Day 4: Linear Classifiers

Summer STEM: Machine Learning

Nikola Janjušević
Akshaj Kumar Veldanda
Jacky Yuan
Tejaishwarya Gagadam

NYU Tandon School of Engineering

June 20, 2019

Learning Objectives

- What is the functional form of logistic regression?
- How do we interpret the output of a binary logistic classifier? Multi-variable classifier?
- How do we determine the threshold for classification?
- What is cross-entropy loss function? Why do we use cross-entropy over MSE?
- What is one-hot vector?
- How to do multi-class classification?
- What is an ROC curve and how do we interpret it?

Outline

Review of Day 3

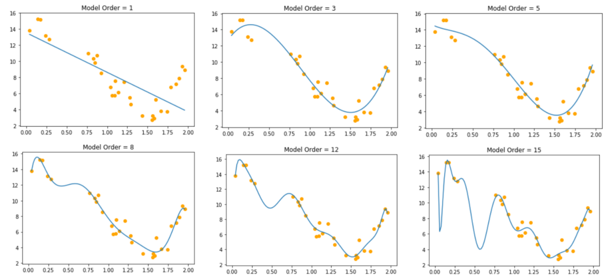
- Yesterday we learned about:
- Polynomial Regression
- Overfitting
- K-folds Validation
- Regularization
- Non-Linear Optimization - Gradient Descent

Review: Polynomial Regression

- Polynomial Model: $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ...\beta_D x^D$
- Design Matrix for Polynomial: $A = \begin{bmatrix} 1 & x_1 & x_2^2 & \cdots & x_1^D \\ 1 & x_2 & x_2^2 & \cdots & x_2^D \\ \vdots & & \ddots & & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^D \end{bmatrix}$
- Think of this design matrix as creating new features that are powers of the original single feature
- The process for calculating the weights β's are same as Linear Regression
- Python function: sklearn or np.polyfit

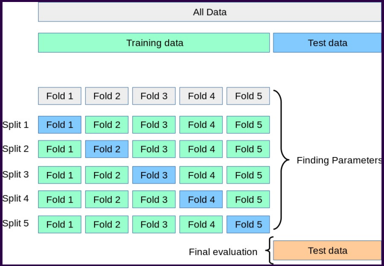
Review: Over-fitting

- Train error always decreases as you use higher order models
- A model that is over-fitted on train data is unlikely to work well on new data



Review: K-folds Validation

- Algorithm to automatically detect the optimal model order
- Split train data into K folds
- On each iteration, evaluate test error using one of the folds
- Calculate the average error among all folds



Review: Regularization

- Another method to combat over-fitting
- High weight terms usually lead to over-fitting
- Introduction of a new term in cost function
-
$$J = \sum_{i=1}^{N} (y_i - y_{i\,pred})^2$$

Review: Regularization

- Another method to combat over-fitting
- High weight terms usually lead to over-fitting
- Introduction of a new term in cost function
- $$J = \sum_{i=1}^{N} (y_i - y_{i\,pred})^2 + \lambda \sum_{j=1}^{D} (w_j)^2$$
- The new term penalizes the magnitude of weights
- Hyperparameter *lambda* determines how much you regularize: higher lambda $\leftarrow$ more regularization

Review: Non-Linear Optimization

- Cannot rely on closed form solutions
  - Computation efficiency
  - Closed form solution not available
- Optimization technique finds optimal solution by iteratively changing parameters toward better solution
- Gradient based method is most commonly used in ML

Review: Gradient Descent

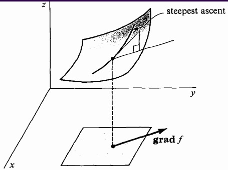
- Average rate of change: $\frac{\Delta y}{\Delta x} = \frac{y_1 - y_0}{x_1 - x_0}$

- Derivative: Instantaneous rate of change of a function
- $\frac{dy}{dx} = \lim_{x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$

- For a 1-dimensional function:
  - $\frac{dy}{dx} > 0$, Function is increasing
  - $\frac{dy}{dx} < 0$, Function is decreasing

