Machine Learning Exercise Sheet 04

Linear Regression

Homework

Least squares regression

Problem 1: Let's assume we have a dataset where each datapoint, $(\boldsymbol{x}_i, y_i)$ is weighted by a scalar factor which we will call t_i. We will assume that $t_i > 0$ for all i. This makes the sum of squares error function look like the following:

$$E_{\text{weighted}}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N} t_i \left[\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) - y_i\right]^2$$

Find the equation for the value of $\boldsymbol{w}$ that minimizes this error function.

Furthermore, explain how this weighting factor, t_i, can be interpreted in terms of
1) the variance of the noise on the data and
2) data points for which there are exact copies in the dataset.

If we define $\boldsymbol{T} = \text{diag}(t_1, \ldots, t_N)$ to be a diagonal matrix containing the weighting coefficients, then we can write the weighted sum-of-squares cost function in the form

$$E_{\text{weighted}}(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{\Phi w} - \boldsymbol{y})^T\boldsymbol{T}(\boldsymbol{\Phi w} - \boldsymbol{y})$$

Setting the derivative with respect to $\boldsymbol{w}$ to zero, and re-arranging, then gives

$$\boldsymbol{w}^*_{\text{weighted}} = (\boldsymbol{\Phi}^T\boldsymbol{T}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{T}\boldsymbol{y}$$

which reduces to the standard solution for the case $\boldsymbol{T} = \boldsymbol{I}$. I.e.

$$\boldsymbol{w}_{\text{ML}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$$

If you remember back to when we modeled the likelihood using a Gaussian, our likelihood had the following form:

$$p(\boldsymbol{y} \mid \boldsymbol{\Phi}, \boldsymbol{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(y_i \mid \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x_i}), \beta^{-1})$$

Upload a single PDF file with your homework solution to Moodle by 10.11.2019, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

After applying the logarithm and using the standard form for the univariate Gaussian our equation looked like this:

$$\ln p(\boldsymbol{y} \mid \boldsymbol{\Phi}, \boldsymbol{w}, \beta) = \sum_{i=1}^{N} \ln \mathcal{N}(y_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i), \beta^{-1})$$
$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_{\mathrm{LS}}(\boldsymbol{w})$$

Where $E_{\mathrm{LS}}(\boldsymbol{w})$ is the standard sum of squares error function (not to be confused with the $E_{\mathrm{weighted}}(\boldsymbol{w})$ we defined earlier). Remember that $E_{\mathrm{LS}}(\boldsymbol{w})$ is defined as follows:

$$E_{\mathrm{LS}}(\boldsymbol{w}) = \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) - y_i)^2$$

When we compare E_{LS} with E_{weighted} and the effect of swapping the two in the previous likelihood equation we can see that T_i can be regarded as a precision (inverse variance) parameter, particular to the data point $(\boldsymbol{x}_i, y_i)$, that either replaces or scales β.

Alternatively, t_i can be regarded as an *effective* number of replicated observations of data point $(\boldsymbol{x}_i, y_i)$; this becomes particularly clear if we consider $E_{\mathrm{weighted}}(\boldsymbol{w})$ with t_i taking positive integer values, although it is valid for any $t_i > 0$.

Ridge regression

Problem 2: Show that the following holds: The ridge regression estimates can be obtained by ordinary least squares regression on an augmented dataset: Augment the design matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}$ with M additional rows $\sqrt{\lambda}\boldsymbol{I}_{M \times M}$ and augment $\boldsymbol{y}$ with M zeros.

Ordinary least squares minimizes $(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})$. For ridge regression we need to minimize $(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}) + \lambda \boldsymbol{w}^T\boldsymbol{w}$. If we define $\hat{\boldsymbol{\Phi}} = \begin{pmatrix} \boldsymbol{\Phi} \\ \sqrt{\lambda}\boldsymbol{I} \end{pmatrix}$ and $\hat{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0}_M \end{pmatrix}$, we can formulate the ridge regression objective as minimizing $(\hat{\boldsymbol{y}} - \hat{\boldsymbol{\Phi}}\boldsymbol{w})^T(\hat{\boldsymbol{y}} - \hat{\boldsymbol{\Phi}}\boldsymbol{w})$.

