Data Cleaning Using Belief Propagation

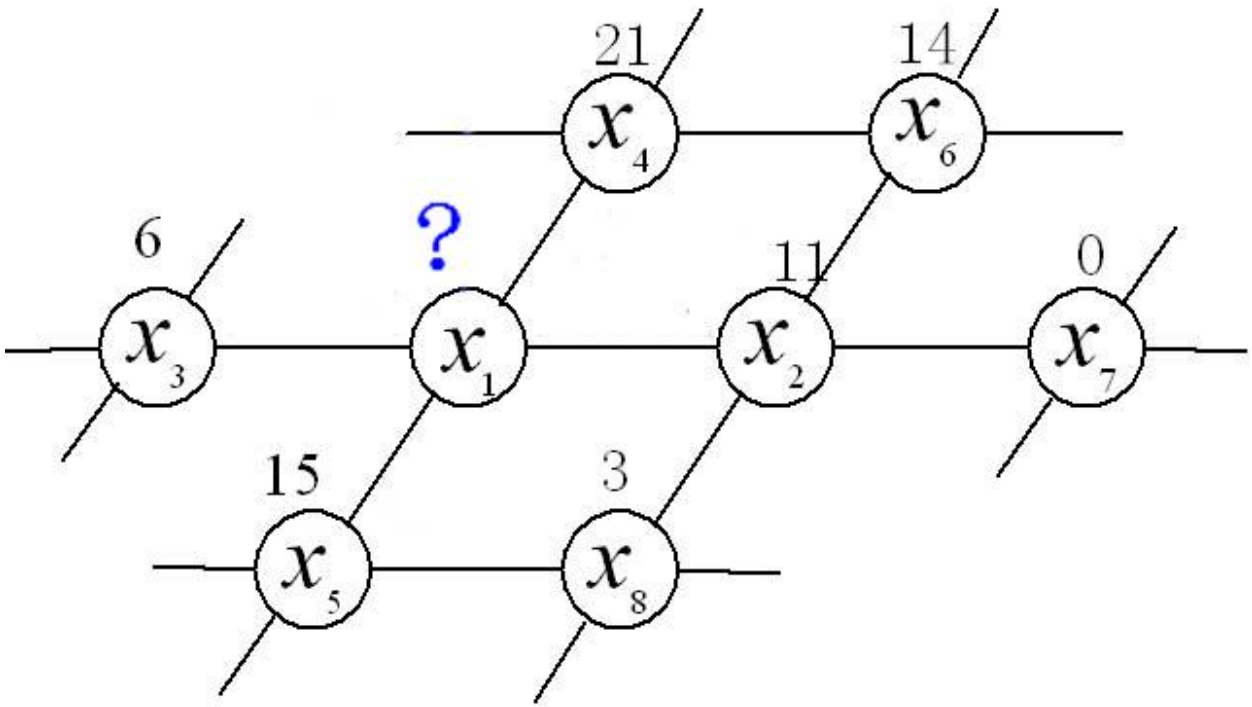
Andrea Fang Chu, Yizhou Wang, D.Stott Parker, Carlo Zaniolo

