Mehryar Mohri
Foundations of Machine Learning 2018
Courant Institute of Mathematical Sciences
Homework assignment 3
11/10, 2018
Due: 11/26, 2018

A. Kernel PCA

Read the *Dimensionality Reduction* Chapter 12 in the course textbook Foundations of ML with a focus on PCA and Kernel PCA. Sections 12.1 and 12.2 are recommended. In this problem we will analyze a hypothesis set based on KPCA projection. Let $K(x, y)$ be a kernel function, $\Phi_K(x)$ be its corresponding feature map and $S = \{x_1, \ldots, x_m\}$ be a sample of m points. When Π is the rank-r KPCA projection, we define the (regularized) hypothesis set of linear separators in the RKHS $\mathbb{H}$ of kernel K as

$$H = \left\{ x \to \langle w, \Pi\Phi_K(x) \rangle_{\mathbb{H}} : \|w\|_{\mathbb{H}} \leq 1 \right\}. \tag{1}$$

This hypothesis set essentially means that the input data is projected onto a smaller dimensional subspace of the RKHS before fitting a separation hyperplane. This problem will show that we can use the eigenvectors and eigenvalues of the sample kernel matrix to give a closed form expression for the functions $h \in H$ without a need for explicit representation of the RKHS itself.

Let $\mathbf{K}$ be the sample kernel matrix for kernel K evaluated on m points of sample S, that is $\mathbf{K}_{i,j} = K(x_i, x_j)$. Let $\lambda_1, \ldots, \lambda_r$ are the top r (nonzero) eigenvalues of $\mathbf{K}$ with the corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$. Denote the j-th element of vector $\mathbf{v}_i$ as $[\mathbf{v}_i]_j$. Follow the subproblems below to derive the explicit representation of $h \in H$.

1. Assume that the feature maps $\Phi_K(x)$ are centered on sample S and recall that the sample covariance operator is $\Sigma = \sum_{i=1}^m \frac{1}{m}\Phi_K(x_i)\Phi_K(x_i)^\top$. Prove that $h(x) = \sum_{i=1}^r \alpha_i \langle \mathbf{u}_i, \Phi_K(x) \rangle_{\mathbb{H}}$ for some $\alpha_i \in \mathbb{R}$, where $\mathbf{u}_1, \cdots, \mathbf{u}_r$ are the eigenvectors of Σ corresponding to its top r eigenvalues.

Solution: This is a direct application of the orthonormal basis $\mathbf{u}_1, \cdots, \mathbf{u}_r$.

$$\begin{aligned} h(x) &= \langle w, \boldsymbol{U}_k \boldsymbol{U}_k^\top \Phi_k(x) \rangle_{\mathbb{H}} \\ &= \langle w, \sum_{i=1}^{r} \boldsymbol{u}_i \boldsymbol{u}_i^\top \Phi_k(x) \rangle_{\mathbb{H}} \\ &= \sum_{i=1}^{r} \langle w, \boldsymbol{u}_i \rangle_{\mathbb{H}}, \langle \boldsymbol{u}_i, \Phi_k(x) \rangle_{\mathbb{H}} \end{aligned}$$

Denoting $\alpha_i = \langle w, \boldsymbol{u}_i \rangle_{\mathbb{H}}$, we obtain the solution.

2. Prove that $\mathbf{u}_i = \mathbf{X} \frac{\mathbf{v}_i}{\sqrt{\lambda_i}}$, where $\mathbf{X} = [\Phi_K(x_1), \ldots, \Phi_K(x_m)]$

   Solution: For more details see Ch12, Section 12.2 of the textbook. The eigenvalue-eigenvector equation for Σ is

   $$\Sigma \mathbf{u}_i = \gamma_i \mathbf{u}_i$$

   Substituting $\Sigma = \frac{1}{m} \boldsymbol{X} \boldsymbol{X}^\top$ and $\mathbf{u}_i = \mathbf{X} w_i$ for some $w_i \in \mathbb{R}^m$ since u_i belongs to the span of $\mathbf{X} = [\Phi_K(x_1), \ldots, \Phi_K(x_m)]$. Also multiplying by $\boldsymbol{X}^\top$ from the left, we get.

   $$\frac{1}{m} (\boldsymbol{X}^\top \boldsymbol{X})(\boldsymbol{X}^\top \boldsymbol{X}) w_i = \gamma_i (\boldsymbol{X}^\top \boldsymbol{X}) w_i$$

   Divide both sides by m.

   $$\left(\frac{1}{m} \boldsymbol{K} \right)^2 w_i = \frac{\gamma_i}{m} \boldsymbol{K} w_i$$

   It can be shown that the solution to the equation above is $w_i = \frac{\mathbf{v}_i}{\sqrt{\lambda_i}}$, which directly leads to $\mathbf{u}_i = \boldsymbol{X} \frac{\mathbf{v}_i}{\sqrt{\lambda_i}}$.

