Biomath 210 Homework 4

Huwenbo Shi (603-778-363) shihuwenbo@ucla.edu

November 9, 2015

Problem 5.6

In this problem, we assume $t > 0$. From **Proposition 5.2.4**, we know that any vector $\mathbf{y}$ can be decomposed as

$$\mathbf{y} = \text{prox}_{tf}(\mathbf{y}) + \text{prox}_{(tf)^*}(\mathbf{y}). \tag{1}$$

Therefore

$$\text{prox}_{tf}(\mathbf{y}) = \mathbf{y} - \text{prox}_{(tf)^*}(\mathbf{y}). \tag{2}$$

From **Example 3.4.4**, we notice that the Fenchel conjugate $(tf)^*$ of $tf(\mathbf{x})$ is δ_{tS}, where $S = \{\mathbf{x} : \mathbf{1}^*\mathbf{x} = 1, x_i \geqslant 0\ \forall i\}$, as shown below

$$\max_i ty_i = \max_{\mathbf{x}\in S} \sum_i tx_i y_i = \max_{\mathbf{z}\in tS} \sum_i z_i y_i = \sup_{\mathbf{z}} [\mathbf{y} * \mathbf{z} - \delta_{tS}(\mathbf{z})]. \tag{3}$$

Since,

$$\text{prox}_{(tf)^*}(\mathbf{y}) = \text{prox}_{\delta_{tS}}(\mathbf{y}) = \underset{\mathbf{x}}{\text{argmin}}[\delta_{tS}(\mathbf{x}) + \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2] = P_{tS}(\mathbf{y}), \tag{4}$$

we conclude that

$$\text{prox}_{tf}(\mathbf{y}) = \mathbf{y} - P_{tS}(\mathbf{y}). \tag{5}$$

Problem 5.8

Matlab script implementing Newton's method for finding $\text{prox}_{cf}(y)$ is attached. Figure 2 shows some results for different initializations of y, c, and z. First 3 iterations of Newton's method is marked on the plot of the function.

```
% initialization
y = 2.0; c = 2.0; x = -5.0; z = 0.0; r = 0.9;
eps = 10^-8; max_iter = 100;

% find prox using newton's method
for i=1:max_iter
```

```
    % compute first and second derivative
    derv1 = -c*z + c*exp(x)/(1+exp(x)) + x - y;
    derv2 = c*exp(x)/(1+exp(x))^2 + 1;

    % backtracking line search
    t = 1; next_x = x - t*derv1/derv2;
    cur_obj = -c*z*x + c*log(1+exp(x)) + 0.5*(y-x)^2;
    while(cur_obj < -c*z*next_x + c*log(1+exp(next_x)) + 0.5*(y-next_x)^2)
        t = t*r; next_x = x - t*derv1/derv2;
    end

    % check stop condition
    new_x = x - t*derv1/derv2;
    new_obj = -c*z*new_x + c*log(1+exp(new_x)) + 0.5*(y-new_x)^2;
    if(cur_obj - new_obj < eps || abs(new_x-x) < eps), break; end

    % update x
    x = new_x;

end
```