Problem 3: Derive the closed form solution for ridge regression error function

$$E_{\mathrm{ridge}}(\boldsymbol{w}) = \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) - y_i)^2 + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w}$$

Additionally, discuss the scenario when the number of training samples N is smaller than the number of basis functions M. What computational issues arise in this case? How does regularization address them?

$$
\begin{aligned}
E_{\text{ridge}}(\boldsymbol{w}) &= \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) - y_i)^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} \\
&= \frac{1}{2}(\boldsymbol{\Phi}\boldsymbol{w} - \boldsymbol{y})^T(\boldsymbol{\Phi}\boldsymbol{w} - \boldsymbol{y}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}
\end{aligned}
$$

Taking the gradient

$$
\begin{aligned}
\nabla_{\boldsymbol{w}} E_{\text{ridge}}(\boldsymbol{w}) &= \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w} - \boldsymbol{\Phi}^T\boldsymbol{y} + \lambda\boldsymbol{w} \\
&= (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I})\boldsymbol{w} - \boldsymbol{\Phi}^T\boldsymbol{y}
\end{aligned}
$$

Set it to zero

$$
\begin{aligned}
(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I})\boldsymbol{w} &= \boldsymbol{\Phi}^T\boldsymbol{y} \\
\boldsymbol{w} &= (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}
\end{aligned}
$$

If $N < M$ the covariance matrix $\boldsymbol{\Phi}^T\boldsymbol{\Phi} \in \mathbb{R}^{M\times M}$ will be singular, therefore not invertible. (this may happen even if $N \geq M$, e.g. when some features are correlated).

When regularization is used, $\lambda\boldsymbol{I}$ is added to the covariance matrix, thus fixing the potential degeneracy issue and making the problem tractable.

Multi-output linear regression

Problem 4: In class, we only considered functions of the form $f : \mathbb{R}^n \to \mathbb{R}$. What about the general case of $f : \mathbb{R}^n \to \mathbb{R}^m$? For linear regression with multiple outputs, write down the loglikelihood formulation and derive the MLE of the parameters.

The observation $\boldsymbol{y}_i$ is a vector with $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{\Sigma})$, $\boldsymbol{W} \in \mathbb{R}^{n\times m}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times m}$, covariance $\boldsymbol{\Sigma}$ is known and fixed for all possible observations. For n i.i.d observed pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, the likelihood is

$$p(\boldsymbol{y}_i|\boldsymbol{W}, \boldsymbol{\Sigma}) = \prod_i \mathcal{N}(\boldsymbol{y}_i|\boldsymbol{W}^T\boldsymbol{x}_i, \boldsymbol{\Sigma}),$$

and thus the negative log-likelihood is

$$\ln p(\boldsymbol{y}_i|\boldsymbol{W}, \boldsymbol{\Sigma}) = \text{const} + \frac{1}{2}\sum_i (\boldsymbol{y}_i - \boldsymbol{W}^T\boldsymbol{x}_i)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{W}^T\boldsymbol{x}_i)$$

Since $\boldsymbol{\Sigma}$ is symmetric and positive semi-definite we can use the Cholesky decomposition $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ to obtain $\boldsymbol{\Sigma}^{-1} = \boldsymbol{L}^{-T}\boldsymbol{L}^{-1}$. With this we can simplify the two factors via

$$\boldsymbol{L}^{-1}(\boldsymbol{y}_i - \boldsymbol{W}^T\boldsymbol{x}_i) = (\boldsymbol{L}^{-1}\boldsymbol{y}_i - \boldsymbol{L}^{-1}\boldsymbol{W}^T\boldsymbol{x}_i) = (\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{W}}^T\boldsymbol{x}_i).$$