3. Using the result above, prove that any function $h \in H$ can be represented as

   $$h(x) = \sum_{i=1}^{r} \sum_{j=1}^{m} \frac{\alpha_i}{\sqrt{\lambda_i}} K(x_j, x) [v_i]_j,$$

   for some $\alpha_i \in \mathbb{R}$.

Solution:

$$\begin{aligned}\langle \mathbf{u}_i, \Phi_K(x)\rangle_{\mathbf{H}} &= \Phi_K^\top(x)\mathbf{X}\frac{\mathbf{v}_i}{\sqrt{\lambda_i}} \\ &= \frac{1}{\sqrt{\lambda_i}}\sum_{j=1}^m K(x_j, x)[\mathbf{v}_i]_j\end{aligned}$$

Substituting the above in the result from part 1 provides the final expression for $h(x)$.

4. Bonus question: derive the Rademacher complexity bound on the hypothesis set H defined in this problem.

   Solution: Use the standard techniques for deriving generalization bounds described in this course, as well as Cauchy-Schwarz inequality and Jensen's inequality. For example, one can derive an upper bound $O\left(\sqrt{\frac{Tr(K)}{m}}\right)$ and even tighter one $O\left(\sqrt{\frac{\sum_{i=1}^r \lambda_i}{m}}\right)$.

B. Multi-class boosting

Lecture 10 introduces the AdaBoost.MH algorithm, which is AdaBoost for multi-class classification. (Consult with Lecture 10's slides if you are unfamiliar with multi-class learning setting.) AdaBoost.MH is defined by objective function $F(\alpha)$:

$$F(\alpha) = \sum_{l=1}^k \sum_{i=1}^m e^{-y_i[l]\sum_{t=1}^n \alpha_t h_t(x_i, l)},$$

where $y_i \in \mathcal{Y} = \{-1, +1\}^k$, and $y_i[l]$ denotes the l-th coordinate of y_i for any $i \in [m]$ and $l \in [k]$. The base classifiers come from $H = \{h : \mathcal{X} \times [k] \to \{-1, +1\}\}$. Consider an alternative objective function for the same problem:

$$G(\alpha) = \sum_{i=1}^m e^{-\frac{1}{k}\sum_{l=1}^k y_i[l]\sum_{t=1}^n \alpha_t h_t(x_i, l)}.$$

1. Compare $G(\alpha)$ with $F(\alpha)$. Show that $F(\alpha) \geq kG(\alpha)$.

   Solution: Since e^{-x} is a convex function, by Jensen's inequality

$$\frac{1}{k}\sum_{l=1}^k e^{-y_i[l]\sum_{t=1}^n \alpha_t h_t(x_i, l)} \geq e^{-\frac{1}{k}\sum_{l=1}^k y_i[l]\sum_{t=1}^n \alpha_t h_t(x_i, l)}$$

thus $F(\alpha) \geq kG(\alpha)$

2. Let $g_n(x_i, l) = \sum_{t=1}^{n} \alpha_t h_t(x_i, l)$. Assume that $|g_n(x_i, l)| \leq 1$ for all $x_i \in \mathcal{X}, l \in [k]$. Show that $kG(\alpha)$ is a convex function upper bounding the multi-label multi-class error:

$$\sum_{i=1}^{m} \sum_{l=1}^{k} 1_{y_i[l] \neq \text{sgn}(g_n(x_i,l))} \leq kG(\alpha).$$

Solution: Since the exponential is linear in α and e^{-x} is convex, $G(\alpha)$ is convex.

