GRADIENT AND NEWTON METHODS

Consider the unconstrained problem

$$\text{minimize} \quad f(x) = -\sum_{i=1}^{m} \log(1 - a_i^T x) - \sum_{i=1}^{n} \log(1 - x_i^2),$$

with variable $x \in \mathbf{R}^n$, and $\mathbf{dom} f = \{x \mid a_i^T x < 1,\ i = 1, \ldots, m,\ |x_i| < 1,\ i = 1, \ldots, n\}$. This is the problem of computing the analytic center of the set of linear inequalities

$$a_i^T x \leq 1, \quad i = 1, \ldots, m, \qquad |x_i| \leq 1, \quad i = 1, \ldots, n.$$

Note that we can choose $x^{(0)} = 0$ as our initial point. You can generate instances of this problem by choosing a_i from some distribution on $\mathbf{R}^n$.

(a) Use the gradient method to solve the problem, using reasonable choices for the backtracking parameters, and a stopping criterion of the form $\|\nabla f(x)\|_2 \leq \eta$. Plot the objective function and step length versus iteration number. (Once you have determined $p^\star$ to high accuracy, you can also plot $f - p^\star$ versus iteration.) Experiment with the backtracking parameters α and β to see their effect on the total number of iterations required. Carry these experiments out for several instances of the problem, of different sizes.

(b) Repeat using Newton's method, with stopping criterion based on the Newton decrement λ^2. Look for quadratic convergence. You do not have to use an efficient method to compute the Newton step, as in exercise 9.27; you can use a general purpose dense solver, although it is better to use one that is based on a Cholesky factorization.

Hints. We recommend the following to generate a problem instance:

```
n = 100;
m = 200;
randn('state',1);
A=randn(m,n);
```

Of course, you should try out your code with different dimensions, and different data as well.

In all cases, be sure that your line search *first* finds a step length for which the tentative point is in $\mathbf{dom} f$; if you attempt to evaluate f outside its domain, you'll get complex numbers, and you'll never recover.

To find expressions for $\nabla f(x)$ and $\nabla^2 f(x)$, use the chain rule (see Appendix A.4); if you attempt to compute $\partial^2 f(x)/\partial x_i \partial x_j$, you will be sorry.

To compute the Newton step, you can use `vnt=-H\g`.

Show Answer

EFFICIENT SOLUTION OF BASIC PORTFOLIO OPTIMIZATION PROBLEM

This problem concerns the simplest possible portfolio optimization problem:

$$\begin{array}{ll} \text{maximize} & \mu^T w - (\lambda/2) w^T \Sigma w \\ \text{subject to} & \mathbf{1}^T w = 1, \end{array}$$

with variable $w \in \mathbf{R}^n$ (the normalized portfolio, with negative entries meaning short positions), and data μ (mean return), $\Sigma \in \mathbf{S}^n_{++}$ (return covariance), and $\lambda > 0$ (the risk aversion parameter). The return covariance has the factor form $\Sigma = FQF^T + D$, where $F \in \mathbf{R}^{n \times k}$ (with rank K) is the *factor loading matrix*, $Q \in \mathbf{S}^k_{++}$ is the factor covariance matrix, and D is a diagonal matrix with positive entries, called the *idiosyncratic risk* (since it describes the risk of each asset that is independent of the factors). This form for Σ is referred to as a 'k-factor risk model'. Some typical dimensions are $n = 2500$ (assets) and $k = 30$ (factors).

(a) What is the flop count for computing the optimal portfolio, if the low-rank plus diagonal structure of Σ is *not* exploited? You can assume that $\lambda = 1$ (which can be arranged by absorbing it into Σ).

(b) Explain how to compute the optimal portfolio more efficiently, and give the flop count for your method. You can assume that $k \ll n$. You do not have to give the best method; any method that has linear complexity in n is fine. You can assume that $\lambda = 1$.

Hints. You may want to introduce a new variable $y = F^T w$ (which is called the vector of factor exposures). You may want to work with the matrix

$$G = \begin{bmatrix} \mathbf{1} & F \\ 0 & -I \end{bmatrix} \in \mathbf{R}^{(n+k)\times(1+k)},$$

treating it as dense, ignoring the (little) exploitable structure in it.

(c) Carry out your method from part (b) on some randomly generated data with dimensions $n=2500$, $k = 30$. For comparison (and as a check on your method), compute the optimal portfolio using the method of part (a) as well. Give the (approximate) CPU time for each method, using `tic` and `toc`.

Hints. After you generate D and Q randomly, you might want to add a positive multiple of the identity to each, to avoid any issues related to poor conditioning. Also, to be able to invert a block diagonal matrix efficiently, you'll need to recast it as sparse.

