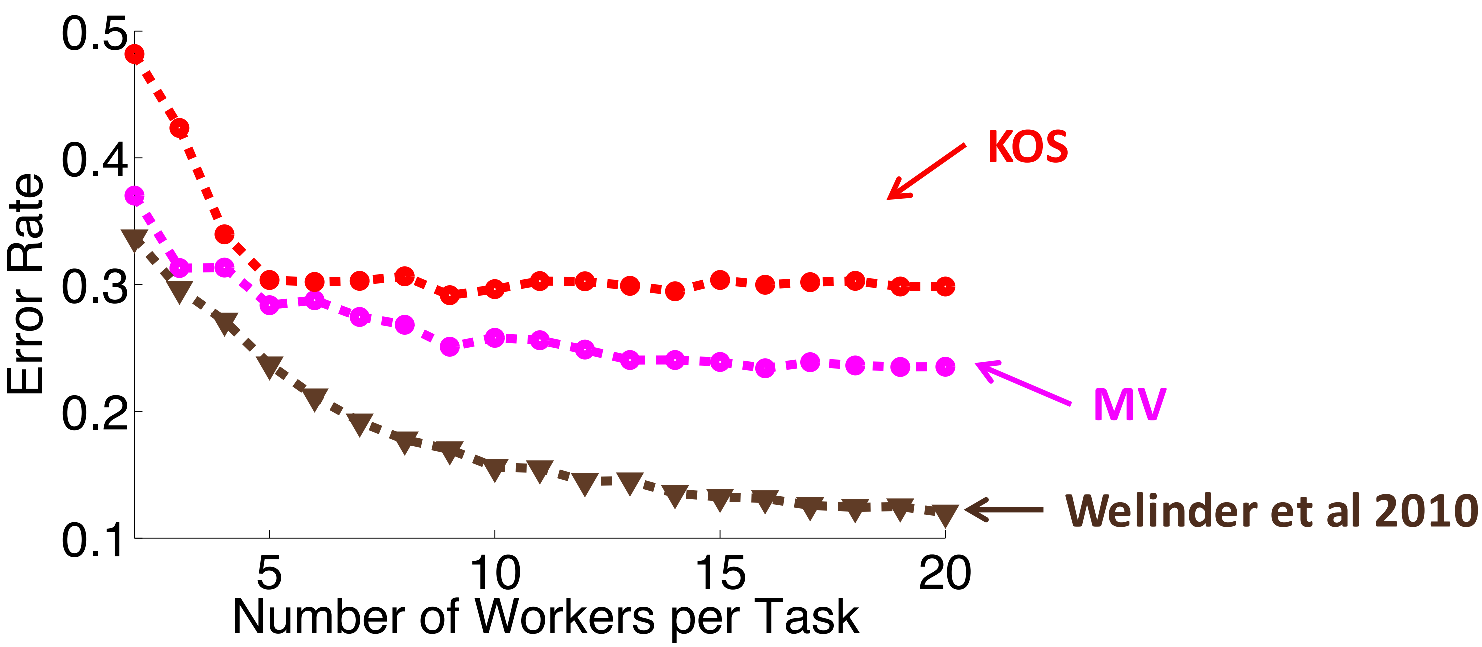
Bluebird dataset (Welinder et al 10)

– 108 tasks

– 39 workers

– Indigo Bunting v.s. Blue GrosBeak?





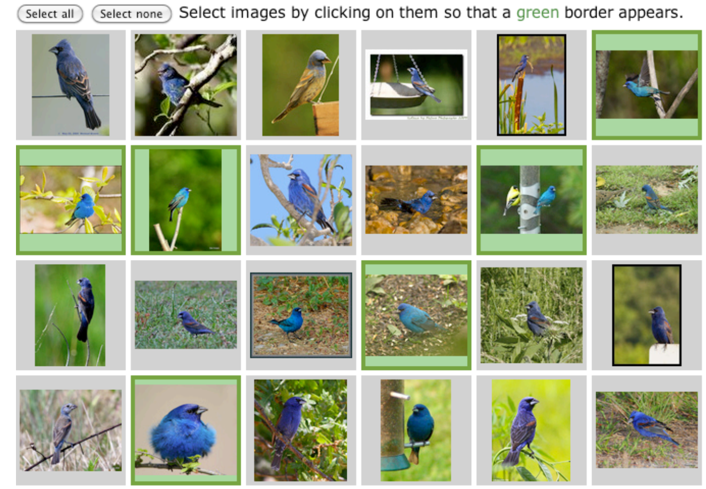
Thanks to P. Welinder and S. Belongie for providing the data and code.

