Bayesian Machine Learning (ML): Modeling And Inference in Big Data

Zhuhua Cai
Google, Rice University
caizhua@gmail.com

Syllabus

- Bayesian ML Concepts (Today)
- Bayesian ML on MapReduce (Next morning)
- Bayesian ML on Spark, SimSQL and Giraph (Next afternoon)
- Discussions (Thursday morning)

ML is Everywhere

Problem 1

Age	Income
18	9127
25	10192
26	12302
26	11467
27	12540
28	12597
28	13136
29	10343
29	11578
30	12828
31	13748
31	14548
32	13160
33	13595
33	13915
35	15695
37	15906
37	16926
43	17428

Age	Income
20	?
21	?
30	?
25	?

Figure. An example for supervised regression.

Problem 2

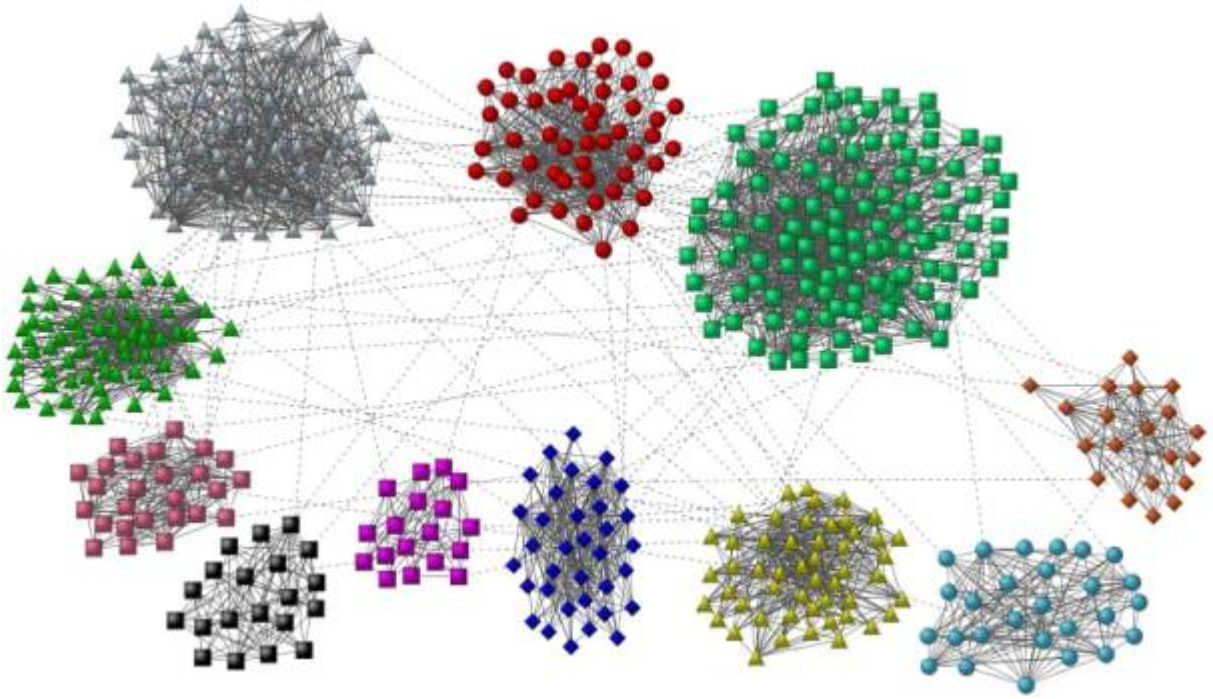


Figure. An example for unsupervised classification.

Bayesian ML

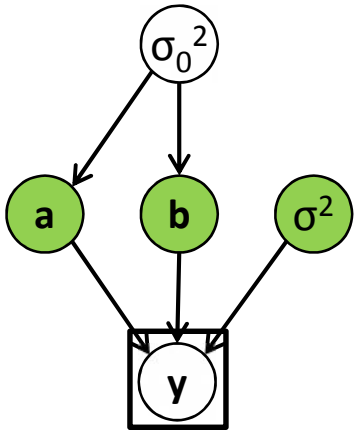
- To apply Bayesian ML:
  - Use statistical process to explain how data is generated, i.e., generative process, $P(D|\Theta)$.
  - Learn parameters/variables in the process given data $P(\Theta|D)$. Use *Bayes' Rule* to learn $P(\Theta|\mathrm{D})$:

$$P(\Theta|\mathrm{D}) = \frac{P(D|\Theta)P(\Theta)}{P(D)} \sim P(D|\Theta)P(\Theta)$$

likelihood **prior**

Solution for Problem 1

- Model: Bayesian LR
  1. Generate $a \sim N\ (0, \sigma_0^2)$
  2. Generate $b \sim N(0, \sigma_0^2)$
  3. Generate $\sigma^2 \sim InvGamma(1, 1)$
  4. Given each age x_i:
     income $y_i \sim N\ (ax_i + b, \sigma^2)$



Age	Income
18	9127
25	10192
26	12302
26	11467
27	12540
28	12597
28	13136
29	10343
29	11578
30	12828
31	13748
31	14548
32	13160
33	13595
33	13915
35	15695
37	15906
37	16926
43	17428

Age	Income
20	?
21	?
30	?
25	?

Figure. Training set and test set

Bayesian Inference

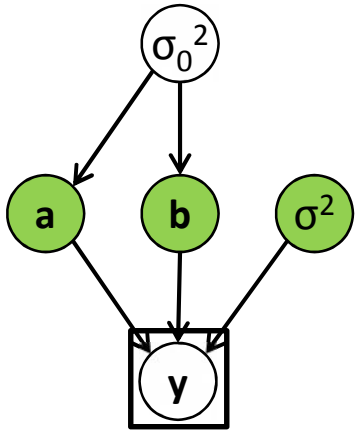
- Using Bayes' theorem

Let $\Theta = \{a, b, \sigma^2\}$, *then*

$$P(\Theta|y) \propto P(y|\Theta) \times P(\Theta)$$

$$P(y|\Theta) = \prod_i N(y_i|ax_i + b, \sigma^2)$$

$$P(\Theta) = N(a|0, \sigma_0^2) \times N(b|0, \sigma_0^2) \times InvGamma(\sigma^2|1, 1)$$



Inference Methods

$$P(a, b, \sigma^2 | x, y) \propto \prod_i N(y_i | ax_i + b, \sigma^2) \times N(a | 0, \sigma_0^2) N(b | 0, \sigma_0^2) InvGamma(\sigma^2 | 1, 1)$$

- Newton-Raphson algorithm, gradient descent algorithm, etc.
- Expectation–maximization (EM) algorithm.
- Approximate inference methods.
- Sampling methods like Monte Carlo Markov chain (MCMC).