Department of Computer Science
University of California, Los Angeles

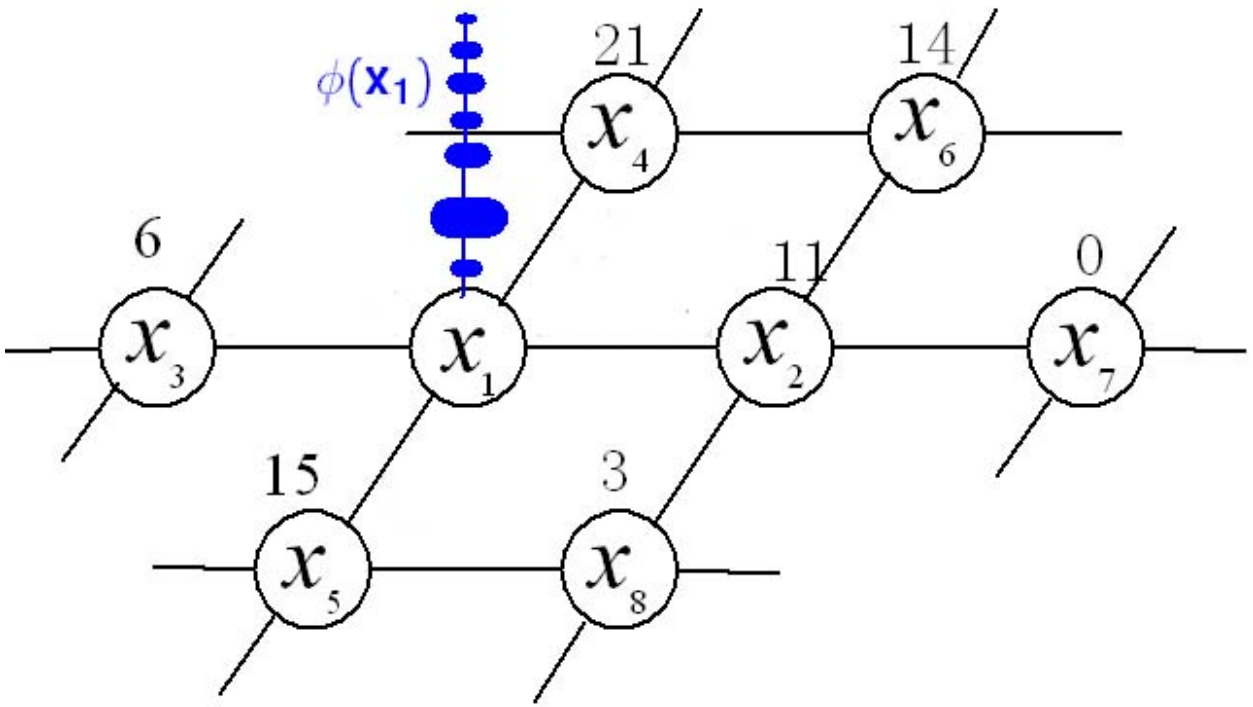
Data Quality

- Data Quality
- Markov Network
- Belief Propagation
- Applications

Example: incomplete sensor readings

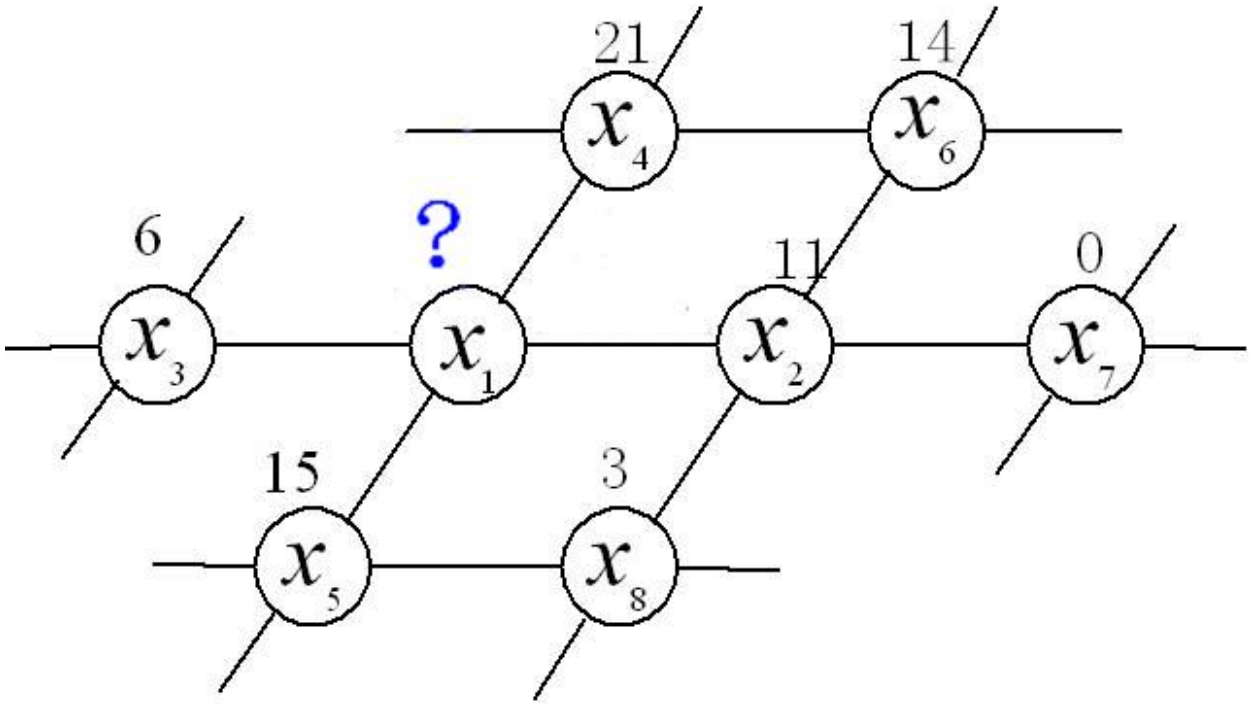


Extreme solution 1: use the marginal probability



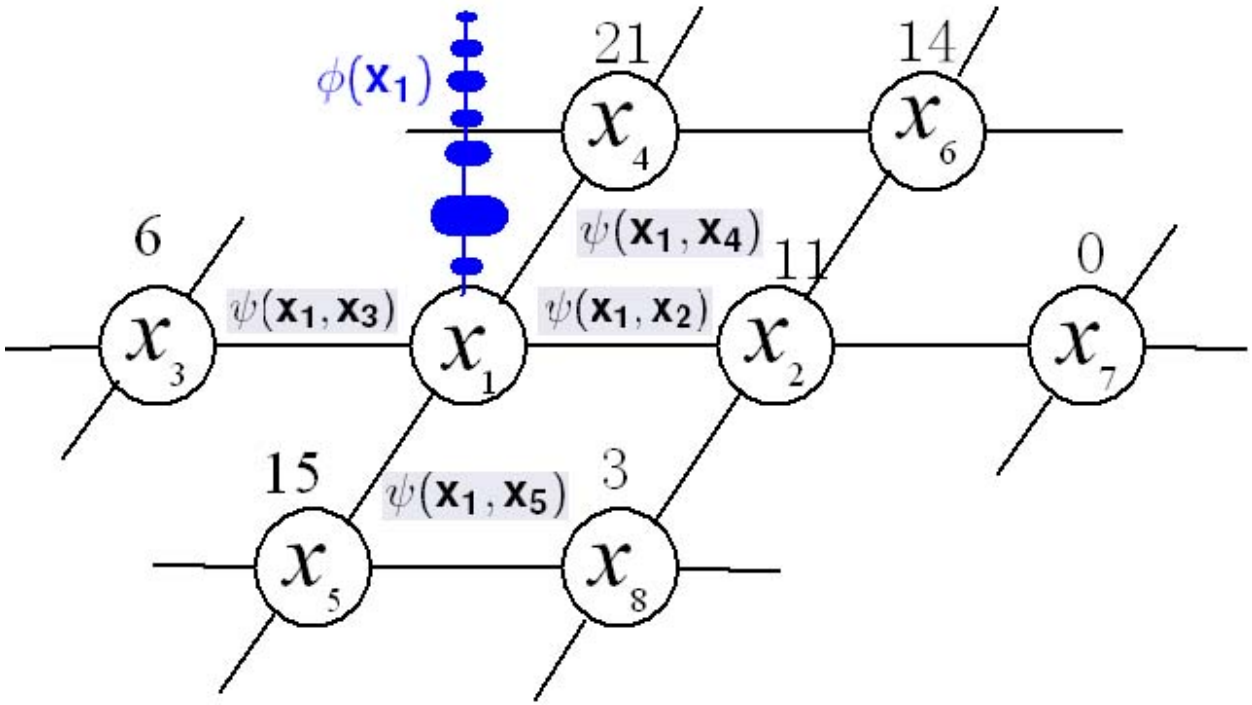
$\hat{x_1}$ is the mean or mode of $\phi(x_1)$.

Extreme solution 2: use the joint probability

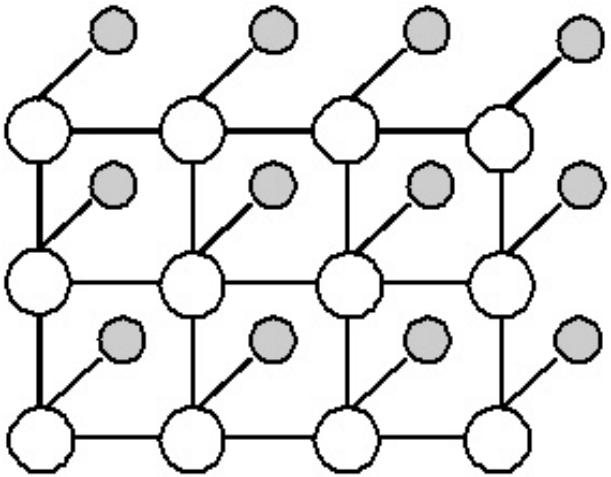


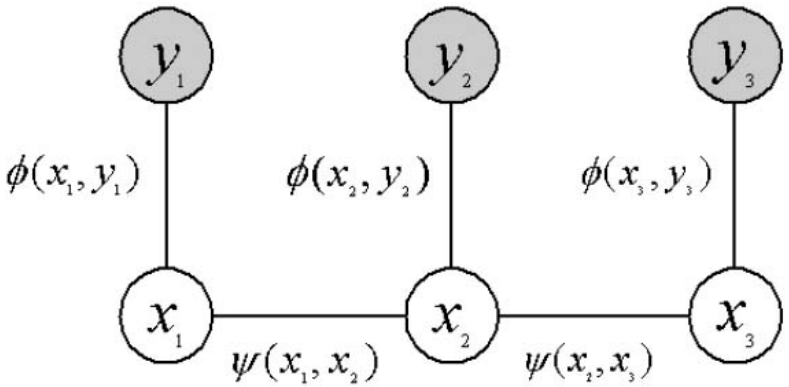
$(\hat{x_1}, x_2, \cdots, x_8)$ is the mean or the mode of the joint posterior probability.

