Exercises
Machine Learning
HS 2016

Machine Learning Laboratory
Dept. of Computer Science, ETH Zürich
Prof. Joachim M. Buhmann
Web https://ml2.inf.ethz.ch/courses/ml/

Series 1, Oct 6th, 2016 (Probability and Regression)

Email questions 1, 2 to: **Karim Labib** labibk@student.ethz.ch
Email questions 3, 4, 5, 6 to: **Qin Wang** qwang@student.ethz.ch

Solution 1 (Various Problems):

1. a) We use the independence to separate the joint distribution into a product:

$$P(\underbrace{H, \ldots, H}_{n \text{ times}}) = \prod_{i=1}^{n} P(H) = p^n.$$

b) We use the same property in a slightly different way:

$$P(\underbrace{H, \ldots, H}_{n-1 \text{ times}}, T) = P(T) \prod_{i=1}^{n-1} P(H) = p^{n-1}(1-p).$$

2. For independent random variables the following holds true $\mathbb{E}[XY] = \mathbb{E}X \cdot \mathbb{E}Y$, so we proceed with the chain (recall that $\mathbb{E}\mathbb{E}X = \mathbb{E}X$)

$$\begin{aligned}
\mathrm{Cov}\,[X, Y] &= \mathbb{E}(X - \mathbb{E}X)(Y - \mathbb{E}Y) \\
&= \mathbb{E}[XY - X\mathbb{E}Y - Y\mathbb{E}X + \mathbb{E}X\mathbb{E}Y] \\
&= \mathbb{E}[XY] - \mathbb{E}X\mathbb{E}Y - \mathbb{E}Y\mathbb{E}X + \mathbb{E}X\mathbb{E}Y \\
&= 0.
\end{aligned}$$

3. From the tutorial slides, the variance of a random variable X is equal to

$$\begin{aligned}
\mathrm{Var}X &= \int_x (x - \mathbb{E}[X])^2 p(x) dx \\
&= \mathbb{E}[(X - \mathbb{E}[X])^2] \\
&= \mathbb{E}[X^2 - 2X\mathbb{E}[X] + (\mathbb{E}[X]])^2 \\
&= \mathbb{E}[X^2] - 2(\mathbb{E}[X])^2 + (\mathbb{E}[X])^2 \quad \text{(by linearity of expectation)} \\
&= \mathbb{E}[X^2] - (\mathbb{E}[X])^2
\end{aligned}$$

4. a) By linearity of expectation $\mathbb{E}[X + Y] = 2 + 4 = 6$.

b) From problem 3 we know that $\mathrm{Var}X = \mathbb{E}[X^2] - [\mathbb{E}X]^2$. The latter term is 4 from the setting, and the first one is

$$\mathbb{E}[X^2 + Y - Y] = \mathbb{E}[X^2 + Y] - \mathbb{E}Y = 8 - 4 = 4,$$

thus making the variance 0.

Offtopic: you have probably already noticed, that zero variance means that X is equal* to its mean, i.e. 2. This fact, together with the constraint $X^2 + Y = 8$, implies that Y is equal† to 4. So, both random variables were just constants.

*equality almost sure (note for those familiar with such notion, otherwise just read as ordinary equality)
†equality almost sure (note for those familiar with such notion, otherwise just read as ordinary equality)

5. First, we compute mean by definition:

$$\mathbb{E}X = \int_a^b x\, p_{\mathsf{unif}}(x)\, dx = \int_a^b \frac{x}{b-a}\, dx = \frac{a+b}{2}.$$

To compute variance, we first evaluate the second uncentralized moment:

$$\mathbb{E}[X^2] = \int_a^b x^2\, p_{\mathsf{unif}}(x)\, dx = \left[\!\left[\text{left to the reader} \right]\!\right] = \frac{a^2+ab+b^2}{3}.$$

Thus, the variance is

$$\mathrm{Var}X = \mathbb{E}[X^2] - [\mathbb{E}X]^2 = \frac{a^2+ab+b^2}{3} - \left(\frac{a+b}{2}\right)^2 = \frac{(b-a)^2}{12}.$$

