Machine Learning
Problem Set 1

Ozaner Hansha

September 29, 2020

For the first 2 questions, consider a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$. We then fit the data using linear least squares, i.e. finding the $\mathbf{w}$ that minimizes $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$ which nets us an optimal parameter vector $\mathbf{w}^*$ for use in the model:

$$y = \mathbf{w}^* \cdot \mathbf{x}$$

Note that each $\mathbf{x}_i$ in the data set has its first component equal to 1 (bias). Also note that $\mathbf{w}^*$ in this case (linear least squares) is given by $\mathbf{X}^+\mathbf{y}$.

Question 1

Problem: Consider a new dataset $\mathcal{D}' = \{(\mathbf{x}_i, y_i')\}_{i=1}^n$, where $y_i' = ay_i + b$ for some constants a, b. Fitting this data using linear least squares, we arrive at a new parameter vector $\mathbf{w}'$.

Can $\mathbf{w}'$ be computed directly from $\mathbf{w}^*$ and the constants a, b without looking at the dataset? That is to say, can we express $\mathbf{w}'$ solely in terms of $\mathbf{w}^*, a$, and b? If yes, how? If no, why not?

Solution: No, $\mathbf{w}'$ cannot be expressed without reference to the dataset. To see this, first note that the action on the labels of the dataset $\mathbf{y}$ is equivalent to the following:

$$\mathbf{y}' = a\mathbf{y} + b\mathbf{1} \qquad (\mathbf{1} \text{ is a vector of 1s})$$

Plugging this into the formula for $\mathbf{w}'$:

$$\begin{aligned} w' &= \mathbf{X}^+\mathbf{y}' && \text{(sol. to linear least squares)} \\ &= \mathbf{X}^+(a\mathbf{y} + b\mathbf{1}) && \text{(def. of } \mathbf{y}') \\ &= a\mathbf{X}^+\mathbf{y} + b\mathbf{X}^+\mathbf{1} \\ &= a\mathbf{w}^* + b\mathbf{X}^+\mathbf{1} && \text{(def. of } \mathbf{w}^*) \end{aligned}$$

We are left with the above sum and while we can express the first term referencing only $\mathbf{w}^*$ and a, the second term must make reference to $\mathbf{X}$. This reference is unavoidable as it is impossible to recover $\mathbf{X}$ from just $\mathbf{w}^*$.

And so, we cannot express $\mathbf{w}'$ without reference to the dataset. That said, as the first term shows, we could if there was no constant term b added to each y_i.

Question 2

Problem: Consider a new dataset $\tilde{\mathcal{D}} = \{(\tilde{\mathbf{x}}_i, y_i)\}_{i=1}^n$, where $\tilde{x}_{ij} = c_j x_{ij}$ for some d dimensional constant vector $\mathbf{c}$. Fitting this data using linear least squares, we arrive at a new parameter vector $\tilde{\mathbf{w}}$.

Can $\tilde{\mathbf{w}}$ be computed directly from $\mathbf{w}^*$ and $\mathbf{c}$ without looking at the dataset? That is to say, can we express $\tilde{\mathbf{w}}$ solely in terms of $\mathbf{w}^*$ and $\mathbf{c}$? If yes, how? If no, why not?

Solution: We can indeed express $\mathbf{w}'$ without reference to the dataset. First note that the action on the features of the dataset $\mathbf{X}$ is equivalent to the following product:

$$\begin{aligned} \tilde{\mathbf{X}} &= \mathbf{X}\,\mathrm{diag}(c_1, \cdots, c_d) \\ &= \mathbf{XC} && (\text{let } C = \mathrm{diag}(c_1, \cdots, c_d)) \end{aligned}$$