Markov Chain Monte Carlo

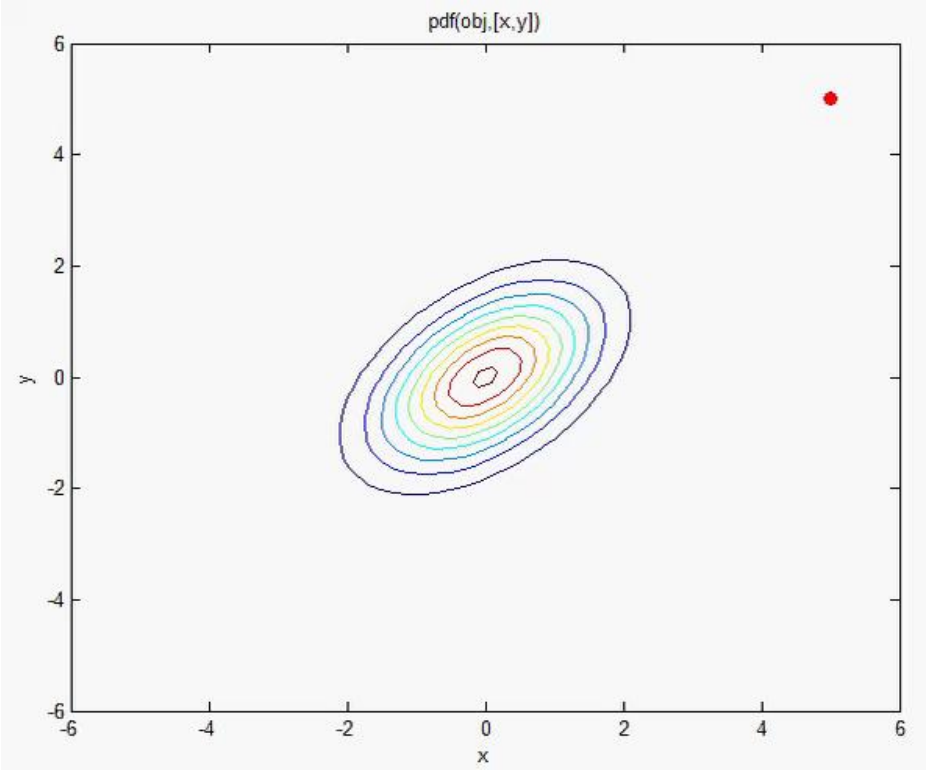
- Markov Chain
  - A collection of random variables $\{X_t\}$ (r = 0, 1, …) having the property that

$$P(X_t|X_0, X_1,..X_{t-1}) = P(X_t|X_{t-1})$$

- Markov Chain Monte Carlo (MCMC)
  - A class of sampling algorithms to sample a distribution by constructing a Markov chain, whose equilibrium states approximate the desired distribution.

Bivariate Normal Distribution

- Input:
  - mean: (0, 0).
  - covariance: [2, 1; 1 2].
- **Output:**
  - A set of sampled data points.
- Steps:
  - Select an initial point (x_0, y_0).
  - Sample (x_{i+1}, y_{i+1}) based on (x_i, y_i) repeatedly.
  - After a burn-in number of steps, collect samples periodically.



Monte Carlo Markov chain

- Gibbs sampling

```
Initialize Θ
while(true){
    choose θ ⊆ Θ;
    sample θ ~ P(θ|Θ − θ, D);
}
```

- Learning parameters in example 1

1) Initialize a and b;
2) $P(\sigma^2|.) \propto \prod_i N(y_i|ax_i + b, \sigma^2) \times InvGamma(\sigma^2|1,1)$;
3) $P(a|.) \propto \prod_i N(y_i|ax_i + b, \sigma^2) \times N(a|0, \sigma_0^2)$;
4) $P(b|.) \propto \prod_i N(y_i|ax_i + b, \sigma^2) \times N(b|0, \sigma_0^2)$;
5) *repeate steps 2) through 4) a number of times.*

Monte Carlo Markov chain

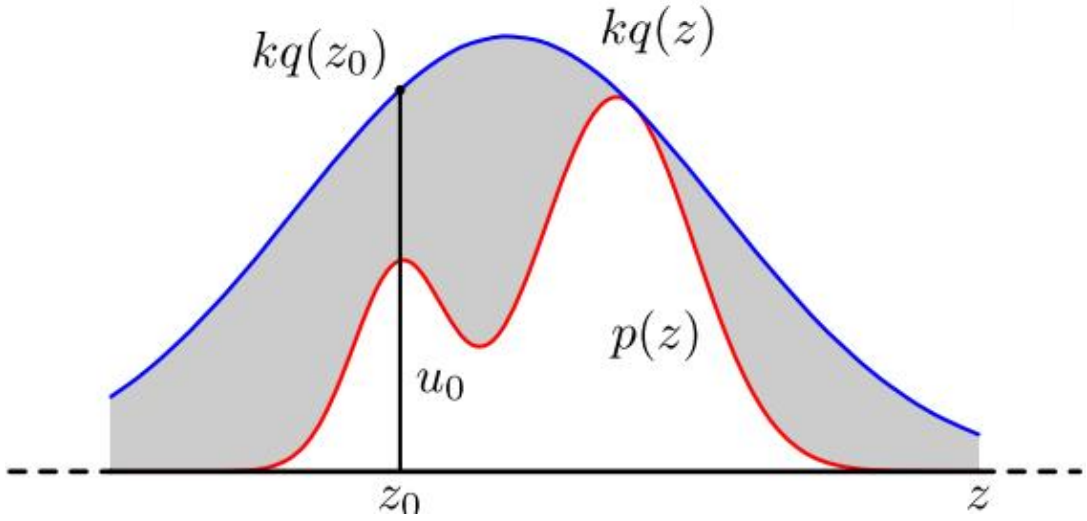
- Each step is a standard distribution.
  - $P(\sigma^2|.) \propto \prod_i N(y_i|ax_i+b,\sigma^2) \times InvGamma(\sigma^2|1,1)$
    $\propto InvGamma(\sigma^2|1+0.5n, 1+0.5\sum_i(y_i-ax_i-b)^2)$
  - $P(a|.) \propto \prod_i N(y_i|ax_i+b,\sigma^2) \times N(a|0,\sigma_0^2)$
    $\propto N\left(a|\frac{\sum_i(y_i-b)x_i}{\sum_i x_i^2}, \frac{\sigma^2}{\sum_i x_i^2}\right) \times N(a|0,\sigma_0^2)$ --- It is a normal distribution.
  - $P(b|.) \propto \prod_i N(y_i|ax_i+b,\sigma^2) \times N(b|0,\sigma_0^2)$
    $\propto N\left(b|\frac{\sum_i(y_i-ax_i)}{\left(\frac{\sigma^2}{\sigma_0^2}\right)+n}, \left(\frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}\right)^{-1}\right)$