6* (Weak Law of Large Numbers). First observe that

$$\mathbb{E}\overline{X} = \frac{1}{n}\mathbb{E}\sum X_i = \frac{1}{n}\cdot n\cdot \mu = \mu.$$

Then, we check the following (recall the i.i.d. and finiteness of variances):

$$\mathrm{Var}\overline{X} = \mathrm{Var}\left[\frac{1}{n}\sum X_i\right] = \frac{1}{n^2}\sum \mathrm{Var}X_i = \frac{n\mathrm{Var}X_1}{n^2} = o(1).$$

Under these observations the Chebychev's inequality turns exactly into the definition of convergence in probability:

$$\mathsf{Pr}\{|\overline{X}-\mu| \geq \varepsilon\} \leq \frac{\mathrm{Var}\overline{X}}{\varepsilon^2} = o(1), \quad (n\to\infty).$$

Solution 2 (Conditional Probability):

1. There are 4 possible outcomes for the two children. Each happens independently with probability $1/2 \cdot 1/2 = 1/4$. The only case with no girls is having two boys. Thus, the probability of having at least one girl is equal to $3/4$.

2. Using same argument as above, probability is equal to $1/4$.

3. P(Both children are girls — First child is girl) $=$ P(second child is a girl) $= 1/2$

4. Let us denote by GG the event of having two girls. $> G$ the event of having at least one girl.

$$\begin{aligned} P(GG \mid > G) &= \frac{P(GG, > G)}{P(> G)} \\ &= \frac{P(GG)}{P(> G)} \\ &= \frac{1/4}{3/4} = 1/3 \end{aligned}$$

5. Let us as before denote by GG the event of having two girls and by $> G$ the event of having at least one girl. Let G denote the event of having exactly one girl and let C be the event of a girl named Cassiopeia. And let a be the probability of having a girl named Cassiopeia.

Thus we want to calculate

$$P(GG| > G, C) = \frac{P(> G, C|GG) \cdot P(GG)}{P(> G, C)} \text{By Bayes' Rule}$$

However,

$$\begin{aligned} P(> G, C|GG) &= P(C| > G, GG) \cdot P(> G|GG) \\ &= P(C| > G, GG) \cdot 1 \\ &= P(C|GG) \end{aligned}$$

Now we have,

$$\begin{aligned} P(GG| > G, C) &= \frac{P(C|GG) \cdot P(GG)}{P(G, C) + P(GG, C)} \\ &= \frac{P(C|GG) \cdot P(GG)}{P(C|G) \cdot P(G) + P(C|GG) \cdot P(GG)} \\ &= \frac{(1 - (1 - a)^2) \cdot 0.25}{a \cdot (2 \cdot 0.5 \cdot 0.5) + (1 - (1 - a)^2) \cdot 0.25} \\ &= \frac{2 - a}{4 - a} \end{aligned}$$

and if we assume that a tends to zero we can see that the probability tends to $1/2$. On a side note, we can see that if the name Cassiopeia is replaced by being a label for a girl instead such that $P(label|G) = 1$, we can see that we end up with probability $1/3$ as the previous problem.

Solution 3 (Regression):