Using this transformation we get

$$\begin{aligned}
\ln p(\boldsymbol{y}_i|\boldsymbol{W}, \boldsymbol{\Sigma}) &= \text{const} + \frac{1}{2}\sum_i (\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{W}}^T\boldsymbol{x}_i)^T(\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{W}}^T\boldsymbol{x}_i) \\
&= \text{const} + \frac{1}{2}\sum_i \sum_j (\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{W}}^T\boldsymbol{x}_i)_j(\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{W}}^T\boldsymbol{x}_i)_j \\
&= \text{const} + \frac{1}{2}\sum_i \sum_j (\hat{\boldsymbol{Y}}_{ij} - (\boldsymbol{X}\hat{\boldsymbol{W}})_{ij})(\hat{\boldsymbol{Y}}_{ij} - (\boldsymbol{X}\hat{\boldsymbol{W}})_{ij}) \\
&= \text{const} + \frac{1}{2}\sum_i \sum_j \left(\left(\hat{\boldsymbol{Y}} - \boldsymbol{X}\hat{\boldsymbol{W}}\right)^T\right)_{ji}\left(\hat{\boldsymbol{Y}} - \boldsymbol{X}\hat{\boldsymbol{W}}\right)_{ij} \\
&= \text{const} + \frac{1}{2}\sum_j \left((\hat{\boldsymbol{Y}} - \boldsymbol{X}\hat{\boldsymbol{W}})^T(\hat{\boldsymbol{Y}} - \boldsymbol{X}\hat{\boldsymbol{W}})\right)_{jj} \\
&= \text{const} + \frac{1}{2}\text{Tr}\left[(\hat{\boldsymbol{Y}} - \boldsymbol{X}\hat{\boldsymbol{W}})^T(\hat{\boldsymbol{Y}} - \boldsymbol{X}\hat{\boldsymbol{W}})\right],
\end{aligned}$$

where $\hat{\boldsymbol{Y}}$ and $\boldsymbol{X}$ are matrices that have the vectors $\hat{\boldsymbol{y}}_i$ and $\boldsymbol{x}_i$ as their rows. Note that the trace essentially just denotes the sum over all dimensions of the output domain. Taking the derivative of the negative log-likelihood with respect to $\hat{\boldsymbol{W}}$ and setting it to 0 gives $\hat{\boldsymbol{W}}_{\text{MLE}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\hat{\boldsymbol{Y}}$. So these are m single least square problems, one for every *column* of $\hat{\boldsymbol{Y}}$. Finally, transforming $\hat{\boldsymbol{W}}_{\text{MLE}}$ back gives $\boldsymbol{W}_{\text{MLE}} = \hat{\boldsymbol{W}}_{\text{MLE}}\boldsymbol{L}^T$.

Comparison of Linear Regression Models

Problem 5: We want to perform regression on a dataset consisting of N samples $\boldsymbol{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} \in \mathbb{R}^N$).

Assume that we have fitted an L_2-regularized linear regression model and obtained the optimal weight vector $\boldsymbol{w}^* \in \mathbb{R}^D$ as

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{x}_i - y_i)^2 + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w}$$

Note that there is no bias term.

Now, assume that we obtained a new data matrix $\boldsymbol{X}_{new}$ by scaling all samples by the same positive factor $a \in (0, \infty)$. That is, $\boldsymbol{X}_{new} = a\boldsymbol{X}$ (and respectively $\boldsymbol{x}_i^{new} = a\boldsymbol{x}_i$).

a) Find the weight vector $\boldsymbol{w}_{new}$ that will produce the same predictions on $\boldsymbol{X}_{new}$ as $\boldsymbol{w}^*$ produces on $\boldsymbol{X}$.

