Linear Regression and Gradient Descent

Prof.Mingkui Tan

South China University of Technology
Southern Artificial Intelligence Laboratory(SAIL)

September 6, 2017

Content

Contents

Machine Learning Setup

- Inputs
  Input space $\mathbf{X} = \mathbb{R}^m$
  feature,covariants,predictors,etc.
- Outputs
  Output space:$\mathbf{Y}$
  many different types of predictions.
- Goal:Learn a hypothesis/model
  $h : \mathbf{X} \mapsto \mathbf{Y}$

Supervised Learning

- Given set of input,output pairs

$$D = (\mathbf{x}_1, y_1)...(\mathbf{x}_n, y_n)$$

- Learn the "best" model based on D.
- Predict $\hat{y}$ for unseen x based on h($\mathbf{x}$).

Linear Regression

Hypothesis:

- there is some good linear function of input to find prediction $\hat{y}$

Assumption:

- Parametric model $h(\mathbf{x}; \mathbf{w})$ is a linear function of $\mathbf{x}$.
- We select this by choosing $\mathbf{w}$.

Linear Regression:Common

Learn $h(\mathbf{x};\mathbf{w})$ with

- Parameters:$\mathbf{w} \in \mathbb{R}^{m-1}, w_0 \in \mathbb{R}$
- Input:$\mathbf{x}$ where $x_j \in \mathbb{R} \quad for \quad j \in 1, ...(m-1)$ features
- Model Function:

$$\begin{aligned} h(\mathbf{x}; w_0, \mathbf{w}) &= w_0 + w_1 x_1 + ... + w_{m-1} x_{m-1} \\ &= \sum_{j=1}^{m-1} w_j x_j + w_0 \\ &= \mathbf{w}^\top \mathbf{x} + w_0 \end{aligned}$$

Performance Measure for Regression

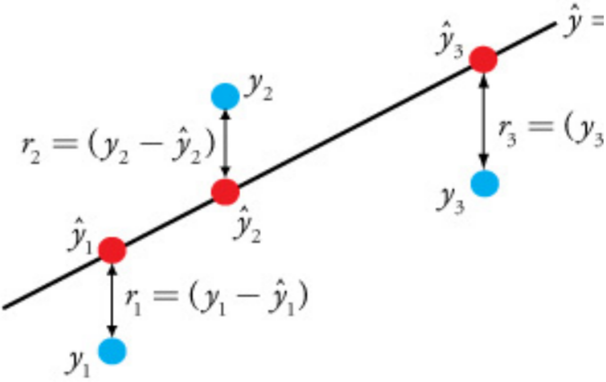
- What makes a good model?
- Squared loss

$$
\begin{aligned}
\mathcal{L}_D(\mathbf{w}) &= \frac{1}{2}\sum_{i=1}^{n}(y_i - h(\mathbf{x}_i; \mathbf{w}))^2 \\
&= \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.
\end{aligned}
$$

- Training:find minimizer of this loss(least squares)

$$
\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}_D(\mathbf{w})
$$

Residual Terms



Contributing loss terms for 1D regression.

Contents

Machine Learning

Training Procedure

- Define a loss criterion $\mathcal{L}$
- Identify a set of hypotheses $\mathbf{h}(\mathbf{x}; \mathbf{w})$
- Pick the best $\mathbf{w}^*$ by minimizing a loss function $\mathcal{L}_D(\mathbf{w})$,i.e

$$\arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$$

Learning is done through optimization.

Optimization

Miningmizing loss function is vary hard...
Therefore optimization is central part of machine learning.

- Gradient descent
- Linear and quadratic programming
- Newton-like methods
- Stochastic optimization
- Lots more...

Main Tool:Gradients

Typical case (with possibly parameterized g)

$$\boldsymbol{\mathcal{L}}(\mathbf{w}) : \mathbb{R}^n \mapsto \mathbb{R}$$

Gradient (vector of partial derivatives)

$$\frac{\partial \boldsymbol{\mathcal{L}}(\mathbf{w})}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{L}}(w_1)}{\partial w_1} \\ \frac{\partial \boldsymbol{\mathcal{L}}(w_2)}{\partial w_2} \\ \vdots \\ \frac{\partial \boldsymbol{\mathcal{L}}(w_n)}{\partial w_n} \end{bmatrix}$$

(We will always write as column vectors)

Gradient Descent

Minimize loss by repeated gradient steps(when no closed form):

- Compute gradient of loss with respect to parameters $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}}$
- Update parameters with rate η

$$\mathbf{w}' \rightarrow \mathbf{w} - \eta \frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}}$$