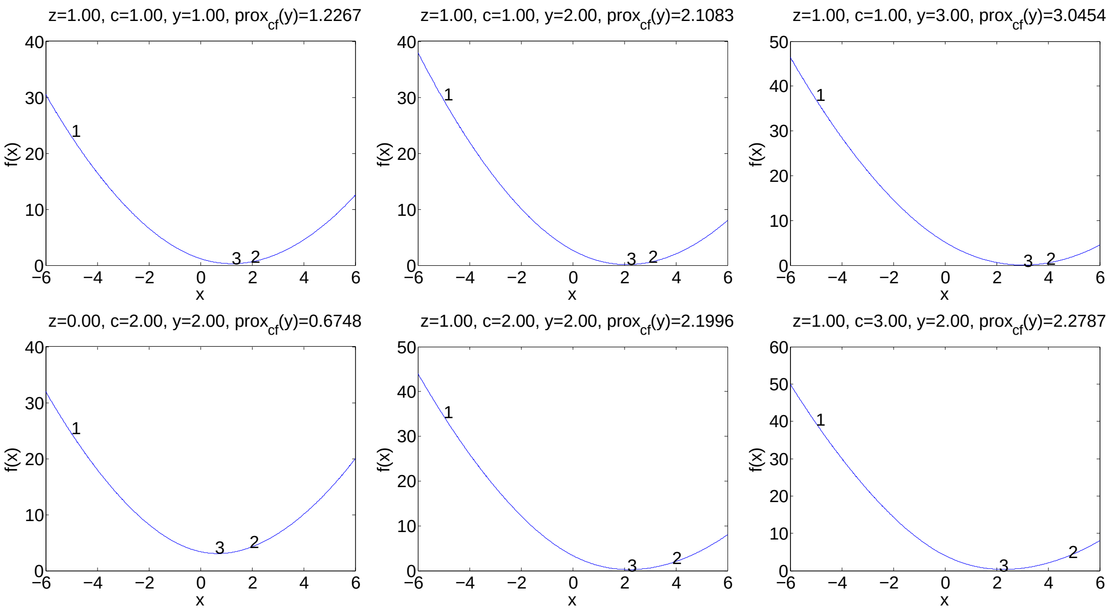

Figure 1: $\text{prox}_{cf}(y)$ for different initializations of y, c, and z

Problem 5.12

To find the projection onto S, we solve the problem of minimizing $\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$ subject to the constraint $\mathbf{1}^*\mathbf{x} = 0$ and $\mathbf{x}^*\mathbf{x} = p$. Because $\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2 = \frac{1}{2}(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2) - \mathbf{y}^*\mathbf{x}$, for $\|\mathbf{x}\|^2$ constrained at p, the problem is equivalent to minimizing the linear term $-\mathbf{y}^*\mathbf{x}$ with the same constraint. The Lagrangian of the transformed problem is

$$\mathcal{L}(\mathbf{x}, \lambda_1, \lambda_2) = -\mathbf{y}^*\mathbf{x} + \lambda_1\mathbf{1}^*\mathbf{x} + \lambda_2\mathbf{x}^*\mathbf{x} - \lambda_2 p. \tag{6}$$

Setting the gradient of the Lagrangian to 0, we have

$$0 = -\mathbf{y} \ + \lambda_1\mathbf{1} + 2\lambda_2\mathbf{x}, \text{ and } \mathbf{x} = \frac{\mathbf{y} - \lambda_1\mathbf{1}}{2\lambda_2}. \tag{7}$$

Because of the constraint $\mathbf{1}^*\mathbf{x} = 0$, we have

$$\frac{\mathbf{1}^*(\mathbf{y} - \lambda_1\mathbf{1})}{2\lambda_2} = 0, \tag{8}$$

which implies

$$\lambda_1 = \frac{1}{p}\mathbf{1}^*\mathbf{y} = \bar{y}. \tag{9}$$

Substituting λ_1 into Equation (7) gives

$$\mathbf{x} = \frac{\mathbf{y} - \bar{y}\mathbf{1}}{2\lambda_2}. \tag{10}$$

The constraint $\mathbf{x}^*\mathbf{x} = p$ entails

$$\frac{\|\mathbf{y} - \bar{y}\mathbf{1}\|^2}{4\lambda_2^2} = p, \text{ and } \lambda_2^2 = \frac{\|\mathbf{y} - \bar{y}\mathbf{1}\|^2}{4p}. \tag{11}$$

Taking $\lambda_2 = \frac{\|\mathbf{y}-\bar{y}\mathbf{1}\|}{2\sqrt{p}}$, and substituting λ_2 into Equation (10) gives the projection

$$P_S(\mathbf{y}) = \frac{\sqrt{p}}{\|\mathbf{y} - \bar{y}\mathbf{1}\|}(\mathbf{y} - \bar{y}\mathbf{1}). \tag{12}$$

When $\mathbf{y} = \bar{y}\mathbf{1}$, for any $\mathbf{x} \in S$ (i.e. $\mathbf{x}^*\mathbf{x} = p$ and $\mathbf{1}^*\mathbf{x} = 0$),

$$\|\mathbf{x} - \bar{y}\mathbf{1}\|^2 = \|\mathbf{x}\|^2 + \|\bar{y}\mathbf{1}\|^2 - 2\mathbf{x}^*(\bar{y}\mathbf{1}) = p + \bar{y}^2 p, \tag{13}$$

therefore, all points of S are equidistant from $\mathbf{y}$.