> Predictions of a linear regression model are generated as $\hat{y} = \boldsymbol{w}^T \boldsymbol{x}$.
>
> This means that we need to ensure that $\boldsymbol{w}^{*T} \boldsymbol{x}_i = \boldsymbol{w}_{new}^T \boldsymbol{x}_i^{new}$ or equivalently $\boldsymbol{w}^{*T} \boldsymbol{x}_i = \boldsymbol{w}_{new}^T a\boldsymbol{x}_i$. Solving for $\boldsymbol{w}_{new}$ we get $\boldsymbol{w}_{new} = \frac{\boldsymbol{w}^*}{a}$

b) Find the regularization factor $\lambda_{new} \in \mathbb{R}$, such that the solution $\boldsymbol{w}_{new}^*$ of the new L_2-regularized linear regression problem

$$\boldsymbol{w}_{new}^* = \arg\min_{\boldsymbol{w}} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{X}_i^{new} - y_i)^2 + \frac{\lambda_{new}}{2} \boldsymbol{w}^T \boldsymbol{w}$$

will produce the same predictions on $\boldsymbol{X}_{new}$ as $\boldsymbol{w}^*$ produces on $\boldsymbol{X}$.

Provide a mathematical justification for your answer.

> The closed form solution for $\boldsymbol{w}^*$ on the original data $\boldsymbol{X}$ is
>
> $$\boldsymbol{w}^* = (\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$
>
> The closed form solution for $\boldsymbol{w}_{new}^*$ on the new data $\boldsymbol{X}_{new}$ is
>
> $$\begin{aligned} \boldsymbol{w}_{new}^* &= (\boldsymbol{X}_{new}^T \boldsymbol{X}_{new} + \lambda_{new} \boldsymbol{I})^{-1} \boldsymbol{X}_{new}^T \boldsymbol{y} \\ &= a(a^2 \boldsymbol{X}^T \boldsymbol{X} + \lambda_{new} \boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{y} \end{aligned}$$
>
> by setting $\lambda_{new} = a^2 \lambda$, we get
>
> $$\begin{aligned} &= a(a^2 \boldsymbol{X}^T \boldsymbol{X} + a^2 \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{y} \\ &= \frac{1}{a} (\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{y} \\ &= \frac{1}{a} \boldsymbol{w}^* \end{aligned}$$

Which (according to our answer in part (a) of this problem) will produce the same predictions on $\boldsymbol{X}_{new}$ as $\boldsymbol{w}^*$ does on $\boldsymbol{X}$, as desired.

Equivalent solution

$$\begin{aligned}
\boldsymbol{w}^*_{new} \overset{!}{=} \frac{\boldsymbol{w}^*}{a} &= \frac{1}{a} \arg\min_{\boldsymbol{w}} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{x}_i - y_i)^2 + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w} \\
&= \frac{1}{a} \arg\min_{\boldsymbol{w}} \frac{1}{2} \sum_{i=1}^{N} (\frac{\boldsymbol{w}^T}{a} a\boldsymbol{x}_i - y_i)^2 + \frac{a^2\lambda}{2} \frac{\boldsymbol{w}^T}{a} \frac{\boldsymbol{w}}{a} \\
&= \frac{a}{a} \arg\min_{\boldsymbol{w}_{new} = \frac{\boldsymbol{w}}{a}} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}_{new}^T \boldsymbol{x}_i^{new} - y_i)^2 + \frac{a^2\lambda}{2} \boldsymbol{w}_{new}^T \boldsymbol{w}_{new} \\
&\overset{!}{=} \boldsymbol{w}^*_{new} = \arg\min_{\boldsymbol{w}_{new}} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}_{new}^T \boldsymbol{x}_i^{new} - y_i)^2 + \frac{\lambda_{new}}{2} \boldsymbol{w}_{new}^T \boldsymbol{w}_{new}
\end{aligned}$$

For this equality to hold we need to match the regularization term by setting $\lambda_{new} = a^2\lambda$.

Programming Task: Least squares regression

Problem 6: Load the notebook `04_homework_linear_regression.ipynb` from Piazza. Fill in the missing code and run the notebook. Convert the evaluated notebook to pdf and add it to the printout of your homework.