1. $L_{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2 = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$
2. $L_{Ridge}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda \beta^T \beta$
   $L_{LASSO}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda \|\beta\|_1$
3. $L_{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) = \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{X}\beta + \beta^T \mathbf{X}^T \mathbf{X}\beta$
   Take the derivative and set it to zero, assuming non singular $\mathbf{X}^T\mathbf{X}$, then
   $$\frac{\partial L_{RSS}}{\partial \beta} = -2\mathbf{X^T y} + 2\mathbf{X^T X}\beta = 0$$
   $$\hat{\beta}_{RSS} = (\mathbf{X^T X})^{\mathbf{-1}} \mathbf{X^T y}$$
   $L_{Ridge}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda \beta^T \beta = \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{X}\beta + \beta^T \mathbf{X}^T \mathbf{X}\beta + \lambda \beta^T \beta$
   Take the derivative and set it to zero, assuming non singular $\mathbf{X}^T\mathbf{X}$, then
   $$\frac{\partial L_{Ridge}}{\partial \beta} = -2\mathbf{X^T y} + 2\mathbf{X^T X}\beta + 2\lambda \mathbf{I}\beta = 0$$
   $$\hat{\beta}_{Ridge} = (\mathbf{X^T X} + \lambda \mathbf{I})^{\mathbf{-1}} \mathbf{X^T y}$$
4. Recall that a (square) matrix is invertible if and only if it does not have a zero eigenvalue. When the norm of the ratio of the largest eigenvalue to the smallest eigenvalue is large, the inversion is numerically unstable.
5. The ridge regression solution adds a positive constant to the diagonal, which moves all eigenvalues further away from zero, thus improves stability of the inversion.
   λ controls the intensity of the regularization. By increasing λ, we further shrink the regression coefficients and reduce model complexity.

6. Let $\mathbf{W}_{ii} = w^{(i)}$, $\mathbf{W}_{ij} = 0$ for $i \neq j$, let $z = \mathbf{X}\beta - \mathbf{y}$, i.e. $z_i = \beta^T x^{(i)} - y^{(i)}$.
   Then we have:
   $$L_{weighted}(\beta) = \sum_{i=1}^{m} w^{(i)} {z_i}^2 = z^T \mathbf{W} z = (\mathbf{X}\beta - \mathbf{y})^T \mathbf{W}(\mathbf{X}\beta - \mathbf{y})$$
7. $\nabla_\beta L_{weighted}(\beta) = \nabla_\beta(\beta^T \mathbf{X}^T \mathbf{W} \mathbf{X} \beta + \mathbf{y}^T \mathbf{W} \mathbf{y} - 2\mathbf{y}^T \mathbf{W} \mathbf{X} \beta) = 2(\mathbf{X}^T \mathbf{W} \mathbf{X} \beta - \mathbf{X}^T \mathbf{W} \mathbf{y}) = 0.$
   From which we can get a closed form formula for β
   $$\beta = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{y}$$

Solution 4 (Bias and variance):

First let us give some definitions. Bias and variance of an estimator $\hat{\theta}$ are defined by

$$Bias(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$$

$$Var(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$$

We use the notation used in the lecture in order to be consistent. Let us first define the data set

$$Z = \{(x_i, y_i), 1 \leq i \leq n\}$$

estimator

$$\hat{f}(X) = f(X, Z)$$

and squared loss

$$l(X, Y) = (\hat{f}(X) - Y)^2$$

Please note that the estimator depends on both X and Z.

$$\begin{aligned}
\mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - Y)^2] &= \mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - \mathbb{E}_Y[Y|X] + \mathbb{E}_Y[Y|X] - Y)^2] \\
&= \mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - \mathbb{E}_Y[Y|X])^2] + \mathbb{E}_{X,Y}[(\mathbb{E}_Y[Y|X] - Y)^2] \\
&+ 2\mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - \mathbb{E}_Y[Y|X])(\mathbb{E}_Y[Y|X] - Y)]
\end{aligned}$$

$\mathbb{E}_{X,Y}[(\mathbb{E}_Y[Y|X] - Y)^2]$ corresponds to noise, this is inherent to the model, there is nothing you can do. Also the cross term vanishes since $(\hat{f}(X) - \mathbb{E}_Y[Y|X])$ does not depend on Y and we can integrate out the second term $\mathbb{E}_Y[Y|X] - Y$ with respect to Y which gives 0. Now we have

$$\begin{aligned}
\mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - Y)^2] &= \text{noise} + \mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - \mathbb{E}_Y[Y|X])^2] \\
&= \text{noise} + \mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - \mathbb{E}_Z[\hat{f}(X)] + \mathbb{E}_Z[\hat{f}(X)] - \mathbb{E}_Y[Y|X])^2] \\
&= \text{noise} + \mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - \mathbb{E}_Z[\hat{f}(X)])^2] + \mathbb{E}_Z \mathbb{E}_{X,Y}[(\mathbb{E}_Z[\hat{f}(X)] - \mathbb{E}_Y[Y|X])^2] \\
&+ 2\mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - \mathbb{E}_Z[\hat{f}(X)])(\mathbb{E}_Z[\hat{f}(X)] - \mathbb{E}_Y[Y|X])]
\end{aligned}$$