And so we have the following chain of equalities:

$$\begin{aligned} \tilde{\mathbf{w}} &= \tilde{\mathbf{X}}^+ \mathbf{y} && (\text{sol. to linear least squares}) \\ &= (\mathbf{XC})^+ \mathbf{y} && (\text{def. of } \tilde{\mathbf{X}}) \\ &= \mathbf{C}^+ \mathbf{X}^+ \mathbf{y} && (\text{see note}^1) \\ &= \mathbf{C}^+ \mathbf{w}^* && (\text{def. of } \mathbf{w}^*) \\ &= \mathbf{C}^{-1} \mathbf{w}^* && (\text{diagonal matrix is invertible}) \\ &= \mathrm{diag}(1/c_1, \cdots, 1/c_d)\mathbf{w}^* && (\text{inverse of diagonal matrix}) \end{aligned}$$

[1] *The property* $(\mathbf{AB})^+ = \mathbf{B}^+\mathbf{A}^+$ *only holds for two matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *if they are of full rank. In this case,* $\mathbf{C}$ *is clearly full rank since it's a diagonal matrix, and we assume that* $\mathbf{X}$ *is also full rank. This is a reasonable assumption as it is increasingly unlikely that the rows of a dataset do not span the full rank as the number of, supposedly i.i.d., samples increase.*

And so each element of the new parameter vector $\tilde{\mathbf{w}}$ is given by:

$$\tilde{w}_i = \frac{w_i^*}{c_i}$$

Question 3

Problem: Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ There is a closed form solution to the optimal MLE estimator of the following model:

$$y = \mathbf{w} \cdot \mathbf{x} + v$$

Where the noise $v \sim \mathcal{N}(0, \sigma^2)$ for some σ^2 that is constant for all recorded samples.

Is there a closed form solution to the optimal MLE estimator of that same model but where the variance of the noise for each sample $\mathbf{x}_i$ is different? The variances are given by $\boldsymbol{\sigma}^{\mathbf{2}} = (\sigma_1^2, \cdots, \sigma_n^2)$.

Solution: There is a closed form solution, and its derivation is similar to that of the homoscedastic error model. Before we produce the closed form solution, we define the following matrix $\mathbf{C}$:

$$\mathbf{C} = \mathrm{diag}(1/\sigma_1^2, \cdots, 1/\sigma_n^2)$$

Also note that, rather than considering v a normal RV with mean 0, we can consider each $Y_i \mid \mathbf{x}_i$ a RV with the following distribution:

$$Y_i \mid \mathbf{x}_i \sim \mathcal{N}(\mathbf{w} \cdot \mathbf{x}_i, \sigma_i^2)$$

Where each label y_i is a realization of $Y_i \mid \mathbf{x}_i$.

With these two facts in mind, we now find the paramater vector $\mathbf{w}^*$ that maximizes the likelihood of the observed data under the given linear model:

$$\begin{aligned}
\mathbf{w}^* &= \arg\max_{\mathbf{w}} p(\mathbf{y} \mid \mathbf{X}; \mathbf{w}, \boldsymbol{\sigma^2}) && \text{(def. of optimal MLE parameter)} \\
&= \arg\max_{\mathbf{w}} \prod_{i=1}^{n} p(y_i \mid \mathbf{x}_i; \mathbf{w}, \sigma_i^2) && (Y_i \text{ are independent RVs}) \\
&= \arg\max_{\mathbf{w}} \log \prod_{i=1}^{n} p(y_i \mid \mathbf{x}_i; \mathbf{w}, \sigma_i^2) && \text{(log is monotone increasing)} \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \log p(y_i \mid \mathbf{x}_i; \mathbf{w}, \sigma_i^2) \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \log \left(\frac{1}{\sigma_i^2 \sqrt{2\pi}} e^{-\frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)}{2\sigma_i^2}} \right) && (Y_i \text{ is a normal RV}) \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} -\log(\sigma_i^2 \sqrt{2\pi}) - \frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}{2\sigma_i^2} \\
&= \arg\max_{\mathbf{w}} -\frac{1}{2} \sum_{i=1}^{n} \frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}{\sigma_i^2} && (\text{red term independent of } \mathbf{w})
\end{aligned}$$

