Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning

Instructor: Purushottam Kar
Date: October 31, 2017
Total: 120 marks

ASSIGNMENT 3

Problem 3.1 (A Consistency Crisis for EM!). Refer to lecture 16 material for this exercise. Let us be given data $X = \left[\mathbf{x}^1, \ldots, \mathbf{x}^n\right]$ which redacts the identities of latent variables $\mathbf{z}^1, \ldots, \mathbf{z}^n$, with the task being to estimate the MLE model $\boldsymbol{\theta}^{\text{MLE}} \in \Theta$ such that $\boldsymbol{\theta}^{\text{MLE}} \in \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ \mathbb{P}\left[X \,|\, \boldsymbol{\theta}\right]$. We have seen how, the EM algorithm proceeds by first finding an estimate $\boldsymbol{\theta}^t$, then constructing a "Q-function" $Q_{\boldsymbol{\theta}^t} : \Theta \to \mathbb{R}$ as

$$Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta}) = \sum_{i=1}^n Q_{i,\boldsymbol{\theta}^t}(\boldsymbol{\theta}),$$

where $Q_{i,\boldsymbol{\theta}^t}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z} \sim \mathbb{P}\left[\mathbf{z} \,|\, \mathbf{x}^i, \boldsymbol{\theta}^t\right]} \log \mathbb{P}\left[\mathbf{x}^i, \mathbf{z} \,|\, \theta\right]$, and then updating $\boldsymbol{\theta}^{t+1} = \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta})$. Given this, show the following *self-consistency* properties of the Q-function

1. $\boldsymbol{\theta}^{\text{MLE}} \in \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ Q_{\boldsymbol{\theta}^{\text{MLE}}}(\boldsymbol{\theta})$
2. If $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2 \in \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ \mathbb{P}\left[X \,|\, \boldsymbol{\theta}\right]$ are two distinct but optimal MLE solutions then $\boldsymbol{\theta}^1 \in \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ Q_{\boldsymbol{\theta}^2}(\boldsymbol{\theta})$ and $\boldsymbol{\theta}^2 \in \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ Q_{\boldsymbol{\theta}^1}(\boldsymbol{\theta})$

You may reuse results proved in class without proving them again. (7+8=15 marks)

Solution. To solve this problem we will need two results that we have proved in lecture 16: if $\boldsymbol{\theta}^t$ and $\boldsymbol{\theta}^{t+1}$ are two successive iterates in the EM algorithm then

1. $Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta}^t) = \log \mathbb{P}\left[X \,|\, \boldsymbol{\theta}^t\right]$ (see lec16.pdf page 43)
2. $\log \mathbb{P}\left[X \,|\, \boldsymbol{\theta}^{t+1}\right] \geq Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta}^{t+1})$ (see lec16.pdf page 44)

We now address the two parts

1. Suppose we take $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{\text{MLE}}$ and execute the EM algorithm for a single step to obtain a new model $\boldsymbol{\theta}^{t+1}$. Suppose $\boldsymbol{\theta}^{\text{MLE}} \notin \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ Q_{\boldsymbol{\theta}^{\text{MLE}}}(\boldsymbol{\theta})$ i.e. $\boldsymbol{\theta}^t \notin \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta})$ since we set $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{\text{MLE}}$. Then since $\boldsymbol{\theta}^{t+1}$ maximizes $Q_{\boldsymbol{\theta}^t}(\cdot)$ because of the M-step, we must have

   $$Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta}^{t+1}) > Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta}^t)$$

   This gives us

   $$\log \mathbb{P}\left[X \,|\, \boldsymbol{\theta}^{t+1}\right] \geq Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta}^{t+1}) > Q_{\boldsymbol{\theta}^t}(\boldsymbol{\theta}^t) = \log \mathbb{P}\left[X \,|\, \boldsymbol{\theta}^t\right]$$

   But this means that $\log \mathbb{P}\left[X \,|\, \boldsymbol{\theta}^{t+1}\right] > \log \mathbb{P}\left[X \,|\, \boldsymbol{\theta}^t\right]$ which contradicts the fact that we set $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{\text{MLE}}$ and no further improvement in observed data log-likelihood should be possible. This means that we must have $\boldsymbol{\theta}^{\text{MLE}} \in \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ Q_{\boldsymbol{\theta}^{\text{MLE}}}(\boldsymbol{\theta})$.