Exploit local dependencies



Pairwise Markov Networks





white circles: random variables $\{x_i\}$.

gray circles: external evidences (or observations) $\{y_i\}$.

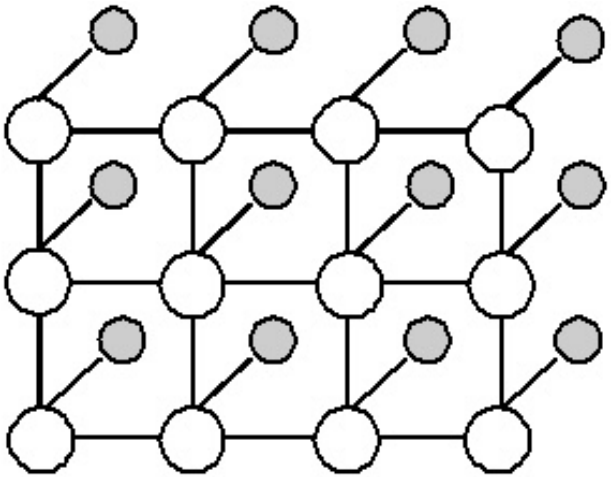
$\phi(x_i, y_i)$: external potential.

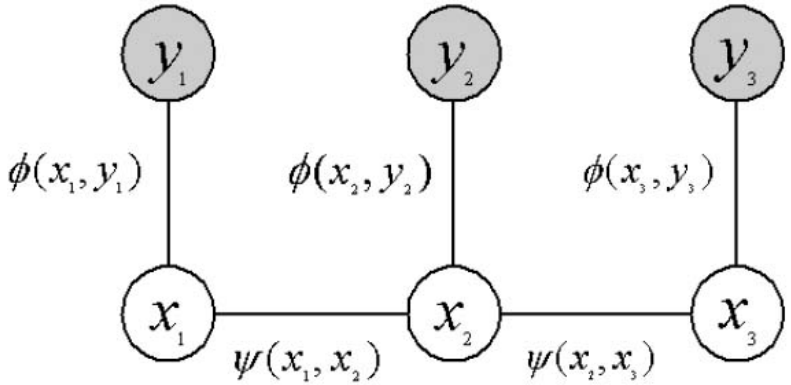
$\psi(x_i, x_j)$: internal binding.

Factorize joint probability:

$$P(\vec{\mathbf{x}}, \vec{\mathbf{y}}) = \frac{1}{Z} \prod_{(i,j)} \psi(x_i, x_j) \prod_i \phi(x_i, y_i)$$

Pairwise Markov Networks





white circles: random variables $\{x_i\}$.

gray circles: external evidences (or observations) $\{y_i\}$.

$\phi(x_i, y_i)$: external potential.

$\psi(x_i, x_j)$: internal binding.

Factorize joint probability:

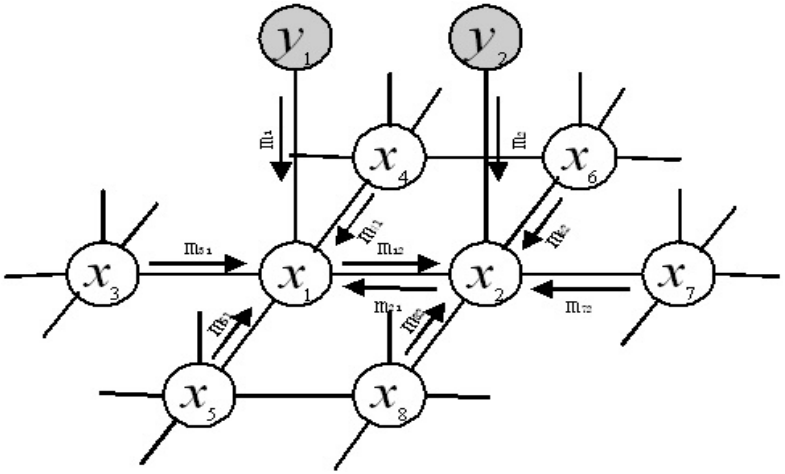
$$P(\vec{\mathbf{x}}, \vec{\mathbf{y}}) = \frac{1}{Z} \prod_{(i,j)} \psi(x_i, x_j) \prod_i \phi(x_i, y_i)$$

Solving a Markov Network

Solving a Markov Network involves two phases:

- *Learning phase:*
  - Determine network structure (variables, neighbors);
  - Learn potential functions: ϕ's and ψ's.
- *Inference phase:*
  - Infer the *mean* or *maximum a posteriori (MAP)* of unknown x_i's, based on ϕ's, ψ's and known y_j's.
  - Efficient inference: Belief Propagation.

Belief Propagation (BP)

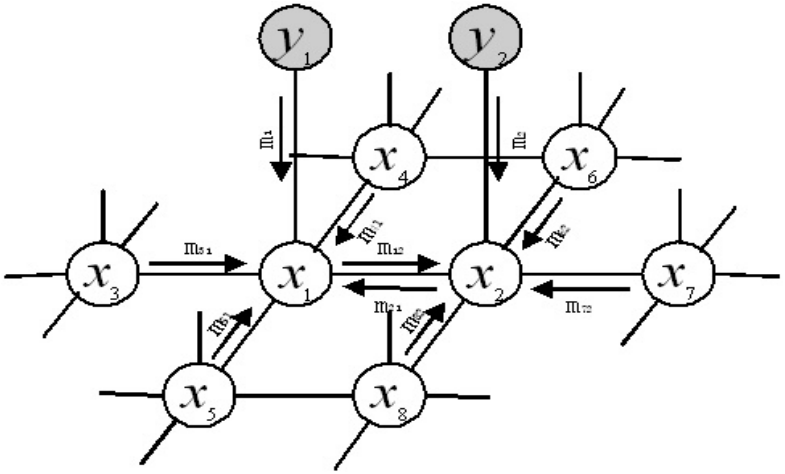


Basics of Belief Propagation: Iterative "message-passing" between neighbors. (J.Yedidia et al, NIPS 94)

$$m_{ij}^{t+1}(x_j) = \sum_{x_i} \phi(x_i, y_i)\psi(x_i, x_j) \prod_{k \in N(i), k \neq j} m_{ki}^{t}(x_i)$$

$$b_i(x_i)_{SUM} = x_i \ \phi(x_i, y_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

