

STSCI 5020: Spring 2009

Statistical Computing

Feb. 9 – Feb. 27 (three weeks)

Instructor: Ping Li

Department of Statistical Science

Cornell University

Syllabus

Schedule: MW 9:05 – 9:55am, Friday 9:05 – 9:55am (lab)

Instructor: Ping Li, Department of Statistical Science, pingli@cornell.edu

Office Hours: Friday 9:55-10:55am

Homework: Two assignments, some programming (in R or Matlab) is required

Course Description Maximum likelihood estimation (MLE), numerical methods for MLE, Convex Optimization, Fisher Information and Variance of MLE, Monte Carlo methods, random number generation, modern massive data sets (MMDS), some recent techniques for MMDS

Maximum Likelihood Estimation (MLE)

Observations $x_i, i = 1$ to n , are i.i.d. samples from a distribution with probability density function $f_X(x; \theta_1, \theta_2, \dots, \theta_k)$, where $\theta_j, j = 1$ to k , are parameters to be estimated.

The maximum likelihood estimator seeks the θ to maximize the joint likelihood

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_{i=1}^n f_X(x_i; \theta)$$

Or, equivalently, to maximize the **log** joint likelihood

$$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{i=1}^n \log f_X(x_i; \theta)$$

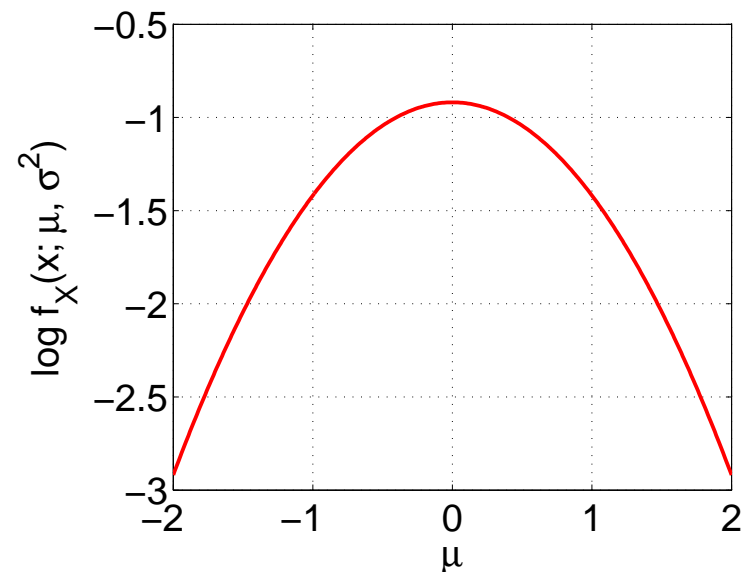
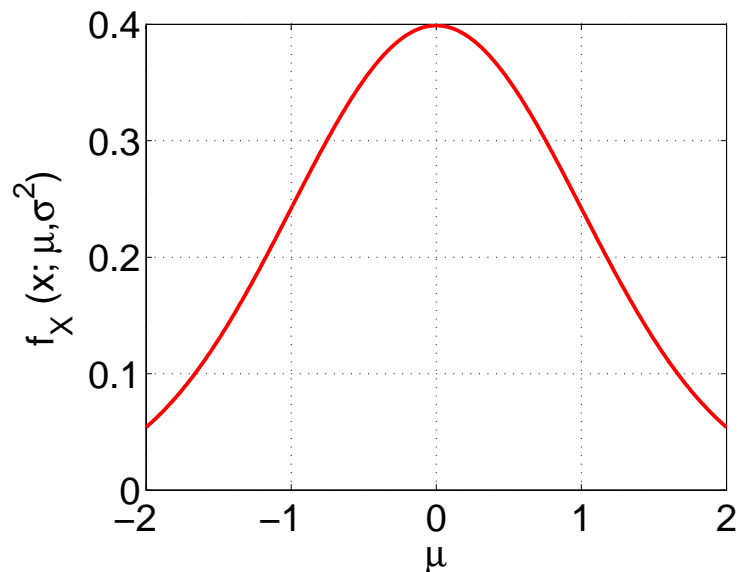
This is a **convex** optimization if f_X is **concave** or **-log-convex**.

An Example: Normal Distribution

If $X \sim N(\mu, \sigma^2)$, then $f_X(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Fix $\sigma^2 = 1, x = 0. f_X(x; \mu, \sigma^2)$

$\log f_X(x; \mu, \sigma^2)$



It is Not concave, but it is a -log convex, i.e., a unique MLE solution exists.

An Example of Exact MLE Solution

Given n i.i.d. samples, $x_i \sim N(\mu, \sigma^2)$, $i = 1$ to n .

$$\begin{aligned} l(x_1, x_2, \dots, x_n; \mu, \sigma^2) &= \sum_{i=1}^n \log f_X(x_i; \mu, \sigma^2) \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 - \frac{1}{2} n \log(2\pi\sigma^2) \end{aligned}$$

$$\frac{\partial l}{\partial \mu} = \frac{1}{2\sigma^2} 2 \sum_{i=1}^n (x_i - \mu) = 0 \implies \hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\frac{\partial l}{\partial \sigma^2} = \frac{1}{2\sigma^4} \sum_{i=1}^n (x_i - \mu)^2 - \frac{n}{2\sigma^2} = 0 \implies \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2.$$

Another Example of Exact MLE Solution

A two-by-two contingency table with **multinomial sampling**.

Observations: $(n_{11}, n_{12}, n_{21}, n_{22})$, $n = n_{11} + n_{12} + n_{21} + n_{22}$.

Parameters $(\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22})$, $(\pi_{11} + \pi_{12} + \pi_{21} + \pi_{22} = 1)$

n_{11}	n_{12}
n_{21}	n_{22}

π_{11}	π_{12}
π_{21}	π_{22}

The likelihood

$$\frac{n!}{n_{11}!n_{12}!n_{21}!n_{22}!} \pi_{11}^{n_{11}} \pi_{12}^{n_{12}} \pi_{21}^{n_{21}} \pi_{22}^{n_{22}}$$

The log likelihood

$$l = \log \frac{n!}{n_{11}!n_{12}!n_{21}!n_{22}!} \quad (\text{which is not important, why?})$$
$$+ n_{11} \log \pi_{11} + n_{12} \log \pi_{12} + n_{21} \log \pi_{21} + n_{22} \log \pi_{22}$$

$$\frac{\partial l}{\partial \pi_{11}} = \frac{n_{11}}{\pi_{11}} + \frac{-n_{22}}{1 - \pi_{11} - \pi_{12} - \pi_{21}} = 0, \implies \frac{n_{11}}{\pi_{11}} = \frac{n_{22}}{\pi_{22}}.$$

Similarly $\frac{n_{11}}{\pi_{11}} = \frac{n_{12}}{\pi_{12}} = \frac{n_{21}}{\pi_{21}} = \frac{n_{22}}{\pi_{22}} = t$. Thus, the MLE solution is

$$\hat{\pi}_{11} = \frac{n_{11}}{n}, \quad \hat{\pi}_{12} = \frac{n_{12}}{n}, \quad \hat{\pi}_{21} = \frac{n_{21}}{n}, \quad \hat{\pi}_{22} = \frac{n_{22}}{n},$$

Contingency Table with Margin Constraints

Total samples : $n = n_{11} + n_{12} + n_{21} + n_{22}$

Total original counts : $N = N_{11} + N_{12} + N_{21} + N_{22}$, i.e., $\pi_{ij} = N_{ij}/N$.

Sample Contingency Table

n_{11}	n_{12}
n_{21}	n_{22}

Original Contingency Table

N_{11}	N_{12}
N_{21}	N_{22}

Margins: $M_1 = N_{11} + N_{12}$, $M_2 = N_{11} + N_{21}$.

Margins are much easier to be counted exactly than interactions.

If margins M_1 and M_2 are known, then only need to estimate N_{11} .

The MLE equation is

$$\frac{n_{11}}{N_{11}} - \frac{n_{12}}{M_1 - N_{11}} - \frac{n_{21}}{M_2 - N_{11}} + \frac{n_{22}}{N - M_1 - M_2 + N_{11}} = 0.$$

This is a cubic equation, which has a closed-form solution.

Alternatively, one can solve the equation by numerical iterative methods.

Possible Homework Questions:

- (1) Show the $-\log$ likelihood equation is a convex function of N_{11} .
- (2) Derive the MLE equation.
- (3) Implement a numerical procedure to solve the MLE equation.
- (4) ...

...

An Example of Contingency Tables with Fixed Margins

Term-by-Document matrix $n = 10^6$ words and $m = 10^{10}$ (Web) documents.

Cell $x_{ij} = 1$ if word i appears in document j . $x_{ij} = 0$ otherwise.

	Doc 1	Doc 2					Doc m	
Word 1	1	0	0	1	0	0	0	1
Word 2	0	1	0	1	0	0	1	0
Word 3								
Word 4								
Word n								

	Word 2	No Word 2
Word 1	N_{11}	N_{12}
No Word 1	N_{21}	N_{22}

N_{11} : number of documents containing both word 1 and word 2.