2. The result demanded in this part does not hold in general. For this reason, this part stands canceled.

Problem 3.2 (ReLU guys! I'm going home!). In this exercise, we will show that a ReLU network always learns a piecewise linear function. An n-partition of a set $\mathcal{X}$ is a collection of n subsets $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ such that each $\mathcal{X}_i \subseteq \mathcal{X}$ and

- $\mathcal{X}_i \cap \mathcal{X}_j = \phi$ if $i \neq j$
- $\bigcup_{i=1}^n \mathcal{X}_i = \mathcal{X}$

A piecewise linear function $f : \mathbb{R}^d \to \mathbb{R}$ with $n > 0$ "pieces" is indexed by an n-partition $\{\Omega_1, \ldots, \Omega_n\}$ of $\mathbb{R}^d$ and n linear models $\mathbf{w}^1, \ldots, \mathbf{w}^n$ such that for any $\mathbf{x} \in \mathbb{R}^d$, we have

$$f(\mathbf{x}) = \sum_{i=1}^n \mathbb{I}\{\mathbf{x} \in \Omega_i\} \cdot \langle \mathbf{w}^i, \mathbf{x} \rangle,$$

where $\mathbb{I}\{E\} = 1$ if E is true and 0 otherwise. Now, let $f_{\mathsf{ReLU}}(v) = \max(v, 0)$ for any $v \in \mathbb{R}$ denote the ReLU activation function. Then show that

1. For any piecewise linear function f, and any scalar $c \in \mathbb{R}$, the function $g(\mathbf{x}) = c \cdot f(\mathbf{x})$ is also piecewise linear.
2. The sum of two piecewise linear functions is piecewise linear. Be careful that the two functions could correspond to different (number of) partitions of $\mathbb{R}^d$.
3. For a piecewise linear function f, the function $g(\mathbf{x}) = f_{\mathsf{ReLU}}(f(\mathbf{x}))$ is also piecewise linear.
4. Any neural network with a ReLU activation function computes a piecewise linear function.
5. **Bonus**: If the network d input nodes, only one hidden layer with D nodes and only one output node, and all nodes except input layer nodes apply the ReLU activation function, how many "pieces" does the function computed by the network correspond to?

(5+10+5+15 = 35 marks)

Solution. We provide the solution in parts below

1. If f corresponds to the parts $\{\Omega_1, \ldots, \Omega_n\}$ and linear models $\mathbf{w}^1, \ldots, \mathbf{w}^n$ then g corresponds to the parts $\{\Omega_1, \ldots, \Omega_n\}$ of $\mathbb{R}^d$ and linear models $c \cdot \mathbf{w}^1, \ldots, c \cdot \mathbf{w}^n$. Clearly g is piecewise-linear too.
2. Let f be a piecewise-linear function corresponding to the parts $\{\Omega_1, \ldots, \Omega_n\}$ and linear models $\mathbf{w}^1, \ldots, \mathbf{w}^n$ and g be another piecewise-linear function corresponding to the parts $\left\{\tilde{\Omega}_1, \ldots, \tilde{\Omega}_m\right\}$ and linear models $\tilde{\mathbf{w}}^1, \ldots, \tilde{\mathbf{w}}^m$. Consider the following partition of $\mathbb{R}^d$ with $m \cdot n$ parts $\{\Pi_{ij} : i \in [n], j \in [m]\}$ with $\Pi_{ij} = \Omega_i \cap \tilde{\Omega}_j$. Since $\{\Omega_1, \ldots, \Omega_n\}$ and $\left\{\tilde{\Omega}_1, \ldots, \tilde{\Omega}_m\right\}$ are proper partitions of $\mathbb{R}^d$, and in particular $\Omega_i \cap \Omega_k = \varphi$ if $i \neq k$ and $\tilde{\Omega}_j \cap \tilde{\Omega}_l = \varphi$ if $j \neq l$, we must have $\Pi_{ij} \cap \Pi_{kl} = \varphi$ if either $i \neq k$ or $j \neq l$ or both. Moreover we have