Belief Propagation (BP)

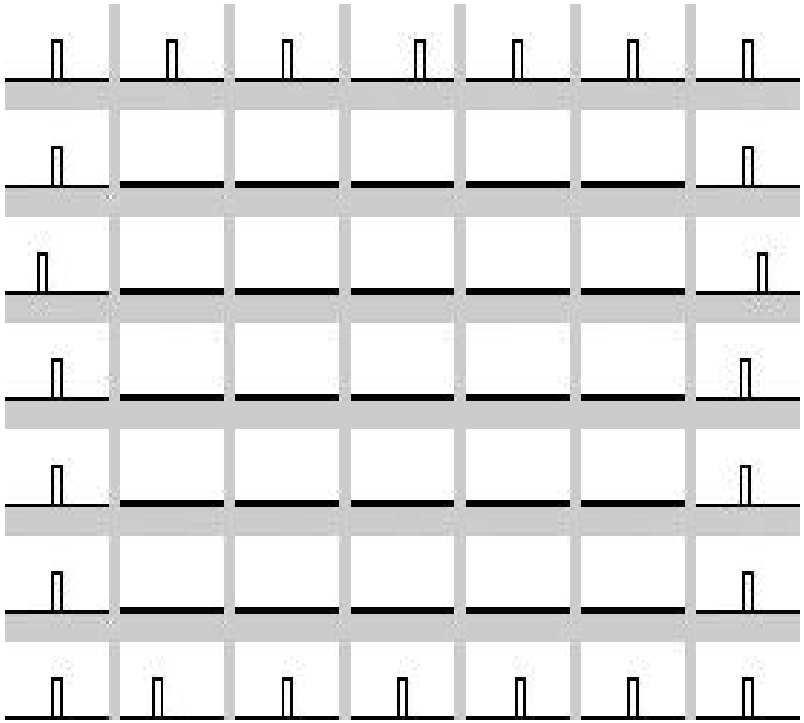


Basics of Belief Propagation: Iterative "message-passing" between neighbors. (J.Yedidia et al, NIPS 94)

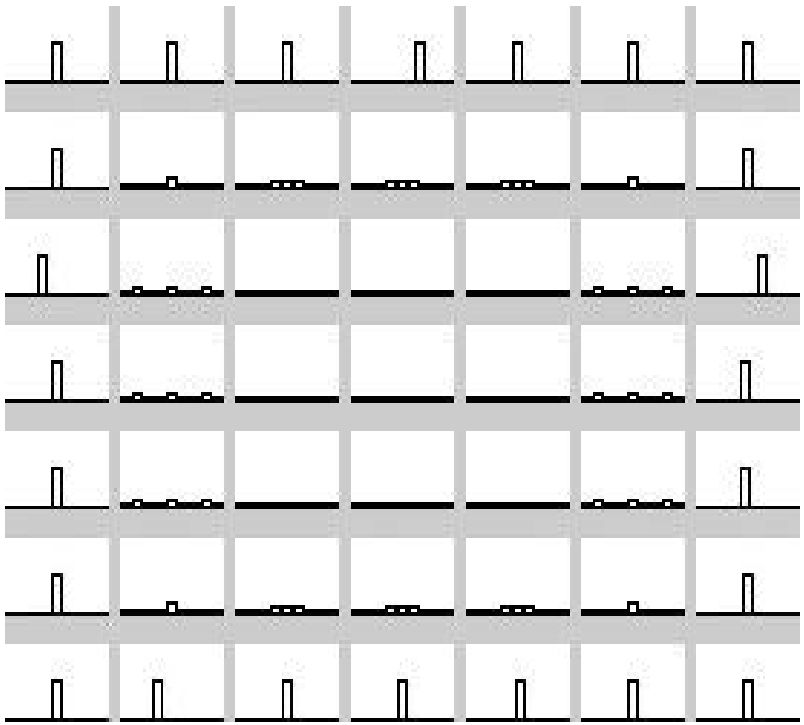
$$m_{ij}^{t+1}(x_j) = \sum_{x_i} \phi(x_i, y_i)\psi(x_i, x_j) \prod_{k \in N(i), k \neq j} m_{ki}^{t}(x_i)$$

$$b_i(x_i)_{SUM} = x_i \ \phi(x_i, y_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