N_{22} : number of documents containing neither word 1 nor word 2.

Margins ($M_1 = N_{11} + N_{12}$, $M_2 = N_{11} + N_{21}$) for all rows costs nm , **easy!**

Interactions (N_{11} , N_{12} , N_{21} , N_{22}) for all pairs costs $n(n-1)m/2$, **difficult!**

A Real Application: Google Pagehits

Google tells user the number of Web pages containing the input query word(s).

“Cornell” : 32,700,000 pages

“University”: 903,000,000 pages

“Cornell University” : 9,610,000 pages

(numbers can change).

It is realistic to believe that the counts for individual words are exact,
but the numbers of co-occurrences may be estimated, eg, from some samples.

(This problem is actually more difficult than it appears, because the number of words and documents are really large, and the co-occurrences are relatively small, e.g., highly sparse)

Bivariate Normal: An (Almost) Convex Example of MLE

Two correlated random variables x_i and y_i are bivariate normal

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \sim N \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix} \right).$$

Σ is the variance-covariance matrix.

Given n i.i.d. samples $\begin{bmatrix} x_i \\ y_i \end{bmatrix}$, $i = 1$ to n . The joint likelihood function

$$lik = (2\pi)^{-n} |\Sigma|^{-\frac{n}{2}} \exp \left(-\frac{1}{2} \sum_{i=1}^n \begin{bmatrix} x_i & y_i \end{bmatrix} \Sigma^{-1} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right).$$

where (assuming $1 > a^2$)

$$|\Sigma| = (1 - a^2), \quad \Sigma^{-1} = \frac{1}{1 - a^2} \begin{bmatrix} 1 & -a \\ -a & 1 \end{bmatrix},$$

The log likelihood function $l(a)$ is

$$l(a) = -\frac{n}{2} \log(1 - a^2) - \frac{1}{2} \frac{1}{1 - a^2} \sum_{i=1}^n (x_i^2 + y_i^2 - 2x_i y_i a).$$

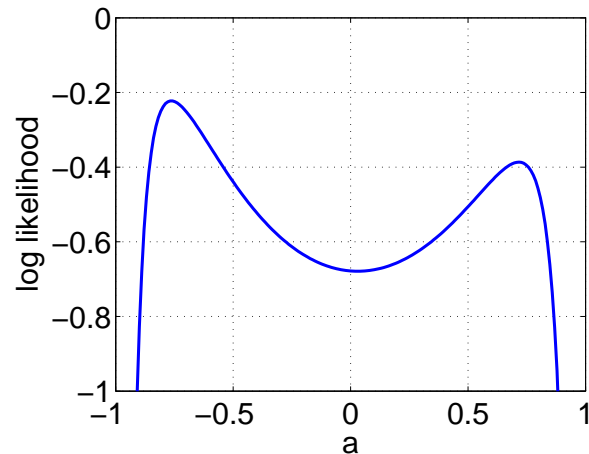
Setting $l'(a) = 0$ to zero yields a **cubic** MLE equation

$$\begin{aligned} l'(a) &= 0 \\ &= -na^3 + a^2 \sum_{i=1}^n x_i y_i - a \left(-n + \sum_{i=1}^n x_i^2 + \sum_{i=1}^n y_i^2 \right) + \sum_{i=1}^n x_i y_i \end{aligned}$$

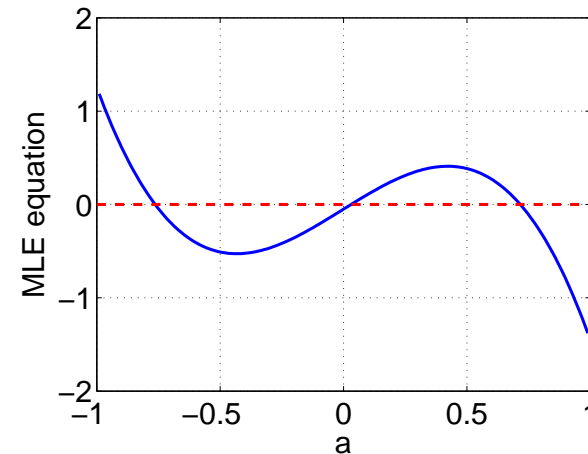
This MLE problem is **almost always** convex, especially when n is large.

Simulations: $n = 3$, $a = 0.55$. After about 20 trials,

log Likelihood function $l(a)$



MLE equation $l'(a) = 0$



When n is small, the MLE equation may admit 3 solutions.

When n is large, this occurs extremely unlikely.

Possible Homework Questions:

(1) Derive $l(a)$ and $l'(a)$.

(2) Implement a numerical procedure for the MLE solution

...

Numerical Techniques & Optimization

- Many statistical problems require numerical solutions.

Maximum Likelihood estimation (MLE), Logistic Regression, Neural Networks, Support Vector Machines (SVM), ...

- Topics

Convex Optimization, Steepest Descent, Newton's Method

Local search, Iterative proportional scaling

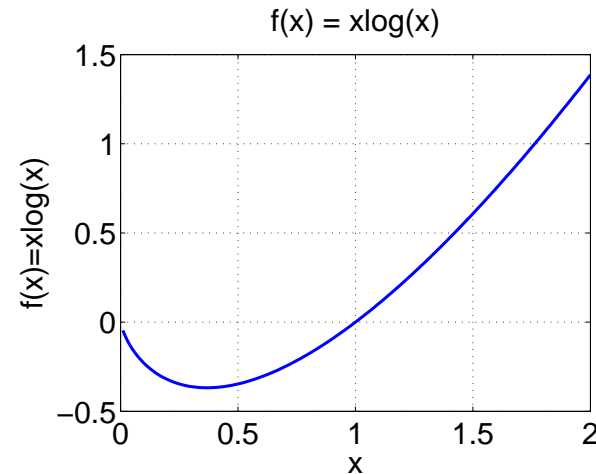
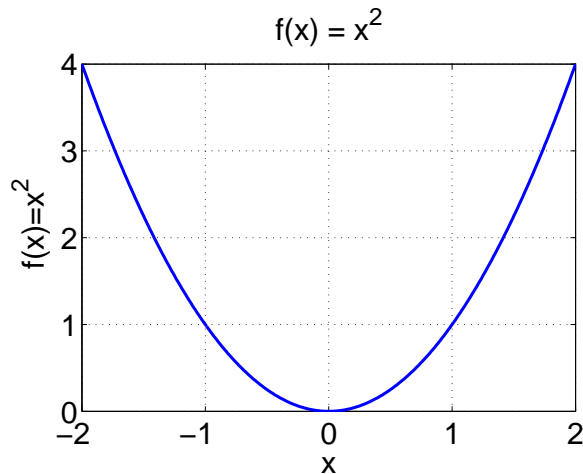
- Possible Homework Ideas

Implement steepest descent and Newton's method for MLE (1-dim only).

Implement iterative proportional scaling

Convex Functions

A function $f(x)$ is convex if the second derivative $f''(x) \geq 0$.



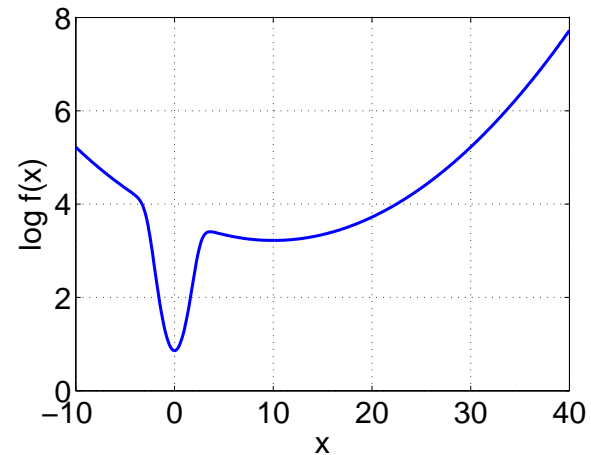
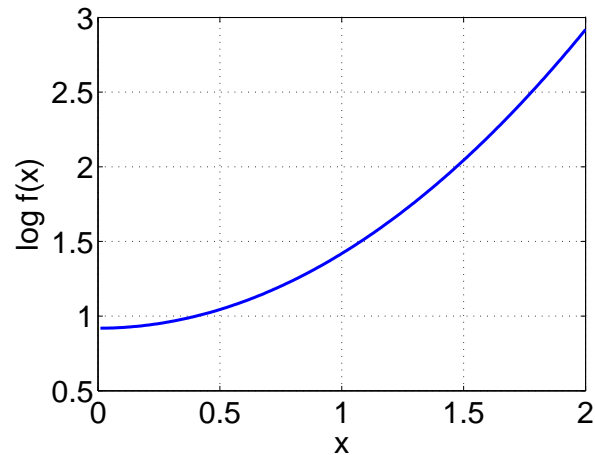
$f(x) = x^2 \implies f'' = 2 > 0$, i.e., $f(x) = x^2$ is convex for all x .

$f(x) = x \log x \implies f'' = \frac{1}{x}$, i.e., $f(x) = x \log x$ is convex if $x > 0$.