$$\bigcup_{i=1}^n \bigcup_{j=1}^m \Pi_{ij} = \bigcup_{i=1}^n \bigcup_{j=1}^m \left(\Omega_i \cap \tilde{\Omega}_j\right) = \bigcup_{i=1}^n \left(\Omega_i \cap \left(\bigcup_{j=1}^m \tilde{\Omega}_j\right)\right) = \bigcup_{i=1}^n \left(\Omega_i \cap \mathbb{R}^d\right) = \bigcup_{i=1}^n \Omega_i = \mathbb{R}^d$$

Thus, $\{\Pi_{ij}\}$ is a proper (mn)-partition of $\mathbb{R}^d$. Note that several of the regions Π_{ij} may be empty but that is perfectly allowed in the definition of a partition. These empty regions simply don't participate in defining the function. The function $f+g$ is piecewise-linear since it corresponds to the partition $\{\Pi_{ij}\}$ with the linear model $\mathbf{w}^i + \tilde{\mathbf{w}}^j$ acting on the piece Π_{ij} (the sum of two linear functions is another linear function). This is true since the linear model $\mathbf{w}^i$ acts on the piece Ω_i and the linear model $\tilde{\mathbf{w}}^j$ acts on the piece $\tilde{\Omega}_j$. Thus, both models act on the intersection region Π_{ij}. Note that there may very well exist a coarser partition corresponding to $f+g$. However, the above proof shows that there at least exists one partition such that $f+g$ act linearly on all parts of the partition.

3. Let f be a piecewise-linear function corresponding to the parts $\{\Omega_1, \ldots, \Omega_n\}$ and linear models $\mathbf{w}^1, \ldots, \mathbf{w}^n$. Then $f_{\mathsf{ReLU}}(f(\mathbf{x})) = \max(f(\mathbf{x}), 0)$. Now, divide each region Ω_i into two parts $\Omega_i^+ = \{\mathbf{x} \in \Omega_i : f(vx) \geq 0\}$ and $\Omega_i^- = \{\mathbf{x} \in \Omega_i : f(vx) < 0\}$. Clearly $\Omega_i^+ \cup \Omega_i^- = \Omega_i$ and $\Omega_i^+ \cap \Omega_i^- = \varphi$. Consider the new partition of $\mathbb{R}^d$ as $\left\{\Omega_i^+ : i \in [n]\right\} \cup \{\Omega^-\}$ where $\Omega^- = \bigcup_{i=1}^n \Omega_i^-$. This new partition has $n+1$ part with the part Ω^- encoding the region where $f(\mathbf{x}) < 0$. The new partition is clearly a proper $(n+1)$-partition of $\mathbb{R}^d$ since the parts are disjoint by construction and they unite to give back $\mathbb{R}^d$ as $\bigcup_{i=1}^n \Omega_i^+ = \{\mathbf{x} : f(\mathbf{x}) \geq 0\}$. $f_{\mathsf{ReLU}}(f(\cdot))$ is a peicewise-linear function since it corresponds to the $(n+1)$-partition $\left\{\Omega_i^+ : i \in [n]\right\} \cup \{\Omega^-\}$ we just constructed with the linear model $\mathbf{w}^i$ acting on the piece Ω_i^+ and the linear model $\mathbf{0}$ acting on the peice Ω^-.