Code for Problem 1

- example1.m

Rejection sampling

- We want to sample from $p(z)$
  1) Find a simple distribution $q(z)$.
  2) Find a constant k such that $\forall z, kq(z) \geq p(z)$.
  3) Generate $z_0 \sim q(z)$, and generate $\mu \sim Uniform(0, kq(z_0))$.
  4) *if* $\mu < p(z)$ use μ as a sample;
     otherwise repeat step 1) *to* 4).



ML with Big Data

- Data is big.
  - Row number n.
  - Column number m.
- Problems
  - It is harder to learn.
  - It is harder to model.

Problem 3

- Problem. Given the 20newsgroup dataset, we want to create a binary classifier to classify religion and non-religion groups.
- Dataset: http://qwone.com/~jason/20Newsgroups/20news-19997.tar.gz

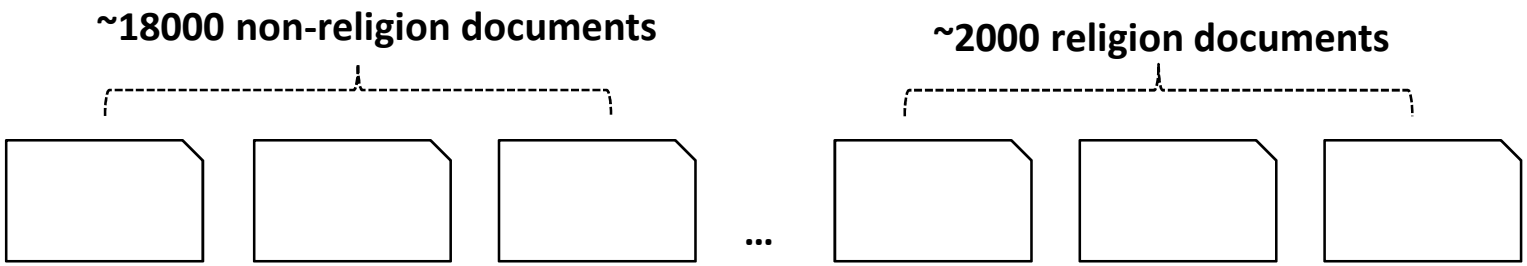
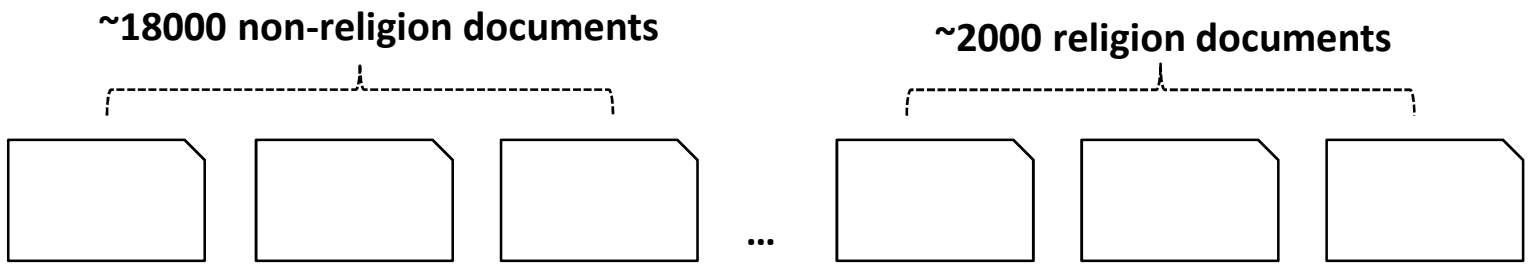


Figure. Data set.

Problem Modeling



$$
\begin{array}{lll}
doc_0 & [word_0: count_{00}, word_1: count_{01}, word_2: count_{02}, \ldots, word_n: count_{0m}] & 1 \\
doc_1 & [word_0: count_{10}, word_1: count_{11}, word_2: count_{12}, \ldots, word_n: count_{1m}] & 0 \\
 & \ldots & \\
doc_n & [word_0: count_{n0}, word_1: count_{n1}, word_2: count_{n2}, \ldots, word_n: count_{nm}] & 1
\end{array}
$$

Bayesian LR

- Create a classifier
  - SVM, KNN, etc. ($both\ m\ and\ n\ can\ be\ really\ large$).
  - I still use Bayesian LR.
- Generative model.
  1. Generate $w \sim N(0, \alpha^{-1}I)$; *// w is a column vector*
  2. Generate $\sigma^2 \sim InvGamma(1,1)$;
  3. Given each doc x_i: *// x_i is a column vector*
     generate outcome $y_i \sim N\ (w^T x_i, \sigma^2)$;

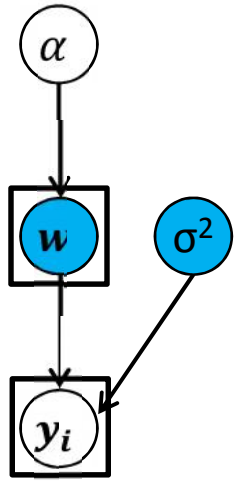
Sample all w_j at once

- $P(\sigma^2|.) \propto \prod_i N(y_i|w^T x_i, \sigma^2) \times InvGamma(\sigma^2|1,1)$

  $\propto InvGamma(\sigma^2|1+\frac{n}{2}, 1+\sum_i \frac{(y_i - w^T x_i)^2}{2})$

- $P(w|.) \propto \prod_i N(y_i|w^T x_i, \sigma^2) \times N(w|0, \alpha^{-1} I)$

  $\propto N\left(w|(\sigma^2 \alpha I + X^T X)^{-1} X^T Y, \left(\alpha I + \frac{1}{\sigma^2} X^T X\right)^{-1}\right)$



where

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_{n-1} \end{bmatrix} \text{ and } Y = \begin{bmatrix} y_0 \\ y_1 \\ \cdots \\ y_{n-1} \end{bmatrix}$$ (X is a matrix, and Y is a vector).