Problem 5.13

Finding the projection $P_S(\mathbf{y})$ is equivalent to solve the problem $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|^2$, with the constraints $\mathbf{x}^*\mathbf{x} = 1$ and $\mathbf{x} \geqslant 0$. We notice that

$$\|\mathbf{y} - \mathbf{x}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\mathbf{x}^*\mathbf{y} = 1 + \|\mathbf{y}\|^2 - 2\mathbf{x}^*\mathbf{y}, \tag{14}$$

where the second equality follows from the constraint $\mathbf{x}^*\mathbf{x} = 1$. For a fixed $\mathbf{y}$, finding the projection is then equivalent to solving $\min_{\mathbf{x}} -\mathbf{y}^*\mathbf{x}$, with the constraint that $\mathbf{x} \in S$. The Lagrangian of the problem is

$$\mathcal{L}(\mathbf{x}, \lambda, \boldsymbol{\mu}) = -\mathbf{y}^*\mathbf{x} + \lambda\mathbf{x}^*\mathbf{x} - \lambda - \boldsymbol{\mu}^*\mathbf{x}. \tag{15}$$

The KKT condition entails the following equalities at a stationary point

$$0 = -y_i + 2\lambda x_i - \mu_i,\ \mu_i \geqslant 0,\ \mu_i x_i = 0\ \forall i, \tag{16}$$

which then entails

$$x_i = \frac{1}{2\lambda}(\mu_i + y_i). \tag{17}$$

From the constraint $\mathbf{x}^*\mathbf{x} = 1$, we have

$$\sum_{i=1}^p x_i^2 = \frac{1}{4\lambda^2}\sum_{i=1}^p(\mu_i + y_i)^2 = 1, \text{and}\ \ \lambda^2 = \frac{\sum_{i=1}^p(\mu_i + y_i)^2}{4}. \tag{18}$$

From complementary slackness, we also have

$$x_i = \frac{1}{2\lambda}y_i,\ \mu_i = 0,\ \text{if}\ x_i > 0. \tag{19}$$

When $y_i < 0$ for all i

When $y_i < 0$ for all i, we must have $\lambda < 0$ (Equation (19)) and $0 \leqslant \mu_i \leqslant -y_i$ (Equation (17)) in order for the constraint $\mathbf{x} \geqslant 0$ to hold, which entails

$$\lambda = -\frac{\sqrt{\sum_{i=1}^p(\mu_i + y_i)^2}}{2},\ \text{and}\ x_i = -\frac{\mu_i + y_i}{\sqrt{\sum_{i=1}^p(\mu_i + y_i)^2}}. \tag{20}$$

From complementary slackness, if $x_i = 0$, we must have $\mu_i = -y_i$, if $x_i > 0$, we must have $\mu_i = 0$. Therefore,

$$x_i = \begin{cases} -\frac{y_i}{\sqrt{\sum_{\{i:x_i>0\}} y_i^2}} & \text{if}\ x_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Let $C = \{i : x_i > 0\}$, then

$$-\mathbf{y}^*\mathbf{x} = \sum_{i\in C}\frac{y_i^2}{\sqrt{\sum_{i\in C} y_i^2}} = \sqrt{\sum_{i\in C} y_i^2}. \tag{22}$$

Clearly, by setting $C = \{i : i = \operatorname*{argmin}_j y_j^2\}$, minimizes $-\mathbf{y}^*\mathbf{x}$. To satisfy the constraint $\mathbf{x}^*\mathbf{x} = 1$, we must have $x_i = 1$ at $i = \operatorname*{argmin}_j -y_j$, and 0 else where. Therefore, the projection $P_S(\mathbf{y}) = \mathbf{e}_i$, where y_i is the least negative.

When $y_i > 0$ for some i

When $y_i > 0$ for some i, we investigate the case $y_i > 0$, $y_i = 0$, and $y_i < 0$ separately.

- For the case $y_i > 0$, we have $x_i = \frac{1}{2\lambda}(\mu_i + y_i)$. Because $\mu_i \geqslant 0$, we must have $\lambda > 0$ and $x_i > 0$. From complementary slackness $x_i\mu_i = 0$, so we must have $\mu_i = 0$. Therefore, for $y_i > 0$, $x_i = \frac{1}{2\lambda}y_i$.
- For $y_i = 0$, we have $x_i = \frac{1}{2\lambda}\mu_i$, and from complementary slackness, we have $x_i = \mu_i = 0$. For $y_i < 0$, we have $x_i = \frac{1}{2\lambda}(\mu_i + y_i)$.
- For $y_i < 0$, we have $x_i = \frac{1}{2\lambda}(\mu_i + y_i)$. If $x_i = 0$, then $0 = \frac{1}{2\lambda}(\mu_i + y_i)$ entails $\mu_i = -y_i$. If $x_i > 0$, then μ_i must be equal to 0, which entails $x_i = \frac{1}{2\lambda}y_i$, contradicting $x_i > 0$. Therefore, for $y_i < 0$, $x_i = 0$.

