Worksheet on "Linear Regression" (version 2)

CS5691 PRML Jul–Nov 2025

October 3, 2025

(Acknowledgment Note: We have used LLMs (ChatGPT and Gemini) in preparing and copy-editing questions and/or answers in this document.)

1. Consider a set of 2D datapoints $D := \{(x_n, y_n)\}_{n=1}^{5} = \{(1,2),(2,3),(3,5),(4,4),(5,6)\}$. We would like to build a linear regression model that approximates y_n using an affine function of x_n.
   (a) In a direct approach to linear regression, we minimize an error/cost function to find the best affine function that predicts y from x. Using this approach, answer the following questions.
      i. Find the optimal weight (w_{LS}) using the least-squares approach to linear regression. What will this resulting model predict for $x_{new} = 10$?
      ii. Find the optimal weight (w_{RLS}) using the regularized least-squares approach to linear regression (specifically ridge regression, with $\lambda = 1$ for simplicity). What will this resulting model predict for $x_{new} = 10$?
      iii. Perform PCA on the same dataset and show how the two linear regression lines given by w_{LS} and w_{RLS} learnt above compare to the first PC of the same dataset.
   (b) In a probabilistic (discriminative model) approach to linear regression, we model the conditional $(y|x)$ as a Gaussian distribution and learn its parameters using density estimation techniques (MLE or Bayesian approach). Specify what conditional distribution $(y|x)$ is learnt from the dataset in this question by filling in these blanks:
      i. Based on an MLE approach where weights w_{ML} (including intercept) are estimated from the dataset D,
      $(y_{new}|x_{new} = 10; D) \sim \mathcal{N}(\_\_\_\_\_, \_\_\_\_\_)$.
      ii. Based on a Bayesian approach where weights w_{MAP} (including intercept) are estimated from the dataset D and hyperparameters α (precision of w's prior) and β (precision of $(y|x)$) are assumed to be fixed known constants (with both assumed to be 1 for this question),
      $(y_{new}|x_{new} = 10; D, \alpha = 1, \beta = 1) \sim \mathcal{N}(\_\_\_\_\_, \_\_\_\_\_)$.
      iii. Based on a Bayesian approach where the full parameter posterior distribution $(w \mid D, \alpha = 1, \beta = 1)$ is utilized to predict the target, the resulting posterior predictive distribution is:
      $(y_{new}|x_{new} = 10; D, \alpha = 1, \beta = 1) \sim \mathcal{N}(\_\_\_\_\_, \_\_\_\_\_)$.

2. Consider the following datasets comprising the design matrix X and target $\mathbf{y}$ such that linear regression amounts to learning the weights w that achieves the approximation $Xw \simeq \mathbf{y}$ (assume that no intercept is allowed in this linear function approximation; so X is used as is and not transformed any further). For each of these datasets, find the least squares solution w_{LS}. If multiple such solutions exist, return the minimum norm solution.
   (a) $X = \begin{bmatrix} -2 & 6 \\ -1 & 3 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$.
   (b) $X = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.
   (c) $X = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

3. Given a high-dimensional dataset $\Phi_{n \times p}$ ($p \gg n$), you may reduce the # of input features (p) when building a predictive linear regression model of a target variable $t_{n \times 1}$. Two common techniques are:

- **PCA-based feature extraction**, which creates new features (that are linear combinations of the original features) without using the target variable, and subsequently use these new features to predict t.
- **LASSO feature selection**, which selects a subset of the original features based on their power to predict the target.

Answer the following:

(a) What is the key conceptual difference between PCA-based feature extraction and LASSO feature selection regarding the use of the target variable?

(b) When multiple features are highly collinear, how do PCA and LASSO differ in handling them? Describe any potential issues LASSO may face in this situation.

(c) Which method generally produces more interpretable features and why?

(d) Which method always admits a closed form solution: PCA or LASSO?

4. The PCA-based feature extraction technique can also be used as a preprocessing step before classification instead of before regression as in the last question. A code to achieve this for a classification method called "logistics regression" is provided here: `https://www.geeksforgeeks.org/data-analysis/principal-component-analysis-with-python/`. Read through and understand this code, and answer the following questions in the context of this code.

   (a) Why is `pca.fit_transform()` function called on the training data, and only `pca.transform()` called on the test data, in this code excerpt taken from the above webpage?

   ```
   from sklearn.decomposition import PCA
   pca = PCA(n_components=2)
   X_train = pca.fit_transform(X_train)
   X_test = pca.transform(X_test)
   ```

   Specifically, report issues if any that will arise from replacing `pca.transform(X_test)` in the above code by `pca.fit_transform(X_test)`?

   (b) The help page of `pca.transform(X)` function says that X is projected on the first (top selected) principal components previously extracted from a training set. Look at the source code of this function to:

      i. Find out whether the data points in X are mean-centered or not before projecting onto the top selected PCs?

      ii. If X is centered before projection onto the top PCs, then what mean vector (training data mean or test data mean) is used to center X?

      iii. What is conceptually the correct way to project, and does the code implement this correct methodology?