Problems with this approach

$$P(w|.) \propto N\left(w|(\sigma^2\alpha I + X^T X)^{-1}X^T Y, \left(\alpha I + \frac{1}{\sigma^2}X^T X\right)^{-1}\right)$$

- The computational complexity of $X^T X$ is $O(nm^2)$.
- The computational complexity of M^{-1} is $O(m^3)$, and it is hard to parallelize.
- Given $m = 10^4$ and $n = 10^6$, both complexity can be up to $O(10^{14})$.
- If we use double for repressors, the complexity can be 10^{15}.
- Up to 5 hours per iteration with Amazon 100 m2.4xlarge nodes.

One dimension at a time

$$P(w_j|.) \propto \prod_i N(y_i|w_j x_{ij} + \sum_{k!=j} w_k x_{ik}, \sigma^2) \times N(w_j|0, \alpha^{-1})$$

$$\propto N(w_j|\frac{\sum_i (y_i - \sum_{k!=j} w_k x_{ik}) x_{ij}}{\sum_i x_{ij}^2}, \frac{\sigma^2}{\sum_i x_{ij}^2}) N(w_j|0, \alpha^{-1})$$

It is a normal distribution.

- Problems:
  - It is too slow: too many scans, and scans follow each other.
  - For large n and m, it takes too much time.

A Block-based Sampler

- Sample a block of dimensions

$$P(w_B|.) \propto \prod_i N\left(y_i \middle| \sum_{j\notin B} w_j x_{ij} + \sum_{j\in B} w_j x_{ij}, \sigma^2\right) \times \prod_{j\in B} N\left(w_j|0,\alpha^{-1}\right)$$

$$\propto \exp\left\{-\frac{1}{2\sigma^2} F(w_B)\right\} \times \prod_{j\in B} N\left(w_j|0,\alpha^{-1}\right)$$

where

$$F(w_B) = \sum_i \left(y_i - \sum_{j\notin B} w_k x_{ij} - \sum_{j\in B} w_k x_{ij}\right)^2$$

$$\sim \sum_{j\in B}\left(\sum_i x_{ij}^2\right) w_j^2 + \sum_{j<k_2, j\in B, k\in B}\left(\sum_i 2x_{ij}x_{ik}\right) w_j w_k -$$

$$\sum_{j\in B}\left(\sum_i 2\left(y_i - \sum_{k\notin B} w_k x_{ik}\right) x_{ij}\right) w_j + \text{const}$$

A Block-based Sampler

$$P(w_B|.) \propto \exp\left\{-\frac{1}{2\sigma^2}F(w_B)\right\} \times \prod_{j\in B} N(w_j|0,\alpha^{-1})$$

where

$$F(w_B) \sim \sum_{j\in B}\left(\sum_i x_{ij}^2\right)w_j^2 + \sum_{j<k_2, j\in B, k\in B}\left(\sum_i 2x_{ij}x_{ik}\right)w_j w_k -$$
$$\sum_{j\in B}\left(\sum_i 2\left(y_i - \sum_{k\notin B} w_k x_{ik}\right)x_{ij}\right)w_j + \text{const}$$

- Given $size(B) = b, m, n$, for each MCMC iteration
  - The number of scans over data: m/b.
  - The number of aggregates: $\frac{(b+3)m}{2}$.
  - Computation cost: $f(b) = k_1 \times \frac{m}{b} + k_2 \times \frac{(b+3)m}{2}$.

Time Cost

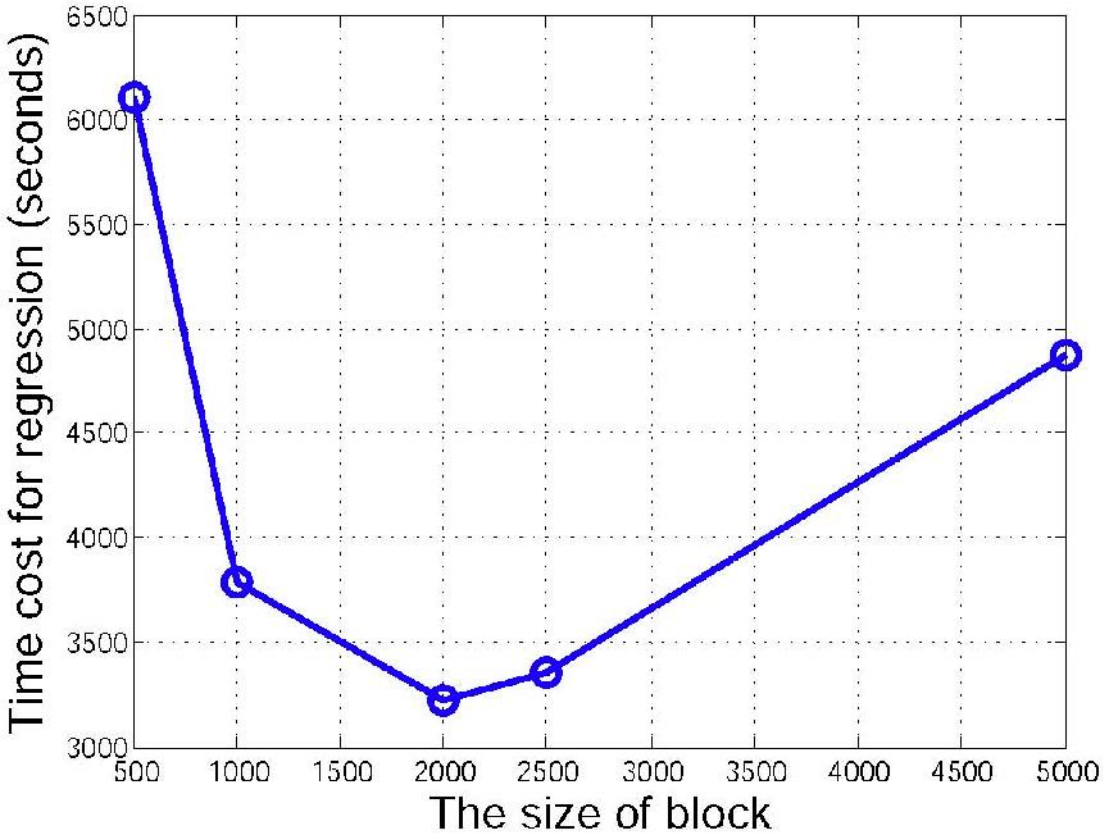


Figure. The time cost per MCMC iteration for linear regression on 20newsgroup dataset, where 10000 distinct words and 2M documents are used.

