Final Examination–2020

ORFE/FIN 525: Statistical Foundations of Data Science

May 15, 2020

Instructions: The exam is open book and open notes, and lasts 24 hours exactly. Work independently in accordance with the honor codes. You may use any results in the notes, homework problems, and our textbook. You can also use any results in the subproblems before the subproblems that you intend to solve.

1. **Real data project** (31%)
   In this problem we work with the data already used in Problem 6 of Homework 3, but this time instead of extracting features we use raw pixels. If you can not run a package, write down the r-command for the problem to get a partial credit.

   (a) (6%) Download the image data `pictures.zip` from the exam folder (or from the instructor's website), and write R-code to read the images. Set random seed to 525: `set.seed(525)`. Randomly shuffle the data and split into training set of size 800 and testing set of size 200. If you do not know how to do this part, we also provide the data matrix in the exam folder: the matrix `X.all` of size 1000×15360 contains vectorized pixels of 1000 pictures, the vector `Y.all` provides 1000 associated labels from $\{0, 1\}$. You can download data there and start working on the problem, but you will miss some points on this part, depending how much you solve this problem.

   (b) (6%) Use SVM with Gaussian kernel to perform classification. Report the out-of-sample misclassification rate. **Hint**: `svm` function in r-package **e1071**.

   (c) (6%) Use Random Forest to perform classification. Report the out-of-sample misclassification rate. **Hint**: `RandomForest` function in r-package **RandomForest**.

   (d) (6%) Forget about the train-test split and apply K-Means clustering (with maximum number of iterations 1000) to `X.all` with number of clusters set to 2. Let $\widehat{y}_i \in \{0, 1\}$ be the label of the cluster to which K-Means assigns i-th picture and y_i be the true label of i-th picture from `Y.all`. Compute and report

   $$\min\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{\widehat{y}_i = y_i\},\ \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{\widehat{y}_i \neq y_i\}\right), \quad \text{where } n = 1000.$$

   Note that this problem has multiple local minima. Run at least five time and report the best one.

   (e) (7%) Use Convolutional Neural Net to perform classification. For this, instead of `X.all` you have to form `X.all.cnn` as a tensor of proper dimensions $1000 \times 160 \times 96 \times 1$ (use function `to.tensor`), and split it into training and testing set similarly to `X.all`. An example of how to work with library 'keras' in R can be found at `tensorflow.rstudio.com/tutorials/advanced/images/cnn/`. The suggested architecture is: two hidden layers both consisting of 16 filters of size 5×5 and 3×3, respectively, with ReLU activation and 2D 2×2 max pooling, and the final layer is the softmax output. Train your NN using SGD with learning rate 0.01 for 100 epochs with batch size 32 and 20% validation split. Report the out-of-sample misclassification rate.

2. (21%) **Basic concepts and understanding.** Answer the following questions briefly.

(a) What does the "curse of dimensionality" mean? Give two specific models where the "curse of dimensionality" is AVOIDED.

(b) Can the k-means algorithm be regarded as the limit of the EM-algorithm? If so, what the model is it for the EM-algorithm to solve?

(c) How many parameters are used in the following FNN? How many hidden layers? How many nodes are there in the output layer? Is this network for classification or regression? Show your work.

```
layer_dense(units = 200, input_shape = c(500)) %>%
layer_activation(activation = 'relu') %>%
layer_dropout(rate = 0.1) %>%
layer_dense(units = 100) %>%
layer_activation(activation = 'relu') %>%
layer_dropout(rate = 0.2) %>%
layer_dense(units = 200) %>%
layer_activation(activation = 'relu') %>%
layer_dropout(rate = 0.2) %>%
layer_dense(units = 6) %>%
layer_activation(activation = 'softmax')
```

(d) What is the RMSprop? Why do we prefer this method over vanilla SDG?

(e) How do we separate approximately the factor part from the idiosyncratic component? What assumptions make this possible?

(f) Show the idea of factor adjustments for model selection in the regression model using SCAD. Give an advantage over the model selection without adjustments.