From the constraint $\mathbf{x}^*\mathbf{x} = 1$, we conclude that $\lambda = \frac{\sqrt{\sum_{\{i:y_i>0\}} y_i^2}}{2}$. To summarize, when $y_i > 0$ for some i

$$x_i = \begin{cases} \frac{y_i}{\sqrt{\sum_{\{i:y_i>0\}} y_i^2}} & \text{if } y_i > 0 \\ 0 & \text{if } y_i \leqslant 0. \end{cases} \tag{23}$$

So $P_S(\mathbf{y})$ is obtained by setting the non-positive part to 0, and then projecting the rest onto a lower-dimensional unit sphere.

Cases where $P_S(\mathbf{y})$ is multivalued

- When $y_i < 0$ for all i, if there are multiple y_i's that have the same value
- When $y_i < 0$ for some i, and all other $y_i = 0$
- If $y_i = 0$ for all i

Problem 5.14

Let $S = \{\mathbf{e}_1, \ldots, \mathbf{e}_p\}$, and we want to prove that

$$P_S(\mathbf{y}) = \underset{\mathbf{x}\in S}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{y}\| = \mathbf{e}_i, \tag{24}$$

where $i = \underset{j\in\{1,\ldots,p\}}{\operatorname{argmax}}\, y_j$.

To prove this, we suppose that $y_k = \max_{j\in\{1,\ldots,p\}} y_j$, but $P_S(\mathbf{y}) = \mathbf{e}_l$ for some $y_l < y_k$. We notice that

$$\begin{aligned} \|\mathbf{y} - \mathbf{e}_l\|^2 &= \sum_{i=1, i\neq k, i\neq l} y_i^2 + y_k^2 + (y_l - 1)^2 \\ \|\mathbf{y} - \mathbf{e}_k\|^2 &= \sum_{i=1, i\neq k, i\neq l} y_i^2 + (y_k - 1)^2 + y_l^2, \end{aligned} \tag{25}$$

and that

$$\begin{aligned} \|\mathbf{y} - \mathbf{e}_k\|^2 - \|\mathbf{y} - \mathbf{e}_l\|^2 &= 2(y_l - y_k) < 0 \\ \|\mathbf{y} - \mathbf{e}_k\| &< \|\mathbf{y} - \mathbf{e}_l\|, \end{aligned} \tag{26}$$

contradicting the assumption that $\mathbf{e}_l$ is the projection of $\mathbf{y}$ onto S. Therefore, the projection $P_S(\mathbf{y})$ must be $\mathbf{e_k}$, at which we have $\|\mathbf{y}-\mathbf{e}_k\| \leqslant \|\mathbf{y}-\mathbf{e}_l\|$ for all l, i.e. $\mathbf{e}_k$ minimizes $\|\mathbf{x}-\mathbf{y}\|$ with the constraint that $\mathbf{x} \in S$.

Problem 5.15

Let $S = \{\mathbf{N} : \mathbf{N} \in \mathbf{R}^{p\times q}, q \leqslant p, \mathbf{N}^*\mathbf{N} = \mathbf{I}_q\}$. The projection $P_S(\mathbf{M})$ can be found by minimizing $\frac{1}{2}\|\mathbf{M}-\mathbf{N}\|_F^2$ subject to the constraint $\mathbf{N}^*\mathbf{N} = \mathbf{I}_q$. Let $\mathbf{M} = \mathbf{UDV}^*$ be the singular value decomposition of $\mathbf{M}$, where $\mathbf{U} \in \mathbf{R}^{p\times q}$ and $\mathbf{V} \in \mathbf{R}^{q\times q}$ are orthogonal and $\mathbf{D} \in \mathbf{R}^{q\times q}$ diagonal. Because Frobenius norm is orthogonal invariant, it can be shown that $\|\mathbf{M}-\mathbf{N}\|_F^2 = \|\mathbf{D}-\mathbf{U}^*\mathbf{NV}\|_F^2$. Then it follows that

$$\begin{aligned}\frac{1}{2}\|\mathbf{M}-\mathbf{N}\|_F^2 &= \frac{1}{2}\|\mathbf{D}-\mathbf{U}^*\mathbf{NV}\|_F^2 \\ &= \frac{1}{2}\mathrm{tr}[(\mathbf{D}-\mathbf{U}^*\mathbf{NV})^*(\mathbf{D}-\mathbf{U}^*\mathbf{NV})] \\ &= \frac{1}{2}\mathrm{tr}[\mathbf{D}^*\mathbf{D}+\mathbf{I}_q] - \mathrm{tr}[\mathbf{D}^*\mathbf{U}^*\mathbf{NV}].\end{aligned} \tag{27}$$

