Tutorial 6

CPSC 340
Overview

- Regularization
- RBF Basis
- Robust Regression
- Gradient descent
Regularization
Overfitting on the training set is a common problem.
Regularization - Motivation

- Overfitting on the training set is a common problem
- Having too many features and little data can lead to overfitting
  - Underdetermined system: fewer equations than unknowns
  - Either no solution or infinitely many solutions
- To address this:
  \[
  f(w) = \frac{1}{2} \|Xw - y\|^2 + \frac{\lambda}{2} \|w\|^2
  \]
Regularization - Motivation

- Overfitting on the training set is a common problem
- Having too many features and little data can lead to overfitting
  - Underdetermined system: fewer equations than unknowns
  - Either no solution or infinitely many solutions
- To address this:
  \[ f(w) = \frac{1}{2} ||Xw - y||^2 + \frac{\lambda}{2} ||w||^2 \]

- Unique solution
Regularization - Motivation

- Overfitting on the training set is a common problem
- Having too many features and little data can lead to overfitting
  - Underdetermined system: fewer equations than unknowns
  - Either no solution or infinitely many solutions
- To address this:
  - Select a subset of features - $L_1$ regularization
    \[ f(w) = \frac{1}{2} ||Xw - y||^2 + \lambda_1 ||w||_1 \]
  - Reduce the magnitude of the weight parameters corresponding to possibly noisy features - $L_2$ and $L_1$ regularization
    \[ f(w) = \frac{1}{2} ||Xw - y||^2 + \lambda_2 ||w||^2 \]
Regularization - Motivation

- Select a subset of features - *L*1 regularization

\[ f(w) = \frac{1}{2}||Xw - y||^2 + \lambda_1||w||_1 \]

- Reduce the magnitude of the weight parameters corresponding to possibly noisy features - *L*2 and *L*1 regularization

\[ f(w) = \frac{1}{2}||Xw - y||^2 + \lambda_2||w||^2 \]
Regularization - Definition

- Regularization is a method that helps in preventing overfitting
- It controls the model complexity
- Small values for the weights leads to a simpler model
- A simpler model is less prone to overfitting
- It penalizes the objective function to avoid the model from closely matching possibly noisy data points
Regularization - Definition

Degree 1

Degree 4

Degree 15
Consider the following L2 regularized least square objective function

\[ f(w) = \frac{1}{2} \|Xw - y\|^2 + \lambda_2 \|w\|^2 \]

How does \( \lambda_2 \) affect the decision boundary?
Regularization - Exercise

- Consider the following L2 regularized least square objective function

\[ f(w) = \frac{1}{2}||Xw - y||^2 + \lambda_2||w||^2 \]

- How does \( \lambda_2 \) affect the decision boundary?
  - \( \lambda_2 \) controls a trade off between fitting the training set well and keeping the weights small
  - Large \( \lambda_2 \) can lead to underfitting (a more linear, simple model)
  - Small \( \lambda_2 \) can lead to overfitting (a more complicated model - larger range of values for the parameters)
Regularization - Exercise

• Consider the following L2 regularized least square objective function

\[ f(w) = \frac{1}{2}||Xw - y||^2 + \lambda_2||w||^2 \]

• How does \( \lambda_2 \) affect the decision boundary?
Radial Basis Function
• Observe the following dataset with two features $X$ and $Y$

• Can we fit a linear regression that separates the two classes (blue and red) sufficiently?
• One approach is to transform the features into a new space where the data is linearly separable.
We transform the data to a higher dimensional space
We can then separate the higher dimensional data using a linear plane.
RBF Basis

- Given $X \in \mathbb{R}^{N \times D}$, transform $X$ to $Z \in \mathbb{R}^{N \times N}$ where

$$Z_{ij} = \exp\left(-\frac{||X_i - X_j||^2}{2\sigma^2}\right)$$

where $\sigma$ controls the influence of nearby points

- Intuitively, $Z_{ij}$ is a similarity value between sample $i$ and sample $j$
RBF Basis - Pros & Cons

- **Pros**
  - Non-linear decision boundary
  - For some applications, such similarity-based features are very robust

- **Cons**
  - Non-parametric - grows with \( N \)
  - Can lead to overfitting
• Consider the following dataset

\[ X = \begin{bmatrix} 3 & 5 \\ 1 & 2 \\ 4 & 6 \end{bmatrix} \]

• Transform the dataset into the RBF space with \( \sigma = 1 \)

\[ X_{rbf} = ? \]
• Least square function

\[ f(w) = \|Xw - y\|^2 \]