Both are widely used in statistics and data mining as loss functions,

\implies computationally tractable algorithms: least square, logistic regression.

Left panel: $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ is -log convex, $\frac{\partial^2[-\log f(x)]}{\partial x^2} = 1 > 0$.



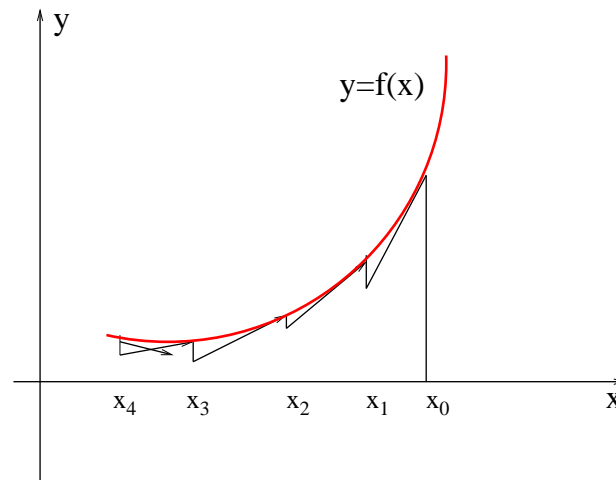
Right panel: a mixture of normals is not -log convex

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} + \frac{1}{\sqrt{2\pi}10} e^{-\frac{(x-10)^2}{200}}$$

The mixture of normals is an extremely useful model in statistics.

In general, only a local minimum can be obtained.

Steepest Descent



Procedure:

Start with an initial guess x_0 .

Compute $x_1 = x_0 - \Delta f'(x_0)$, where Δ is the step size.

Continue the process $x_{t+1} = x_t - \Delta f'(x_t)$.

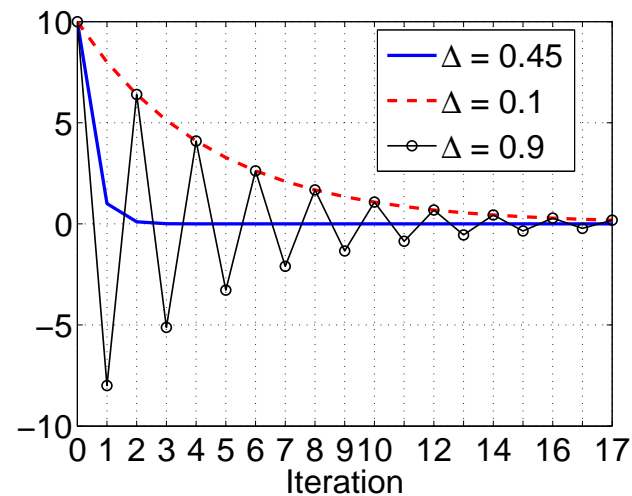
Until some criterion is met, e.g., $f(x_{t+1}) \approx f(x_t)$

The meaning of “**steepest**” is more clear in the two-dimensional situation.

An Example of Steepest Descent: $f(x) = x^2$

$f(x) = x^2$. The minimum is attained at $x = 0$, $f'(x) = 2x$.

The steepest descent iteration formula $x_{t+1} = x_t - \Delta f'(x_t) = x_t - 2\Delta x_t$.



Choosing the step size Δ is important (even when $f(x)$ is convex).

Too small $\Delta \implies$ slow convergence, i.e., many iterations,

Too large $\Delta \implies$ oscillations, i.e., also many iterations.

Steepest Descent in Practice

Steepest descent is one of the most widely techniques in real world

- It is extremely simple
- It only requires knowing the first derivative
- It is numerically stable (for above reasons)
- For real applications, it is often affordable to use very small Δ
- In machine learning, Δ is often called **learning rate**
- It is used in Neural Nets and Gradient Boosting (MART)

Newton's Method

Recall the goal is to find the x^* to minimize $f(x)$.

If $f(x)$ is convex, it is equivalent to finding the x^* such that $f'(x^*) = 0$.

Let $h(x) = f'(x)$. Take Taylor expansion about the optimum solution x^* :

$$h(x^*) = h(x) + (x^* - x)h'(x) + \text{“negligible” higher order terms}$$

Because $h(x^*) = f'(x^*) = 0$, we know approximately

$$0 \approx h(x) + (x^* - x)h'(x) \implies x^* \approx x - \frac{h(x)}{h'(x)}$$

The procedure of Newton's Method

Start with an initial guess x_0

$$\text{Update } x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

$$\text{Repeat } x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)}$$

Until some stopping criterion is reached, e.g., $x_{t+1} \approx x_t$.

An example: $f(x) = (x - c)^2$. $f'(x) = 2(x - c)$, $f''(x) = 2$.

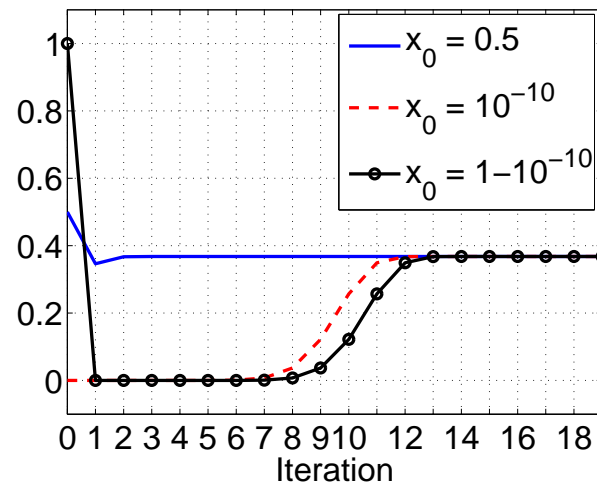
$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} \implies x_1 = x_0 - \frac{2(x_0 - c)}{2} = c$$

But we already know that $x = c$ minimizes $f(x) = (x - c)^2$.

Newton's method may find the minimum solution using only one step.

An Example of Newton's Method: $f(x) = x \log x$

$$f'(x) = \log x + 1, \quad f''(x) = \frac{1}{x}. \quad x_{t+1} = x_t - \frac{\log x_t + 1}{1/x_t}$$



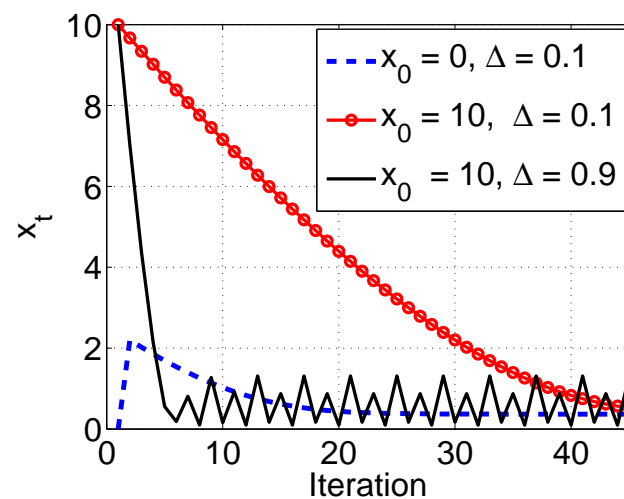
When x_0 is close to optimum solution, the convergence is very fast

When x_0 is far from the optimum, the convergence is slow initially

When x_0 is badly chosen, no convergence. This example requires $0 < x_0 < 1$.

Steepest Descent for $f(x) = x \log x$

$$f'(x) = \log x + 1, \quad x_{t+1} = x_t - \Delta(\log x_t + 1)$$



Regardless of x_0 , convergence is guaranteed if $f(x)$ is convex.

May be oscillating if step size Δ is too large

Convergence is slow near the optimum solution.

Newton's Method in Practice

Newton's Method is not as widely used as methods based gradient descent.

- It requires the second derivative of a model function, which is only a crude approximation to the (unobservable) truth. The first derivative may be ok, but the second derivative can be too noisy.
- Consequently, algorithms based on Newton's method can be unstable.
- The convergence can be slow if the solution is far from the optimum.
- It can be combined with other methods.

For example, start with gradient descent, change to Newton's method when we expect the optimum solution is close.

Convex Optimization in Multi-Dimensions

- Steepest descent and other gradient descent methods are probably the most popular in practice.
- A general optimization solver usually can only handle $< 10^4$ variables, except for linear programming, which may handle (a lot) more.
- Certain application areas may develop their own, specially tailored, optimization procedure. For example, the SMO algorithm and SVM^{light} algorithm for support vector machines (SVM).

Logistic Regression

The logistic regression model (with intercept)

n observations: $\{x_i, y_i\}, i = 1$ to n . $y_i \in \{0, 1\}$, ie two classes .