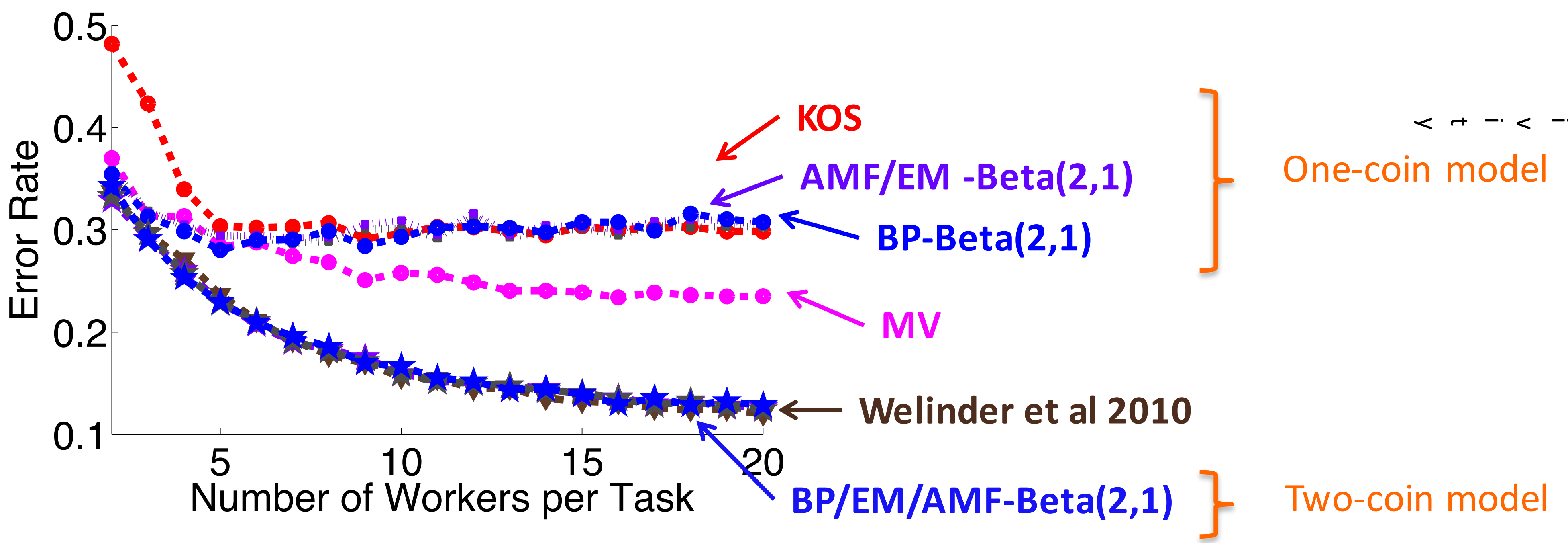
Bluebird dataset (Welinder et al 10)

– 108 tasks

– 39 workers

– Indigo Bunting v.s. Blue GrosBeak?





Thanks to P. Welinder and S. Belongie for providing the data and code.

Natural language datasets
(snow et al 08)

