Solutions to Graded Homework 2 (HW6)
CS-526 Learning Theory

Problem 1. Gradient Descent for Positive Semi-definite Matrices

1. Use the spectral decomposition $B = \sum_{j=1}^{n} \lambda_j u_j u_j^T$ and since B is positive definite all $\lambda_j > 0$ (and we can take eigenvectors with real components). Then

$$\begin{aligned} F(X) &= \sum_{j=1}^{n} \lambda_j \mathrm{Tr} X^T u_j u_j^T X = \sum_{j=1}^{n} \lambda_j \mathrm{Tr}(X^T u_j)(X^T u_j)^T \\ &= \sum_{j=1}^{n} \lambda_j (X^T u_j)^T (X^T u_j) = \sum_{j=1}^{n} \lambda_j \|X^T u_j\|^2 \geq 0 \end{aligned}$$

   since $\lambda_j > 0$ for all j.

2. We find

$$\begin{aligned} f^{''}(s) &= 2\mathrm{Tr} X^T B X + 2\mathrm{Tr} Y^T B Y - \mathrm{Tr} X^T B Y - \mathrm{Tr} Y^T B X \\ &= 2\mathrm{Tr}(X-Y)^T B (X-Y) \geq 0 \end{aligned}$$

   Thus f is convex. Since $f(s) = f((1-s).0 + s.1)$ we have $f(s) \leq (1-s)f(0) + sf(1)$. This inequality reads

$$F((sX + (1-s)Y) \leq sF(X) + (1-s)F(Y)$$

3. The gradient of $F(X)$ is the matrix

$$\nabla_X F(X) = BX$$

   This can be computed using components $\frac{\partial}{\partial X_{ij}} F(X)$. Since F is convex it is above its tangent and this shows (see class)

$$F(Y) - F(X) \geq \langle \nabla_X F(X), Y - X \rangle = \mathrm{Tr}(BX)^T (Y - X)$$

   Note the last result can also be found working with components.

   The function is not Lipschitz because the gradient BX is not bounded (locally it is Lipschitz but we did not talk about this in class).

4. For L the gradient is $\nabla L(X) = BX + AX - A$. The gradient descent algorithm is as follows: initialize with X_1 and for $t = 1, \cdots, T$ do

$$X_{t+1} = X_t - \eta(BX_t + AX_t - A)$$

Summing over $t = 1, \cdots, T$ we get

$$\frac{1}{T}(X_{T+1} - X_1) = -\eta((B+A)\frac{1}{T}\sum_{t=1}^{T} X_t - A)$$

Since we assume $\|X_t\| \leq M$ uniformly in t, we can use $\|X_1\| \leq M$ and $\|X_{T+1}\| \leq M$ to get

$$\|\frac{1}{T}\sum_{t=1}^{T} X_t - (B+A)^{-1}A\| \leq \frac{2M}{\eta T}\|(B+A)^{-1}\|$$

Problem 2. Gradient Descent

Let $S^{-1} = U^T\Lambda^{-1}U$ with U an orthogonal matrix, and $\Lambda = \text{Diag}\,(\lambda_1 \cdots \lambda_d)$. With $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x^t$, we have

$$\begin{aligned}
f(\bar{x}) - f(x^*) &\leqslant \frac{1}{T}\sum_{t=1}^{T}\left(f\left(x^t\right) - f\left(x^*\right)\right) \qquad \text{convexity} \\
&\leqslant \frac{1}{T}\sum_{t=1}^{T}\left\langle \nabla f\left(x^t\right), x^t - x^*\right\rangle \qquad \text{convexity} \\
&= \frac{1}{T}\sum_{t=1}^{T}\left\langle U\nabla f\left(x^t\right), Ux^t - Ux^*\right\rangle \\
&= \sum_{k=1}^{d}\frac{1}{T}\sum_{t=1}^{T}(U\nabla f)_k(x^t)\left(U\left(x^t - x^*\right)\right)_k \\
&= \sum_{k=1}^{d}\frac{\lambda_k}{\eta T}\sum_{t=1}^{T}\left(\frac{\eta}{\lambda_k}\right)(U\nabla f)_k(x^t)\left(U\left(x^t - x^*\right)\right)_k \\
= \sum_{k=1}^{d}\frac{\lambda_k}{2\eta T}\sum_{t=1}^{T}&\left\{-\left(\left(U\left(x^t - x^*\right)\right)_k - \frac{\eta}{\lambda_k}(U\nabla f)_k(x^t)\right)^2 + \left(U\left(x^t - x^*\right)\right)_k^2 + \frac{\eta^2}{\lambda_k^2}\left(U\nabla_f\right)_k(x^t)^2\right\}
\end{aligned}$$