$p_0(x_i) = \mathbf{Pr}(\text{Class} = 0|x_i)$, $p_1(x_i) = \mathbf{Pr}(\text{Class} = 1|x_i) = 1 - p_0(x_i)$,

Assume

$$\log \frac{p_0(x_i)}{p_1(x_i)} = \log \frac{p_0(x_i)}{1 - p_0(x_i)} = \beta_0 + \beta_1 x_i$$

Equivalently,

$$p_0(x_i) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}, \quad p_1(x_i) = 1 - p_0(x_i) = \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}},$$

Log likelihood for the i th observation: Let $\beta = (\beta_0, \beta_1)$

$l_i(\beta|x_i) = \log p_j(x_i)$ if $y_i = j \in \{0, 1\}$. Equivalently,

$$l_i(\beta|x_i) = (1 - y_i) \log p_0(x_i) + y_i \log p_1(x_i)$$

Joint log likelihood for n observations:

$$\begin{aligned} l(\beta|x_1, \dots, x_n) &= \sum_{i=1}^n l_i(\beta|x_i) \\ &= \sum_{i=1}^n (1 - y_i) \log p_0(x_i) + y_i \log p_1(x_i) \\ &= \sum_{i=1}^n y_i (\beta_0 + \beta_1 x_i) - \log (1 + e^{\beta_0 + \beta_1 x_i}) \end{aligned}$$

First derivatives

$$\frac{\partial l(\beta)}{\partial \beta_0} = \sum_{i=1}^n y_i - p_0(x_i), \quad \frac{\partial l(\beta)}{\partial \beta_1} = \sum_{i=1}^n x_i (y_i - p_0(x_i)),$$

Second derivatives

$$\frac{\partial^2 l(\beta)}{\partial \beta_0^2} = - \sum_{i=1}^n p_0(x_i) (1 - p_0(x_i)),$$

$$\frac{\partial^2 l(\beta)}{\partial \beta_1^2} = - \sum_{i=1}^n x_i^2 p_0(x_i) (1 - p_0(x_i)),$$

$$\frac{\partial^2 l(\beta)}{\partial \beta_0 \beta_1} = - \sum_{i=1}^n x_i p_0(x_i) (1 - p_0(x_i))$$

Solve the MLE by Newton's Method or steepest descent (two-dim problem).

Logistic Regression without Intercept ($\beta_0 = 0$)

The simplified model

$$\log \frac{p_0(x_i)}{p_1(x_i)} = \log \frac{p_0(x_i)}{1 - p_0(x_i)} = \beta x_i$$

Equivalently,

$$p_0(x_i) = \frac{e^{\beta x_i}}{1 + e^{\beta x_i}}, \quad p_1(x_i) = 1 - p_0(x_i) = \frac{1}{1 + e^{\beta x_i}},$$

Joint log likelihood for n observations:

$$l(\beta|x_1, \dots, x_n) = \sum_{i=1}^n x_i y_i \beta - \log (1 + e^{\beta x_i})$$

First derivative

$$l'(\beta) = \sum_{i=1}^n x_i (y_i - p_0(x_i)),$$

Second derivative

$$l''(\beta) = - \sum_{i=1}^n x_i^2 p_0(x_i) (1 - p_0(x_i)),$$

Newton's Method updating formula

$$\beta_{t+1} = \beta_t - \frac{l'(\beta_t)}{l''(\beta_t)}$$

Steepest descent (in fact ascent) updating formula

$$\beta_{t+1} = \beta_t + \Delta l'(\beta_t)$$

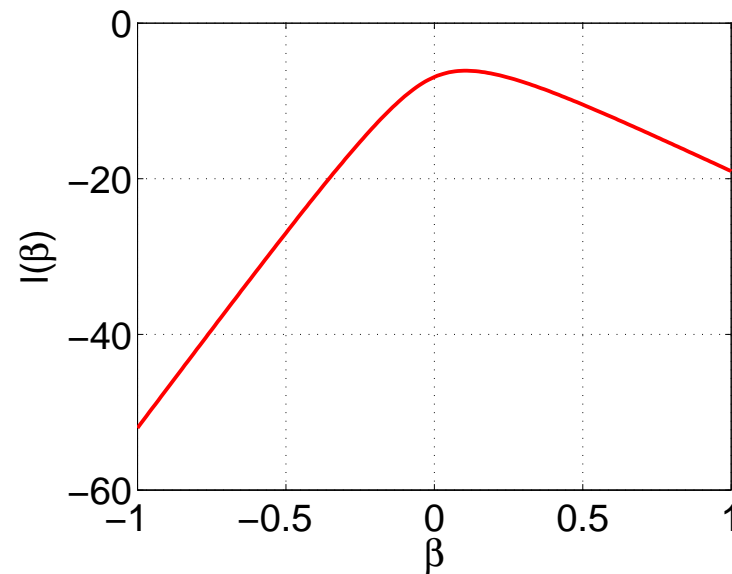
A Numerical Example of Logistic Regression

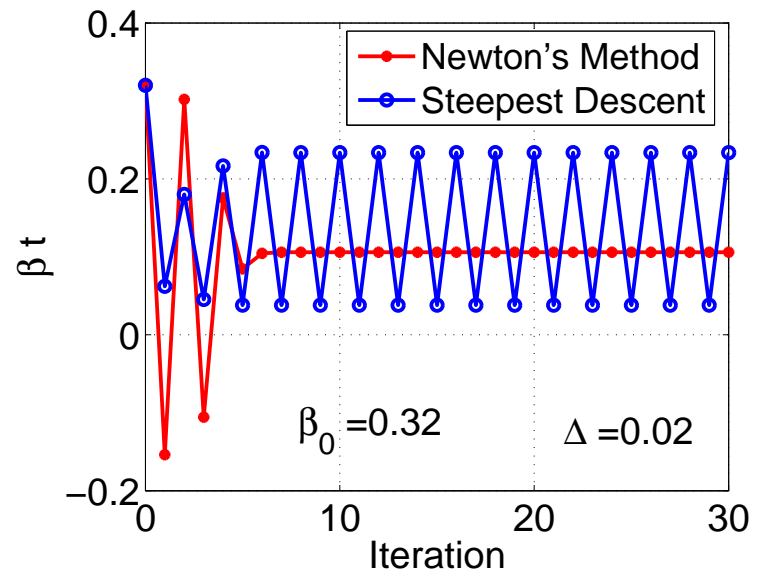
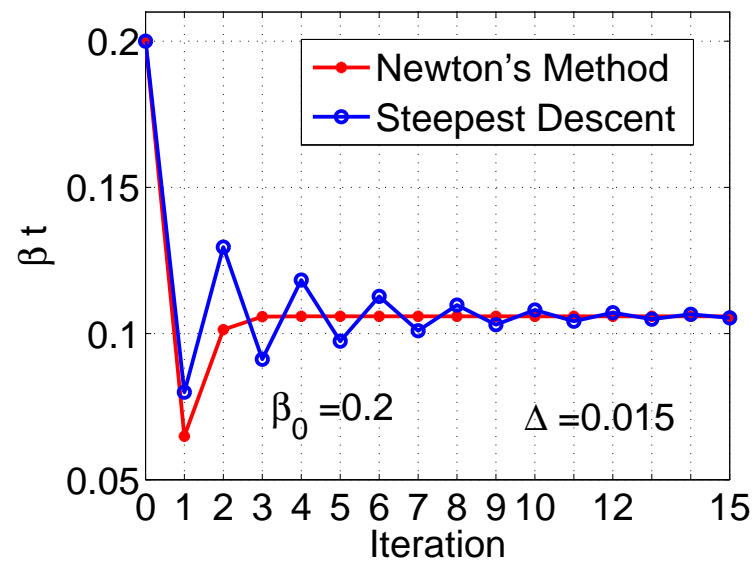
Data