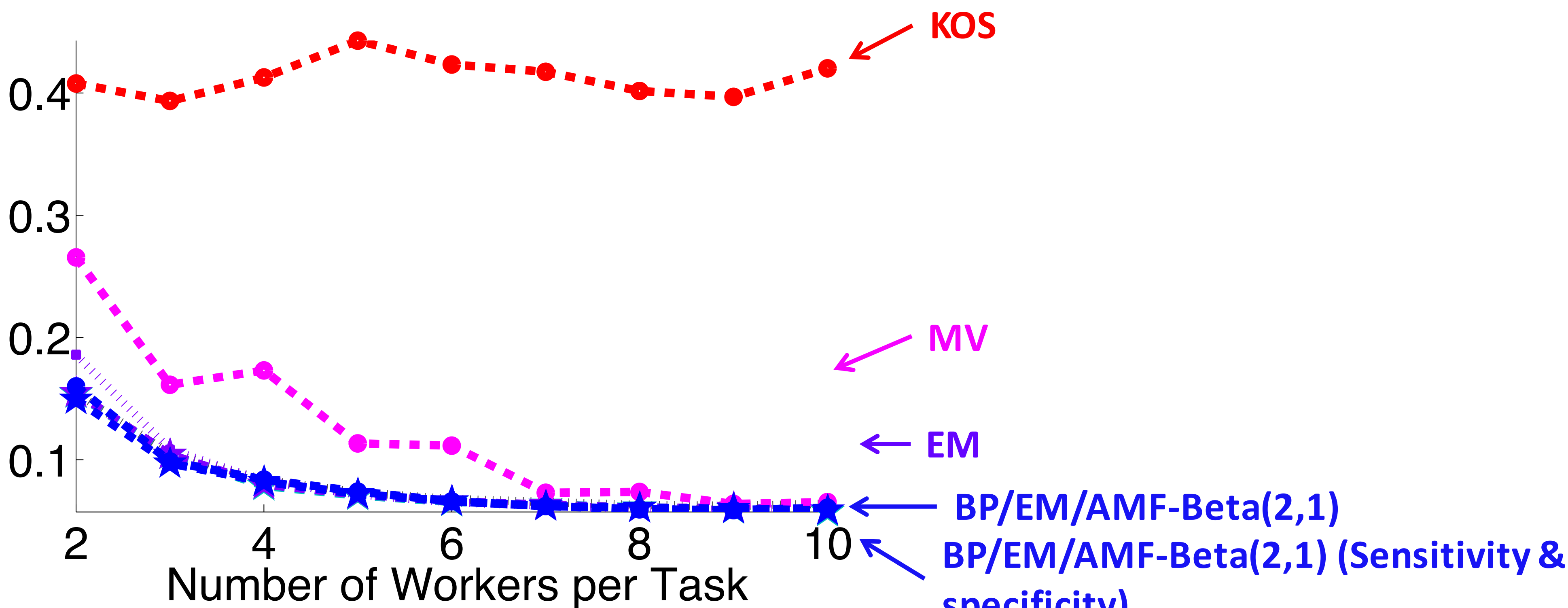
- 800 tasks
- 164 workers

Temporal ordering (TEMP):

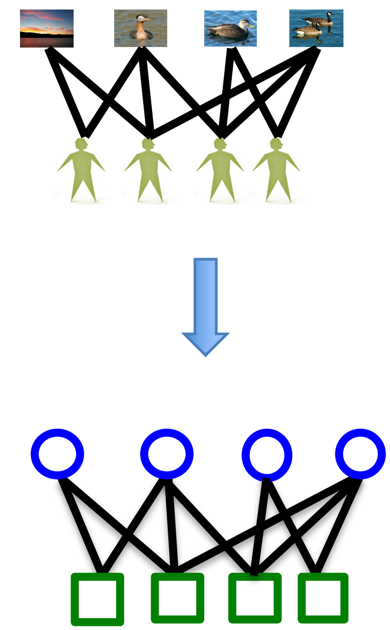
In the following, which event happens first: fell or pushed?

John fell. Sam pushed him.

