Take-Home Mid-Term Examination

BY YALIN LIAO

Solution to 1.

(a) No. If yes, $J^*(p)=0$ when $p=d$. In this case, the optimal linear mapping is the idenity function over the dataset. This does not make sense for any learning tasks. In image denoising tasks, we could split the data into training and testing dataset. We regard p as a hyper-parameter and use the testing reconstruction error to select p. If the training reconstruction error is very small but the testing error is relatively large, we have the overfitting issue. Then we decrease p's value. If the training error is large, we have the underfitting isue and thus should increase the value of p. Finally, we will obtain the right p, at which both training and testing error are relatively samll.

(b) $p \geq d$ can achieve $J^*(p)=0$. If $V=\begin{bmatrix} U \\ 0 \end{bmatrix}$ or $V^T=\begin{bmatrix} U^T & 0 \end{bmatrix}$, then

$$\begin{aligned} \|\begin{bmatrix} (Ux_i)^T & 1 \end{bmatrix} V - x_i^T\|_2^2 &= \left\| V^T \begin{bmatrix} Ux_i \\ 1 \end{bmatrix} - x_i \right\|_2^2 \\ &= \left\| \begin{bmatrix} U^T & 0 \end{bmatrix} \begin{bmatrix} Ux_i \\ 1 \end{bmatrix} - x_i \right\|_2^2 \\ &= \|U^T U x_i - x_i\|_2^2 \end{aligned}$$

So further if U is semi-orthogonal, i.e., $U^T U = I_d$ then $J^*(p)=0$. Note $U^T U = I_d$ implies

$$\text{Rank}(U) \geq \text{Rank}(UU^T) = \text{Rank}(I_d) = d$$

Also, $\text{Rank}(U) \leq \min\{p,d\} \leq p$. Therefore, $p \geq d$.

(c) If $p=0$ then terms Ux_i disappear and $V \in \mathbb{R}^{1 \times d}$. Further,

$$\begin{aligned} J(U,V) &= \sum_{i=1}^n \|V - x_i^T\|_2^2 \\ &= \sum_{i=1}^n VV^T - 2\sum_{i=1}^n Vx_i^T + \sum_{i=1}^n x_i^T x_i \\ &= nVV^T - 2V\sum_{i=1}^n x_i^T + \sum_{i=1}^n x_i^T x_i \end{aligned}$$

which is quadratic with repsect to V. Then (the gradient is a column vector)

$$\begin{aligned} \frac{\partial J}{\partial V} &= 2nV^T - 2\sum_{i=1}^n x_i \\ \frac{\partial^2 J}{\partial V^2} &= 2nI \succnsim 0 \end{aligned}$$

Since $\frac{\partial^2 J}{\partial V^2}$ is strictly positive definite, $J(U,V)$ admits a unique minimum. Let $\frac{\partial J}{\partial V}=0$. Then

$$V = \frac{\sum_{i=1}^n x_i}{n}$$

So when V is equal to the sample mean $J(U,V)$ achieves its minimum.

(d) If $p=1$ then $U \in \mathbb{R}^{1\times d}$ and $V \in \mathbb{R}^{2\times d}$. Let $V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1d} \\ v_{21} & v_{22} & \cdots & v_{2d} \end{bmatrix}$. Then

$$\begin{aligned}
J(U,V) &= \sum_{i=1}^{n} \left\| \begin{bmatrix} x_i^T U^T & 1 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1d} \\ v_{21} & v_{22} & \cdots & v_{2d} \end{bmatrix} - x_i^T \right\|_2^2 \\
&= \sum_{i=1}^{n} \left\| \begin{bmatrix} x_i^T U^T v_{11} + v_{21} & \cdots & x_i^T U^T v_{1d} + v_{2d} \end{bmatrix} - \begin{bmatrix} x_{i1} & \cdots & x_{id} \end{bmatrix} \right\|_2^2 \\
&= \sum_{i=1}^{n} [(x_i^T U^T v_{11} + v_{21} - x_{i1})^2 + \cdots + (x_i^T U^T v_{1d} + v_{2d} - x_{id})^2] \\
&= \sum_{i=1}^{n} (x_i^T U^T v_{11} + v_{21} - x_{i1})^2 + \sum_{i=1}^{n} (x_i^T U^T v_{1d} + v_{2d} - x_{id})^2
\end{aligned}$$

where x_{ij} is the j-th component of x_i. To minimize $J(U,V)$ with repsect to V, we could do it by optimizing

$$J_1 = \sum_{i=1}^{n} (x_i^T U^T v_{11} + v_{21} - x_{i1})^2$$

since only J_1 contains v_{11} and v_{21} in $J(U,V)$. Calculate partial derivatives:

$$\begin{aligned}
\frac{\partial J_1}{\partial v_{11}} &= 2\sum_{i=1}^{n} (x_i^T U^T v_{11} + v_{21} - x_{i1}) x_i^T U^T \\
&= 2\sum_{i=1}^{n} x_i^T U^T v_{11} x_i^T U^T + 2\sum_{i=1}^{n} v_{21} x_i^T U^T - 2\sum_{i=1}^{n} x_{i1} x_i^T U^T \\
&= 2\left[\sum_{i=1}^{n} (U x_i)^2\right] v_{11} + 2\left[\sum_{i=1}^{n} U x_i\right] v_{21} - 2\sum_{i=1}^{n} U x_i x_{i1}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial J_1}{\partial v_{21}} &= 2\sum_{i=1}^{n} (x_i^T U^T v_{11} + v_{21} - x_{i1}) \\
&= 2\left[\sum_{i=1}^{n} U x_i\right] v_{11} + 2 n v_{21} - 2\sum_{i=1}^{n} x_{i1}
\end{aligned}$$

Let $\frac{\partial J_1}{\partial v_{11}} = 0$ and $\frac{\partial J_1}{\partial v_{21}} = 0$. Then we obtain

$$\begin{aligned}
v_{11} &= \frac{n\sum_{i=1}^{n} U x_i x_{i1} - (\sum_{i=1}^{n} U x_i) \sum_{i=1}^{n} x_{i1}}{n\sum_{i=1}^{n} (U x_i)^2 - (\sum_{i=1}^{n} U x_i)^2} \\
v_{21} &= \frac{[\sum_{i=1}^{n} (U x_i)^2](\sum_{i=1}^{n} x_{i1}) - (\sum_{i=1}^{n} U x_i)(\sum_{i=1}^{n} U x_i x_{i1})}{n\sum_{i=1}^{n} (U x_i)^2 - (\sum_{i=1}^{n} U x_i)^2}
\end{aligned}$$

Solution to 2.

In decesion trees, the main node is referred to as the parent node, whereas sub-nodes are known as child nodes. We can use information gain to determine how good the splitting of nodes in a decision tree. The information gain is defined as

$$I_{\text{gain}} = E_{\text{parent}} - E_{\text{children}}$$

The more the entropy removed, the greater the information gain. The higher the information gain, the better the split. Learning a decision is realized by maximizing the information gain.

Solution to 3

From the optimization perspective, we could choose linear models, like SVM or kernel SVM. SVM is suitable for small-sample-size learning tasks as its model complexity is low (as it only depends on support vectors). Moreover, kernel SVM can handle nonlinear problems.

Based on randomization method, I thought of Random Fourier feature embedding. It projects the training data points to a lower dimensional space (the mapping is stochastic) and meanwhile the low-dimensional representation is meaningful, which approximates a shift-invariant kernel function. Then train a linear model on the low-dimensional representations.

Solution to 4

A kernel regression model is defined as

$$y = \langle w, \phi(x) \rangle$$

where w is the model parameter and the feature mapping $\phi(x_i)$ is deterined by the radial baiss function kernel $K(x, y)$, i.e., $\phi(x) = K(x, \cdot)$. Since the function $y = \langle w, \phi(x) \rangle$ is linear with respect to the model parameter w, the kernel regression is a linear model. However, the function $y = \langle w, \phi(x) \rangle$ is nonlinear with respect to the input x as $\phi(x)$ is nonlinear with respect to x. So kernel regression is a linear model but can provide a nonlinear prediction.

Solution to 5

Kernel SVM only needs a small subset of training points (the support vectors) to define the classification rule, making it often more memory efficient and less computationally demanding when predicting the class of a new observation. In contrast, KNN typically requires higher computation and memory resources because it needs to use all input variables and training samples for each new observation to be classified. Also, Kernel SVM can SVM can achieve good prediction accuracy for new observations in high dimensional space, especially when data dimension is far larger than the number of data points. By contrast, the classification performance of kNN rapidly decreases in handling high-dimensional data points as many input features may be unrelated to the classification or contribute only small amounts of information and thus the nearest neighbors may be defined by irrelevant variables. This is so-called curse of dimensionality.

Solution to 6

Kernel SVM can produce non-linear classifier boundary while linear SVM cannot. Thus, when the target classification problem is non-linear, linear SVM cannot realize good classification performance. One disadvantage of kernel SVM is its higher model complexity compared to linear SVM, so it is easier to overfit trainin dataset than linear SVM. Also, kernel SVM loses the model interpretation, which linear SVM does not.