Problem 4

- Problem. Given the 20-newsgroup dataset with a large fraction of missing values, the task is to recover such values.
- Dataset: http://qwone.com/~jason/20Newsgroups/20news-19997.tar.gz

$$
\begin{array}{ll}
doc_0 & [word_0: count_{00}, word_1: ?, word_2: count_{02}, \ldots, word_m: ?] \\
doc_1 & [word_0: ?, word_1: count_{11}, word_2: ?, \ldots, word_n: count_{1m}] \\
 & \ldots \\
doc_n & [word_0: ?, word_1: count_{n1}, word_2: ?, \ldots, word_n: count_{nm}]
\end{array}
$$

Problem Modeling

- It is modelled as an imputation problem.
  - Really old topic, and there are more than twenty of methods.
  - One of the most widely used methods is Multi-Gaussian distribution.

Challenges

$$\mathrm{N}(\mu, \Sigma)$$

- Σ^{-1} computed from Σ takes $\Theta(m^3)$, m is the size of dimensions.
- Σ or Σ^{-1} from data takes $\Theta(nm^2)$, n is the number of data points.
- Σ should be ***positive-definite***.
- GMRF(Gaussian Markov Random Field) has similar problems.

Solution

- PGRF (Pairwise Gaussian Random Field)
  - Sidestep the covariance.
  - High performance.
  - Algorithm complexity is linear with the scale of data.

Simple Case 1

$$Let\ x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix},\ and\ \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 \end{bmatrix}$$

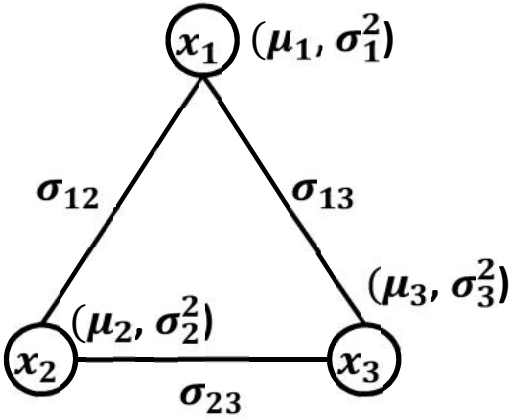


Figure. Three dimensions are strongly correlated.

- Multi-Gaussian model

$$p(x) = \frac{1}{(2\ )^{1.5}|\Sigma|^{0.5}} \exp\{-0.5(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

- PGRF model for complete correlations
  - Let $\psi_{1,2}(x) = N_2\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}\right)$, and similar definition for $\psi_{1,3}(x)$ and $\psi_{2,3}(x)$:

$$p(x) = \frac{\psi_{1,2}(x)\psi_{1,3}(x)\psi_{2,3}(x)}{\iiint \psi_{1,2}(x)\psi_{1,3}(x)\psi_{2,3}(x)dx_1dx_2dx_3}$$

Simple Case 2

- PGRF model with two correlations
  - Let $\psi_{1,2}(x) = N_2\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}\right)$, and

similar definition for $\psi_{1,3}(x)$ and $\psi_{2,3}(x)$:

$$p(x) = \frac{\psi_{1,2}(x)\psi_{2,3}(x)}{\iiint \psi_{1,2}(x)\psi_{2,3}(x)dx_1dx_2dx_3}$$

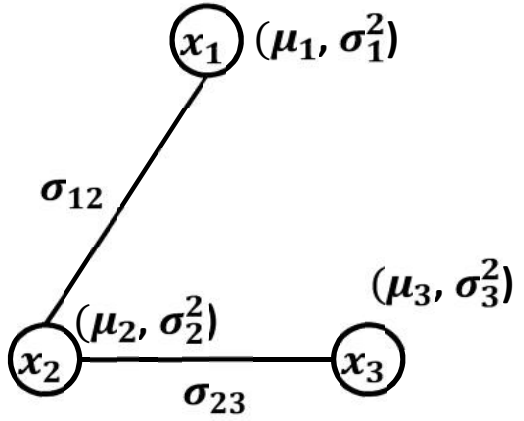


Figure. Two dimensions are correlated.

Simple Case 3

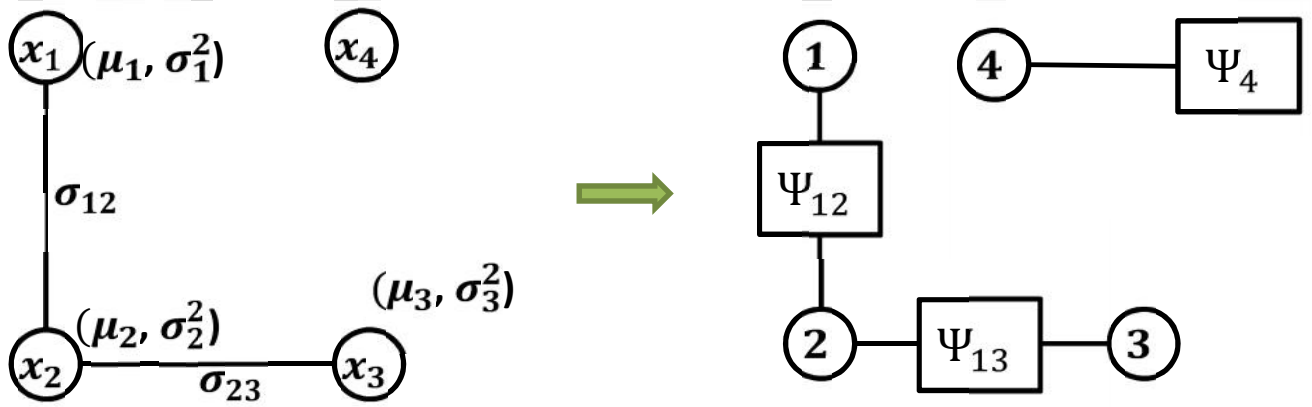


Figure. Four dimensions with two correlations.