Therefore, the projection problem is equivalent to minimizing $-\mathrm{tr}[\mathbf{D}^*\mathbf{U}^*\mathbf{NV}]$, with the constraint $\mathbf{N}^*\mathbf{N} = \mathbf{I}_q$. The Lagrangian of this problem is

$$\mathcal{L}(\mathbf{N},\mathbf{\Lambda}) = -\mathrm{tr}[\mathbf{D}^*\mathbf{U}^*\mathbf{NV}] + \mathrm{tr}[\mathbf{\Lambda}\mathbf{N}^*\mathbf{N} - \mathbf{\Lambda}\mathbf{I}_q], \tag{28}$$

where $\mathbf{\Lambda}$ is the symmetric matrix of Lagrange multipliers. Setting the gradient of the Lagrangian to 0, we get

$$\begin{aligned} 0 &= -\mathbf{UDV}^* + \mathbf{N}(\mathbf{\Lambda}+\mathbf{\Lambda}^*) = -\mathbf{UDV}^* + 2\mathbf{\Lambda}\mathbf{N} \\ \mathbf{N} &= \frac{1}{2}\mathbf{\Lambda}^{-1}\mathbf{UDV}^*.\end{aligned} \tag{29}$$

The constraint $\mathbf{N}^*\mathbf{N}$ entails

$$\frac{1}{4}(\mathbf{VDU}^*\mathbf{\Lambda}^{-2}\mathbf{UDV}^*) = \mathbf{I}_q. \tag{30}$$

The solution $\mathbf{\Lambda}^{-1} = 2\mathbf{UD}^{-1}\mathbf{U}^*$ clearly satisfies the equality above, which gives

$$\mathbf{N} = \frac{1}{2}\mathbf{\Lambda}^{-1}\mathbf{UDV}^* = \mathbf{UD}^{-1}\mathbf{U}^*\mathbf{UDV}^* = \mathbf{UV}^*. \tag{31}$$

Problem 5.25

Show that $\|\mathbf{x}\|_{1,2}$ has properties of a norm

First we show that $\|\mathbf{x}\|_{1,2}$ has the properties a norm.

1. Because each $\|\mathbf{x}_{\sigma_g}\| \geqslant 0$, the sum $\|\mathbf{x}\|_{1,2} = \sum_g \|\mathbf{x}_{\sigma_g}\| \geqslant 0$.

2. If $\mathbf{x} = 0$, then each $\|\mathbf{x}_{\sigma_g}\| = 0$, and therefore $\|\mathbf{x}\|_{1,2} = \sum_g \|\mathbf{x}_{\sigma_g}\| = 0$. To prove in the other direction, if $\|\mathbf{x}\|_{1,2} = \sum_g \|\mathbf{x}_{\sigma_g}\| = 0$, then each $\|\mathbf{x}_{\sigma_g}\| = 0$, because of the property $\|\mathbf{x}_{\sigma_g}\| \geqslant 0$. Therefore, each $\mathbf{x}_{\sigma_g} = 0$, and so $\mathbf{x} = 0$.
3. Also, $\|c\mathbf{x}\|_{1,2} = \sum_g \|c\mathbf{x}_{\sigma_g}\| = |c| \sum_g \|\mathbf{x}_{\sigma_g}\| = |c|\|\mathbf{x}\|_{1,2}$.
4. Finally, $\|\mathbf{x}+\mathbf{y}\|_{1,2} = \sum_g \|\mathbf{x}_{\sigma_g} + \mathbf{y}_{\sigma_g}\| \leqslant \sum_g [\|\mathbf{x}_{\sigma_g}\| + \|\mathbf{y}_{\sigma_g}\|] = \|\mathbf{x}\|_{1,2} + \|\mathbf{y}\|_{1,2}$

In conclusion, $\|\mathbf{x}\|_{1,2}$ has the properties of a norm.

The projection of $\mathbf{x}$ when $\mathbf{x} \in B_r$

If $\sum_g c_g \leqslant r$, then $\mathbf{x} \in B_r$, and the projection of $\mathbf{x}$ onto B_r is $\mathbf{x}$.

Show that if $c_g = 0$, then $\mathbf{y}_{\sigma_g} = 0$

The problem of finding the projection $\mathbf{y}$ of $\mathbf{x}$ onto B_r can be solved by minimizing $\|\mathbf{y}-\mathbf{x}\|^2$ with the constraint $\sum_g \|\mathbf{y}_{\sigma_g}\| \leqslant r$. First, we notice that

$$\begin{aligned}\|\mathbf{y}-\mathbf{x}\|^2 &= \mathbf{x}^*\mathbf{x} + \mathbf{y}^*\mathbf{y} - 2\mathbf{x}^*\mathbf{y} \\ &= \sum_g [\mathbf{x}^*_{\sigma_g}\mathbf{x}_{\sigma_g} + \mathbf{y}^*_{\sigma_g}\mathbf{y}_{\sigma_g} - 2\mathbf{y}^*_{\sigma_g}\mathbf{x}_{\sigma_g}] \\ &= \sum_g \|\mathbf{x}_{\sigma_g} - \mathbf{y}_{\sigma_g}\|^2.\end{aligned} \tag{32}$$