We have

$$\frac{1}{k} \sum_{l=1}^{k} 1_{y_i[l] \neq \text{sgn}(g_n(x_i,l))} = \frac{1}{k} \sum_{l=1}^{k} 1_{y_i[l] g_n(x_i,l) \leq 0} \leq 1 - \frac{1}{k} \sum_{l=1}^{k} y_i[l] g_n(x_i, l).$$

The last inequality holds because

$$1_{y_i[l] g_n(x_i,l) \leq 0} + y_i[l] g_n(x_i, l) \leq 1,$$

where we use the fact that $|g_n(x_i, l)| \leq 1$ and thus $y_i[l] g_n(x_i, l) \leq 1$. Finally,

$$1 - \frac{1}{k} \sum_{l=1}^{k} y_i[l] g_n(x_i, l) \leq e^{-\frac{1}{k} \sum_{l=1}^{k} y_i[l] g_n(x_i,l)},$$

which concludes the proof.

3. Drive an algorithm defined by the application of coordinate descent to $G(\alpha)$. You should give a full description of your algorithm, including the pseudocode, details for the choice of the step and direction, as well as a generalization bound.

Solution: Define $G_i(\boldsymbol{\alpha}) = e^{-\frac{1}{k} \sum_{l=1}^{k} y_i[l] \sum_{j=1}^{n} \alpha_j h_j(x_i,l)}$ then $G(\boldsymbol{\alpha}) = \sum_{i=1}^{m} G_i(\boldsymbol{\alpha})$. we denote $\boldsymbol{\alpha}_t = (\alpha_1, ..., \alpha_t, 0, ...0)$

For descent direction,

$$\frac{d}{d\eta} G(\boldsymbol{\alpha}_t + \eta e_{t+1}) = -\frac{1}{k} \sum_{i=1}^{m} \sum_{l=1}^{k} y_i[l] h_{t+1}(x_i, l) G_i(\boldsymbol{\alpha}_t + \eta e_{t+1})$$

thus

$$\frac{d}{d\eta}G(\boldsymbol{\alpha}_t + \eta e_{t+1})|_{\eta=0} = -\frac{1}{k}\sum_{i=1}^{m}\sum_{l=1}^{k} y_i[l]h_{t+1}(x_i,l)G_i(\boldsymbol{\alpha}_t)$$
$$= -\sum_{i=1}^{m}\sum_{l=1}^{k} y_i[l]h_{t+1}(x_i,l)D_{t+1}(i)m\Pi_{s=1}^{t}Z_s$$
$$= (2\epsilon_{t+1}-1)m\Pi_{s=1}^{t}Z_s$$

where $D_{t+1}(i) = \frac{D_t(i)e^{-\frac{1}{k}\sum_{j=1}^{k} y_i[j]\alpha_t h_t(x_i,j)}}{Z_t}$ and

$$Z_t = \sum_{i=1}^{m} D_t(i)e^{\alpha_t(2\epsilon_t^i - 1)}$$

where $\epsilon_t^i = Pr_{j\sim U(k)}[y_i[j] \neq h_t(x_i,j)]$.

Also,

$$\epsilon_{t+1} = Pr_{(i,l)\sim D_{t+1}\times U(k)}[y_i[l] \neq h_{t+1}(x_i,l)] = \mathbb{E}_{i\sim D_{t+1}}\epsilon_{t+1}^i$$

Our h_{t+1} minimize ϵ_{t+1}.

For step size note that

$$\frac{d}{d\eta}G(\boldsymbol{\alpha}_t + \eta e_{t+1}) = -\frac{1}{k}\sum_{i=1}^{m}\sum_{l=1}^{k} y_i[l]h_{t+1}(x_i,l)G_i(\boldsymbol{\alpha}_t + \eta e_{t+1})$$
$$= -\frac{1}{k}\sum_{i=1}^{m}\sum_{l=1}^{k} y_i[l]h_{t+1}(x_i,l)G_i(\boldsymbol{\alpha}_t)\exp(-\frac{1}{k}\sum_{j=1}^{k} y_i[j]\eta h_{t+1}(x_i,j))$$
$$= -\sum_{i=1}^{m}\sum_{l=1}^{k} y_i[l]h_{t+1}(x_i,l)\exp(-\frac{1}{k}\sum_{j=1}^{k} y_i[j]\eta h_{t+1}(x_i,j))D_{t+1}(i)m\Pi_{s=1}^{t}Z_s$$
$$= -\sum_{i=1}^{m}\sum_{l=1}^{k} y_i[l]h_{t+1}(x_i,l)\exp(\eta(2\epsilon_{t+1}^i - 1))D_{t+1}(i)m\Pi_{s=1}^{t}Z_s$$

