CSCE 478/878 Recitation 9 Handout

Linear Regression: Stochastic Gradient Descent Linear Regression & OLS Polynomial Regression

---

- This is a **graded recitation**. Submit your work for grading.
- Your Jupyter notebook should be submitted via webhandin by 4:45PM, **March 15**.
- Use the following naming convention: `<lastname>_<firstname>_9.ipynb`

Go through the following two jupyter notebooks from my Github "***Linear Regression - Extensive Adventure***" repository. Carefully understand how to use sklearn's Stochastic Gradient Descent Regressor (SGDRegressor) for Linear Regression and the ordinary least squares (OLS) method for Polynomial Regression.

- Linear Regression - Boston Housing Dataset - Gradient Descent
- Linear Regression - Boston Housing Dataset - Polynomial OLS

You should understand how to choose optimal complexity (degree of polynomial) in Polynomial Regression by using the following two curves: "root mean square error vs degree" curve and learning curves.

Score Distribution:

1. Hyperparameter Tuning for SGD Regressor (**pts 20**)

   Report the best score (negative mean squared error) & optimal hyperparameter values.

2. Select the Best Model for the SGD Regressor (**pts 5**)

   Report the following two evaluation metrics.
   - Mean Squared Error (MSE)
   - Coefficient of Determination or R^2

3. Evaluate Model Performance Using Test Data (**pts 5**)

Report the following two evaluation metrics.

- Mean Squared Error (MSE)
- Coefficient of Determination or R^2

4. Choose the Optimal Degree (Model Complexity) of the Polynomial Regression (**pts 20**)

   Plot the "rmse vs. degree" for degree 1, 2, 3 and 4. Report the optimal degree (that gives the smallest generalization error).

5. Determine Model Complexity using Learning Curve (**pts 40**)

   You will generate following two set of learning curves.

   - Linear Model
   - 4^{th} Degree Polynomial Model

6. Then, based on the learning curves, determine whether your models are overfitting or underfitting. (**pts 10**)