Now, from the backward equation we have:

$$\begin{aligned}
&x^{t+1} = x^t - \eta U^T\Lambda^{-1}U\nabla(x^t) \\
&\Rightarrow Ux^{t+1} = Ux^t - \eta\Lambda^{-1}U\nabla f(x^t) \\
&\left(Ux^{t+1}\right)_k = \left(Ux^t\right)_k - \frac{\eta}{\lambda_k}(U\nabla f)_k(x^t)
\end{aligned}$$

From which we get

$$\begin{aligned}
f(\bar{x}) - f(x^*) &\leq \sum_{k=1}^{d} \frac{\lambda_k}{2\eta T} \sum_{t=1}^{T} \left\{ -\left(U\left(x^{t+1} - x^*\right)\right)_k^2 + \left(U\left(x^t - x^*\right)\right)_k^2 + \frac{\eta^2}{\lambda_k^2} \left(U\nabla_f\right)_k (x^t)^2 \right\} \\
&= \sum_{k=1}^{d} \frac{\lambda_k}{2\eta T} \left[\left(U\left(x^1 - x^*\right)\right)_k^2 - \left(U\left(x^{T+1} - x^*\right)\right)_k^2 \right] + \sum_{k=1}^{d} \frac{\lambda_k}{2\eta T} \sum_{t=1}^{T} \frac{\eta^2}{\lambda_k^2} (U\nabla f)_k (x^t)^2 \\
&\leq \frac{\lambda_{\max}}{2\eta T} \sum_{k=1}^{d} \left(U\left(x^1 - x^*\right)\right)_k^2 + \frac{\eta}{2T\lambda_{\min}} \sum_{t=1}^{T} \|U\nabla f\|^2 \\
&= \frac{\lambda_{\max}}{2\eta T} \left\|U\left(x^1 - x^*\right)\right\|^2 + \frac{\eta}{2\lambda_{\min}} \|\nabla f\|^2 \\
&\leq \frac{\lambda_{\max}}{2\eta T} R^2 + \frac{\eta}{2\lambda_{\min}} \rho^2
\end{aligned}$$

where we used that $x^1 = 0$ and $\|x^*\|^2 \leq R^2$ (by assumption) in the last inequality.

Set

$$\eta^2 = \frac{\lambda_{\max}\lambda_{\min} R^2}{\rho^2 T}$$

Then, we find:

$$\begin{aligned}
f(\bar{x}) - f(x^*) &\leq \frac{\lambda_{\max} R^2 \rho \sqrt{T}}{2\sqrt{\lambda_{\max}\lambda_{\min}} RT} + \frac{\sqrt{\lambda_{\max}\lambda_{\min}} R}{\rho\sqrt{T}} \frac{\rho^2}{2\lambda_{\min}} \\
&= \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \frac{\rho R}{2\sqrt{T}} + \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \frac{\rho R}{2\sqrt{T}} \\
&= \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \frac{\rho R}{\sqrt{T}}
\end{aligned}$$

Problem 6. (adapted from 14.3, *Understanding Machine Learning*)