4. We prove this part using induction on the number of hidden layers in the neural network. A neural network (NN) with 0 hidden layers is a piecewise-linear function since it is of the form $f_{\mathsf{ReLU}}(\mathbf{w}^\top \mathbf{x})$. Since the function $\mathbf{w}^\top \mathbf{x}$ is linear (hence piecewise linear with just one "piece"), using part 3, we get that a network with 0 hidden layers is a piecewise-linear function. Now suppose all NNs with L hidden layers are also peicewise-linear functions. We will show that all NNs with $L+1$ hidden layers are also peicewise-linear functions.

   To see this, let the layer previous to the output layer have n nodes. The output layer itself has just one node. Chop off the output layer and all the edges connecting the n "preoutput" nodes to the output node. We can think of this new network as a network with L hidden layers and n output nodes. However, by the induction hypothesis, each of these n nodes is computing a piecewise-linear function. Let the i-th node compute the function $f_i(\cdot)$. Now bring back the original output node and all edges connecting the n preoutput nodes to the output node. The function computed at the output node is $f_{\mathsf{ReLU}}\left(\sum_{i=1}^n w_i \cdot f_i(\mathbf{x})\right)$ where w_i is the weight on the edge joining the i-th preoutput node to the output node. Using part 1 we get that $w_i \cdot f_i(\cdot)$ is piecewise-linear since $f_i(\cdot)$ is piecewise-linear. Using part 2 we get that $\sum_{i=1}^n w_i \cdot f_i(\cdot)$ is piecewise-linear since it is a sum of piecewise-linear functions. Using part 3 we get that $f_{\mathsf{ReLU}}\left(\sum_{i=1}^n w_i \cdot f_i(\cdot)\right)$ is piecewise-linear which concludes the argument.

5. Each of the hidden nodes computes a function of the form $f_i(\mathbf{x}) = f_{\mathsf{ReLU}}(\langle \mathbf{w}^i, \mathbf{x} \rangle)$ which is piecewise-linear with just 2 pieces. To see this notice that part 3 proves that applying the ReLU function increases the number of peices by atmost one and $\mathbf{w}^\top \mathbf{x}$ is a linear function corresponding to just one piece. The output node computes $f_{\mathsf{ReLU}}(\sum_{i=1}^D v_i \cdot f_i(\mathbf{x}))$ where v_i is the weight of the edge joining the i-th hidden node to the output node. The function $v_i \cdot f_i(\mathbf{x})$ still has 2 pieces since part 1 shows that multiplication by a scalar does not change the pieces. The sum of D piecewise-linear functions, each with 2 pieces can have atmost 2^D pieces. Applying the final ReLU adds one more piece. So the NN corresponds to at most $2^D + 1$ pieces. Using a more careful argument we can show that the number of pieces is atmost $2^{\min\{d,D\}} + 1$.

Problem 3.3 (Kernel Perceptron). Develop a variant of the perceptron algorithm that can work in an RKHS corresponding to a Mercer kernel K. Your algorithm is forbidden from explicitly computing the feature map corresponding to K even once. Your perceptron should at every time step (see lecture 10), receive a data point $(x^t, y^t) \in \mathcal{X} \times \{-1, +1\}$ and perform updates. Just state your final algorithm cleanly giving all details in pseudo-code format - no derivations needed. (25 marks)

Solution. The algorithm is given below

Algorithm 1: KTron: The Kernelized Perceptron

```
Input: A Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, step lengths $\eta_t, t > 0$
$S \leftarrow \varphi$                                         //Initialize empty set
for $t = 1, 2, \ldots,$ do
    Receive data point $(x^t, y^t) \in \mathcal{X} \times \{-1, +1\}$
    if $|S| = 0$ then
        $\hat{y}^t \leftarrow 0$                                   //Default prediction
    else
        $\hat{y}^t \leftarrow \sum_{(\alpha,x)\in S} \alpha \cdot K(x^t, x)$          //Calculate $\langle \mathbf{w}, \phi_K(x^t) \rangle$
    end if
    if $y^t \cdot \hat{y}^t \leq 0$ then
        $\alpha^t \leftarrow \eta_t \cdot y^t$          //Some people prefer $y^t \cdot \hat{y}^t \leq 0$ others use $y^t \cdot \hat{y}^t < 0$
        $S \leftarrow S \cup (\alpha^t, x^t)$                //Update $\mathbf{w} \leftarrow \mathbf{w} + \eta_t \cdot y^t \cdot \phi_K(x^t)$
    end if
end for
```