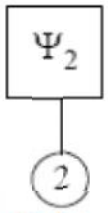
- PGRF model with two correlations
  - Let $\psi_{1,2}(x) = N_2\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}\right)$, and

$\psi_4(x) = N(x_4|\mu_4, \sigma_4^2)$, then:

$$p(x) = \frac{\psi_{1,2}(x)\psi_{2,3}(x)\psi_4(x)}{\iiint \psi_{1,2}(x)\psi_{2,3}(x)\psi_4(x)dx_1dx_2dx_3dx_4}$$

Simple Case 4

- Input
  - Data: x.
  - $\Psi = \{(1,3),(1,5),(3,5)\ (4,5)\}$.



- Model
  - $f_{\Omega}(x) = \frac{1}{Z}\psi_2(x)\psi_{1,3}(x)\psi_{1,5}(x)\psi_{3,5}(x)\psi_{4,5}(x)$
  - $Z = \int \psi_2(x)\psi_{1,3}(x)\psi_{1,5}(x)\psi_{3,5}(x)\psi_{4,5}(x)\ dx_1 \ldots x_5$

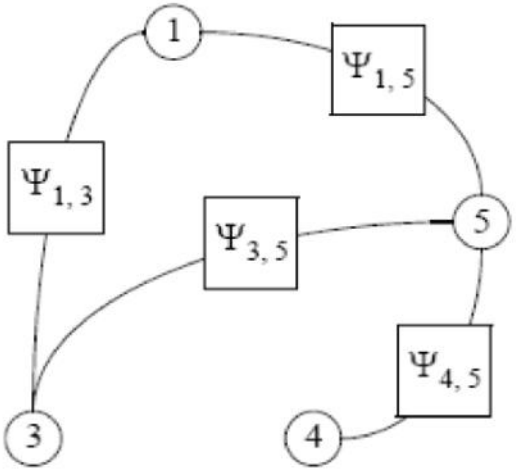


Figure. A PGRF model for 5 dimensional variables.

Generative Model

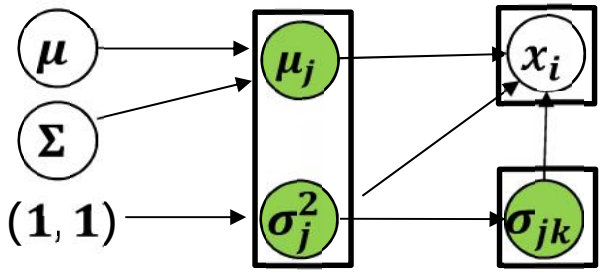


Figure. Graphical model for Markov random field. Ψ is known.

1. $for\ i \in \{1 \dots m\}$:

   $\sigma_j^2 \sim InvGamma(1, 1);$

   $\mu_j \sim Normal(\mu, \Sigma);$

2. $for\ (j,k) \in \Psi:$

   $\sigma_{j,k} \sim Uniform(-\sqrt{\sigma_j^2 \sigma_k^2}, \sqrt{\sigma_j^2 \sigma_k^2});$

3. $for\ i \in \{1 \dots n\}$:

   $f(x_i|\Omega\} = \frac{1}{Z}\left(\prod_{j \square \bar{\Psi}}^{n} \psi_j(x_i)\right)\left(\prod_{(j,k) \square \Psi} \psi_{j,k}(x_i)\right)$

Figure 6. Generative process

Inference of PGRF

variables	parallel	complexity
x'	yes	$\Theta(mn)$
μ_j	An independent subgraph of variables can be parallelized.	$\Theta(mn)$
σ_j^2		$\Theta((m+p)n)$
$\sigma_{j,k}^2$		$\Theta(pn)$

m : the number of dimensions,
n: the number of data points,
p: the number of input correlations.

Sample σ_j^2 and $\sigma_{j,k}^2$

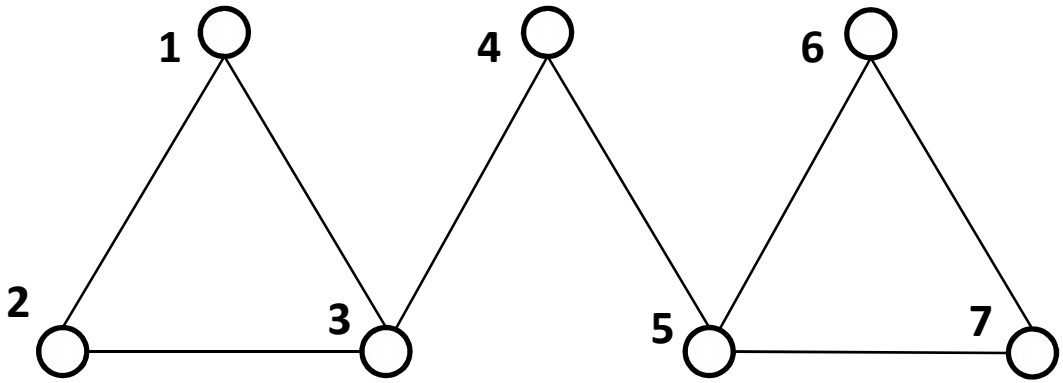


Figure. The correlation graph.

1. Find the maximum independent set (MIS).

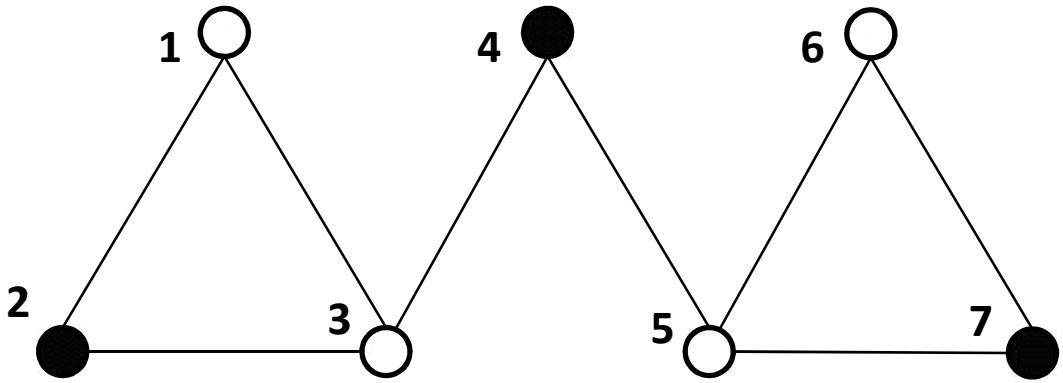


Figure. The correlation graph.