$$x = \{8, 14, -7, 6, 5, 6, -5, 1, 0, -17\}$$

$$y = \{1, 1, 0, 0, 1, 0, 1, 0, 0, 0\}$$

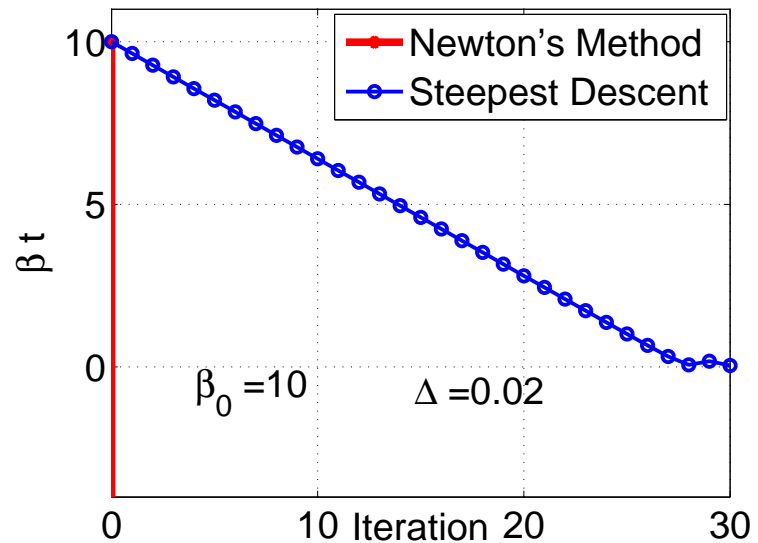
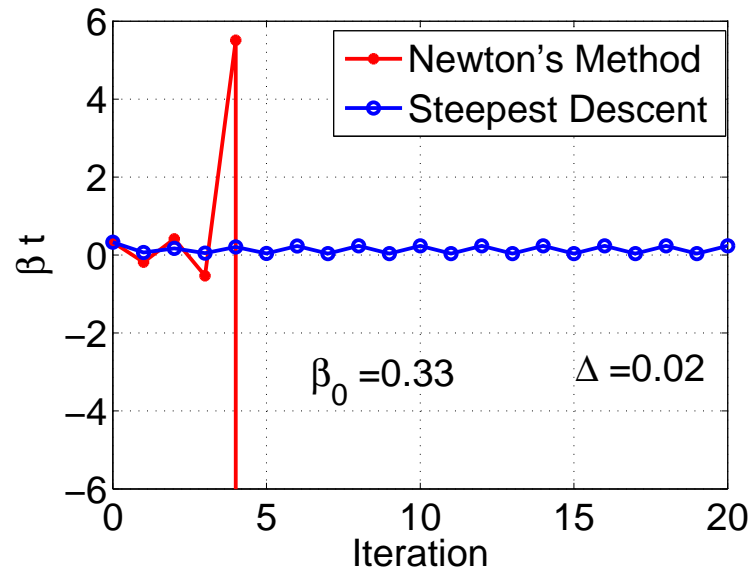
Log likelihood function





Steepest descent is quite sensitive to the step size Δ .

Too large Δ leads to oscillation.



Newton's Method is sensitive to the starting point β_0 . May not converge at all.

The starting point (mostly) only affects computing time of steepest descent.

Homework Idea: Implement both methods for one-parameter logistic regression.

General Comments on Numerical Optimization

Numerical Optimization is **tricky!**, even for convex problems.

Multivariate optimization is much **trickier!**

Whenever possible, try to avoid intensive numerical optimization, even maybe at the cost of losing some accuracy.

Two examples

- One-Step Newton Update
- Iterative Proportional Scaling

MLE Using Newtons' Method for Estimating Gamma Parameters

$X_i \sim \text{Gamma}(\alpha, \lambda)$, i.i.d. $i = 1$ to n .

The log likelihood function

$$l(\alpha, \lambda) = \sum_{i=1}^n -\log \Gamma(\alpha) + \alpha \log \lambda + (\alpha - 1) \log X_i - \lambda X_i$$

First derivatives

$$\frac{\partial l(\alpha, \lambda)}{\partial \alpha} = -n \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + n \log \lambda + \sum_{i=1}^n \log X_i$$

$$\frac{\partial l(\alpha, \lambda)}{\partial \lambda} = n \frac{\alpha}{\lambda} - \sum_{i=1}^n X_i$$

Second derivatives

$$\frac{\partial^2 l(\alpha, \lambda)}{\partial \alpha^2} = -n\psi'(\alpha), \quad \psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$$

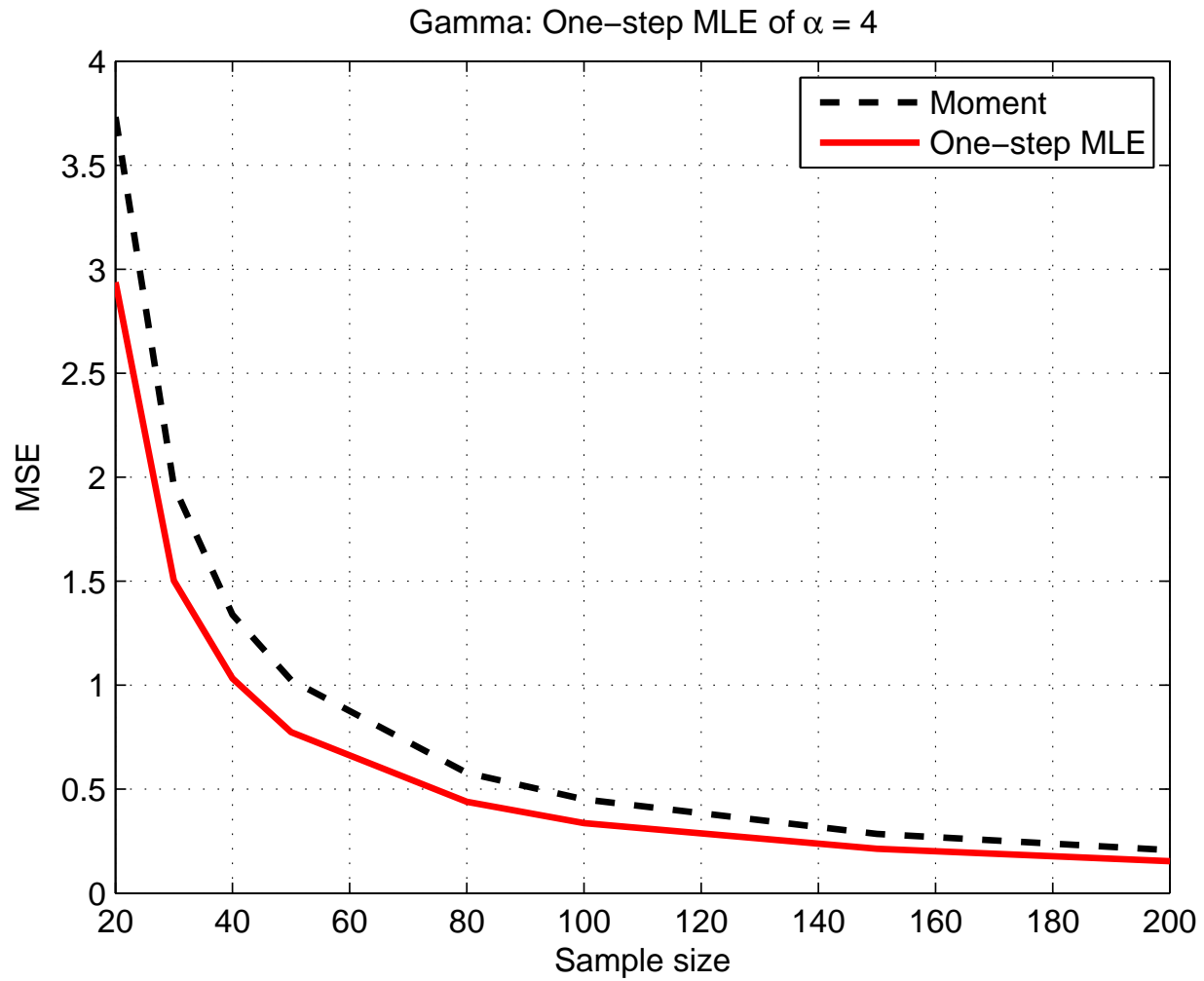
$$\frac{\partial^2 l(\alpha, \lambda)}{\partial \lambda^2} = -n \frac{\alpha}{\lambda^2}$$

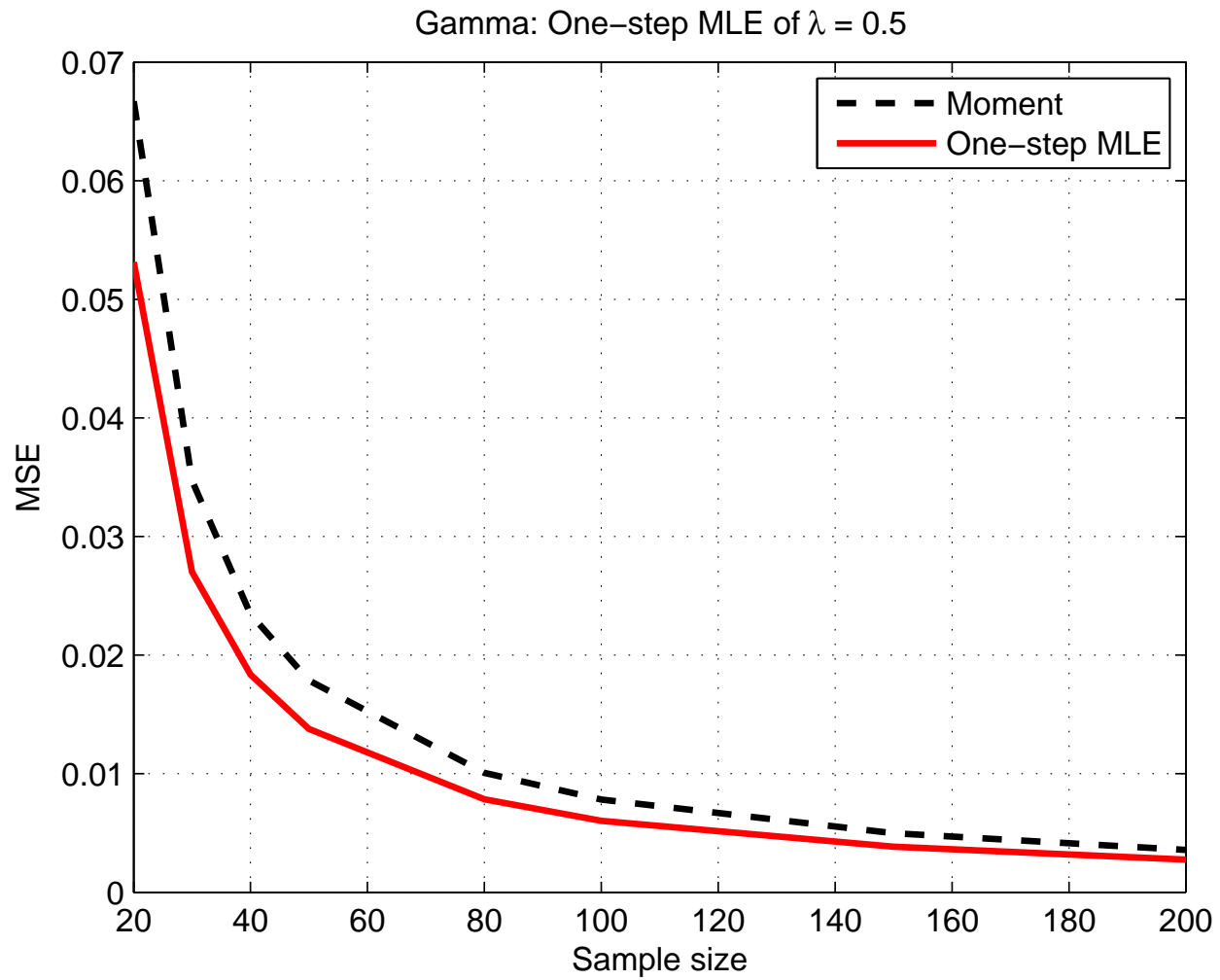
$$\frac{\partial^2 l(\alpha, \lambda)}{\partial \lambda \alpha} = n \frac{1}{\lambda}$$