Show Answer

SIZING A GRAVITY FEED WATER SUPPLY NETWORK

A water supply network connects water supplies (such as reservoirs) to consumers via a network of pipes. Water flow in the network is due to gravity (as opposed to pumps, which could also be added to the formulation). The network is composed of a set of n nodes and m directed edges between pairs of nodes. The first k nodes are supply or reservoir nodes, and the remaining $n - k$ are consumer nodes. The edges correspond to the pipes in the water supply network.

We let $f_j \geq 0$ denote the water flow in pipe (edge) j, and h_i denote the (known) altitude or height of node i (say, above ea level). At nodes $i = 1, \ldots, k$, we let $s_i \geq 0$ denote the flow into the network from the supply. For $= 1, \ldots, n - k$, we let $c_i \geq 0$ denote the water flow taken out of the network (by consumers) at node $k + i$. onservation of flow can be expressed as

Help

$$Af = \begin{bmatrix} -s \\ c \end{bmatrix},$$

where $A \in \mathbf{R}^{n \times m}$ is the incidence matrix for the supply network, given by

$$A_{ij} = \begin{cases} -1 & \text{if edge } j \text{ leaves node } i \\ +1 & \text{if edge } j \text{ enters node } i \\ 0 & \text{otherwise.} \end{cases}$$

We assume that each edge is oriented from a node of higher altitude to a node of lower altitude; if edge j goes from node i to node l, we have $h_i > h_l$. The pipe flows are determined by

$$f_j = \frac{\alpha \theta_j R_j^2 (h_i - h_l)}{L_j},$$

where edge j goes from node i to node l, $\alpha > 0$ is a known constant, $L_j > 0$ is the (known) length of pipe j, $R_j > 0$ is the radius of pipe j, and $\theta_j \in [0, 1]$ corresponds to the valve opening in pipe j.

Finally, we have a few more constraints. The supply feed rates are limited: we have $s_i \leq S_i^{\max}$. The pipe radii are limited: we have $R_j^{\min} \leq R_j \leq R_j^{\max}$. (These limits are all known.)

(a) *Supportable consumption vectors.* Suppose that the pipe radii are fixed and known. We say that $c \in \mathbf{R}_+^{n-k}$ is supportable if there is a choice of f, s, and θ for which all constraints and conditions above are satisfied. Show that the set of supportable consumption vectors is a polyhedron, and explain how to determine whether or not a given consumption vector is supportable.

Typesetting math: 31%

(b) *Optimal pipe sizing.* You must select the pipe radii R_j to minimize the cost, which we take to be (proportional to) the total volume of the pipes, $L_1 R_1^2 + \cdots + L_m R_m^2$, subject to being able to support a set of consumption vectors, denoted $c^{(1)}, \ldots, c^{(N)}$, which we refer to as consumption scenarios. (This means that any consumption vector in the convex hull of $\{c^{(1)}, \ldots, c^{(N)}\}$ will be supportable.) Show how to formulate this as a convex optimization problem. *Note.* You are asked to choose *one* set of pipe radii, and N sets of valve parameters, flow vectors, and source vectors; one for each consumption scenario.

(c) Solve the instance of the optimal pipe sizing problem with data defined in the file `grav_feed_network_data.m`, and report the optimal value and the optimal pipe radii. The columns of the matrix C in the data file are the consumption vectors $c^{(1)}, \ldots, c^{(N)}$.

Hint. $-A^T h$ gives a vector containing the height differences across the edges.

Show Answer

FLUX BALANCE ANALYSIS IN SYSTEMS BIOLOGY (50 points possible)

Flux balance analysis is based on a very simple model of the reactions going on in a cell, keeping track only of the gross rate of consumption and production of various chemical species within the cell. Based on the known stoichiometry of the reactions, and known upper bounds on some of the reaction rates, we can compute bounds on the other reaction rates, or cell growth, for example.

We focus on m metabolites in a cell, labeled $M_1, \ldots, M_m$. There are n reactions going on, labeled $R_1, \ldots, R_n$, with nonnegative reaction rates $v_1, \ldots, v_n$. Each reaction has a (known) stoichiometry, which tells us the rate of consumption and production of the metabolites per unit of reaction rate. The stoichiometry data is given by the *stoichiometry matrix* $S \in \mathbf{R}^{m \times n}$, defined as follows: S_{ij} is the rate of production of M_i due to unit reaction rate $v_j = 1$. Here we consider consumption of a metabolite as negative production; so $S_{ij} = -2$, for example, means that reaction R_j causes metabolite M_i to be consumed at a rate $2v_j$.