Now to find $\mathbf{w}$ we can simply set the derivative of the expression we are trying to maximize equal to $\mathbf{0}$, as quadratic polynomials have a single maximum:

$$\begin{aligned}
\mathbf{0} &= \frac{\partial}{\partial \mathbf{w}} - \frac{1}{2} \sum_{i=1}^{n} \frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}{\sigma_i^2} \\
&= \frac{\partial}{\partial \mathbf{w}} \sum_{i=1}^{n} \frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}{\sigma_i^2} \\
&= \frac{\partial}{\partial \mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top \mathbf{C} (\mathbf{y} - \mathbf{X}\mathbf{w}) && (\mathbf{C} = \text{diag}(1/\sigma_1^2, \cdots, 1/\sigma_n^2)) \\
&= \frac{\partial}{\partial \mathbf{w}} (\mathbf{y}^\top - \mathbf{w}^\top \mathbf{X}^\top) \mathbf{C} (\mathbf{y} - \mathbf{X}\mathbf{w}) \\
&= \frac{\partial}{\partial \mathbf{w}} \mathbf{y}^\top \mathbf{C}\mathbf{y} - \mathbf{w}^\top \mathbf{X}^\top \mathbf{C}\mathbf{y} - \mathbf{y}^\top \mathbf{C}\mathbf{X}\mathbf{w} + \mathbf{w}^\top \mathbf{X}^\top \mathbf{C}\mathbf{X}\mathbf{w} \\
&= -\mathbf{X}^\top \mathbf{C}\mathbf{y} - (\mathbf{y}^\top \mathbf{C}\mathbf{X})^\top + 2\mathbf{X}^\top \mathbf{C}\mathbf{X}\mathbf{w} \\
&= -\mathbf{X}^\top \mathbf{C}\mathbf{y} - \mathbf{X}^\top \mathbf{C}\mathbf{y} + 2\mathbf{X}^\top \mathbf{C}\mathbf{X}\mathbf{w} && (\mathbf{C} \text{ is symmetric}) \\
&= \mathbf{X}^\top \mathbf{C}\mathbf{y} - \mathbf{X}^\top \mathbf{C}\mathbf{X}\mathbf{w} \\
\mathbf{X}^\top \mathbf{C}\mathbf{X}\mathbf{w} &= \mathbf{X}^\top \mathbf{C}\mathbf{y} \\
\mathbf{w} &= (\mathbf{X}^\top \mathbf{C}\mathbf{X})^{-1} \mathbf{X}^\top \mathbf{C}\mathbf{y} && (\mathbf{X} \text{ is full rank})
\end{aligned}$$

And so the MLE estimator is given by:

$$\begin{aligned}
y &= \mathbf{w}^* \cdot \mathbf{x} \\
&= ((\mathbf{X}^\top \mathbf{C}\mathbf{X})^{-1} \mathbf{X}^\top \mathbf{C}\mathbf{y}) \cdot \mathbf{x}
\end{aligned}$$

Note that this is identical to the weighted least squares (WLS) estimator with a weight matrix of $\mathbf{C}$.

Question 4

Problem: After filling in the missing code for the gradient descent function, choose a polynomial model that best fits the population based off its RMSE on the validation set.

Solution: Based on the RMSE, the best polynomial model I found was the sixth order one. That is, the feature transformation $\boldsymbol{\phi}$ given by:

$$\boldsymbol{\phi}_6\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}\right) = \begin{bmatrix} 1 \\ x_1 \\ x_1^2 \\ \vdots \\ x_1^6 \\ x_2 \\ x_2^2 \\ \vdots \\ x_m^5 \\ x_m^6 \end{bmatrix}$$

My methodology for determining this was to to compare the RMSE of increasingly large polynomial models until the error found a local minima. In this case it was at sixth degree features:

```
Model (0,): train RMSE 16.0295, val RMSE 17.1968
Model (0, 1): train RMSE 14.4201, val RMSE 15.3118
Model (0, 1, 2): train RMSE 14.2399, val RMSE 14.8646
Model (0, 1, 2, 3): train RMSE 14.1662, val RMSE 14.6580
Model range(0, 5): train RMSE 14.1187, val RMSE 14.5578
Model range(0, 6): train RMSE 14.0804, val RMSE 14.5141
Model range(0, 7): train RMSE 14.0494, val RMSE 14.5032
Model range(0, 8): train RMSE 14.0253, val RMSE 14.5117
Model range(0, 9): train RMSE 14.0079, val RMSE 14.5314
```