Solution to 7

Although SVMs (support vector machines) can approximate any function or decision boundary arbitrarily well with enough training data, they scale poorly with the size of the training dataset for computing Gram matrix of the data. The random Fourier feature embedding maps the data into a lower dimensional space and the embedding itself is nonlinear and meaningful, which approximates a shift-invariant kernel. Then when learning linear regressions, we only need to handle the covariance matrix of the training dataset (independent of the size of training data), which is a small-size matrix.

Solution to 8

Nystrom low-rank approximation depends crucially on the quantization error induced by encoding the sample set with the landmark points. The landmark points are chosen as the k-means cluster centers, which finds a local minimum of the quantization error.

Solution to 9

The objective for multi-learning task has the form

$$\min_{W,C} \sum_{t=1}^{T} L(W_t, C_t | X_t, Y_t) + \lambda_1 \Omega(W) + \lambda_2 \|W\|_F^2$$

where $L(\cdot)$ is the loss function, $C = (C_1, \cdots, C_T)$ is the cost function for all tasks, $W = (W_1, \cdots, W_T)$ is the model paramter for all tasks, $\Omega(\cdot)$ is the cross-task regularization for for knowledge transfer, and $\|W\|_F^2$ is used for improving the generalization. The cross-task regularization $\Omega(\cdot)$ could be $L1$ norm for sparse structure, or trace norm (nuclear norm) for low-rank structure.

Solution to 10

Mann-Whitney Test

Solution to 11

Pearson Chi-Squared Test is popular for testing the independence between categorical random variables. Let$X$ and Y denote the discrete random variables, π represent their joint pmf, and $\pi_{i.}$ and $\pi_{.j}$ stand for its marginals. The null hypothesis is H_0 : There is no association between X and Y, i.e., the two categorical variables are independent. The Pearson χ^2 test statistic is constructed as follows:

$$Q_P = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \frac{(\text{observed}_{ij} - \text{expected}_{ij})^2}{\text{expected}_{ij}}$$

where $\text{observed}_{ij} = n_{ij}$ is the observed count of $X = i$ and $Y = j$ in the dataset and the $\text{expected}_{ij} = np_{i.}p_{.j} \equiv \hat{\mu}_{ij}$, which is the MLE of $\mu_{ij} = \mathbb{E}[n_{ij}] = n\pi_{i.}\pi_{j.}$ under H_0, the assumed independent model. It is easy to show that under H_0, the MLEs of $\pi_{i.}$ and $\pi_{.j}$ are

$$p_{i.} = \frac{n_{i.}}{n} \text{ and } p_{j.} = \frac{n_{j.}}{n}$$

where n is the total sample size and $n_{i.}$ is the observation of $X = i$. Thus, the Pearson χ^2 test statistic reduces to

$$Q_P = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \frac{(n_{ij} - \hat{\mu}_{ij})^2}{\hat{\mu}_{ij}}$$

where $\hat{\mu}_{ij} = \frac{n_{i.}n_{.j}}{n}$. Asymptotically, we have

$$Q_P \sim^{\text{approx.}} \chi_1^2$$

assuming all n_{ij}'s are large ($n_{ij} > 5$).

Solution to 12

No. Consider $X\sim\mathcal{N}(0,1)$ and $Y\sim U[-\sqrt{3},\sqrt{3}]$. Then $\mathbb{E}[Y]=0=\mathbb{E}[X]$ and $\mathrm{Var}(Y)=\frac{(2\sqrt{3})^2}{12}=1=\mathrm{Var}(X)$ but obvisouly X and Y have different probability density functions.

Solution to 13

Yes. Assume that X and Y share the same cummulative distribution $F(x)$. Then their density functions $p_X(x)$ and $p_Y(x)$ should be equal since $p_X(x)=F'(x)=p_Y(y)$ except for points where $F(x)$ is not differentiable (the set of points where density functions are not equal is a zero measure set). So X and Y have same probability distribution. But note this does not say X and Y are identical random variable.

Remark. We don't need to distinguish $X\sim U(0,1)$ and $Y\sim U[0,1]$ even though X and Y share same cdf and $p_X(0)=0\neq 1=p_Y(0)$.

Solution to 14

(1) Sliced Wasserstein distance

Randomly project data onto the real line; compute the Wasserstein distance between the two projected input distributions (empirical distribution) via the closed-form formula (or sorting data approach); take the average or maximum over all projections as an alternative to the original Wasserstein distance.

(2) Entropy-regularized Wasserstien distance

Regularize Wasserstein distance by adding the negative entropy of the joint couple potentials; compute the entropy-regularized Wasserstien distance efficiently via the so called Sinkhorn algorithm.