2. Sample σ_j^2 and $\sigma_{j,k}^2$ for selected vertices.

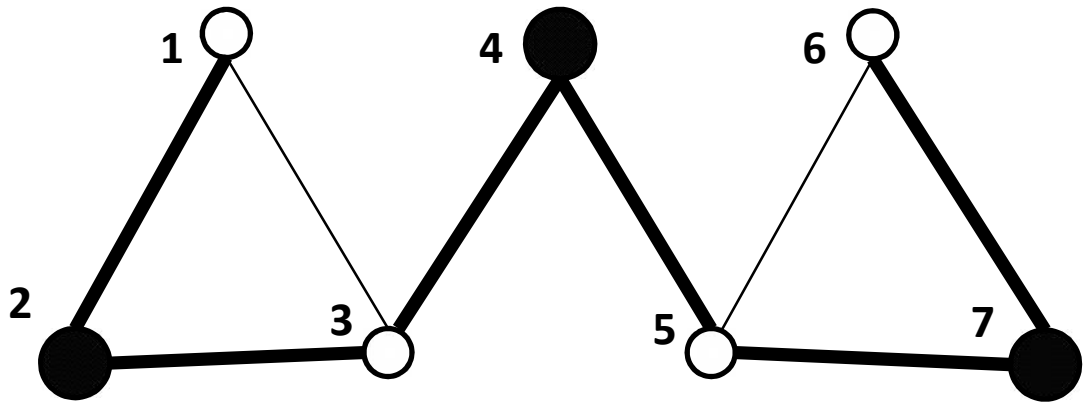


Figure. The correlation graph.

3. Find the MIS in the remaining graph.

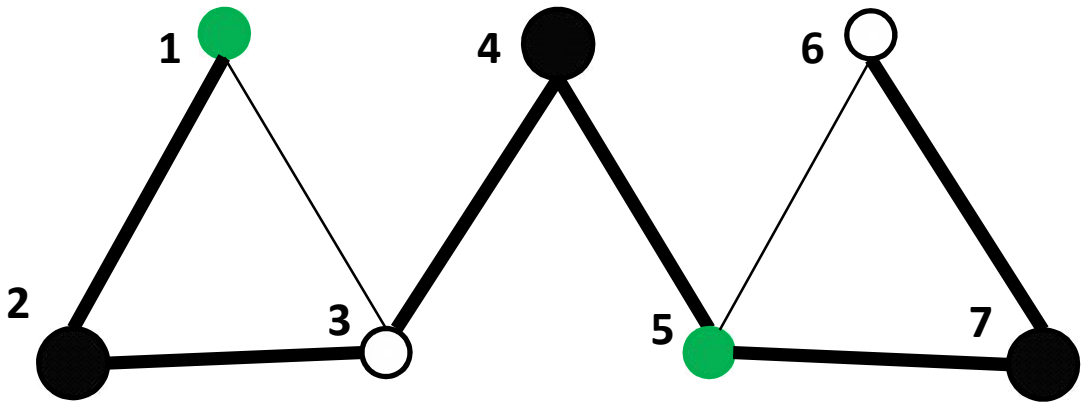


Figure. The correlation graph.

4. Sample σ_j^2 and $\sigma_{j,k}^2$ for selected vertices.

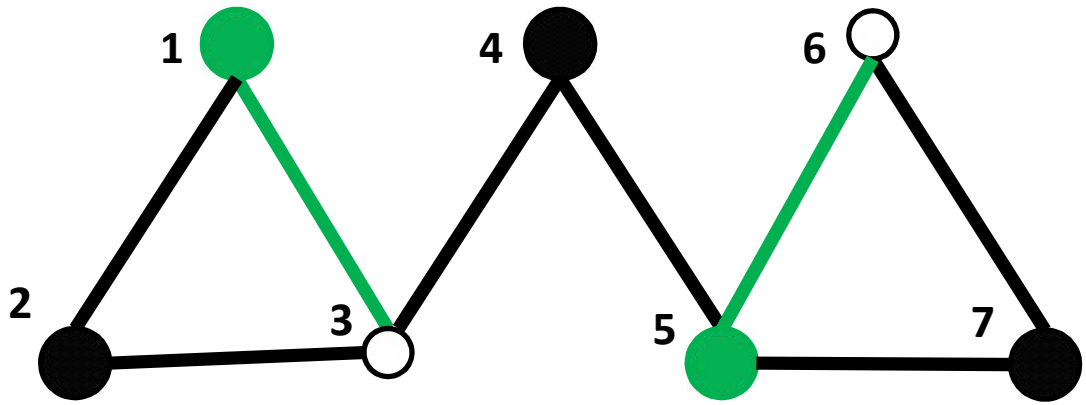


Figure. The correlation graph.