1. We have $\min_{\|\mathbf{w}\| \leq \|\mathbf{w}^*\|} f(\mathbf{w}) \leq f(\mathbf{w}^*) \leq 0$ because $\forall i \in [m] : y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle \geq 1$. Suppose there exists $\mathbf{w}$ satisfying both $\|\mathbf{w}\| \leq \|\mathbf{w}^*\|$ and $f(\mathbf{w}) < 0$. Then $\mathbf{w}$ can be slightly modify to obtain a vector $\tilde{\mathbf{w}}$ such that $\|\tilde{\mathbf{w}}\| < \|\mathbf{w}^*\|$, while still having $f(\tilde{\mathbf{w}}) \leq 0$. It contradicts $\mathbf{w}^*$'s definition, hence $\min_{\|\mathbf{w}\| \leq \|\mathbf{w}^*\|} f(\mathbf{w}) \geq 0$. It proves $\min_{\|\mathbf{w}\| \leq \|\mathbf{w}^*\|} f(\mathbf{w}) = 0$.

2. If $f(\mathbf{w}) < 1$ then $\forall i \in [m] : y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle > 0$, i.e., $\mathbf{w}$ separates the examples.

3. For all $i \in [m]$ the gradient of $f_i : \mathbf{w} \mapsto 1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle$ is $-y_i \mathbf{x}_i$. Applying Claim 14.6, we get that a subgradient of f at $\mathbf{w}$ is given by $-y_{i^*} \mathbf{x}_{i^*}$ where $i^* \in \arg\max_{i \in [m]} \{1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle\}$.

4. The algorithm is inialized with $\mathbf{w}^{(1)} = 0$. At each iteration, if $f(\mathbf{w}^{(t)}) \geq 1$ then it chooses $i^* \in \arg\min_{i \in [m]} \{y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle\}$ and updates $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta y_{i^*} \mathbf{x}_{i^*}$. Otherwise, if $f(\mathbf{w}^{(t)}) < 1$, $\mathbf{w}^{(t)}$ separates all the examples and we stop. To analyze the speed of

convergence of the subgradient algorithm, first notice that $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle - \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle = \eta y_{i^*} \langle \mathbf{w}^*, \mathbf{x}_{i^*} \rangle \geq \eta$. Therefore, after performing T iterations, we have

$$\langle \mathbf{w}^*, \mathbf{w}^{(T+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(T+1)} \rangle - \langle \mathbf{w}^*, \mathbf{w}^{(1)} \rangle = \sum_{t=1}^{T} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle - \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle \geq \eta T \,. \quad (1)$$

Besides, $\|\mathbf{w}^{(t+1)}\|^2 = \|\mathbf{w}^{(t)}\|^2 + \eta^2 y_{i^*}^2 \|\mathbf{x}_i\|^2 + 2\eta y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_{i^*} \rangle \leq \|\mathbf{w}^{(t)}\|^2 + \eta^2 R^2$. The last inequality follows from $\|\mathbf{x}_i\| \leq R$ and $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_{i^*} \rangle \leq 0$ (we update only if $f(\mathbf{w}^{(t)}) \geq 1$). Then

$$\|\mathbf{w}^{(T+1)}\| \leq \eta R \sqrt{T} \,. \quad (2)$$

Combining Cauchy-Schwarz inequality, (1) and (2), we obtain

$$1 \geq \frac{\langle \mathbf{w}^*, \mathbf{w}^{(T+1)} \rangle}{\|\mathbf{w}^{(T+1)}\| \|\mathbf{w}^*\|} \geq \frac{\sqrt{T}}{R \|\mathbf{w}^*\|} \,. \quad (3)$$

The subgradient algorithm must stop in less than $R^2 \|\mathbf{w}^*\|^2$ iterations. We see that η does not affect the speed of convergence.

5. The algorithm is almost identical to the Batch Perceptron algorithm with two modifications. First, the Batch Perceptron updates with any example for which $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0$, while the current algorithm chooses the example for which $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle$ is minimal. Second, the current algorithm employs the parameter η. However, the only difference with the case $\eta = 1$ is that it scales $\mathbf{w}^{(t)}$ by η.