As an example, suppose reaction R_1 has the form $M_1 \rightarrow M_2 + 2M_3$. The consumption rate of M_1, due to this reaction, is v_1; the production rate of M_2 is v_1; and the production rate of M_3 is $2v_1$. (The reaction R_1 has no effect on metabolites $M_4, \ldots, M_m$.) This corresponds to a first column of S of the form $(-1, 1, 2, 0, \ldots, 0)$.

Reactions are also used to model flow of metabolites into and out of the cell. For example, suppose that reaction R_2 corresponds to the flow of metabolite M_1 into the cell, with v_2 giving the flow rate. This corresponds to a second column of S of the form $(1, 0, \ldots, 0)$.

The last reaction, R_n, corresponds to biomass creation, or cell growth, so the reaction rate v_n is the cell growth rate. The last column of S gives the amounts of metabolites used or created per unit of cell growth rate.

Since our reactions include metabolites entering or leaving the cell, as well as those converted to biomass within the cell, we have conservation of the metabolites, which can be expressed as $Sv = 0$. In addition, we are given upper limits on *some* of the reaction rates, which we express as $v \preceq v^{\max}$, where we set $v_j^{\max} = \infty$ if no upper limit on reaction rate j is known. The goal is to find the maximum possible cell growth rate (i.e., largest possible value of v_n) consistent with the constraints

$$Sv = 0, \qquad v \succeq 0, \qquad v \preceq v^{\max}.$$

The questions below pertain to the data found in `fba_data.m`.

(a) What the maximum possible cell growth rate $G^\star$?

Which of the reaction rate limits have nonzero optimal Lagrange multipliers?

- □ R_1
- □ R_2
- □ R_3
- □ R_4
- □ R_5
- □ R_6
- □ R_7
- □ R_8
- □ R_9

Which of the reaction rate limits is the maximum growth rate most sensitive to?

- ○ R_1
- ○ R_2
- ○ R_3

○ R_4

○ R_5

○ R_6

○ R_7

○ R_8

○ R_9

(b) *Essential genes and synthetic lethals.* For simplicity, we'll assume that each reaction is controlled by an associated gene, i.e., gene G_i controls reaction R_i. Knocking out a set of genes associated with some reactions has the effect of setting the reaction rates (or equivalently, the associated $v^{\max}$ entries) to zero, which of course reduces the maximum possible growth rate. If the maximum growth rate becomes small enough or zero, it is reasonable to guess that knocking out the set of genes will kill the cell. An *essential gene* is one that when knocked out reduces the maximum growth rate below a given threshold $G^{\min}$. (Note that G_n is always an essential gene.) A *synthetic lethal* is a pair of non-essential genes that when knocked out reduces the maximum growth rate below the threshold. Find all essential genes and synthetic lethals for the given problem instance, using .

Given the threshold $G^{\min} = 0.2G^{\star}$, which of the following genes are essential?

□ G_1

□ G_2

□ G_3

□ G_4

□ G_5

□ G_6

□ G_7

□ G_8

□ G_9

Given the threshold $G^{\min} = 0.2G^{\star}$, which of the following gene pairs are synthetic lethals?

□ $\{G_2, G_3\}$

□ $\{G_2, G_5\}$

□ $\{G_2, G_7\}$

□ $\{G_3, G_4\}$

□ $\{G_4, G_6\}$

□ $\{G_4, G_7\}$

□ $\{G_5, G_6\}$

□ $\{G_5, G_7\}$

□ $\{G_6, G_8\}$

□ $\{G_7, G_8\}$

Show Answer *You have used 0 of 2 submissions*

ONLINE ADVERTISING DISPLAYS (60 points possible)

When a user goes to a website, one of a set of n ads, labeled $1, \ldots, n$, is displayed. This is called an *impression*. We divide some time interval (say, one day) into T periods, labeled $t = 1, \ldots, T$. Let $N_{it} \geq 0$ denote the number of impressions in period t for which we display ad i. In period t there will be a total of $I_t > 0$ impressions, so we must have $\sum_{i=1}^n N_{it} = I_t$, for $t = 1, \ldots, T$. (The numbers I_t might be known from past history.) You can treat all these numbers as real. (This is justified since they are typically very large.)

The revenue for displaying ad i in period t is $R_{it} \geq 0$ per impression. (This might come from click-through payments, for example.) The total revenue is $\sum_{t=1}^T \sum_{i=1}^n R_{it} N_{it}$. To maximize revenue, we would simply display the ad with the highest revenue per impression, and no other, in each display period.