• Transform this objective function to one that uses RBF features
• Least square function

\[ f(w) = \|Xw - y\|_2^2 \]

• Transform this objective function to one that uses RBF features

\[ f(w) = \|X_{rbf} w - y\|_2^2 \]
RBF Basis - Exercises

- Least square function
  
  \[ f(w) = \|Xw - y\|^2 \]

- Transform this objective function to one that uses RBF features
  
  \[ f(w) = \|X_{\text{rbf}}w - y\|^2 \]
• Least square function

\[ f(w) = ||Xw - y||^2_2 \]

• Transform this objective function to one that uses RBF features

\[ f(w) = ||X_{rbf}w - y||^2_2 \]

• Recall that RBF can lead to a model that is too complicated for the dataset - potentially causing overfitting

• Regularization helps against overfitting

• Add the $L1$ and $L2$ regularization terms to $f(w)$
• Least square function

\[ f(w) = \|Xw - y\|_2^2 \]

• Transform this objective function to one that uses RBF features

\[ f(w) = \|X_{rbf}w - y\|_2^2 \]

• Recall that RBF can lead to a model that is too complicated for the dataset - potentially causing overfitting

• Regularization helps against overfitting

• Add the L1 and L2 regularization terms to \( f(w) \)

\[ f(w) = \|X_{rbf}w - y\|_2^2 + \lambda_1\|w\|_1 + \lambda_2\|w\|_2^2 \]

• Suggest one way to choose the values for \( \lambda_1 \) and \( \lambda_2 \)
RBF Basis - Exercises

- Least square function
  \[ f(w) = \|Xw - y\|^2_2 \]

- Transform this objective function to one that uses RBF
  \[ f_{rbf}(w) = \|X_{rbf}w - y\|^2_2 \]

- Recall that RBF can lead to a model that is too complicated for the dataset - potentially causing overfitting

- Regularization helps against overfitting

- Add the L1 and L2 regularization terms to \( f(w) \)
  \[ f_{rbf}(w) = \|X_{rbf}w - y\|^2_2 + \lambda_1\|w\|_1 + \lambda_2\|w\|^2_2 \]

- How do we choose the values for \( \lambda_1 \) and \( \lambda_2 \) ?
RBF Basis - Exercises

• Least square function

\[ f(w) = \|Xw - y\|_2^2 \]

• Transform this objective function to one that uses RBF

\[ f_{rbf}(w) = \|X_{rbf}w - y\|_2^2 \]

• Recall that RBF can lead to a model that is too complicated for the dataset - potentially causing overfitting

• Regularization helps against overfitting

• Add the $L1$ and $L2$ regularization terms to $f(w)$

\[ f_{rbf}(w) = \|X_{rbf}w - y\|_2^2 + \lambda_1\|w\|_1 + \lambda_2\|w\|_2^2 \]

• How do we choose the values for $\lambda_1$ and $\lambda_2$?

  • Cross-validation
Given the regularized RBF model,

\[ f_{rbf}(w) = \frac{1}{2} \| X_{rbf} w - y \|_2^2 + \frac{\lambda}{2} \| w \|_2^2 \]

solve for \( w \).
Given the regularized RBF model,

\[ f_{\text{rbf}}(w) = \frac{1}{2} \| X_{\text{rbf}} w - y \|_2^2 + \frac{\lambda}{2} \| w \|_2^2 \]

solve for \( w \)
Given the regularized RBF model,

\[ f_{rbf}(w) = \frac{1}{2} \| X_{rbf} w - y \|_2^2 + \frac{\lambda_2}{2} \| w \|_2^2 \]

solve for \( w \)

\[ w = (X_{rbf}^T X_{rbf} + I \lambda_2)^{-1} X_{rbf}^T y \]
Robust Regression
Weighted Least-Squares

- Least-squares estimates assumes that the residuals \( (w^T x_i - y_i) \) are normally distributed.
- Outliers violate this assumption which can cause poor least-square models.
Weighted Least-Squares

- Weighted least squares error assigns a weight $z_i$ to each training example $x_i$

$$f(w) = \frac{1}{2} \sum_{i=1}^{n} z_i (w^T x_i - y_i)^2$$

- To reduce the influence of outliers on the decision boundary, assign lower $z_i$ to the outlier observations
Weighted Least-Squares

To compute \( w \) that minimizes \( f(w) \) we need to derive the partial derivatives of \( f(w) \) w.r.t each \( w_j \) and update \( w_j \) using gradient descent.