We can use Newton's method (two dimensions), starting with moment estimators.

The problem is actually more complicated because we have a constrained optimization problem. The constraints: $\alpha \geq 0$ and $\lambda \geq 0$ may not be satisfied during iterations, especially when sample size n is not large.

One the other hand, **One-Step Newton's method** usually works well, starting with an (already pretty good) estimator. Often more iterations do not help much.





Iterative Proportional Scaling (IPS)

Estimating contingency tables with known (original) margins.

Sample 2 by 2 Table

n_{11}	n_{12}
n_{21}	n_{22}

Original 2 by 2 Table

N_{11}	N_{12}
N_{21}	N_{22}

The margins $M_1 = N_{11} + N_{12}$ and $M_2 = N_{11} + N_{21}$ are known.

The MLE equation is cubic, easy!

What about 3 by 3 tables? Multi-dimensional convex optimization!

Iterative Proportional Scaling (IPS) provides a simple alternative.

An Example of IPS for 2 by 2 Tables

n_{11}	n_{12}
n_{21}	n_{22}

The steps of IPS

- (1) Modify the counts to satisfy the **row** margins.
- (2) Modify the counts to satisfy the **column** margins.
- (3) Iterate until some stopping criterion is met.

An example: $n_{11} = 30$, $n_{12} = 5$, $n_{21} = 10$, $n_{22} = 10$, $D = 600$.

$$M_1 = N_{11} + N_{12} = 400,$$

$$M_2 = N_{11} + N_{21} = 300.$$

Iteration 1

342.8571 57.1429

100.0000 100.0000

232.2581 109.0909

67.7419 190.9091

Iteration 2

272.1649 127.8351

52.3810 147.6190

251.5807 139.2265

48.4193 160.7735

Iteration 3

257.4985 142.5015

46.2916 153.7084

254.2860 144.3248

45.7140 155.6752

Iteration 4

255.1722 144.8278

45.3987 154.6013

254.6875 145.1039

45.3125 154.8961

Iteration 5

254.8204 145.1796

45.2653 154.7347

254.7477 145.2211

45.2523 154.7789

Iteration 6

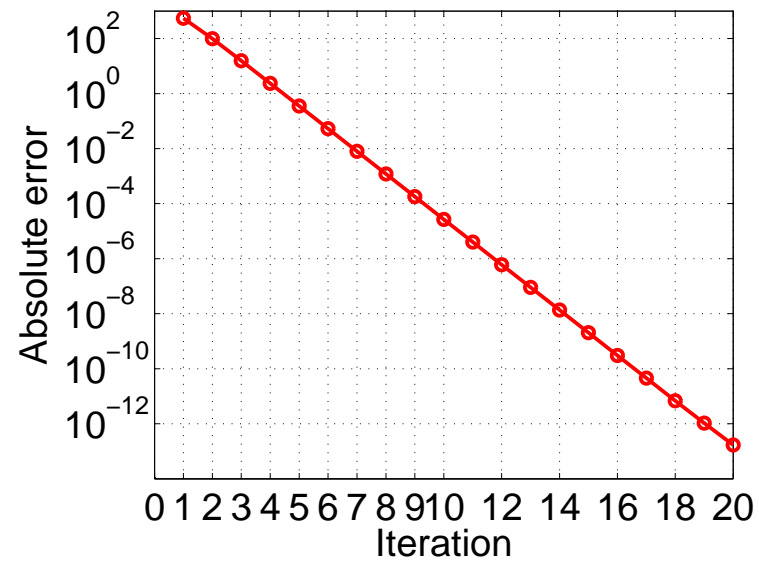
254.7676 145.2324

45.2453 154.7547

254.7567 145.2386

45.2433 154.7614

Error = $|\text{current step} - \text{previous step counts}|$, sum over four cells.



IPS converges fast and it always converges.

But how good are the estimates?

An Example of Contingency Tables with Fixed Margins

Term-by-Document matrix $n = 10^6$ words and $m = 10^{10}$ (Web) documents.

Cell $x_{ij} = 1$ if word i appears in document j . $x_{ij} = 0$ otherwise.

	Doc 1	Doc 2					Doc m	
Word 1	1	0	0	1	0	0	0	1
Word 2	0	1	0	1	0	0	1	0
Word 3								
Word 4								
Word n								

	Word 2	No Word 2
Word 1	N_{11}	N_{12}
No Word 1	N_{21}	N_{22}

N_{11} : number of documents containing both word 1 and word 2.

N_{22} : number of documents containing neither word 1 nor word 2.

Margins ($M_1 = N_{11} + N_{12}$, $M_2 = N_{11} + N_{21}$) for all rows costs nm , **easy!**

Interactions ($N_{11}, N_{12}, N_{21}, N_{22}$) for all pairs costs $n(n-1)m/2$, **difficult!**

⇒ Sampling a small number of columns to estimate interactions.

A Real Data Example for IPS

W_1, W_2 : Two vector of length $D = 2^{16}$.

$W_{1,i}$ = number of times "THIS" appeared in document i , $i = 1$ to D .

$W_{2,i}$ = number of times "HAVE" appeared in document i , $i = 1$ to D .

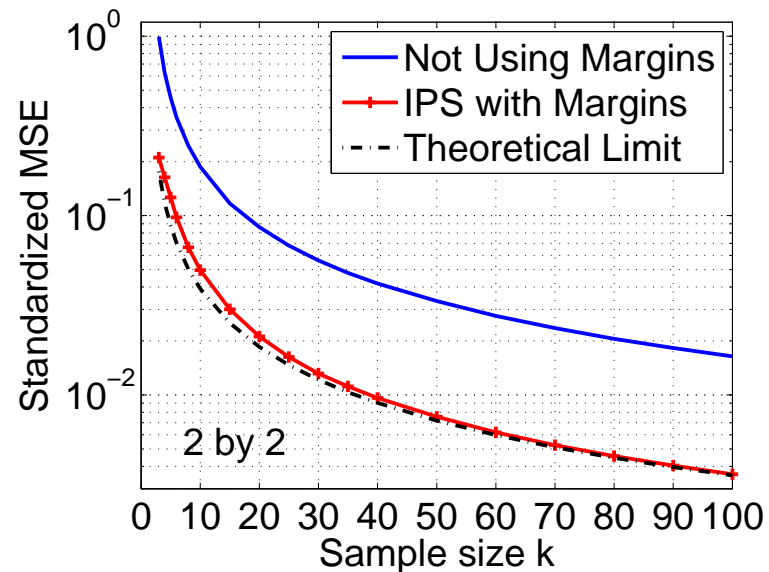
We are interested in the inner product:

$$a = \sum_{i=1}^D W_{1,i} \times W_{2,i}$$

If $W_{1,i}, W_{2,i} \in \{0, 1\}$, ie binary, \implies a 2 by 2 table.

Binary quantize $W_{1,i}, W_{2,i}$ to be $\in \{0, 1\}$, ie. if $W_{1,i} \geq 1$, then let $W_{1,i} = 1$.

This is 2 by 2 table. Suppose we know the margins, i.e., $\sum_{i=1}^D W_{1,i}$.