We also have in place a set of m contracts that require us to display certain numbers of ads, or mixes of ads (say, associated with the products of one company), over certain periods, with a penalty for any shortfalls. Contract j is characterized by a set of ads $\mathcal{A}_j \subseteq \{1, \ldots, n\}$ (while it does not affect the math, these are often disjoint), a set of periods $\mathcal{T}_j \subseteq \{1, \ldots, T\}$, a target number of impressions $q_j \geq 0$, and a shortfall penalty rate $p_j > 0$.

The *shortfall* s_j for contract j is

$$s_j = \left(q_j - \sum_{t \in \mathcal{T}_j} \sum_{i \in A_j} N_{it} \right)_+ ,$$

where $(u)_+$ means $\max\{u, 0\}$. (This is the number of impressions by which we fall short of the target value q_j.) Our contracts require a total penalty payment equal to $\sum_{j=1}^m p_j s_j$. Our net profit is the total revenue minus the total penalty payment.

Use convex optimization to find the display numbers N_{it} that maximize net profit. The data in this problem are $R \in \mathbf{R}^{n \times T}$, $I \in \mathbf{R}^T$ (here I is the vector of impressions, not the identity matrix), and the contract data $\mathcal{A}_j$, $\mathcal{T}_j$, q_j, and p_j, $j = 1, \ldots, m$.

Carry out your method on the problem with data given in ad_disp_data.m. The data $\mathcal{A}_j$ and $\mathcal{T}_j$, for $j = 1, \ldots, m$ are given by matrices $A^{\text{contr}} \in \mathbf{R}^{n \times m}$ and $T^{\text{contr}} \in \mathbf{R}^{T \times m}$, with

$$A_{ij}^{\text{contr}} = \begin{cases} 1 & i \in \mathcal{A}_j \\ 0 & \text{otherwise,} \end{cases} \qquad T_{tj}^{\text{contr}} = \begin{cases} 1 & t \in \mathcal{T}_j \\ 0 & \text{otherwise.} \end{cases}$$

What is the optimal net profit?

□

□

What is the revenue associated with the optimal net profit?

□

□

What is the total penalty payment associated with the optimal net profit?

What is the net profit if we were to display only the ad with the largest revenue per impression?

What is the revenue if we were to display only the ad with the largest revenue per impression?

What is the total penalty if we were to display only the ad with the largest revenue per impression?

Show Answer *You have used 0 of 2 submissions*

RANKING BY AGGREGATING PREFERENCES (40 points possible)

We have n objects, labeled 1, \ldots, n. Our goal is to assign a real valued rank r_i to the objects. A *preference* is an ordered pair (i,j), meaning that object i is preferred over object j. The ranking r \in \mathbf{R}^n and preference (i,j) are *consistent* if r_i \geq r_j +1. (This sets the scale of the ranking: a gap of one in ranking is the threshold for preferring one item over another.) We define the *preference violation* of preference (i,j) with ranking r\in \mathbf{R}^n as

v = (r_j+1-r_i)_+ = \max \{ r_j+1-r_i, 0\}.

We have a set of m preferences among the objects, (i^{(1)},j^{(1)}), \ldots ,(i^{(m)}, j^{(m)}). (These may come from several different evaluators of the objects, but this won't matter here.)

We will select our ranking r as a minimizer of the total preference violation penalty, defined as

J = \sum_{k=1}^m \phi (v^{(k)}),

where v^{(k)} is the preference violation of (i^{(k)},j^{(k)}) with r, and \phi is a nondecreasing convex penalty function that satisfies \phi(u)=0 for u \leq 0.

(a) Say we don't mind some small violations, but we really want to avoid large violations. Which choice of \phi will likely produce a solution that matches this criterion?

○ \phi(u)=u_+

○ \phi(u)=u_+^2

(b) Say we want as many preferences as possible to be consistent with the ranking, but will accept some (hopefully, few) larger preference violations. Which choice of \phi will likely produce a solution that matches this criterion?

○ \phi(u)=u_+

○ \phi(u)=u_+^2

(c) Find the rankings obtained using the penalty functions proposed in part (a) and (b), on the data set found in `rank_aggr_data.m`. Plot a histogram of preference violations for each.

How many more positive preference violations does the penalty function of (a) have than that of (b)? (Use `sum(v>0.001)` to determine this number.)

The penalty function of (a) has more *large* preference violations than the penalty function of (b).

○ true

○ false

Show Answer *You have used 0 of 2 submissions*

Terms of Service (Revised 06/14/14) Privacy Policy (Revised 06/14/14) Honor Code Copyright Careers Contact Help

SU Home
Maps & Directions
Search Stanford
Terms of Use
Copyright Complaints