In other words, the minimization can be done separately for each g. Define $r_g \geqslant 0$ and suppose $\sum_g r_g \leqslant r$, then the above minimization problem can be separated into minimizing $\|\mathbf{x}_{\sigma_g} - \mathbf{y}_{\sigma_g}\|^2$ with the constraint $\|\mathbf{y}_{\sigma_g}\| \leqslant r_g$ for each g. Therefore, for g the optimal $\mathbf{y}_{\sigma_g}$ is the projection $\mathbf{x}_{\sigma_g}$ onto the Euclidean ball with radius r_g. (From the hint, we also know that the optimal r_g can be found by ordinary projection and satisfies $r_g \leqslant c_g$ for all g.) Obviously, if $c_g = \|\mathbf{x}_{\sigma_g}\| = 0$, then $\mathbf{x}_{\sigma_g} = 0$, and is inside the Euclidean ball with radius r_g. Hence, the projection $\mathbf{y}_{\sigma_g} = \mathbf{x}_{\sigma_g} = 0$.

The projection when all $c_g > 0$

From the previous subsection, we know that the optimal $\mathbf{y}_{\sigma_g}$ can be found by projecting $\mathbf{x}_{\sigma_g}$ onto the Euclidean ball with radius r_g. Therefore,

$$\mathbf{y}_{\sigma_g} = \frac{r_g}{\|\mathbf{x}_{\sigma_g}\|}\mathbf{x}_{\sigma_g} = r_g c_g^{-1}\mathbf{x}_{\sigma_g}. \tag{33}$$

As stated in the hint, the variables r_g, can be found by projecting $\mathbf{c}$ onto the intersection of the non-negative orthant and the half-space $\{\mathbf{z} : \mathbf{z} \geqslant 0, \mathbf{1}^*\mathbf{z} \leqslant r\}$. For the case $\sum_g c_g > r$, this projection is equivalent to projecting $\mathbf{c}$ onto the simplex $\{\mathbf{z} : \mathbf{z} \geqslant 0, \mathbf{1}^*\mathbf{z} = r\}$. Since $c_g \geqslant 0$, the projection onto simplex is equivalent to projecting $\mathbf{c}$ onto the ℓ_1 ball $\{\mathbf{z} : \|\mathbf{z}\|_1 = r\}$. Therefore, $r_g = d_g$, where d_g is the pertinent element in $\mathbf{d}$, obtained by projecting $\mathbf{c}$ onto the ℓ_1 norm ball with radius r. In conclusion, $\mathbf{y}_{\sigma_g} = d_g c_g^{-1}\mathbf{x}_{\sigma_g}$.

Problem 5.26

Because $f(\mathbf{x})$ is a convex function, it must satisfy the supporting hyperplane inequality $f(\mathbf{x}) \geqslant f(\mathbf{x}) + \mathbf{g}^*(\mathbf{x}-\mathbf{y})$, where $\mathbf{g}$ is the subgradient of $f(\mathbf{x})$ at $\mathbf{x}$. (On the other hand, if $f(\mathbf{x})$ satisfies the supporting hyperplane inequality, it's convex.) Therefore, at $\mathbf{x}_i$ and $\mathbf{x}_j$, the following inequality must be satisfied for $f(\mathbf{x})$ to be convex

$$f(\mathbf{x}_j) \geqslant f(\mathbf{x}_i) + \mathbf{g}_j^*(\mathbf{x}_j - \mathbf{x}_i), \tag{34}$$

where $\mathbf{g}_j$ denotes the subgradient of $f(\mathbf{x})$ at $\mathbf{x}_j$. Let $z_i = f(\mathbf{x}_i)$, the problem of finding a convex function $f(\mathbf{x})$ that minimizes the sum of squares $\sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2$, is therefore equivalent to solving the problem of minimizing $\sum_{i=1}^n [y_i - z_i]^2$, with the constraints $z_j \geqslant z_i + \mathbf{g}_j^*(\mathbf{x}_j - \mathbf{x}_i)$. In this problem, the variables that need to be found are z_i, and g_i. Because the objective of this problem is quadratic, and the constraints are linear, it can be solved using convex optimization techniques.