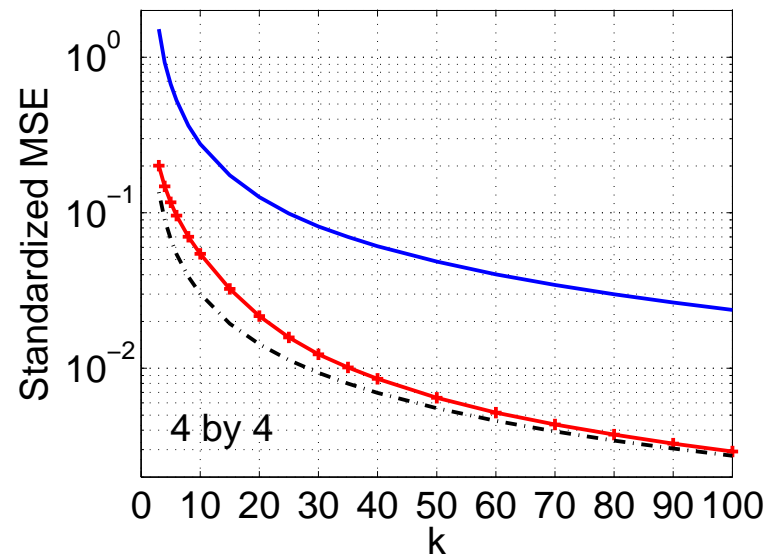


Using known margins considerably reduces the errors.

Using IPS does not lose much accuracy

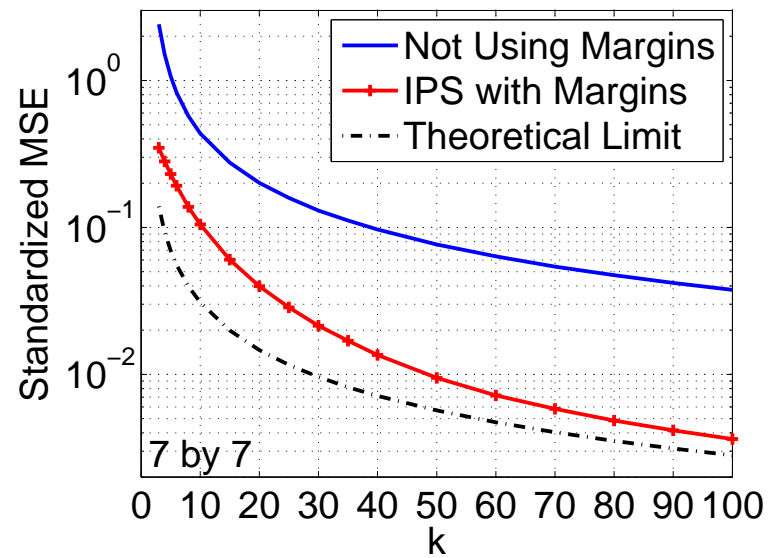
Theoretical limit is computed from Fisher Information

If $W_{1,i} \geq 3$, then let $W_{1,i} = 3$. \implies a 4 by 4 table.



There is more noticeable loss of accuracy by IPS

If $W_{1,i} \geq 6$, then let $W_{1,i} = 6$. \implies a 7 by 7 table.



IPS still helps (a lot), but MLE should be considerably better

Variance of MLE and Fisher Information

Suppose a continuous random variable X has a density function $f_X(x; \theta)$.

Fisher information is defined as

$$I(\theta) = E \left(\frac{\partial \log f_X(x; \theta)}{\partial \theta} \right)^2$$

Equivalently

$$I(\theta) = -E \left(\frac{\partial^2 \log f_X(x; \theta)}{\partial \theta^2} \right)$$

Fisher Information is extremely useful, e.g., for analyzing variances of MLE.

The Asymptotic Behavior of MLE

Given n i.i.d. samples x_1, x_2, \dots, x_n , from a continuous distribution with density $f_X(x; \theta)$. Assume some regularity conditions. Suppose $\hat{\theta}_{MLE}$ is the MLE of θ , then as $n \rightarrow \infty$,

$$\sqrt{n} \left(\hat{\theta}_{MLE} - \theta \right) \xrightarrow{D} N \left(0, \frac{1}{I(\theta)} \right)$$

Approximately, $\hat{\theta}_{MLE}$ is normally distributed with mean θ and variance $\frac{1}{nI(\theta)}$.

In some cases, the variance of the MLE is exactly (not approximately) $\frac{1}{nI(\theta)}$.

Example: Normal Distribution

Given n i.i.d. samples, $x_i \sim N(\mu, \sigma^2)$, $i = 1$ to n .

$$\log f_X(x; \mu, \sigma^2) = -\frac{1}{2\sigma^2}(x - \mu)^2 - \frac{1}{2} \log(2\pi\sigma^2)$$

$$\frac{\partial^2 \log f_X(x; \mu, \sigma^2)}{\partial \mu^2} = -\frac{1}{\sigma^2} \implies I(\mu) = \frac{1}{\sigma^2}$$

$$\frac{\partial^2 \log f_X(x; \mu, \sigma^2)}{\partial (\sigma^2)^2} = -\frac{(x - \mu)^2}{\sigma^6} + \frac{1}{2\sigma^4}$$

$$\implies I(\sigma^2) = \frac{\sigma^2}{\sigma^6} - \frac{1}{2\sigma^4} = \frac{1}{2\sigma^4}$$

Example: Binomial Distribution

$$x \sim \text{Binomial}(p, n): \Pr(x = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Log likelihood and Fisher Information:

$$l(p) = k \log p + (n - k) \log(1 - p)$$

$$l'(p) = \frac{k}{p} - \frac{n - k}{1 - p}$$

$$l''(p) = -\frac{k}{p^2} - \frac{n - k}{(1 - p)^2}$$

$$I(p) = -\mathbf{E}(l''(p)) = \frac{np}{p^2} + \frac{n - np}{(1 - p)^2} = \frac{n}{p(1 - p)}$$

$$\hat{p}_{MLE} = \frac{k}{n} \text{ and } \text{Var}(\hat{p}_{MLE}) = \frac{p(1-p)}{n}, \text{ which is exactly } \frac{1}{I(p)}.$$

Example: Multinomial Distribution

Observations: $(n_{11}, n_{12}, n_{21}, n_{22})$, $n = n_{11} + n_{12} + n_{21} + n_{22}$.

Parameters $(\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22})$, $(\pi_{11} + \pi_{12} + \pi_{21} + \pi_{22} = 1)$

n_{11}	n_{12}
n_{21}	n_{22}

π_{11}	π_{12}
π_{21}	π_{22}

Log likelihood

$$l = n_{11} \log \pi_{11} + n_{12} \log \pi_{12} + n_{21} \log \pi_{21} + n_{22} \log \pi_{22}$$

Derivatives

$$\frac{\partial l}{\pi_{11}} = \frac{n_{11}}{\pi_{11}} - \frac{n_{22}}{1 - \pi_{11} - \pi_{12} - \pi_{21}},$$

$$\frac{\partial^2 l}{\pi_{11}^2} = -\frac{n_{11}}{\pi_{11}^2} - \frac{n_{22}}{(1 - \pi_{11} - \pi_{12} - \pi_{21})^2}.$$

Because $E(n_{11}) = \pi_{11}n$, $E(n_{22}) = \pi_{22}n$,

$$E\left(-\frac{\partial^2 l}{\pi_{11}^2}\right) = E\left(\frac{n_{11}}{\pi_{11}^2} + \frac{n_{22}}{(1 - \pi_{11} - \pi_{12} - \pi_{21})^2}\right) = \frac{n}{\pi_{11}} + \frac{n}{\pi_{22}}$$

Similarly obtain $E\left(-\frac{\partial^2 l}{\pi_{12}^2}\right)$, $E\left(-\frac{\partial^2 l}{\pi_{11}\pi_{12}}\right)$, etc.

Need to invert a matrix to obtain the covariance matrix.

A more direct method (not covered in this lecture) can show the variance

$$\text{Var}(\hat{\pi}_{11}) = \frac{1}{n}\pi_{11}(1 - \pi_{11}) = \frac{1/n}{\frac{1}{\pi_{11}} + \frac{1}{1-\pi_{11}}}$$

Example: Contingency Table with Known Margins

$$n = n_{11} + n_{12} + n_{21} + n_{22}$$

n_{11}	n_{12}
n_{21}	n_{22}

$$N = N_{11} + N_{12} + N_{21} + N_{22}$$

N_{11}	N_{12}
N_{21}	N_{22}

Margins: $M_1 = N_{11} + N_{12}$, $M_2 = N_{11} + N_{21}$, are known.