Problem 3.4 (A Kernel is All You Need). We will denote a 2-dimensional vector as $\mathbf{z} = (x, y) \in \mathbb{R}^2$ where $x, y \in \mathbb{R}$ are the coordinates of the point. Consider the quadratic kernel over these points

$$K(\mathbf{z}^1, \mathbf{z}^2) = (\langle \mathbf{z}^1, \mathbf{z}^2 \rangle + 1)^2.$$

Let $\mathcal{H}_K$ denote the RKHS of the kernel K and let φ_K be the feature map for K. A quadratic function over $\mathbb{R}^2$ is parameterized as $(A, \mathbf{b}, c) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^2 \times \mathbb{R}$ as

$$f_{(A,\mathbf{b},c)}(\mathbf{z}) = \langle \mathbf{z}, A\mathbf{z} \rangle + \langle \mathbf{b}, \mathbf{z} \rangle + c$$

1. Show that the kernel K is Mercer by giving an explicit construction for $\varphi : \mathbb{R}^2 \to \mathcal{H}_K$. You will need to set $\mathcal{H}_K \equiv \mathbb{R}^D$ for an appropriate value of D. What D did you choose?

2. For every quadratic function $f_{(A,\mathbf{b},c)}$ over $\mathbb{R}^2$, construct a $\mathbf{w} \in \mathcal{H}_K$ such that for all $\mathbf{z} \in \mathbb{R}^2$

$$f_{(A,\mathbf{b},c)}(\mathbf{z}) = \langle \mathbf{w}, \varphi_K(\mathbf{z}) \rangle$$

3. For every $\mathbf{w} \in \mathcal{H}_K$, construct a triplet $(A, \mathbf{b}, c) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^2 \times \mathbb{R}$ such that for all $\mathbf{z} \in \mathbb{R}^2$

$$f_{(A,\mathbf{b},c)}(\mathbf{z}) = \langle \mathbf{w}, \varphi_K(\mathbf{z}) \rangle$$

4. Given a regression dataset $(Z, \mathbf{u}) \in \mathbb{R}^{2\times n} \times \mathbb{R}^n$, show that as the regularization parameter $\lambda \to 0^+$, the output of kernel ridge regression over $(Z, \mathbf{u})$ using the kernel K, is a quadratic function $\hat{f}$ over $\mathbb{R}^2$ that offers a least squares error that is arbitrarily close to the smallest least squares error achievable over the dataset by any quadratic function over $\mathbb{R}^2$ i.e.

$$\sum_{i=1}^{n} (u^i - \hat{f}(\mathbf{z}^i))^2 \leq \min_{(A,\mathbf{b},c)\in\mathbb{R}^{2\times 2}\times\mathbb{R}^2\times\mathbb{R}} \sum_{i=1}^{n} (u^i - f_{(A,\mathbf{b},c)}(\mathbf{z}^i))^2 + \epsilon$$

where $\epsilon \to 0$ as $\lambda \to 0^+$. (10+7+8+10=35 marks)

Solution. We give the solution in parts below. We will follow the modified notation as used in the Piazza post `https://piazza.com/class/j5toxxryhdx56k?cid=471`

1. Consider the following 6-dimensional map for a point $\mathbf{z} = (x, y)$

$$\varphi_K(\mathbf{z}) = [x^2,\ y^2,\ \sqrt{2} \cdot xy,\ \sqrt{2} \cdot x,\ \sqrt{2} \cdot y,\ 1]^\top$$