Answer: Pushed



Conclusions & Future Work



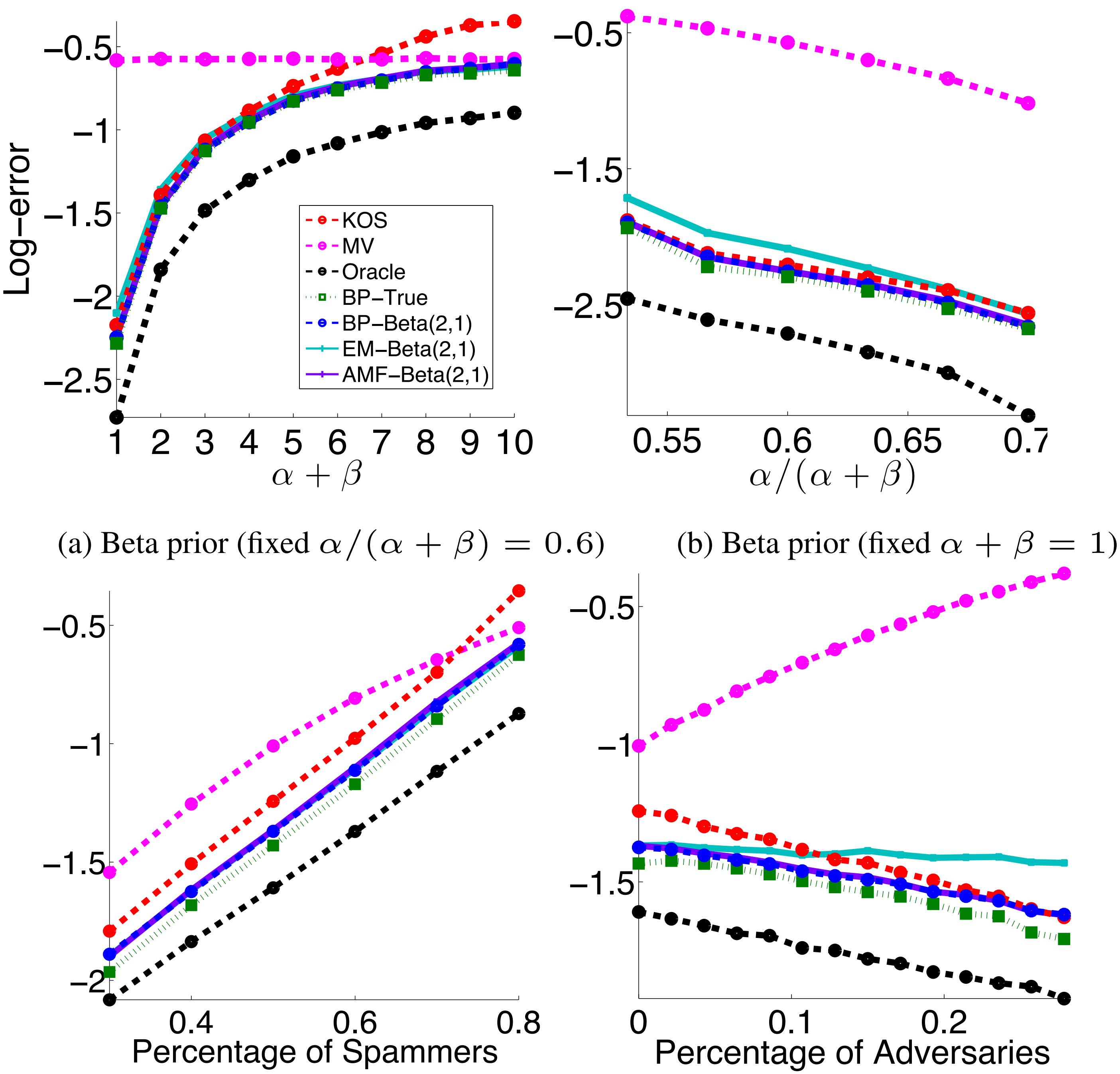
- Crowdsourcing + graphical models
- Belief propagation (KOS, MV), mean field (EM)

- Choice of priors is critical
- Modeling choices
- Inference algorithms

Thanks ☺

Acknowledgements. Work supported in part by two Microsoft Research Fellowships and NSF IIS-1065618.

More results on Different priors



(a) Beta prior (fixed $\alpha/(\alpha+\beta) = 0.6$) (b) Beta prior (fixed $\alpha+\beta = 1$)

(c) Spammer-hammer prior (d) Adversary-spammer-hammer prior

Belief Propagation

Variables -> Factors: $$m_{i\to j}(z_i) \propto \prod_{j'\neq j} m_{j'\to i}(z_i)$$

Factors -> Variables: $$m_{j\to i}(z_i) \propto \sum_{z_{\partial_j \setminus \{i\}}} \psi_j \prod_{i'\neq i} m_{i'\to j}(z_{i'})$$

Marginal probabilities: $$b_i(z_i) = \prod_{j\in\partial_i} m_{j\to i}(z_i)$$

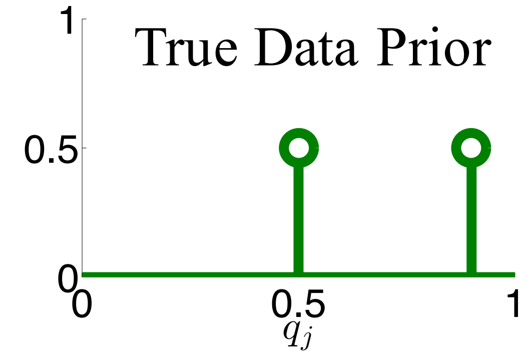
Decode solution: $$z_i = \arg\max_{z_i} b_i(z_i),$$