Thus

$$\frac{d}{d\eta}G(\boldsymbol{\alpha}_t + \eta e_{t+1}) = 0 \Leftrightarrow \sum_{i=1}^{m}(2\epsilon_{t+1}^i - 1)D_{t+1}(i)\exp(\eta(2\epsilon_{t+1}^i - 1)) = 0 \quad (2)$$

Algorithm 1 Alternative ADABOOST.MH($S = ((x_1, y_1), ...(x_m, y_m))$)

for $i \leftarrow 1$ to m **do**
  $D_1(i, l) = \frac{1}{m}$
  for $h \in H$ **do**
    $\epsilon_h^i \leftarrow Pr_{j \sim U(k)}[y_i[j] \neq h(x_i, j)]$
  end for
end for
for $t \leftarrow 1$ to T **do**
  $h_t \leftarrow$ base classifier minimize $\mathbb{E}_{i \sim D_t} \epsilon_h^i$
  $\eta_t \leftarrow$ solution of (2)
  $Z_t \leftarrow \mathbb{E}_{i \sim D_t} e^{\eta_t (2\epsilon_t^i - 1)}$
  for $i \leftarrow 1$ to m **do**
    $D_{t+1}(i) \leftarrow \frac{D_t(i) e^{-\frac{1}{k} \sum_{j=1}^{k} y_i[j] \eta_t h_t(x_i, j)}}{Z_t}$
  end for
end for
$g \leftarrow \sum_{t=1}^{T} \eta_t h_t$
return $sgng$

Note that ϵ_t^i are multiple of $\frac{1}{k}$ so by change of variable $x = e^{\frac{\eta}{k}}$ we can transform it into a polynomial equation.

The above analysis gives us algorithm 1.

Note that in this case our weak learning condition becomes $\mathbb{E}_{i \sim D_t} \epsilon_h^i < \frac{1}{2}$ for any distribution D_t and $h \in H$. Also when k is large this alternative algorithm is more efficient than the original ADABOOST.MH.

For generalization bound, note that we are dealing with multi-label classification. For any hypotheses h we can see it as a vector of binary classifiers $(h_1, ...h_k)$, where $h_l(x) = h(x, l)$. We denote $\Pi_l(H) = \{h(\cdot, l) : h \in H\}$

$$R(h) = \mathbb{E}_{x \sim D} d(h(x), y) = \sum_{l=1}^{k} \mathbb{E}_{x \sim D} 1_{h_l(x) \neq y[l]} = \sum_{l=1}^{k} R(h_l)$$

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} d(h(x_i), y_i) = \sum_{l=1}^{k} \frac{1}{m} 1_{h_l(x_i) \neq y_i[l]} = \sum_{l=1}^{k} \hat{R}(h_l)$$

where d is Hamming distance.

We then can use corllary 6.1 on textbook for every $l \in [k]$. Fix ρ and then for any $\delta > 0$, with prob at least $1 - \delta$ the following holds for all $h_l \in conv(\Pi_l(H))$

$$R(h_l) \leq \widehat{R}_\rho(h_l) + \frac{2}{\rho}\mathfrak{R}_m(\Pi_l(H)) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$
$$R(h_l) \leq \widehat{R}_\rho(h_l) + \frac{2}{\rho}\widehat{\mathfrak{R}}_S(\Pi_l(H)) + 3\sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

Thus fix ρ and then for any $\delta > 0$, with prob at least $1 - k\delta$ the following holds for all $g \in conv(H)$

$$R(g) \leq \sum_{l=1}^{k} \widehat{R}_\rho(g_l/\|\alpha\|_1) + \frac{2}{\rho}\sum_{l=1}^{k}\mathfrak{R}_m(\Pi_l(H)) + k\sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$
$$R(g) \leq \sum_{l=1}^{k} \widehat{R}_\rho(g_l/\|\alpha\|_1) + \frac{2}{\rho}\sum_{l=1}^{k}\widehat{\mathfrak{R}}_S(\Pi_l(H)) + 3k\sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$