Problem 5.27

To prove majorization, we notice that

$$\frac{1}{2}\|\mathbf{D}\|^2\|\mathbf{x}-\mathbf{x}_n\|^2 \geqslant \frac{1}{2}\|\mathbf{D}\ (\mathbf{x}-\mathbf{x}_n)\|^2 = \frac{1}{2}\mathbf{x}_n^*\mathbf{D}^*\mathbf{D}\mathbf{x}_n + \frac{1}{2}\mathbf{x}^*\mathbf{D}^*\mathbf{D}\mathbf{x} - \mathbf{x}_n^*\mathbf{D}^*\mathbf{D}\mathbf{x}, \tag{35}$$

which entails

$$\begin{aligned}
&\frac{1}{2}\|\mathbf{D}\mathbf{x}_n\|^2 + \mathbf{x}_n\mathbf{D}^*\mathbf{D}(\mathbf{x}-\mathbf{x}_n) + \frac{1}{2}\|\mathbf{D}\|^2\|\mathbf{x}-\mathbf{x}_n\|^2 \\
&\geqslant \frac{1}{2}\|\mathbf{D}\mathbf{x}_n\|^2 + \mathbf{x}_n\mathbf{D}^*\mathbf{D}(\mathbf{x}-\mathbf{x}_n) + \frac{1}{2}\|\mathbf{D}\ (\mathbf{x}-\mathbf{x}_n)\|^2 \\
&= \frac{1}{2}\mathbf{x}_n^*\mathbf{D}^*\mathbf{D}\mathbf{x}_n + \mathbf{x}_n\mathbf{D}^*\mathbf{D}\mathbf{x} - \mathbf{x}_n\mathbf{D}^*\mathbf{D}\mathbf{x}_n + \frac{1}{2}\mathbf{x}_n^*\mathbf{D}^*\mathbf{D}\mathbf{x}_n + \frac{1}{2}\mathbf{x}^*\mathbf{D}^*\mathbf{D}\mathbf{x} - \mathbf{x}_n^*\mathbf{D}^*\mathbf{D}\mathbf{x} \\
&= \frac{1}{2}\mathbf{x}^*\mathbf{D}^*\mathbf{D}\mathbf{x} = \frac{1}{2}\|\mathbf{D}\mathbf{x}\|^2.
\end{aligned} \tag{36}$$

Equality is satisfied at $\mathbf{x} = \mathbf{x}_n$. Problem 18 of chapter 1, in which $\mathbf{D} = (1, 1, -1)$, is a special case of this problem.

Problem 5.28

I implemented the sparse precision matrix estimation algorithm in Matlab, and tested it on a 100×100 covariance matrix computed from simulated data. Figures that show objective values across iteration as well sparsity pattern of the solution is also attached. Overall, the parameter $2k$, the number of non-zero off-diagonal elements, seems to have little impact on the objective value. The Matlab code is also attached.