After this, I randomly removed certain degrees from the sixth order model and compared its RSME (e.g. instead of using the degrees $\{0, 1, 2, 3, 4, 5, 6\}$ I removed the fifth degree terms: $\{0, 1, 2, 3, 4, 6\}$). However, all of these attempts resulted in a lower RSME and so I stuck with the complete sixth order model.

Question 5

Problem: After filling in the missing code for the asymmetric loss function, choose a polynomial model that best fits the population based off its root mean asymmetric loss (RMAE), with $\alpha = .05$, on the validation set.

Solution: Based on the RMAE, the best polynomial model I found was the 29th order one. That is, the feature transformation $\boldsymbol{\phi}$ given by:

$$\boldsymbol{\phi}_{29}\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}\right) = \begin{bmatrix} 1 \\ x_1 \\ x_1^2 \\ \vdots \\ x_1^6 \\ x_2 \\ x_2^2 \\ \vdots \\ x_m^{28} \\ x_m^{29} \end{bmatrix}$$

My methodology for determining this was the same as in question 4, to compare the RMAE of increasingly large polynomial models until the error found a local minima. In this case it was at 29th degree features:

```
Model (0,): train RMAE 5.2725, val RMAE 5.4921
Model range(0, 20): train RMAE 5.0524, val RMAE 5.1497
Model range(0, 21): train RMAE 5.0511, val RMAE 5.1429
Model range(0, 22): train RMAE 5.0502, val RMAE 5.1366
Model range(0, 23): train RMAE 5.0493, val RMAE 5.1306
Model range(0, 24): train RMAE 5.0487, val RMAE 5.1250
Model range(0, 25): train RMAE 5.0482, val RMAE 5.1195
Model range(0, 26): train RMAE 5.0477, val RMAE 5.1143
Model range(0, 27): train RMAE 5.0474, val RMAE 5.1095
Model range(0, 28): train RMAE 5.0472, val RMAE 5.1062
Model range(0, 29): train RMAE 5.0470, val RMAE 5.1045
Model range(0, 30): train RMAE 5.0469, val RMAE 5.1042
Model range(0, 31): train RMAE 5.0468, val RMAE 5.1050
Model range(0, 32): train RMAE 5.0468, val RMAE 5.1069
Model range(0, 33): train RMAE 5.0468, val RMAE 5.1097
Model range(0, 34): train RMAE 5.0468, val RMAE 5.1133
```

Question 6

Problem: Evaluate the two chosen models from question 4 and 5 on the test set with both the RMSE and the RMAE. Based on these results, discuss the relative merits of the two models for the data and task at hand. If you had to choose one, given the description of the prediction problem and the assumptions above, which one would you choose and why?

Solution: The results of running the 6th and 29th degree models on the test data are given below:

```
Model range(0, 7): test RMSE 23.3769, test RMAE 5.2272
Model range(0, 30): test RMSE 23.0375, test RMAE 5.1514
```

As we can see, the 29th degree model does better in both RMSE *and* RMAE. As a result, it seems reasonable to dub it the better model choice.

However performance isn't the only concern for such a model. In particular note that the 29th degree model is still very close to the 6th degree model in terms of performance. And that this slight increase in performance is contrasted by the huge increase in feature dimensionality:

$$\dim(\boldsymbol{\phi}_6(\mathbf{x})) = 14 * 6 + 1 = 79$$
$$\dim(\boldsymbol{\phi}_{29}(\mathbf{x})) = 14 * 29 + 1 = 378$$

With greater dimensionality comes greater computational overhead. Although in this case, computing a prediction using the 29th degree model still isn't prohibitively expensive with reasonable hardware.

And so, it seems that, sans any other testing data, the 29th degree model is the better of the two models.