TTIC 31020: Introduction to Statistical Machine Learning
Autumn 2017

Problem Set #1

Out: September 29, 2017

Due: Thursday October 12, 11:59pm

Instructions

How and what to submit? Please submit your solutions electronically via Canvas. Please submit two files:

1. A PDF file with the written component of your solution including derivations, explanations, etc. You can create this PDF in any way you want: typeset the solution in LaTeX (recommended), type it in Word or a similar program and convert/export to PDF, or even hand write the solution (legibly!) and scan it to PDF. Please name this document ⟨`firstname-lastname`⟩`-sol1.pdf`.

2. The empirical component of the solution (Python code and the documentation of the experiments you are asked to run, including figures) in a Jupyter notebook file. Name the notebook ⟨`firstname-lastname`⟩`-sol1.ipynb`.

Late submissions: there will be a penalty of 25 points for any solution submitted within 24 hours past the deadlne. No submissions will be accepted past then.

What is the required level of detail? When asked to derive something, please clearly state the assumptions, if any, and strive for balance: justify any non-obvious steps, but try to avoid superfluous explanations. When asked to plot something, please include int the `ipynb` file the figure as well as the code used to plot it. If multiple entities appear on a plot, make sure that they are clearly distinguishable (by color or style of lines and markers). When asked to provide a brief explanation or description, try to make your answers concise, but do not omit anything you believe is important. If there is a mathematical answer, provide is precisely (and accompany by only succinct words, if appropriate).

When submitting code, please make sure it's reasonably documented, and describe succinctly in the written component of the solution what is done where.

Collaboration policy : collaboration is allowed and encouraged, as long as you (1) write your own solution entirely on your own, (2) specify names of student(s) you collaborated with on the PDF.

1 Linear regression

In this set of problems we will look at the regression problem and the maximum likelihood (ML) approach, with the goal to understand a bit better some of their properties.