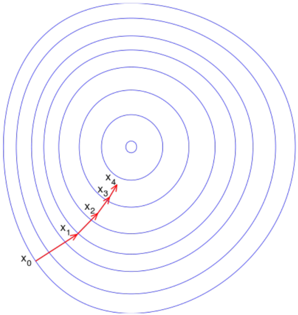
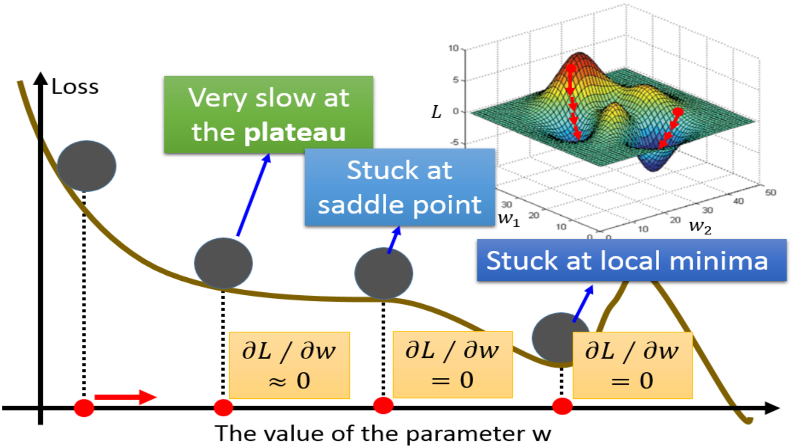


Figure: Gradient steps on a simple m = 2 loss function.

More Limitation of Gradient Descent



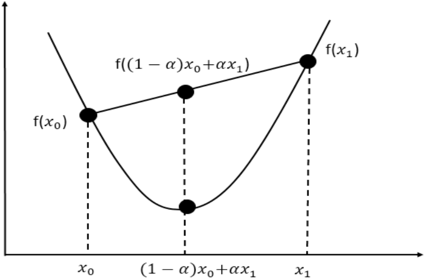
Convex Function

If the cost fuction is convex,then a locally optimal point is globally optimal

D ,a domain in $\mathbb{R}^n$.

A convex function $f : D \mapsto \mathbb{R}$ is one that satisfies,for any x_0 and x_1 in D:

$$f((1-\alpha)x_0 + \alpha x_1) \leq (1-\alpha)f(x_0) + \alpha f(x_1)$$



Gradient descent algorithm for Regression

The loss function $\mathcal{L}(\mathbf{w})$:

$$\sum_{i=1}^{n}(y_i - h(\mathbf{x}_i;\mathbf{w}))^2$$

Gradient descent algorithm for Regression

To minimize a cost function $\mathcal{L}(\mathbf{w})$ use the iterative update

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \frac{\partial \mathcal{L}(\mathbf{w}_t)}{\partial \mathbf{w}_t}$$

where η is the learning rate and $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}}$ is :

$$\begin{aligned}
\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} &= -\sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)\mathbf{x}_i \\
&= -\sum_{i=1}^{n} y_i \mathbf{x}_i + (\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top)\mathbf{w}
\end{aligned}$$

Matrix Version

$$D = \{(x_1, y_1), ...(x_n, y_n)\}$$

- Inputs: $\mathbf{X} \in \mathbb{R}^{n \times m}$

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & ... & x_{1,m} \\ x_{2,1} & x_{2,2} & ... & x_{2,m} \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ x_{n,1} & x_{n,2} & ... & x_{n,m} \end{bmatrix}$$

Matrix Version

- Outputs(target vector):$\mathbf{y} \in \mathbb{R}^{n \times 1}$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Least Square Loss(Matrix Form)

Same as above,but using matrix caculus identities

$$
\begin{aligned}
L(\mathbf{w}) &= \frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top(\mathbf{y}-\mathbf{X}\mathbf{w}) \\
&= (\mathbf{y}^\top\mathbf{y} - 2\mathbf{w}^\top\mathbf{X}^\top\mathbf{y} + \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}) \\
\tfrac{\partial L(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{2}\frac{\partial \mathbf{y}^\top\mathbf{y}}{\partial \mathbf{w}} - \frac{\partial 2\mathbf{w}^\top\mathbf{X}^\top\mathbf{y}}{\partial \mathbf{w}} + \frac{\partial \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}}{\partial \mathbf{w}} \\
&= \frac{1}{2}(-2\mathbf{X}^\top\mathbf{y} + (\mathbf{X}^\top\mathbf{X} + (\mathbf{X}^\top\mathbf{X})^\top)\mathbf{w}) \\
&= -\mathbf{X}^\top\mathbf{y} + \mathbf{X}^\top\mathbf{X}\mathbf{w}
\end{aligned}
$$

Least Square Loss(Matrix Form)

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = -\mathbf{X}^\top \mathbf{y} + \mathbf{X}^\top \mathbf{X}\mathbf{w}$$

Set to 0,and solve for optimal parameters $\mathbf{w}^*$

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y} = \arg\min_{\mathbf{w}} L_D(\mathbf{w})$$

(Known as Moore-Penrose pseudo-inverse

$$(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top$$

Generalization of inverse for non-square matrix).

THANK YOU!