Homework: show the (asymptotic) variance of the MLE (for N_{11}) is

$$\text{Var} \left(\hat{N}_{11,MLE} \right) = \frac{N/n}{\frac{1}{N_{11}} + \frac{1}{M_1 - N_{11}} + \frac{1}{M_2 - N_{11}} + \frac{1}{N - M_1 - M_2 + N_{11}}}$$

Margin Helps (A Lot)

Without using margins

$$\text{Var} \left(\hat{N}_{11} \right) = \frac{N/n}{\frac{1}{N_{11}} + \frac{1}{N - N_{11}}}$$

Using margins

$$\text{Var} \left(\hat{N}_{11,MLE} \right) = \frac{N/n}{\frac{1}{N_{11}} + \frac{1}{M_1 - N_{11}} + \frac{1}{M_2 - N_{11}} + \frac{1}{N - M_1 - M_2 + N_{11}}}$$

$$\text{Var} \left(\hat{N}_{11,MLE} \right) \leq \text{Var} \left(\hat{N}_{11} \right).$$

Monte Carlo Methods

- Pseudo random number generations

Linear congruential generator, Mersenne twister

- Nonuniform sampling

The method of inversion, Box-Muller transform, correlated normal vectors

- Simulate Brownian motions

- Monte Carlo methods for numerical integrations

Linear congruential generator

The sequence $Z_1, Z_2, \dots, Z_i, \dots$ generated from

$$Z_{i+1} = (a \times Z_i + b) \text{ mod } p,$$

may resemble random numbers for carefully chosen p , a and b .

p is a large number, e.g., $p = 2^{32}$.

Advantages: Simple, fast.

Disadvantages

- Sensitive to the choice of p , a , b .
- May not produce high-quality random numbers.
- Relatively short period, which is at most p .

Facts about Mersenne twister

- The name was from **Mersenne prime** : $2^n - 1$ and is a prime number.
 $n = 19937$ in Mersenne twister.
- Has a very long period $2^{19937} - 1$.
- Passed numerous stringent randomness tests.
- Reasonably fast, portable, and freely available online.
- Not suitable for cryptography.

Nonuniform Sampling by Inversion

The goal: Sample X from a distribution $F(x)$.

The inversion transform sampling:

- Sample $U \sim \text{Uniform}(0, 1)$.
- Output $X = F^{-1}(U)$

$$\Pr(X \leq x) = \Pr(F^{-1}(U) \leq x) = \Pr(U \leq F(x)) = F(x)$$

Limitation: Need a closed-form F^{-1} .

Examples of Inversion Transform Sampling

- $X \sim \text{Exponential}(\lambda)$, i.e., $F(x) = 1 - e^{-\lambda x}$, $x \geq 0$.
Let $U \sim \text{Uniform}(0, 1)$, then $\frac{\log(1-U)}{-\lambda} \sim \text{Exponential}(\lambda)$
- $X \sim \text{Pareto}(\alpha)$, i.e., $F(x) = 1 - \frac{1}{x^\alpha}$, $x \geq 1$.
Let $U \sim \text{Uniform}(0, 1)$, then $\frac{1}{(1-U)^{1/\alpha}} \sim \text{Pareto}(\alpha)$.

A small trick:

If $U \sim \text{Uniform}(0, 1)$, then $1 - U \sim \text{Uniform}(0, 1)$.

Thus, we can replace $1 - U$ by U .

Most common distributions (eg, normal) do not have closed-form F^{-1} .

The Box-Muller Transform

U_1 and U_2 are i.i.d. samples from $\text{Uniform}(0, 1)$. Then

$$N_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2)$$

$$N_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2)$$

are two i.i.d samples from the standard normal $N(0, 1)$.

Sample Correlated Normal Vectors

To generate $Y \sim MN(\mu, \Sigma)$ (MN for multivariate normal)

- Generate a vector $X \sim N(0, I_n)$
- Output $Y = \Sigma^{1/2}X + \mu$

$\Sigma^{1/2} \times \Sigma^{1/2} = \Sigma$: Matrix square root.

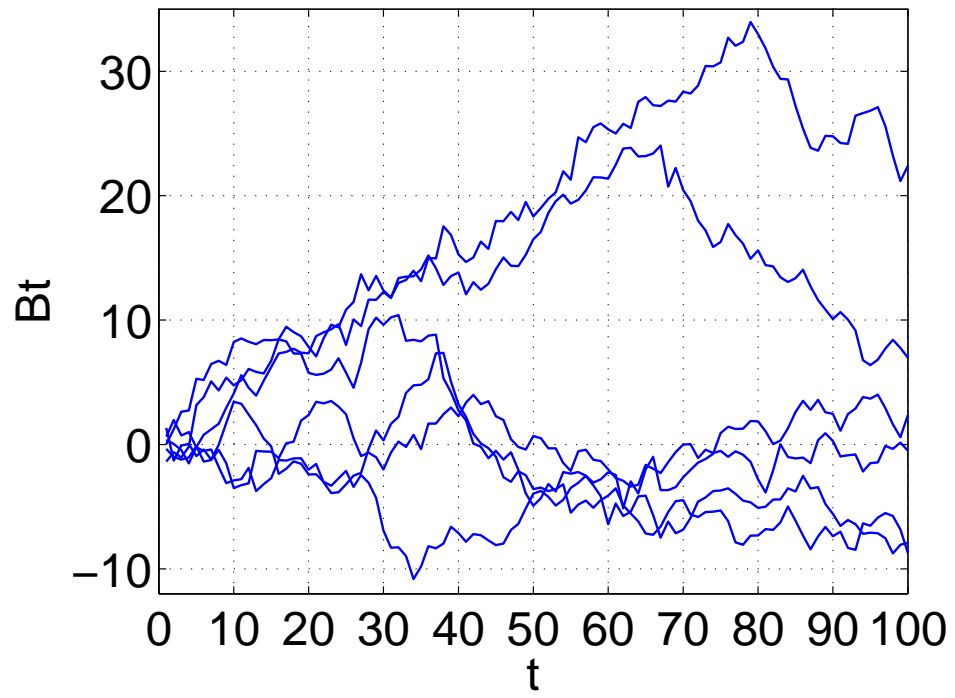
Brownian Motion

A Brownian motion $\mathbf{B} = \{B_t, 0 \leq t < \infty\}$ is characterized by

- $B_0 = 0$
- Independent, normally distributed, increment
 $B_t - B_s \sim N(0, t - s)$, independent of B_s .

Brownian motions are widely used in Finance, among numerous applications.

Homework: Simulate Brownian motions from uniform random samples (using Box-Muller transform).



Monte Carlo Numerical Integrations

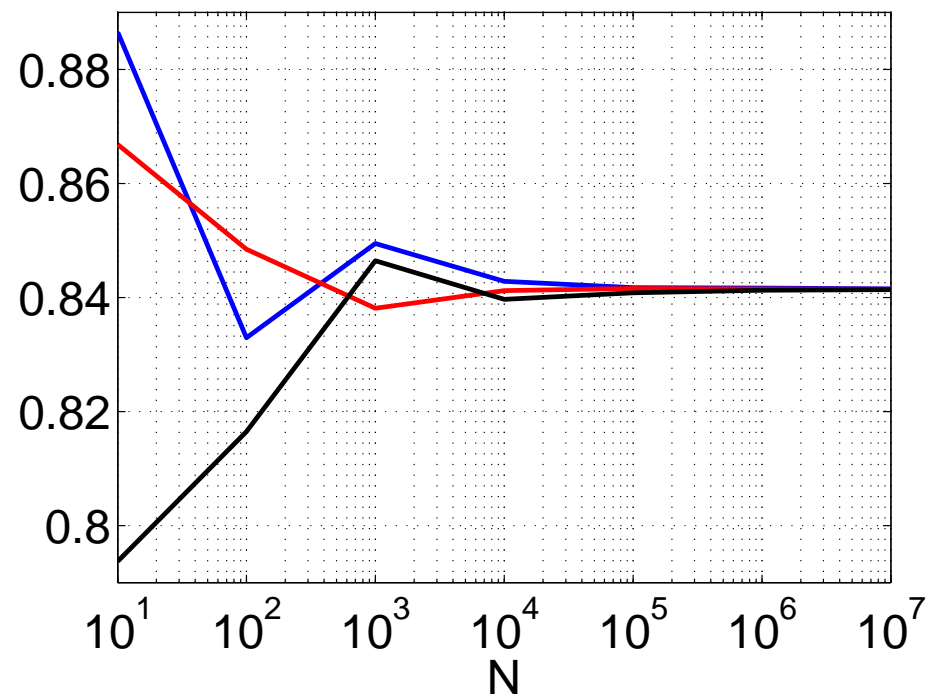
Treat $\int_0^1 \cos x dx$ as an **expectation**:

$$\int_0^1 \cos x dx = \int_0^1 1 \times \cos x dx = \mathbf{E}(\cos(x)), \quad x \sim \text{Uniform } U(0, 1)$$

Monte Carlo integration procedure:

- Generate N i.i.d. samples $x_i \sim \text{Uniform } U(0, 1)$, $i = 1$ to N .
- Use empirical expectation $\frac{1}{N} \sum_{i=1}^N \cos(x_i)$ to approximate $\mathbf{E}(\cos(x))$.

True value: $\int_0^1 \cos x dx = \sin(1) = 0.8415$



- Requires sample size N be “large enough”
- Nice method for complicated functions, instead of $\cos(x)$
- Often the only method for multi-dimensional integrations

$$\int_0^1 \frac{\log^2(x + 0.1)}{\sqrt{\sin(x + 0.1)}} e^{-x^{0.15}} dx$$