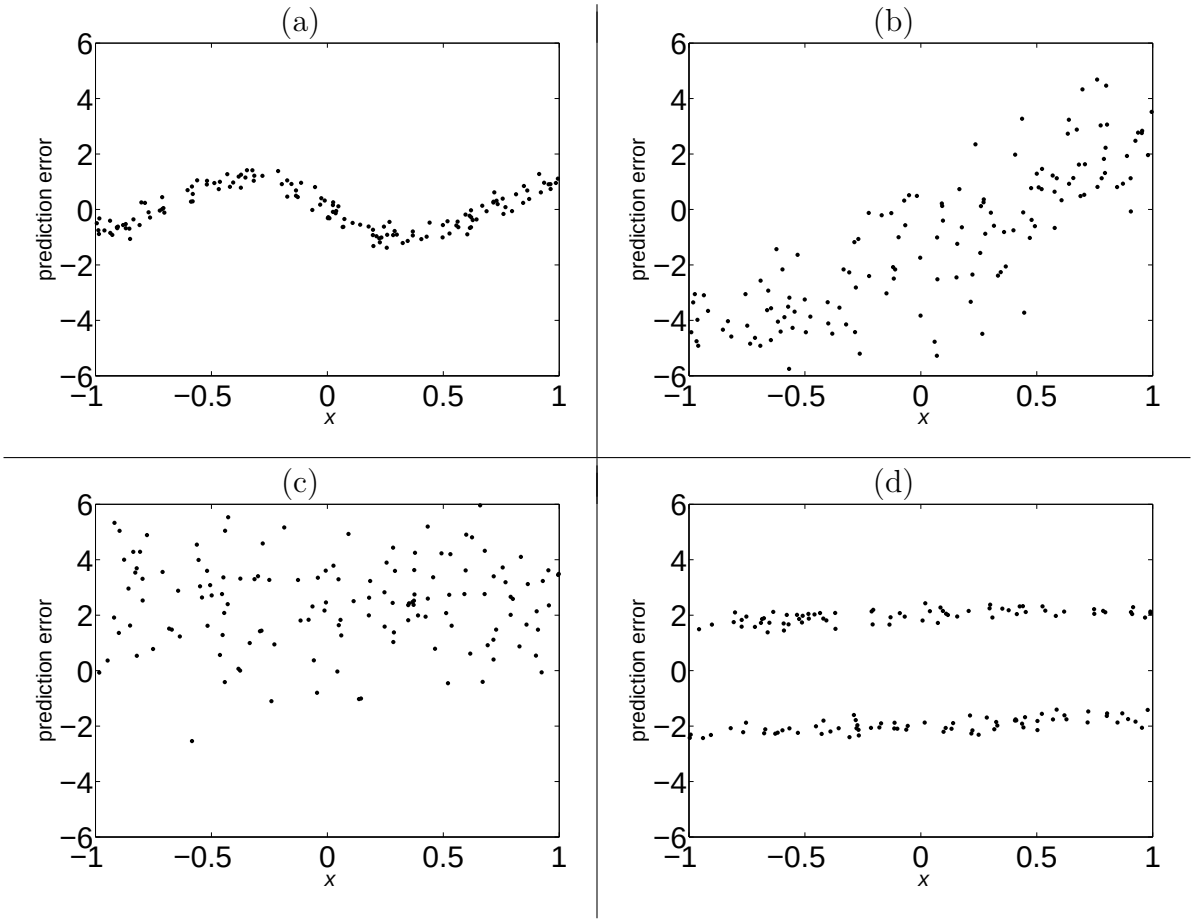

Figure 1: Are these plausible plots for residuals of least squares linear regression on its training set?

Problem 1 **[10 points]**

We are concerned with the distribution of the prediction errors (residuals) $e_i = \hat{y}(x_i) - y_i$ for a linear regression model $\hat{y}(x) = w_1 \cdot x$. Suppose we fit a least squares linear regression model on a training set $\{(x_i, y_i)\}_{i=1}^N$ of (scalar) inputs x and the corresponding target values y. We can then compute e_i for every $i = 1 \ldots, N$ and plot the values vs. x_i.

Figure 1 shows four plots. Each plot may, or may not be showing the distribution of e_i vs. x_i on the training set of a linear least squares regression model.

For each plot, either explain why it can not be showing residuals of least squares regression

on the training set, or describe the form of (x_i, y_i) data set that would produce such residuals (that is, describe the dependence of y on x that the plot at hand implies).

End of problem 1

Advice: Pay attention to both the shape and position of point clouds and the axis labels/values. Don't worry about potential tiny numerical effects not noticeable by eye – this is not a trick question.

In the next two problems we will consider the effect that transforming the training data has on the optimal regression model. Suppose we fit least squares linear model to a data set $\{\mathbf{x}_i, y_i\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, yielding the parameter vector $\mathbf{w}^*$.

Problem 2 [10 points]
Now we take the same data set, and for every i, change y_i to $y'_i = ay_i + b$, for some constants a and b (same a and b for all the is). Now you fit least squares model to the new data set $\{\mathbf{x}_i, y'_i\}$, yielding the parameter vector $\mathbf{w}'$.

Can $\mathbf{w}'$ be computed directly from $\mathbf{w}^*$, without looking at the data? If yes, how exactly? If not, why not?

End of problem 2

Problem 3 [10 points]
Now, instead of modifying ys, we will modify $\mathbf{x}$s. For every feature (dimension of $\mathbf{x}$) we will change x_{ij} to $\tilde{x}_{ij} = c_j x_{ij}$ for some constant c_j (same set of $c_1, \ldots, c_d$ for all is). Again, we fit least squares model to the new data set $\{\tilde{\mathbf{x}}_i, y_i\}$, and get $\tilde{\mathbf{w}}$.

Can $\mathbf{w}'$ be computed directly from $\mathbf{w}^*$, without looking at the data? If yes, how exactly? If not, why not?