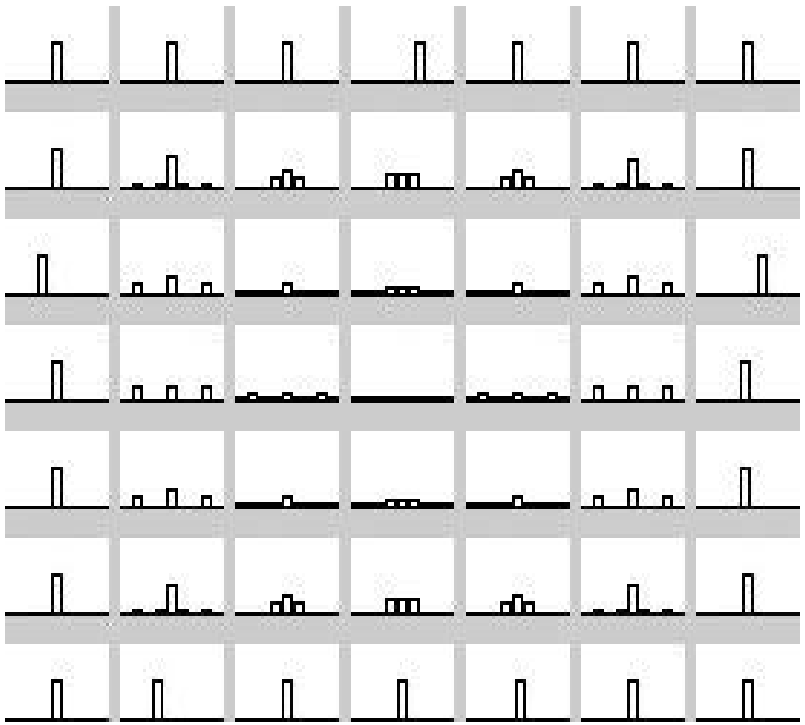
How BP Works



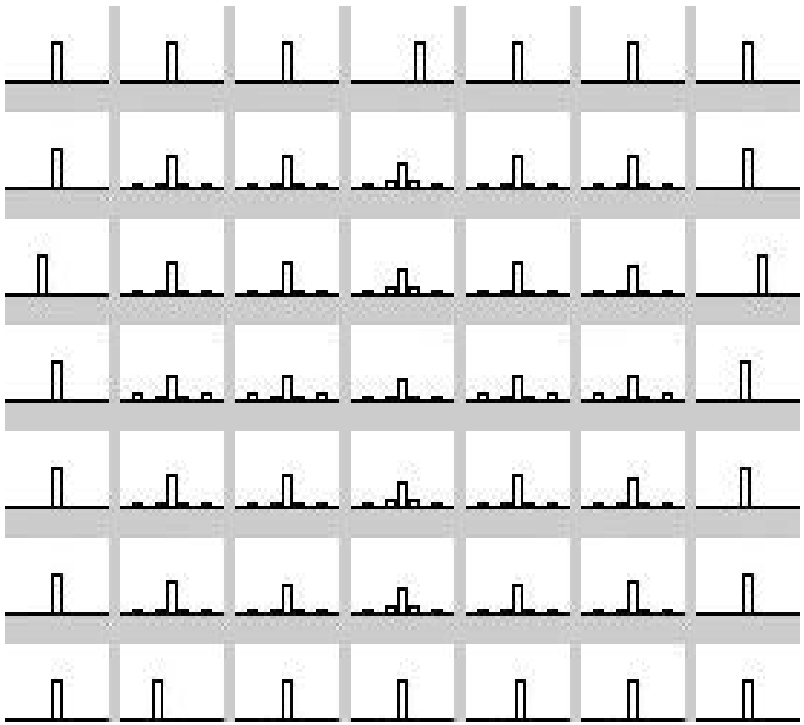
How BP Works



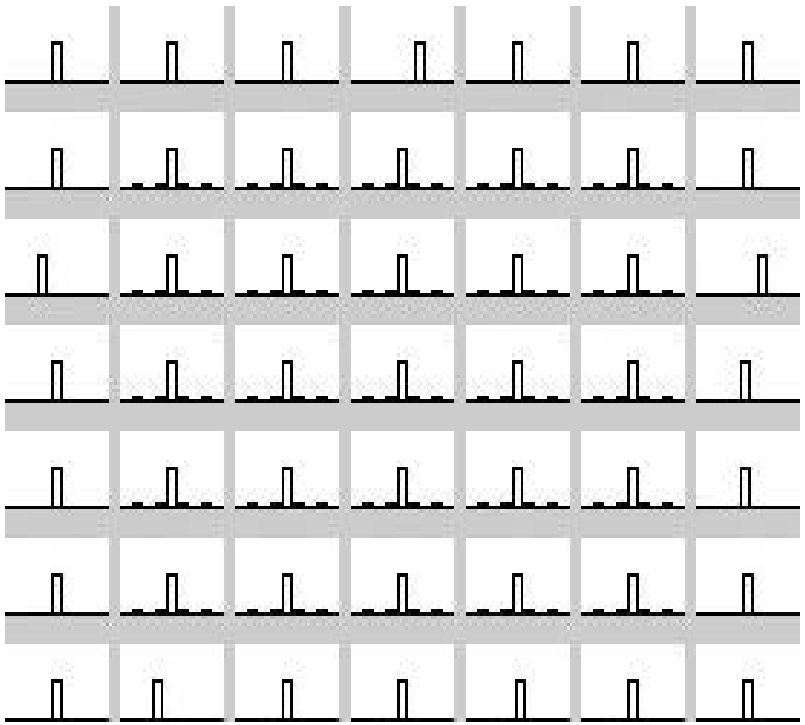
How BP Works



How BP Works

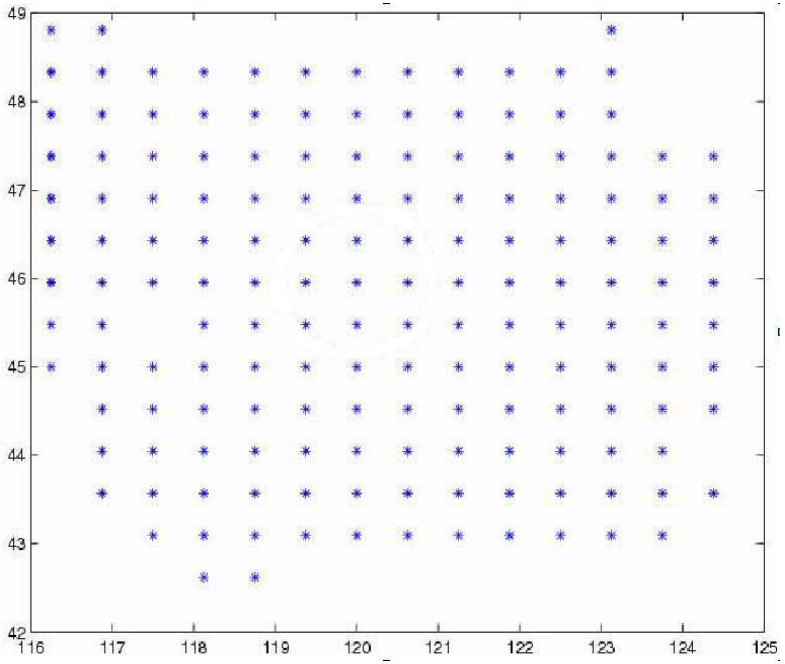


How BP Works



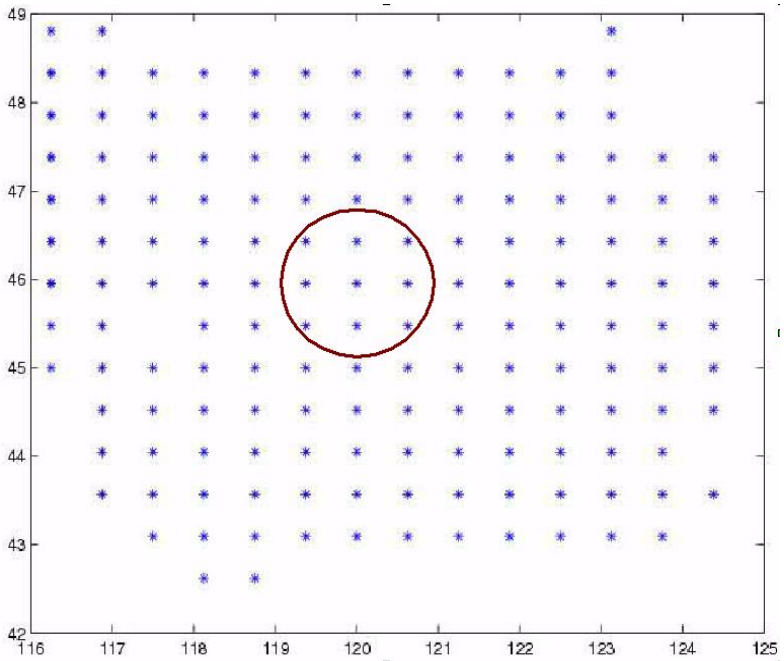
Application: Sensor Probing

sensor map



Application: Sensor Probing

define the neighbors

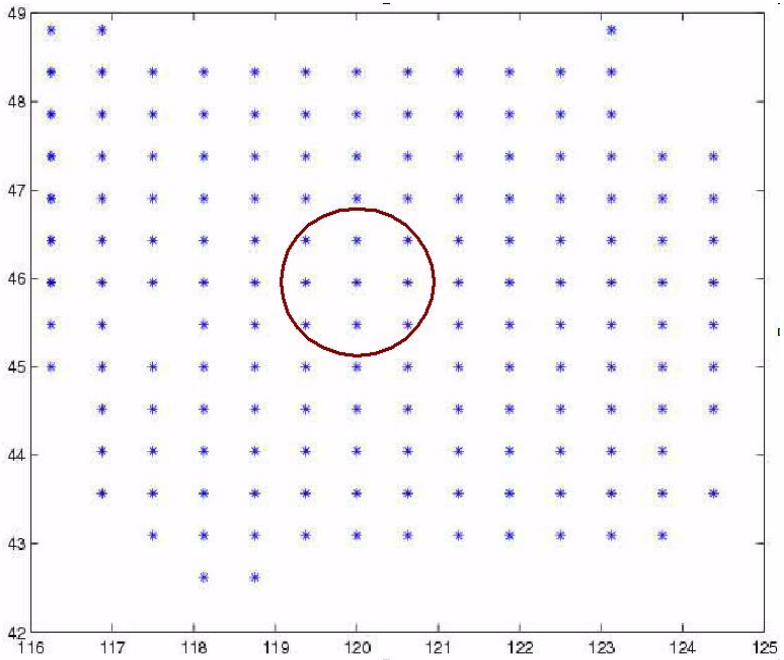


estimate potentials based on history:

$$\phi_i(x_i, y_i) = P(y_i|x_i),$$
$$\psi_{ij}(x_i, x_j) = P(x_j|x_i).$$

Application: Sensor Probing

define the neighbors



estimate potentials based on history:

$$\phi_i(x_i, y_i) = P(y_i|x_i),$$
$$\psi_{ij}(x_i, x_j) = P(x_j|x_i).$$

Top-K Queries for Sensor Probing

Naive probing:

1: Init: compute expected sensor, and pick the top N;
2: Probe selected sensors;
3: Pick the top K out of the probed;

BP-based probing:

1: Init: compute expected sensor readings, and pick the top M;
2: **while** Beliefs not converge **do**
3: Probe selected sensors;
4: Propagate beliefs and update expectations;
5: Pick sensors with top expectations;
6: **end while**
7: Pick the top K out of the probed;

Eg. $\#sensors = 167, K = 10, N = 40, \quad M = 8$

Top-K Queries for Sensor Probing

Naive probing:

1: Init: compute expected sensor, and pick the top N;
2: Probe selected sensors;
3: Pick the top K out of the probed;

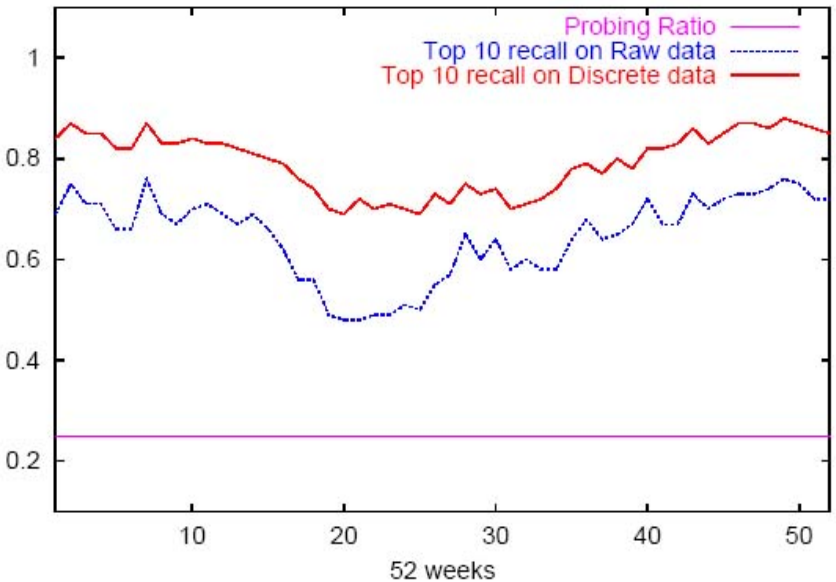
BP-based probing:

1: Init: compute expected sensor readings, and pick the top M;
2: **while** Beliefs not converge **do**
3: Probe selected sensors;
4: Propagate beliefs and update expectations;
5: Pick sensors with top expectations;
6: **end while**
7: Pick the top K out of the probed;

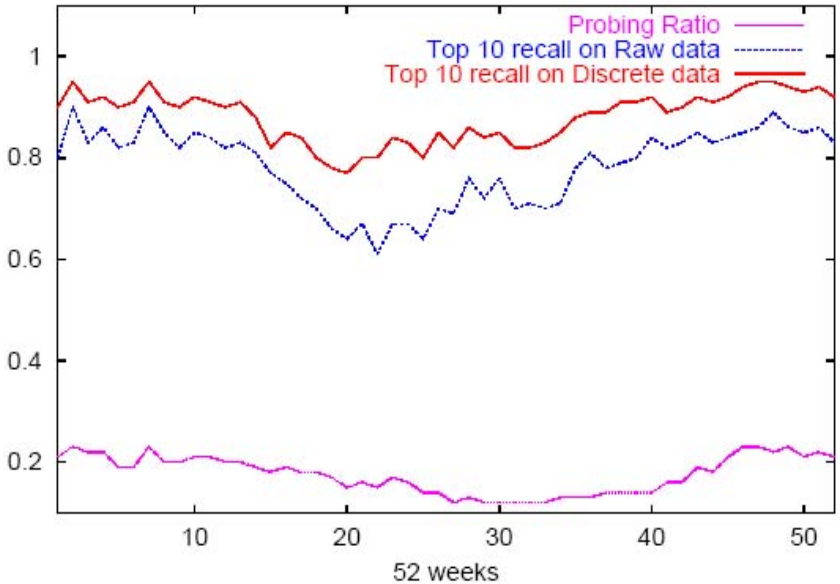
Eg. $\#sensors = 167, K = 10, N = 40, \quad M = 8$