   For any two points $\mathbf{z}^1 = (x_1, y_1)$ and $\mathbf{z}^2 = (x_2, y_2)$ we have

$$\begin{aligned}\left\langle \varphi_K(\mathbf{z}^1), \varphi_K(\mathbf{z}^2) \right\rangle &= x_1^2 x_2^2 + y_1^2 y_2^2 + 2x_1 y_1 x_2 y_2 + 2x_1 x_2 + 2y_1 y_2 + 1 \\ &= (x_1 x_2 + y_1 y_2 + 1)^2 \\ &= (\left\langle \mathbf{z}^1, \mathbf{z}^2 \right\rangle + 1)^2 \\ &= K(\mathbf{z}^1, \mathbf{z}^2)\end{aligned}$$

   This establishes that K is a Mercer kernel.

2. Let $A = \begin{bmatrix} p & q \\ r & s \end{bmatrix}$ and $\mathbf{b} = [k,\ l]^\top$. Given this we have, for any point $\mathbf{z} = (x, y)$

$$f_{(A,\mathbf{b},c)}(\mathbf{z}) = p \cdot x^2 + s \cdot y^2 + (q + r) \cdot xy + k \cdot x + l \cdot y + c$$

   Thus, $f_{(A,\mathbf{b},c)}(\mathbf{z}) = \langle \mathbf{w}, \varphi_K(\mathbf{z}) \rangle$ if we define $\mathbf{w}$ as

$$\mathbf{w} = \left[p,\ s,\ \frac{q + r}{\sqrt{2}},\ \frac{k}{\sqrt{2}},\ \frac{l}{\sqrt{2}},\ c\right]^\top$$

   Note that there is a different $\mathbf{w}$ for each triplet $(A, \mathbf{b}, c)$.

3. Given a vector $\mathbf{w} \in \mathcal{H}_K$ we can actually have many triplets $(A, \mathbf{b}, c)$ such that $f_{(A,\mathbf{b},c)}(\mathbf{z}) = \langle \mathbf{w}, \varphi_K(\mathbf{z}) \rangle$. We give one such construction below. Let $\mathbf{w} = [\alpha, \beta, \gamma, \delta, \zeta, \eta]^\top \in \mathcal{H}_K$. Then let $A = \begin{bmatrix} \alpha & \frac{\gamma}{\sqrt{2}} \\ \frac{\gamma}{\sqrt{2}} & \beta \end{bmatrix}$, $\mathbf{b} = [\delta\sqrt{2},\ \zeta\sqrt{2}]^\top$ and $c = \eta$. In particular, the off-diagonal entries of A could be any two numbers that add up to $\gamma\sqrt{2}$.

4. Since kernel ridge regression learns a linear model in $\mathcal{H}_K$, part 3 shows that it learns a quadratic function over $\mathbb{R}^2$. Let $\hat{\mathbf{w}}$ be the linear function learnt by kernel ridge regression

$$\hat{\mathbf{w}} \in \underset{\mathbf{w} \in \mathcal{H}_K}{\arg\min} \sum_{i=1}^{n} (u^i - \left\langle \mathbf{w}, \varphi_K(\mathbf{z}^i) \right\rangle)^2 + \lambda \cdot \|\mathbf{w}\|_2^2$$

   Let $(\hat{A}, \hat{\mathbf{b}}, \hat{c})$ be the optimal quadratic function for the least squares problem

$$(\hat{A}, \hat{\mathbf{b}}, \hat{c}) \in \underset{(A,\mathbf{b},c) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^2 \times \mathbb{R}}{\arg\min} \sum_{i=1}^{n} (u^i - f_{(A,\mathbf{b},c)}(\mathbf{z}^i))^2$$