Results from simulated data

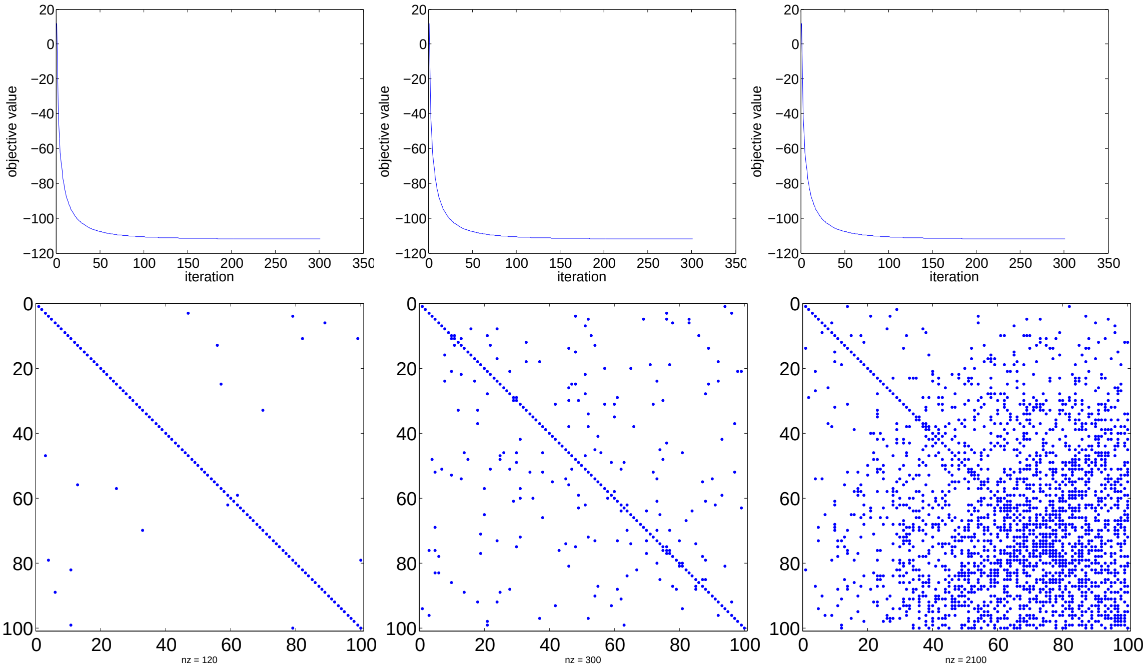

Figure 2: Objective value $-\ln\det\boldsymbol{\Theta} + \mathrm{tr}(\mathbf{S}\boldsymbol{\Theta})$ across iterations and sparsity pattern of the solution for $k = 10, 100, 1000$

spme.m (implements sparse precision matrix estimation)

```
function [theta, obj_val] = spme(S, num_nonzero, rho, r, max_iter)
%SPPME Sparse precision matrix estimation
    %   Uses proximal distance algorithm

    % initialization
    p = size(S,1);
    theta = eye(p);                        % initialization of solution
    obj_val = zeros(max_iter+1, 1);        % store objective values

    % minimize surrogate for each rho
    for i=1:max_iter

        % compute objective value
        obj_val(i) = -log(det(theta))+trace(S*theta);

        % project theta
        theta = project(theta, num_nonzero);
```

```
        % update theta
        [U, D] = eig(S - rho*theta);
        e_vector = zeros(p,1);
        for j=1:p
            e_vector(j) = (-D(j,j)+sqrt(4*rho+D(j,j)^2))/(2*rho);
        end
        theta = U'*diag(e_vector)*U;

        % update rho
        rho = rho * r;
    end

    % project theta onto T^p_k
    theta = project(theta, num_nonzero);
    obj_val(max_iter+1) = -log(det(theta))+trace(S*theta);

end
```

project.m (implements the projection onto T_k^p)

```
function theta_proj = project(theta, num_nonzero)
%PROJECT projects theta onto the set T_k^p
%   follows Example 5.5.3

    p = size(theta, 1);
    theta_proj = theta;
    idx = 1;
    abov_diag = zeros((p*p-p)/2, 3);
    for j=1:p
        for k=j+1:p
            abov_diag(idx,1) = abs(theta(j,k));
            abov_diag(idx,2) = j;
            abov_diag(idx,3) = k;
            idx = idx + 1;
        end
    end
    abov_diag = sortrows(abov_diag, -1);
    for j=(num_nonzero+1):length(abov_diag)
        theta_proj(abov_diag(j,2), abov_diag(j,3)) = 0.0;
    end

    % for below diagonal elements
    idx = 1;
    below_diag = zeros((p*p-p)/2, 3);
```

```
    for j=1:p
        for k=j+1:p
            below_diag(idx,1) = abs(theta(k,j));
            below_diag(idx,2) = k;
            below_diag(idx,3) = j;
            idx = idx + 1;
        end
    end
    below_diag = sortrows(below_diag, -1);
    for j=(num_nonzero+1):length(below_diag)
        theta_proj(below_diag(j,2), below_diag(j,3)) = 0.0;
    end

end
```

main.m (for testing the spme function)

```
% set up problem
p = 100;                      % dimension of matrix
n = 1000;                     % number of samples
samples = 1.2*rand(n, p);     % simulated random samples
S = cov(samples);             % covariance matrix

% ietrate through different number of k
all_num_nonzero = [10; 100; 1000];
for ii=1:length(all_num_nonzero)

    % set up parameters
    rho = 1.0;                            % factor appended for penalty
    r = 1.01;                             % rate of increase for rho
    num_nonzero = all_num_nonzero(ii);    % number of non-zero entries
    max_iter = 300;                       % maximum number of iterations

    % estimate sparse precision matrix
    [theta, obj_val] = spme(S, num_nonzero, rho, r, max_iter);

    % create objective value plots
    figure('visible', 'off');
    plot(obj_val, 'b-');
    xlabel('iteration', 'fontsize', 20);
    ylabel('objective value', 'fontsize', 20);
    set(gca,'FontSize',20)
    print(sprintf('prob_28_obj_k_%d', num_nonzero),'-depsc','-r0');
```

```
    % create sparsity pattern plot
    figure('visible', 'off');
    spy(theta);
    set(gca,'FontSize',20)
    print(sprintf('prob_28_sp_k_%d', num_nonzero),'-depsc','-r0');

end
```