Since the second cross term $\mathbb{E}_Z[\hat{f}(X)] - \mathbb{E}_Y[Y|X]$ does not depend on Z we can integrate out the first term $\hat{f}(X) - \mathbb{E}_Z[\hat{f}(X)]$ with respect to Z which gives 0. Now we have

$$\begin{aligned}
\mathbb{E}_Z \mathbb{E}_{X,Y}[(\hat{f}(X) - Y)^2] &= \text{noise} + \mathbb{E}_Z \mathbb{E}_X[(\hat{f}(X) - \mathbb{E}_Z[\hat{f}(X)])^2] + \mathbb{E}_X[(\mathbb{E}_Z[\hat{f}(X)] - \mathbb{E}_Y[Y|X])^2] \\
&= \text{noise} + \text{variance} + \text{bias}^2
\end{aligned}$$

by integrating out independent variables.

Solution 5 (Combination of Individual Regression Models):

1. Figure 1: $\lambda = 13.5$ (Large λ pulls the weight parameters toward zero)
   Figure 3: $\lambda = 0.09$ (Individual models tend to overfit(high variance))

2. Recall the last slide of Regression Lecture, if we combine different regressors,
   $bias[\hat{f}(x)] = \frac{1}{B}\sum_{i=1}^{B} bias[\hat{f}_i(x)]$
   $\mathbb{V}[\hat{f}(x)] \approx \frac{\sigma^2}{B}$, assuming small covariances and similar variances.
   Figure 1,2: The individual models have high bias and low variance. By combining the models, the bias is averaged, the variance is reduced, leading to a high bias, low variance model.
   Figure 3,4: The individual models have high variance and low bias. By combining the models, we significantly reduce the variance, leading to a low variance and low bias model.

Solution 6 (Python Exercise):

Here we provide some hints for the Python exercise.

1. You can use sklearn.linear model.ElasticNet to do the regression.

```
#Assume we have X_train, y_train, X_vali, y_vali, X_test, y_test.
#X, y are the union of training and validation set
#Notice that in the cost function we define,
#l1_ratio controls the regularization tradeoff between l1 and l2
#l1_ratio=1 corresponds to Lasso penalty

import numpy as np
from sklearn.metrics import mean_squared_error
from sklearn import linear_model

l1_ratios = np.linspace(0.1, 1, 100)
enet = linear_model.ElasticNet()
train_errors = list()
vali_errors = list()
for l1_ratio in l1_ratios:
    enet.set_params(l1_ratio=l1_ratio)
    enet.fit(X_train, y_train)
    train_errors.append(enet.score(X_train, y_train))
    vali_errors.append(enet.score(X_vali, y_vali))

l1_ratio_optim  = l1_ratios[np.argmax(vali_errors)]

# Estimate the coef_ on both training and validation data with optimal regularizatio
enet.set_params(l1_ratio=l1_ratio_optim)
coef_ = enet.fit(X, y).coef_
# Prediction
y_predict = enet.fit(X, y).predict(X_test)
# MSE
mean_squared_error(y_test, y_predict)
```

2. You can also use cross-validation to choose the parameter. Check:
   `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNetCV.html`
   `http://scikit-learn.org/stable/modules/cross_validation.html`

3. Check `http://scikit-learn.org/stable/modules/preprocessing.html` to see how we usually encode categorical features, impute missing values, generate new polynomial features... You may also use Panda Library to get more compact codes.