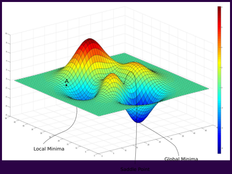
Review: Gradient Descent

- In a two dimensional function $z = f(x, y)$, we can take derivative in more than one direction
- Gradient a vector that points to the direction of maximum increase
- $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T$



Review: Gradient Descent

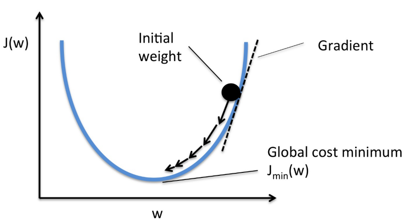
- Weight Space
- We may visualize the loss function as surface in a multi-dimensional space
- Locally, the function may be viewd as a paraboloid
- There are local minima that we want to avoid because they are not optimal solution

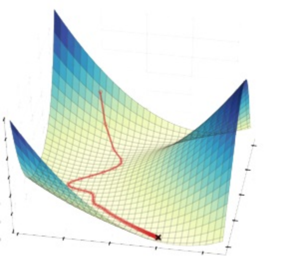


Review: Gradient Descent

- Key idea: For each iteration
  - 1. Calculate the gradient of the loss function with respect to the weights
    $\nabla J(\boldsymbol{w}) = \begin{bmatrix} \frac{\partial J}{\partial \beta_0} & \dots & \frac{\partial J}{\partial \beta_N} \end{bmatrix}^T$
  - 2. Update the weights by taking a step opposite to the gradient $\boldsymbol{w_{new}} = \boldsymbol{w_{old}} - \alpha \nabla J(\boldsymbol{w})$
    where *alpha* is called the learning rate: how big of a step you take

Review: Gradient Descent





Outline

Demo: Diagnosing Breast Cancer

- We're going to use the breast cancer dataset to predict whether the patients' scans show a malignant tumour or a benign tumour.
- Let's try to find the best linear classifier using linear regression.

Outline

Classification

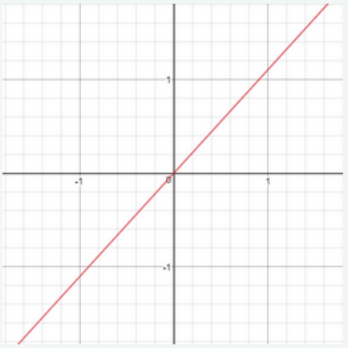
- One method is to use linear regression
  - Map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0
  - this method doesn't work well because classification is not actually a linear function
- Classification takes the only discrete values for prediction
  - For binary classification problem, y can only take two values, 0 and 1
  - Ex: If we want to build a spam classifier for email, then x may be some features of the email
    - The $y = 1$ if it is a spam
    - Otherwise, $y = 0$

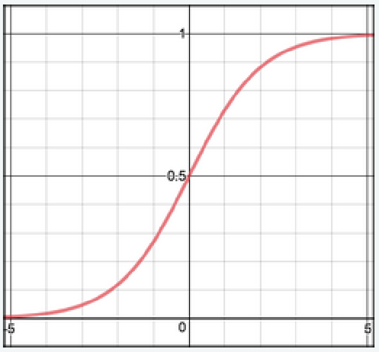
Hypothesis Representation

- Approach classification as old linear regression problem, ignoring the fact that y is discrete
  - We have seen that this approach performs poorly
- To fix this, develop an hypothesis such that $0 \leq \hat{y} \leq 1$
  - This is accomplished by using the Sigmoid function

Sigmoid Function

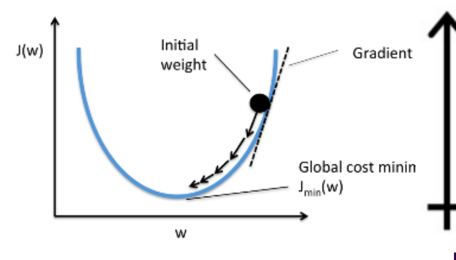
- Recall from linear regression $z = \beta_0 + \beta_1 x$
- On application of sigmoid function to z, we force $0 \leq \hat{y} \leq 1$
  - $\hat{y} = sigmoid(z) = \frac{1}{1+e^{-z}}$