End of problem 3

Now we are going to look at a noise model which is a bit different from the i.i.d. Gaussian noise model described in class. Suppose that for every $\mathbf{x}$, the noise that affects y is Gaussian, but the variance of this noise depends on $\mathbf{x}$:

$$y, = \mathbf{w} \cdot \mathbf{x} + \nu, \qquad \nu \sim \mathcal{N}\left(0, \sigma_{\mathbf{x}}^2\right). \tag{1}$$

Problem 4 [10 points]
Without knowing anything else besides the assumptions in 1, can we compute the maximum likelihood estimate for the linear regression parameters $\mathbf{w}^*$ under this noise model? If yes, describe the procedure as precisely as you can; if not, explain why not.

End of problem 4

Problem 5 **[10 points]**
Now suppose we *know* the value of the noise variance $\sigma^2_{\mathbf{x}_i}$ at every training input $\mathbf{x}_i$ for $i = 1, \ldots, N$. Can we now compute the maximum likelihood estimate for linear regression parameters $\mathbf{w}^*$ under this noise model? If yes, describe the procedure as precisely as you can; if not, explain why not.

End of problem 5

2 Loss functions

In this section we will explore the interaction between noise models and the loss functions used in regression. We will work with a slice of Boston Housing data set[1] in which the task is to predict median home prices in various areas of Boston suburbs (in 1990s) from a variety of features. Here we will use just one feature, called LSTAT in the original data set – roughly speaking, it's the percentage of people without high school education living within the "tract" for which we predict home prices.

We are going to assume that in our task, it is much worse to over-estimate the price of a house than to underestimate it. (Perhaps we really want to be able to sell houses quickly...) Technically, we will express it as a *asymmetric squared loss*,

$$\ell_\alpha(\widehat{y}, y) = \begin{cases} \alpha(\widehat{y} - y)^2 & \text{if} \quad \widehat{y} \leq y, \\ (\widehat{y} - y)^2 & \text{if} \quad \widehat{y} \geq y. \end{cases} \tag{2}$$

where $\alpha < 1$ specifies how much more we worry about $\widehat{y}$ over- than under-estimating the house price y. For example, if $\alpha = 0.1$ then over-estimating by $\$D$ is 10 times worse than under-estimating the same amount.

We will now consider fitting polynomial regression models to predict house prices y from LSTAT values x. The Python notebook provided with this assignment contains most of the code you need to conduct experiments in this section; you will need to fill in some missing pieces of the code, however.

Problem 6 **[15 points]**
Complete the missing code in function implementing the (regular symmetric) loss calculation, and the calculation of the log-likelihood of a model under the Gaussian noise model. Using the provided code and following the instructions in the notebook, fit polynomial models of degrees 1,2,3 and 5 to the training set; include the plot of the resulting functions (the code for that is already included).

Use the validation set to select one of the four models; explain your selection criteria.

End of problem 6

[1] `https://archive.ics.uci.edu/ml/machine-learning-databases/housing/`

Now we will move the asymmetric loss. The optimization problem involving asymmetric loss is convex, but unfortunately does not have close form solution. Instead, we will rely on gradient descent to solve it.

Problem 7 **[15 points]**

Complete the missing elements of the code necessary to run the gradient descent with the asymmetric loss. It includes the functions computing the asymmetric loss, the likelihood under the corresponding model, and the gradient descent procedure.

Setting $\alpha = 0.05$, fit the polynomial models of degree 1,2,3 and 5 to the training set; include the plot of the resulting functions (the code for that is already included).

Use the validation set to select one of the four models; explain your selection criteria.

End of problem 7

Problem 8 **[20 points]**

Finally, evaluate your two chosen models (one trained with symmetric loss and the other trained with asymmetric loss) on the test set. For each model compute: the mean symmetric loss, the mean asymmetric loss and the likelihood under the (symmetric) Gaussian noise model. Based on these results, discuss the relative merits of the two models for the data and task at hand.

End of problem 8