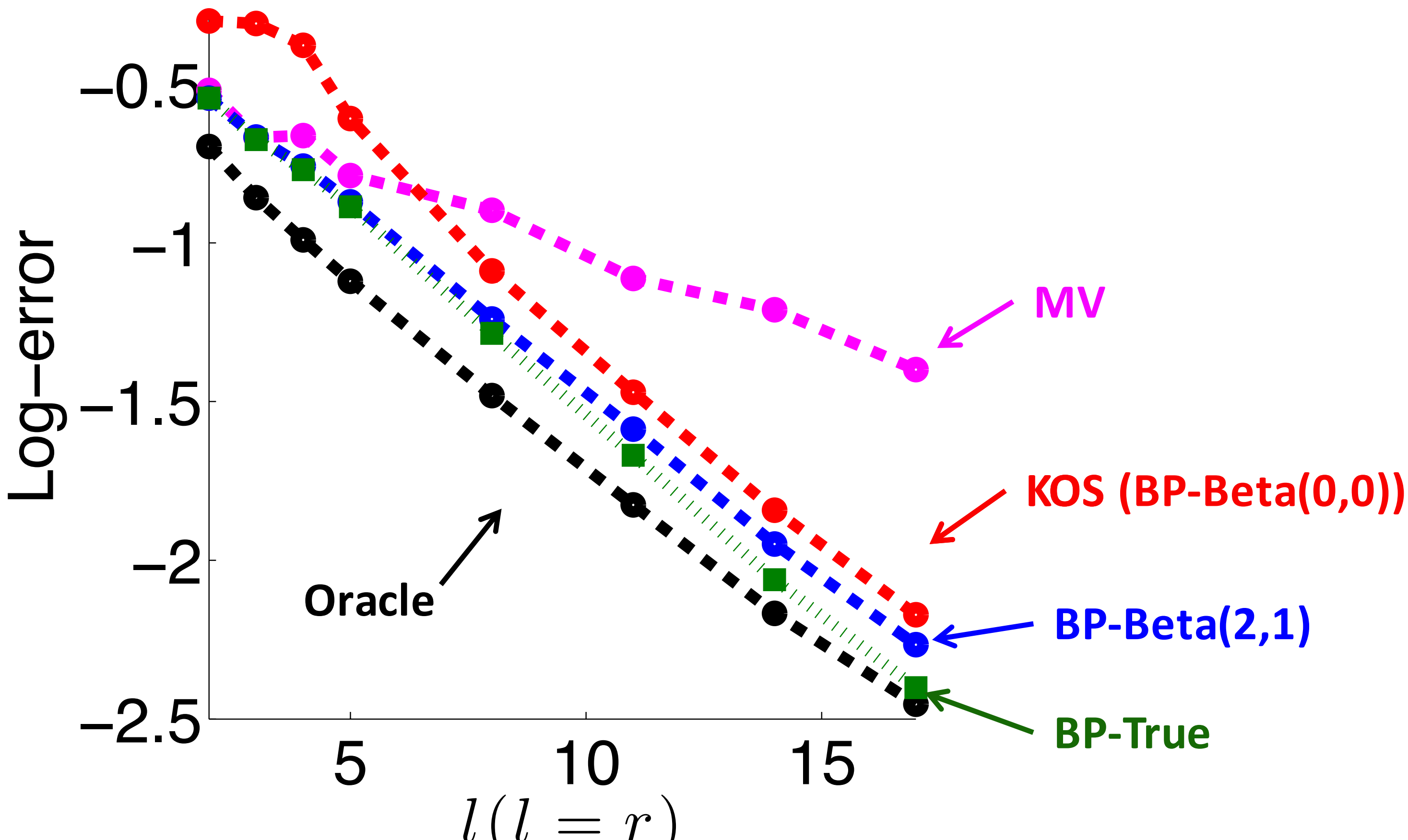
- Log-odds form:

$$\hat{x}_i = \log \frac{b_i(+1)}{b_i(-1)}, \qquad x_{i\to j} = \log \frac{m_{i\to j}(+1)}{m_{i\to j}(-1)}, \qquad y_{j\to i} = \log \frac{m_{j\to i}(+L_{ij})}{m_{j\to i}(-L_{ij})}$$

Random (l, r)-regular bipartite graph

- 1000 tasks
- True data prior: spammer-hammer prior
- Varying both l and r, with $l = r$.





Natural language datasets
(snow et al 08)

- 462 tasks
- 76 workers

Recognizing Textual Entailment:

Text: *Many experts think that there is likely to be another terrorist attack on American soil within the next five years.*
Hypothesis: *There will be another terrorist attack on American soil within the next five years.*
Answer: *NO*