   Let $\tilde{\mathbf{w}}$ be the RKHS model corresponding to the quadratic function described by $(\hat{A}, \hat{\mathbf{b}}, \hat{c})$ as constructed in part 2. Note that for all $\mathbf{z}$, we have

$$f_{(\hat{A},\hat{\mathbf{b}},\hat{c})}(\mathbf{z}) = \langle \tilde{\mathbf{w}}, \varphi_K(\mathbf{z}) \rangle$$

Then we have, by optimality of the kernel RR output,

$$\begin{aligned}\sum_{i=1}^n (u^i - \langle \hat{\mathbf{w}}, \varphi_K(\mathbf{z}^i)\rangle)^2 &\leq \sum_{i=1}^n (u^i - \langle \hat{\mathbf{w}}, \varphi_K(\mathbf{z}^i)\rangle)^2 + \lambda \cdot \|\hat{\mathbf{w}}\|_2^2 \\ &\leq \sum_{i=1}^n (u^i - \langle \tilde{\mathbf{w}}, \varphi_K(\mathbf{z}^i)\rangle)^2 + \lambda \cdot \|\tilde{\mathbf{w}}\|_2^2 \\ &= \sum_{i=1}^n (u^i - f_{(\hat{A},\hat{\mathbf{b}},\hat{c})}(\mathbf{z})(\mathbf{z}^i))^2 + \lambda \cdot \|\tilde{\mathbf{w}}\|_2^2 \\ &= \min_{(A,\mathbf{b},c)\in\mathbb{R}^{2\times 2}\times\mathbb{R}^2\times\mathbb{R}} \sum_{i=1}^n (u^i - f_{(A,\mathbf{b},c)}(\mathbf{z}^i))^2 + \lambda \cdot \|\tilde{\mathbf{w}}\|_2^2\end{aligned}$$

Since $\|\tilde{\mathbf{w}}\|_2^2$ is a finite quantity, as we take $\lambda \to 0^+$, we also have $\lambda \cdot \|\tilde{\mathbf{w}}\|_2^2 \to 0^+$ which proves the result. Note that this last argument can be made more rigorous. The details are given below but feel free to skip them if you are not interested. Using the equivalence between quadratic models in $\mathbb{R}^2$ and linear models in $\mathcal{H}_K \equiv \mathbb{R}^6$, we can see that $\tilde{\mathbf{w}}$ can be taken to be the solution to the least squares problem in $\mathcal{H}_K$ i.e. if we let $\Phi = [\varphi_K(\mathbf{z}^1), \ldots, \varphi_K(\mathbf{z}^n)] \in \mathbb{R}^{6\times n}$, then

$$\tilde{\mathbf{w}} = \left(\Phi\Phi^\top\right)^\dagger \Phi\mathbf{u},$$

where $A^\dagger$ is the Moore-Penrose pseudoinverse of the square matrix A. If $\Phi = U\Sigma V^\top$ is the SVD of the matrix Φ where $U, \Sigma \in \mathbb{R}^{6\times 6}$ and $V \in \mathbb{R}^{n\times 6}$, then we have (assuming $n > 6$)

$$\tilde{\mathbf{w}} = \left(\Phi\Phi^\top\right)^\dagger \Phi\mathbf{u} = U\Sigma^{-1}V^\top\mathbf{u},$$

where in the term Σ^{-1}, any zero diagonal entries in Σ are retained as zero and non-zero entries are inverted. This gives us $\|\tilde{\mathbf{w}}\|_2 \leq \frac{1}{\sigma_{\min}} \|\mathbf{u}\|_2$ where $\sigma_{\min}$ is the smallest non-zero diagonal entry in Σ. This shows us that $\|\tilde{\mathbf{w}}\|_2$ is a bounded quantity and we will have $\lambda \cdot \|\tilde{\mathbf{w}}\|_2^2 \to 0^+$ as $\lambda \to 0^+$.