(g) Let $\mathbf{\Omega} = (\omega_{ij})$ be the precision matrix of $\mathbf{X} \sim N(0, \mathbf{\Omega}^{-1})$. What does ω_{ij} represent? The precision matrix of five Gaussian random variables is given below. Construct its corresponding graph.

```
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,]  3.07 -0.77   0     0     0
[2,] -0.77  4.93   0   -1.68   0
[3,]    0     0  2.98   0     0
[4,]    0   -1.68  0   3.20  0.69
[5,]    0    0     0   0.69  2.09
```

3. **Risk approxiations.** (16%)
Suppose that we have a random sample $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ from a population $(\mathbf{X}, Y)$. Let $f(\mathbf{X})$ be a predictor of Y with loss $\ell(f(\mathbf{X}), Y)$ for a given loss function $\ell(\cdot, \cdot)$ (e.g. $\ell(u, v) = (u - v)^2$). The empirical and expected risks (the latter also called generalization error) are defined respectively as

$$R_n(f) = n^{-1} \sum_{i=1}^{n} \ell(f(\mathbf{X}_i), Y_i) \quad \text{and} \quad R(f) = E\ell(f(\mathbf{X}), Y).$$

Let $\mathcal{F}$ be a given class of decision functions and $\widehat{f}_n$ and f^* be respectively the empirical optimal and theoretical optimal solutions:

$$\widehat{f}_n = \text{argmin}_{f \in \mathcal{F}} R_n(f) \quad \text{and} \quad f^* = \text{argmin}_{f \in \mathcal{F}} R(f).$$

Denote by $\Delta_n = \sup_{f \in \mathcal{F}} |R_n(f) - R(f)|$.

(a) Show the generalization error $R(\widehat{f}_n) \le R_n(\widehat{f}_n) + \Delta_n$, which indicates that generalization error is controlled by the empirical one so long as Δ_n is small.

(b) Show that $R(\widehat{f}_n) \le R(f^*) + 2\Delta_n$, which deomostrates that the generalization error is close to the theoretical optimal once Δ_n is small.

(c) For classification problem, let us take the 0-1 loss $\ell(u, v) = I(u \neq v)$. What are $R(f)$ and $R_n(f)$?

(d) Now consider the class $\mathcal{F} = \{f : f(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\beta}, \forall \|\boldsymbol{\beta}\|_1 \le c\}$ for a given constant c and the quadratic loss. Show that $\Delta_n \le \|\mathbf{S}_n - \boldsymbol{\Sigma}\|_{\max}(1+c)^2$, where

$$\mathbf{S}_n = n^{-1}\sum_{i=1}^n \begin{pmatrix}\mathbf{X}_i \\ Y_i\end{pmatrix}\begin{pmatrix}\mathbf{X}_i \\ Y_i\end{pmatrix}^\top \qquad \text{and} \qquad \boldsymbol{\Sigma} = E\left[\begin{pmatrix}\mathbf{X} \\ Y\end{pmatrix}\begin{pmatrix}\mathbf{X} \\ Y\end{pmatrix}^\top\right].$$

4. **Classification** (16%)
Consider the binary classification problem $(\mathbf{X}|Y = j) \sim f_j(\mathbf{x})$ $(j = 0, 1)$, namely, the classes 0 and 1 have densities $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$, respectively.

(a) Consider an ensemble of classifiers $\{\widehat{\delta}_j(\mathbf{x})\}_{j=1}^m$ based on a random sample $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$. Suppose that we wish to combine them by using logistic regression $\beta_0 + \sum_{j=1}^m \beta_j \widehat{\delta}_j(\mathbf{x})$. Show how you would determine the coefficients $\beta_0, \ldots, \beta_m$ for the given data $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ and what the emsemble classifier is.

(b) Suppose that we have $\mathbf{X}_1, \ldots, \mathbf{X}_n \sim N(\boldsymbol{\mu}_0, \mathbf{I}_p)$ and $\mathbf{Y}_1, \ldots, \mathbf{Y}_n \sim N(\boldsymbol{\mu}_1, \mathbf{I}_p)$. Then, the Fisher discriminator is $I(\boldsymbol{\mu}_d^T(\mathbf{x} - \boldsymbol{\mu}_a) > 0)$ where $\boldsymbol{\mu}_a = (\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)/2$ and $\boldsymbol{\mu}_d = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. In practice, we use the estimates $\widehat{\boldsymbol{\mu}}_a$ and $\widehat{\boldsymbol{\mu}}_d$ by substituting the sample means. According to Problem 1(a), Homework #4, the misclassification rate for class 1 is $\Phi\Big(\frac{\widehat{\boldsymbol{\mu}}_d^T(\widehat{\boldsymbol{\mu}}_a - \boldsymbol{\mu}_1)}{\|\widehat{\boldsymbol{\mu}}_d\|}\Big)$. Show that $\frac{E\widehat{\boldsymbol{\mu}}_d^T(\widehat{\boldsymbol{\mu}}_a - \boldsymbol{\mu}_1)}{[E\|\widehat{\boldsymbol{\mu}}_d\|^2]^{1/2}} = -\frac{\|\boldsymbol{\mu}_d\|^2/2}{[\|\boldsymbol{\mu}_d\|^2 + 2p/n]^{1/2}}$. Explain briefly how this result shows the adverse impact of dimensionality in classification. If you can impose reasonable conditions to show that $\frac{\widehat{\boldsymbol{\mu}}_d^T(\widehat{\boldsymbol{\mu}}_a - \boldsymbol{\mu}_1)}{\|\boldsymbol{\mu}_d\|^2} \to -\frac{1}{2}$ in probability, you will get additional 3 bonus points. **Hint:** You can use $\widehat{\boldsymbol{\mu}}_d$ and $\widehat{\boldsymbol{\mu}}_a$ are independent.