Top-K Queries for Sensor Probing

Naive



BP-based

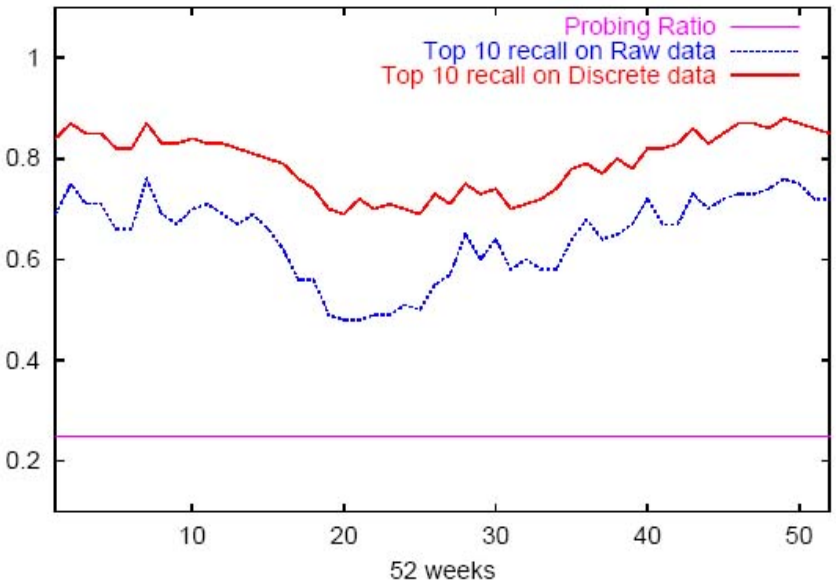


BP-based approach against Naive on average:

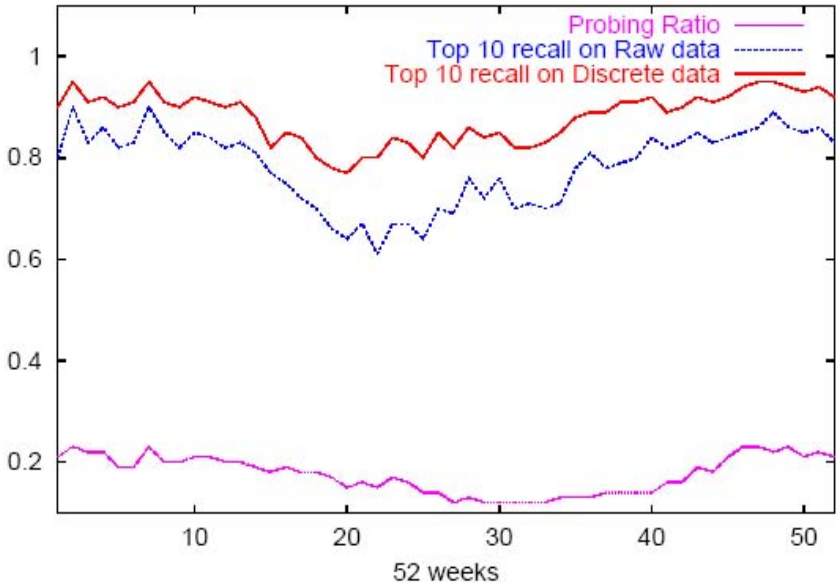
- 8% less probing;
- 13.6% higher recall for raw values; and
- 7.7% higher recall for discrete values.

Top-K Queries for Sensor Probing

Naive



BP-based

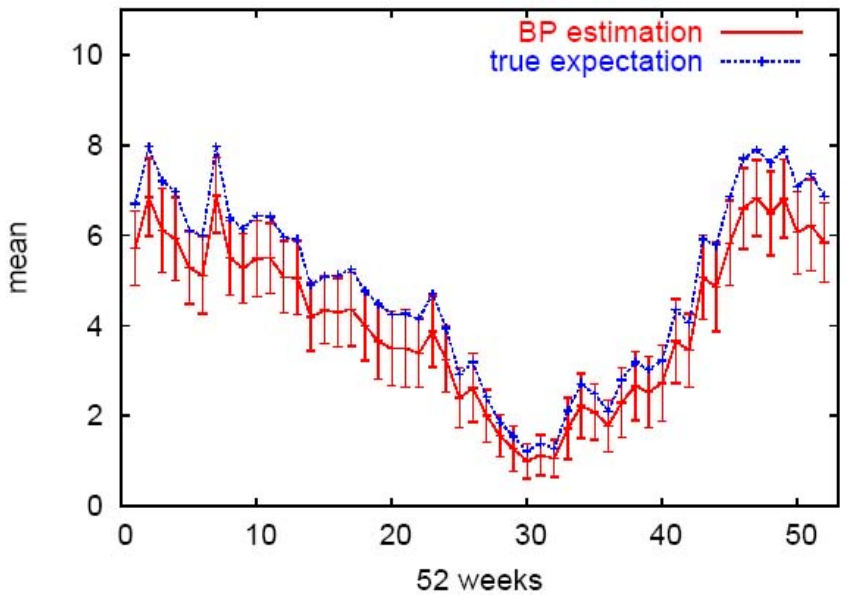


BP-based approach against Naive on average:

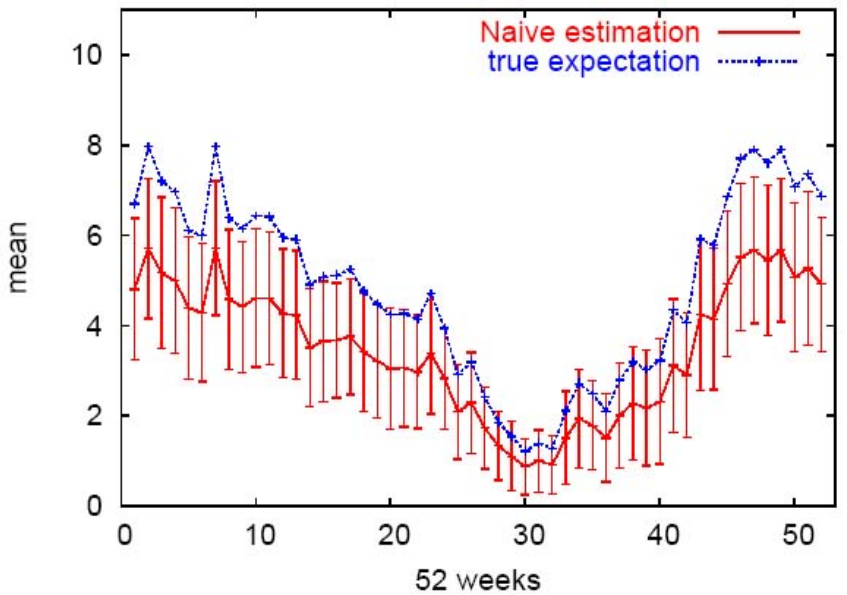
- 8% less probing;
- 13.6% higher recall for raw values; and
- 7.7% higher recall for discrete values.