5. Sample σ_j^2 and $\sigma_{j,k}^2$ for remaining vertices.

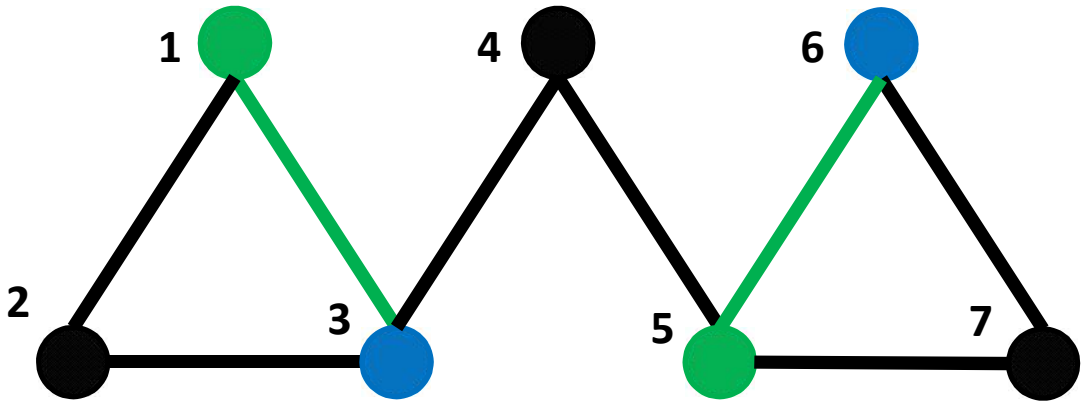
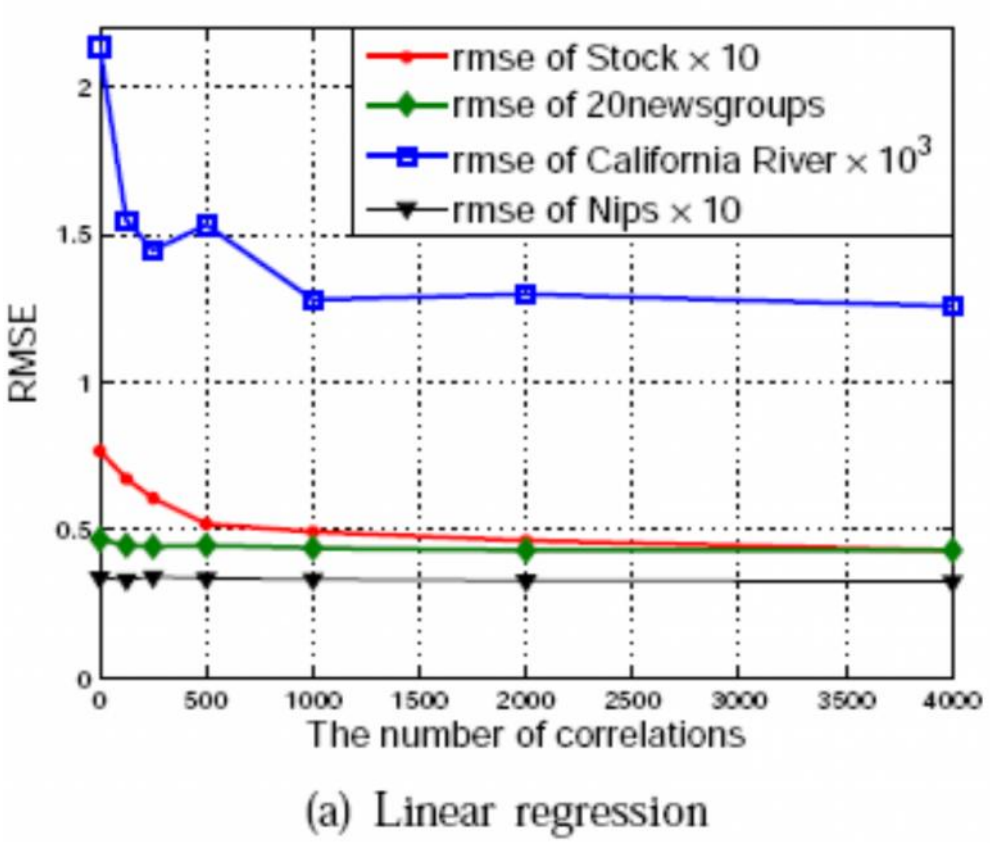


Figure. The correlation graph.

Evaluation

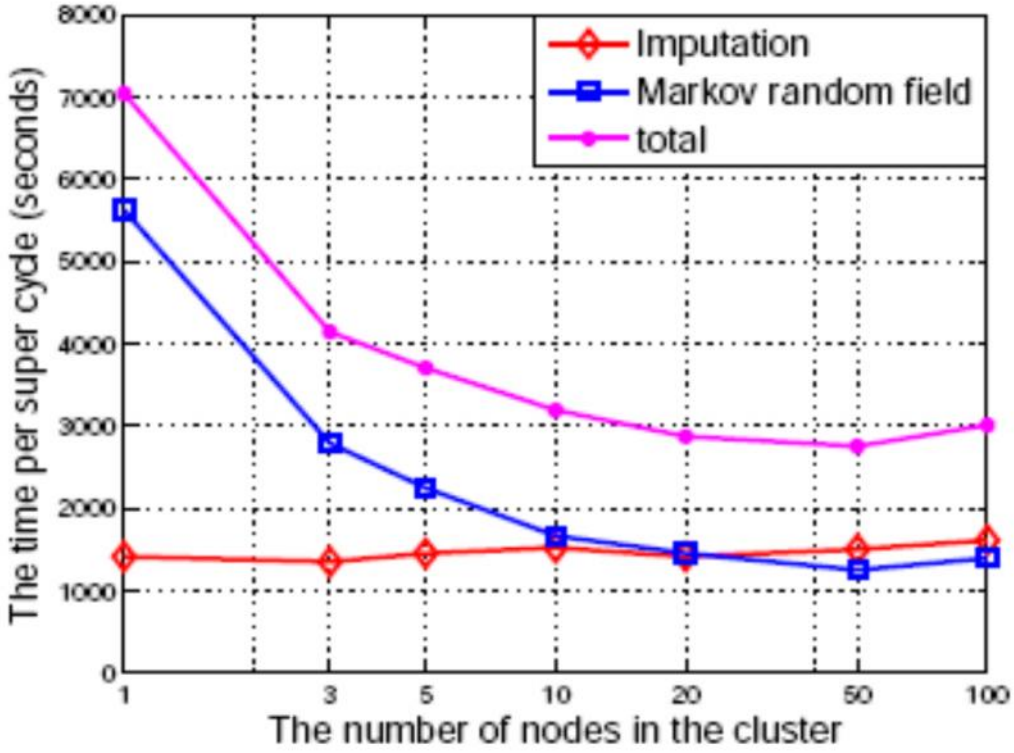
- Impact of the number of correlations.
  - Linear regression with PGRF.



(a) Linear regression

Evaluation

- **Scalability** of our approach (PGRF).
  - Dataset: 20newsgroups, Dimensions: 10000
  - Each machine has a simulated copy of dataset.



Conclusion

- In “Big Data”, computation efficiency should be considered in both modelling and inference.