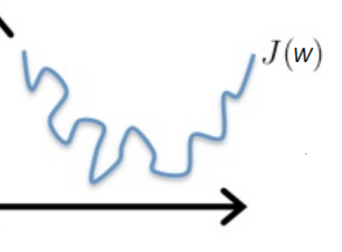




Classification Loss Function

- Cannot use the same cost function that we used for linear regression
  - Logistic function has many local optima
- Logistic cost function is $\frac{1}{m}\Sigma_{i=1}^{m}[-ylog(\hat{y}) - (1-y)log(1-\hat{y})$
  - This loss function is called binary cross entropy loss
  - This loss function has only one optimum point





Outline

Types of Errors in Classification

- Correct predictions:
  - True Positive (TP) : Predict y = 1 when y =1
  - True Negative (TN) : Predict y = 0 when y = 0
- Two types of errors:
  - False Positive/False Alarm (FP): Predict y=1 when y=0
  - False Negative/Missed Detection (FN): Predict y=0 when y=1
- Confusion Matrix:

- Accuracy of classifier can be measured by:
  - $TPR = P(\hat{y} = 1 | y = 1)$
  - $FPR = P(\hat{y} = 1 | y = 0)$
  - Accuracy=$P(\hat{y} = 1 | y = 1)$+ $P(\hat{y} = 0 | y = 0)$
    - (percentage of correct classification)

predicted→ real↓	*Class_pos*	*Class_neg*
Class_pos	TP	FN
Class_neg	FP	TN

$$\text{TPR (sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR (1-specificity)} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

Different Metrics for Error

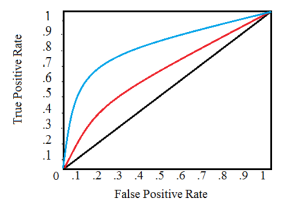
- Metrics to measure the error rate:
  - Recall/Sensitivity/TPR = TP/(TP+FN) (How many positives are detected among all positive?)
  - Precision = TP/(TP+FP) (How many detected positive is actually positive?)
  - Accuracy = (TP+TF)/(TP+FP+TN+FN) (percentage of correct classification)
  - F1-score = $\frac{Precision*Recall}{(Precision+Recall)/2}$
- Why accuracy alone is not a good measure for assessing the model
  - There might be an overwhelming proportion of one class over another
  - Example: A rare disease occurs 1 in ten thousand people
  - A test that classifies everyone as free of the disease can achieve 99.999% accuracy when tested with people drawn randomly from the entire population

Thresholding and ROC

- We can trade-off TPR (sensitivity) and FPR by changing the threshold
- Increasing $t \rightarrow$ Decreases false positives, but also reduces sensitivity
- Decreasing $t \rightarrow$ Increases sensitivity, but also increases false positive
- Why do we want this trade-off?
- Example:
  1. Detection for burglary into a building, need high sensitivity, we can tolerate a few false alarms $\rightarrow$ decrease t
  2. Making decision to buy a stock, making a false positive decision will lose millions $\rightarrow$ increase t

Thresholding and ROC

- ROC (Receiver Operating Characteristics) curve:
- Plot the change between TPR and FPR by varying the threshold
- Allow you to choose the threshold to meet a target TPR/FPR
- A good classifier will have large area under the curve
- A classifier with a higher area under the curve means that under same FPR, it has higher TPR



Outline

Outline

Outline

Learning Objectives

- What is the functional form of logistic regression?
- How do we interpret the output of a binary logistic classifier? Multi-variable classifier?
- How do we determine the threshold for classification?
- What is cross-entropy loss function? Why do we use cross-entropy over MSE?
- What is one-hot vector?
- How to do multi-class classification?
- What is an ROC curve and how do we interpret it?

Thank You!

- Next Class: Linear Regression
- The real machine learning will begin!