Note: We suggest that you use Anaconda for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks and how to convert them to other formats, consult the Jupyter documentation and nbconvert documentation.

In-class Exercises

Problem 7: Assume that we are given a dataset, where each sample x_i and regression target y_i is generated according to the following process

$$x_i \sim \text{Uniform}(-10, 10)$$
$$y_i = ax_i^3 + bx_i^2 + cx_i + d + \epsilon_i, \quad \text{where} \quad \epsilon_i \sim \mathcal{N}(0, 1) \quad \text{and} \quad a, b, c, d \in \mathbb{R}.$$

The 3 regression algorithms below are applied to the given data. Your task is to say what the bias and variance of these models are (low or high). Provide a 1-2 sentence explanation to each of your answers.

a) Linear regression

Bias: high. Variance: low.

A straight line cannot capture a degree 3 polynomial (thus underfitting the data).

b) Polynomial regression with degree 3

Bias: low. Variance: low.

The model is same as the data generating process. We can achieve a good fit.

c) Polynomial regression with degree 10

Bias: low. Variance: high.

Since we are using a polynomial regression with a degree much higher compared to the data generating process, the model will overfit the data.

Problem 8: Given is a training set consisting of samples $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ with respective regression targets $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$.

Alice fits a linear regression model $f(\boldsymbol{x}_i) = \boldsymbol{w}^T\boldsymbol{x}_i$ to the dataset using the closed form solution for linear regression (normal equations).

Bob has heard that by transforming the inputs $\boldsymbol{x}_i$ with a vector-valued function $\boldsymbol{\phi}$, he can fit an alternative function, $g(\boldsymbol{x}_i) = \boldsymbol{v}^T\boldsymbol{\phi}(\boldsymbol{x_i})$, using the same procedure (solving the normal equations). He decides to use a linear transformation $\boldsymbol{\phi}(\boldsymbol{x_i}) = \boldsymbol{A}^T\boldsymbol{x}_i$, where $\boldsymbol{A} \in \mathbb{R}^{D\times D}$ has full rank.

a) Show that Bob's procedure will fit the same function as Alice's original procedure, that is $f(\boldsymbol{x}) = g(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^D$ (given that $\boldsymbol{w}$ and $\boldsymbol{v}$ minimize the training set error).

By fitting $g(\boldsymbol{x}) = \boldsymbol{v}^T\boldsymbol{\phi}(\boldsymbol{x})$, Bob can only set the function equal to linear combinations of the inputs $(\boldsymbol{v}^T\boldsymbol{A}^T)\boldsymbol{x} = \boldsymbol{w}^T\boldsymbol{x}$, where $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}$.

Moreover, just like Alice, all linear combinations are available: any function Alice fits can be matched by setting $\boldsymbol{v} = \boldsymbol{A}^{-1}\boldsymbol{w}$.

Upload a single PDF file with your homework solution to Moodle by 10.11.2019, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

As both Alice and Bob are selecting the function that best matches the outputs from the same set of functions, and with the same cost function, they will select the same function.

b) Can Bob's procedure lead to a lower training set error than Alice's if the matrix $\boldsymbol{A}$ is not invertible? Explain your answer.

Since $\boldsymbol{A}$ is square and not invertible, then multiple input vectors are transformed to the same output vector. It's not possible for Bob to assign different function values to two such inputs, whereas Alice can. Bob can no longer fit all the same functions as Alice. Furthermore, Alice can still match any of Bob's solutions by setting $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}$. Since the error function is quadratic and PSD it has a unique (set of) optima. So either Alice's (set of) optima will include Bob's solution or it will be lower. In summary, Bob's training error might be worse than Alice's, but it can't be better.

Note that we are only talking about training error in this example, not test error. Bob might manage to find a model that generalizes better than Alice's, but Alice will always be able to fit the training data at least as well as Bob.

Problem 9: See Jupyter notebook `inclass_04_notebook.ipynb`.