Average Queries for Sensor Probing

BP-based



Naive

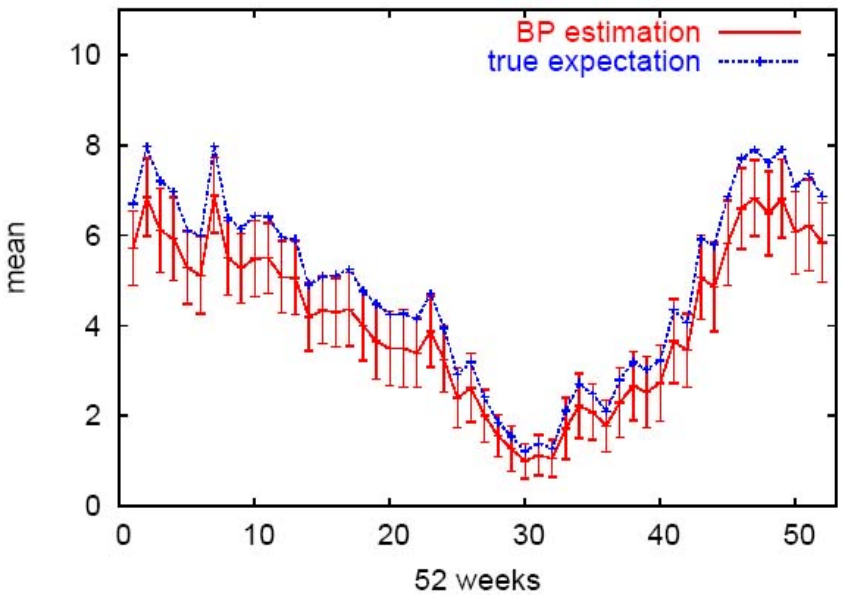


Average estimation error bars, on discrete values (0-11). BP-based approach against Naive on average:

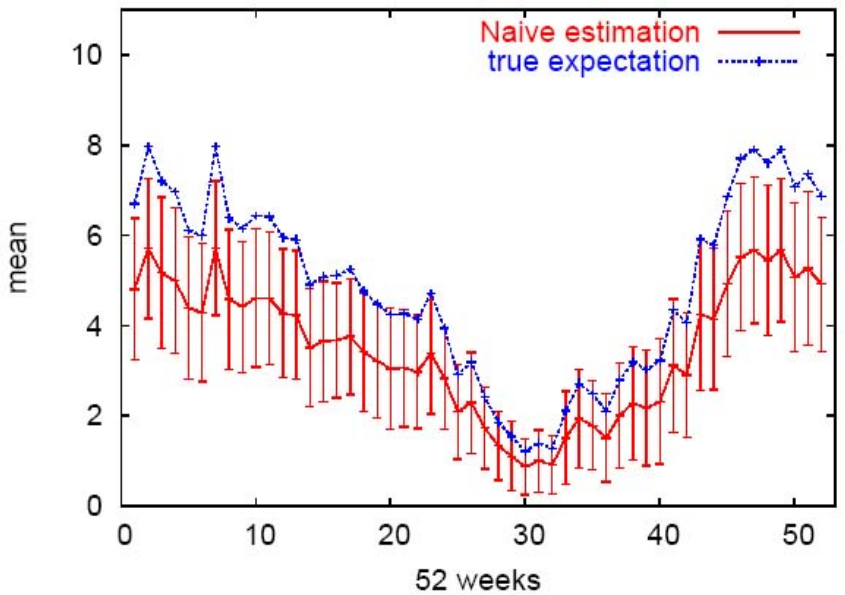
- 8% less probing;
- BP has a mean error of -0.75, deviation 0.79.
- Naive has a mean error of -1.39, deviation 1.35.

Average Queries for Sensor Probing

BP-based



Naive



Average estimation error bars, on discrete values (0-11). BP-based approach against Naive on average:

- 8% less probing;
- BP has a mean error of -0.75, deviation 0.79.
- Naive has a mean error of -1.39, deviation 1.35.

Application: Text Denoising

rule	mutation prob.	# errors	% corrected
x $\longrightarrow$ k	100%	56	91%
f $\longrightarrow$ d	30%	123	92%
f $\longrightarrow$ z	28%	118	87%
th $\longrightarrow$ tn	52%	220	96%
se $\longrightarrow$ ue	18%	51	93%
se $\longrightarrow$ le	25%	69	94%
se $\longrightarrow$ ie	21%	58	95%
tio $\longrightarrow$ tho	20%	35	100%
tio $\longrightarrow$ txo	20%	35	100%
tio $\longrightarrow$ two	31%	57	98%
total words/errors: 3459/822		overall accuracy: 94%	

Distortion rules and error correction

Application: Text Denoising

rule	mutation prob.	# errors	% corrected
x $\rightarrow$ k	100%	56	91%
f $\rightarrow$ d	30%	123	92%
f $\rightarrow$ z	28%	118	87%
th $\rightarrow$ tn	52%	220	96%
se $\rightarrow$ ue	18%	51	93%
se $\rightarrow$ le	25%	69	94%
se $\rightarrow$ ie	21%	58	95%
tio $\rightarrow$ tho	20%	35	100%
tio $\rightarrow$ txo	20%	35	100%
tio $\rightarrow$ two	31%	57	98%
total words/errors: 3459/822		overall accuracy: 94%	

Distortion rules and error correction

Related Work

- Sensor probing
  - BBQ: Global multivariate Gaussian (A.Deshpande, et al, vldb04)
- Text cleaning
  - NLP (G.Salton et al, McGraw Hill 83)
- Data cleaning
  - Special purpose data cleaning
  - Classification rule-based
  - Outlier detection

Conclusion

- Unified Approach to Data Cleaning
  - By exploiting data dependency
  - Inferring missing values
  - Correcting noisy values
- Future Work
  - Extending to dynamic graph structure
  - Application: Web usage analysis

Thank you !