(c) Suppose p features of $\mathbf{X}$ are i.i.d., namely $f_j(\mathbf{x}) = \prod_{k=1}^p g_j(x_k)$, $j = 0, 1$. For the Bayes classifier $c(\mathbf{x}) = I(f_1(\mathbf{x}) > f_0(\mathbf{x}))$, show that its misclassification error

$$P_0(f_1(\mathbf{X}) > f_0(\mathbf{X})) \approx \Phi\Big(-\sqrt{p}\mu_p/\sigma_p\Big), \qquad \mu_p = E_{X\sim g_0}\Big(\log\frac{g_0(X)}{g_1(X)}\Big), \quad \sigma_p^2 = \mathrm{var}_{X\sim g_0}\Big(\log\frac{g_0(X)}{g_1(X)}\Big),$$

as $p \to \infty$ under some regularity conditions that make the central limit theorem hold.

(d) It is well known that in the generative adversarial network, when discriminator is unconstrained, we find the generator $g(\cdot)$ to minimize the Jensen-Shanon divergence:

$$\int \Big[f_0(\mathbf{x})\log\frac{f_0(\mathbf{x})}{f_0(\mathbf{x}) + g(\mathbf{x})} + g(\mathbf{x})\log\frac{g(\mathbf{x})}{f_0(\mathbf{x}) + g(\mathbf{x})}\Big]d\mathbf{x},$$

where $f_0(\mathbf{x})$ is true data generting density. Show that the minimizer of the generator $g(\mathbf{x})$ (without additional constraints other than it is a density function) is obtained at $g(\mathbf{x}) = f_0(\mathbf{x})$. **Hint**: Use the constraint $\int g(\mathbf{x})d\mathbf{x} = 1$.

5. **Factor models and their applications** (16%)
Consider the factor model $\mathbf{X} = \mathbf{B}\mathbf{f} + \mathbf{u}$ with $E\mathbf{u} = 0$ and $\mathrm{cov}(\mathbf{f}, \mathbf{u}) = 0$. Let $\{\mathbf{X}_i\}_{i=1}^n$ be a random sample from the model.

(a) Show that $\boldsymbol{\Sigma} = 0.5E(\mathbf{X}_1 - \mathbf{X}_2)(\mathbf{X}_1 - \mathbf{X}_2)^T$ if $\mathbf{X}_1$ and $\mathbf{X}_2$ are two independent copies with covariance matrix $\boldsymbol{\Sigma}$. What is the $(1,2)^{th}$ element of the robust U-statistic $\widehat{\boldsymbol{\Sigma}}_U$?

(b) With $\widehat{\boldsymbol{\Sigma}}_U$ as initial covariance input, show how $n \times K$ matrix $\mathbf{B}$ is estimated.

(c) Due to indeterminacy of $\mathbf{B}$, let us rotate the columns of $\widehat{\mathbf{B}}$ so that it is as close to $\mathbf{B}$ as possible. Show that among all $K \times K$ othornormal matrices $\mathbf{O}$, namely $\mathbf{O}^T\mathbf{O} = \mathbf{I}_K$, the one that minimizes

$$\|\widehat{\mathbf{B}}\mathbf{O} - \mathbf{B}\|_F^2$$

is given by $\widehat{\mathbf{O}} = \mathbf{LR}$, where $\mathbf{L}$ and $\mathbf{R}$ are the left and right singular vectors of $\widehat{\mathbf{B}}^T\mathbf{B}$, i.e. $\widehat{\mathbf{B}}^T\mathbf{B} = \mathbf{L}\boldsymbol{\Lambda}\mathbf{R}$ where $\boldsymbol{\Lambda}$ are the singular values.

(d) The connectivity of networks is very important for community detection, matrix completion, and item ranking. For the Erdös-Rényi model $G(n, p)$, show that the random graph is connected with probability tending to one if $p = a\frac{\log n}{n}$ for the constant $a > 2$ (indeed it holds for $a > 1$ by modifying slightly the argument for $a \in (1, 2]$, but this is not required). What is the expected number of connections for a given node?