Problem 3.5 (Why PCA does Mean-centering)**.** Recall that we advocated a mean-centering preprocessing step to ensure optimal performance for PCA and PPCA routines. Lets see why is it that the mean is chosen to center. Suppose the low-dimensional latent factors are generated as

$$\mathbb{P}[\mathbf{z}] = \mathcal{N}(\mathbf{0}, I_k) \in \mathbb{R}^k,$$

whereupon an affine transformation is applied to them and noise is added to produce the observed data point, i.e. for $W \in \mathbb{R}^{d\times k}, \boldsymbol{\mu} \in \mathbb{R}^d, \sigma \geq 0$

$$\mathbb{P}[\mathbf{x} \,|\, \mathbf{z}] = \mathcal{N}(\mathbf{x} \,|\, W\mathbf{z} + \mu, \sigma^2 \cdot I_d) \in \mathbb{R}^d.$$

Note that the transformation is affine $W\mathbf{z}^i + \mu$ instead of linear in this example. Now using conjugacy properties of the Gaussian (see [**BIS**] Chapter 12), we can show that

$$\mathbb{P}[\mathbf{x}] = \int_{\mathbf{z}} \mathbb{P}[\mathbf{x} \,|\, \mathbf{z}]\, \mathbb{P}[\mathbf{z}]\ d\mathbf{z} = \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, C),$$

where $C = WW^\top + \sigma^2 \cdot I_d$. For a dataset $X = \left[\mathbf{x}^1, \ldots, \mathbf{x}^n\right]$, write down the complete expression for the data log-likelihood $\mathbb{P}[X \,|\, \boldsymbol{\mu}, W, \sigma]$. Do not ignore constants in your expression. Then derive an expression for $\boldsymbol{\mu}^{\text{MLE}} = \arg\max_{\boldsymbol{\mu}\in\mathbb{R}^d} \mathbb{P}[X \,|\, \boldsymbol{\mu}, W, \sigma]$. Show all steps. (3+7=10 marks)

Solution. The point likelihood expression is simply that of a multivariate normal as is given in the problem statement

$$\mathbb{P}\left[\mathbf{x}^i \mid \boldsymbol{\mu}, W, \sigma\right] = \frac{1}{\sqrt{(2\pi)^d \left|C\right|}} \exp\left(-\frac{1}{2}(\mathbf{x}^i - \boldsymbol{\mu})^\top C^{-1}(\mathbf{x}^i - \boldsymbol{\mu})\right)$$

The data log-likelihood expression thus becomes

$$\begin{aligned}\log \mathbb{P}\left[X \mid \boldsymbol{\mu}, W, \sigma\right] &= \sum_{i=1}^n \log \mathbb{P}\left[\mathbf{x}^i \mid \boldsymbol{\mu}, W, \sigma\right] \\ &= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\left|C\right| - \frac{1}{2}\sum_{i=1}^n (\mathbf{x}^i - \boldsymbol{\mu})^\top C^{-1}(\mathbf{x}^i - \boldsymbol{\mu})\end{aligned}$$

Relegating portions of this expression that do not depend on $\boldsymbol{\mu}$ we get

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^d}{\arg\max}\ \mathbb{P}\left[X \mid \boldsymbol{\mu}, W, \sigma\right] = \underset{\boldsymbol{\mu}\in\mathbb{R}^d}{\arg\min}\ n \cdot \boldsymbol{\mu}^\top C^{-1}\boldsymbol{\mu} - 2\boldsymbol{\mu}^\top C^{-1}\left(\sum_{i=1}^n \mathbf{x}^i\right),$$

where we have used the fact that since C is symmetric, $\boldsymbol{\mu}^\top C^{-1}\mathbf{x}^i = (\mathbf{x}^i)^\top C^{-1}\boldsymbol{\mu}$. Using first order optimality tells us that at the optima we must have

$$2n \cdot C^{-1}\boldsymbol{\mu} - 2C^{-1}\left(\sum_{i=1}^n \mathbf{x}^i\right) = \mathbf{0}$$

Since C^{-1} is invertible (with inverse C), we get a unique solution

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}^i$$