Given the one-dimensional weighted least square error function

\[
f(w) = \frac{1}{2} \sum_{i=1}^{n} z_i (w x_i - y_i)^2
\]

derive \( \frac{\partial f(w)}{\partial w} \).
Weighted Least-Squares

- Weighted least square error function

\[ f(w) = \frac{1}{2} \sum_{i=1}^{n} z_i (w^T x_i - y_i)^2 \]
Robust regression - lasso

- Problem: weighted least squares requires us to know the identity of the outliers
- We can change the least square error function

\[
f(w) = \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2
\]

to the L1-norm error function that is robust to outliers

\[
f(w) = \sum_{i=1}^{n} |y_i - w^T x_i|
\]
Robust regression - lasso

- Problem: the L1 norm is not differentiable
Robust regression - lasso

- Problem: the L1 norm is not differentiable
- Solution: approximate the L1 norm and obtain a differential objective function
- We can change the L1-norm objective function

\[ f(w) = \sum_{i=1}^{n} |y_i - w^T x_i| \]

to the approximated objective function that is differentiable

\[ f(w) = \sum_{i=1}^{n} \sqrt{(y_i - w^T x_i)^2 + \epsilon} \]

- \(|r| \approx \sqrt{r^2 + \epsilon}\) where \(\epsilon\) is a small value
Robust Regression - Exercise

- Given the approximation

\[ f(w) = \sum_{i=1}^{n} \sqrt{(y_i - w^T x_i)^2 + \epsilon} \]

derive \( \frac{\partial f(w)}{\partial w_j} \)
Robust Regression - Exercise

• Given the approximation

\[ f(w) = \sum_{i=1}^{n} \sqrt{(y_i - w^T x_i)^2} + \epsilon \]

Let \( r_i = y_i - w^T x_i \)

\[ \frac{\partial \sqrt{r^2 + \epsilon}}{\partial r} = \frac{2r}{2\sqrt{r^2 + \epsilon}} = \frac{r}{\sqrt{r^2 + \epsilon}} \]

\[ \frac{\partial f}{\partial w_j} = -\sum_{i=1}^{n} \frac{(y_i - w^T x_i)x_{ij}}{\sqrt{(y_i - w^T x_i)^2} + \epsilon} \]

Let \( v_i = \frac{y_i - w^T x_i}{\sqrt{(y_i - w^T x_i)^2} + \epsilon} \)

\[ \nabla f(w) = -X^T v \]
Gradient Descent with minFunc
Gradient Descent

- Given the least square error function

\[ f(w) = ||Xw - y||^2_2 \]

we want our model prediction \( Xw \) to be as close to \( y \) as possible.

- The minimum is attained when \( \nabla_w f(w) = 0 \)

- We can minimize \( f(w) \) by using gradient descent
Gradient Descent

- Gradient descent is an iterative method
- The idea is to compute a better estimation of $w$ each iteration
- Each iteration, we update $w_i$ as follows

$$w_i = w_i - \alpha \frac{\partial f(w)}{\partial w_i}$$

where $\alpha$ is the step size
Gradient Descent

![Graph showing gradient descent](image)

- **J(w)**: Cost function
- **w**: Weight
- **Initial weight**: Starting point
- **Gradient**: Direction of steepest ascent
- **Global cost minimum**: Minimum point
- **J_{\text{min}}(w)**: Minimum cost function value
Gradient Descent

In the file robustRegression.m

```matlab
21 % Solve least squares problem
22 w = findMin(@funObj,w,100,X,y);
23 model.w = w;
24 model.predict = @predict;
25 end
26
27 function [yhat] = predict(model,Xtest)
28 w = model.w;
29 yhat = Xtest*w;
30 end
31
32 function [f,g] = funObj(w,X,y)
33 end

• What should we write under funObj to minimize,

\[ f(w) = \sum_{i=1}^{n} \sqrt{(y_i - w^T x_i)^2 + \epsilon} \]
Gradient Descent

\[ f(w) = \sum_{i=1}^{n} \sqrt{(y_i - w^T x_i)^2} + \epsilon \]

```matlab
34 function [f,g] = funObj(w,X,y)
35 % Compute residual
36 r = X*w - y;
37
38 % Compute objective function
39 f = sum(sqrt(r.^2 + epsilon));
40
41 % Compute sign-of-residual approximation
42 v = r./(sqrt(r.^2 + epsilon));
43
44 % Compute gradient
45 g = X'*v;
46 end```