Machine learning - HT 2016
2. Linear Regression

Varun Kanade

University of Oxford
January 22, 2016

Outline

Supervised Learning Setting

- Data consists of input- output pairs
- Inputs (also covariates, independent variables, predictors, features)
- Output (also variates, dependent variable, targets, labels)

Goals

- Understand the supervised learning setting
- Understand linear regression (aka least squares)
- Derivation of the least squares estimate

Why study linear regression?

- "Least squares" is at least 200 years old (Legendre, Gauss)
- Francis Galton: Regression to mediocrity (1886)

- Often real processes can be approximated by linear models
- More complicated models require understanding linear regression

- Closed form analytic solutions can be obtained
- Many key notions of machine learning can be introduced

A toy example

Want to predict commute time into city centre

What variables would be useful?

- Distance to city centre
- Day of the week

Data

dist (km)	**day**	**commute time** (min)
2.7	fri	25
4.1	mon	33
1.0	sun	15
5.2	tue	45
2.8	sat	22

Linear Models

Suppose the input is a vector $\mathbf{x} \in \mathbb{R}^n$ and the output is $y \in \mathbb{R}$.

We have data $\langle \mathbf{x}_i, y_i \rangle_{i=1}^m$

Notation: dimension n, size of dataset m, column vectors

Linear model:

$$y = w_0 + x_1 w_1 + \cdots + x_n w_n + \textit{noise}$$

Linear Models

Linear model:

$$y = w_0 + x_1 w_1 + \cdots + x_n w_n + \textit{noise}$$

Input encoding: mon-sun has to be converted to a number

- monday: 0, tuesday: 1, . . . , sunday: 6 (This is a bad encoding Use 7 binary features instead)
- 0 if weekend, 1 if weekday

Assume $x_1 = 1$ for all data. So model can be succinctly represented as

$$y = \mathbf{x} \cdot \mathbf{w} + \text{noise} = \hat{y} + \text{noise}$$

$\bar{x} \in \mathbb{R}^n$, $\bar{w} \in \mathbb{R}^n$, $\quad \bar{x} \cdot \bar{w} = \sum_{i=1}^{n} x_i \cdot w_i$

Learning Linear Models

Data: $\langle (\mathbf{x}_i, y_i) \rangle_{i=1}^{m}$

Model parameter $\mathbf{w}$

Training phase: (learning/estimation of $\mathbf{w}$)

$$\langle (x_i, y_i) \rangle_{i=1}^{m} \xrightarrow[\text{DATA}]{} \boxed{\text{ALGORITHM}} \longrightarrow w \text{ (estimate)}$$

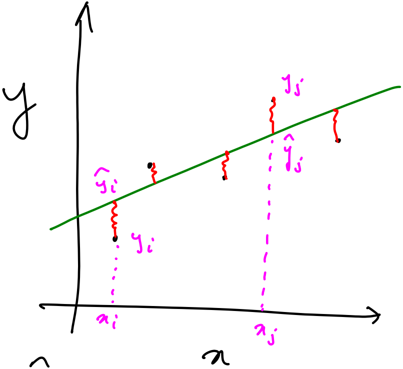
Testing phase: (predict $\hat{y}_{m+1} = \mathbf{x}_{m+1} \cdot \mathbf{w}$)

should keep aside some data for testing

$$\hat{y}(x) = w_0 + xw_1$$

$$L(\mathbf{w}) = L(w_0, w_1) = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{m} (w_0 + x_i w_1 - y_i)^2$$

LOSS FUNCTION
OBJECTIVE
ENERGY



$\hat{y} = w_0 + w_1 x$

RSS : Residual Sum of Squares

$$\hat{y}(x) = w_0 + x w_1$$

$$L(\mathbf{w}) = L(w_0, w_1) = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{m} (w_0 + x_i w_1 - y_i)^2$$

$$\frac{1}{2m} \frac{\partial L}{\partial w_0} = \frac{1}{m} \sum_{i=1}^{m} (w_0 + w_1 x_i - y_i) = 0$$

i.e. $$w_0 + w_1 \left(\frac{1}{m} \sum_{i=1}^{m} x_i \right) = \frac{1}{m} \sum_{i=1}^{m} y_i \quad (1)$$

NORMAL EQUATIONS

$$\frac{1}{2m} \frac{\partial L}{\partial w_1} = \frac{1}{m} \sum_{i=0}^{m} (w_0 + w_1 x_i - y_i) x_i = 0$$

i.e. $$w_0 \left(\frac{1}{m} \sum_{i=1}^{m} x_i \right) + w_1 \left(\frac{1}{m} \sum_{i=1}^{m} x_i^2 \right) = \frac{1}{m} \sum_{i=1}^{m} x_i y_i \quad (2)$$

Solving for w_1:

$$w_1 = \frac{\frac{1}{m} \sum_{i=1}^{m} x_i y_i - \left(\frac{1}{m} \sum_{i=1}^{m} x_i \right) \left(\frac{1}{m} \sum_{i=1}^{m} y_i \right)}{\frac{1}{m} \sum_{i=1}^{m} x_i^2 - \left(\frac{1}{m} \sum_{i=1}^{m} x_i \right)^2}$$

Linear Regression: General Case

Recall that the linear model is

$$\hat{y}_i = \sum_{j=1}^{n} x_{ij} w_j$$

where we assume that $x_{i1} = 1$ for all $\mathbf{x}_i$. So w_1 is the bias or offset term.

Expressing everything in matrix notation

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$$

Here we have $\hat{\mathbf{y}} \in \mathbb{R}^{m \times 1}$, $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $\mathbf{w} \in \mathbb{R}^{n \times 1}$

$$\begin{array}{c} \hat{\mathbf{y}}_{m\times 1} \\ \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} \end{array} = \begin{array}{c} \mathbf{X}_{m\times n} \\ \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_m^T \end{bmatrix} \end{array} \begin{array}{c} \mathbf{w}_{n\times 1} \\ \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \end{array} = \begin{array}{c} \mathbf{X}_{m\times n} \\ \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \end{array} \begin{array}{c} \mathbf{w}_{n\times 1} \\ \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \end{array}$$

Back to toy example

dist (km)	**weekday?**	**commute time** (min)
2.7	1 (fri)	25
4.1	1 (mon)	33
1.0	0 (sun)	15
5.2	1 (tue)	45
2.8	0 (sat)	22

We have $m = 5$, $n = 3$ and so we get

$$\mathbf{y} = \begin{bmatrix} 25 \\ 33 \\ 15 \\ 45 \\ 22 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & 2.7 & 1 \\ 1 & 4.1 & 1 \\ 1 & 1.0 & 0 \\ 1 & 5.2 & 1 \\ 1 & 2.8 & 0 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Suppose we get $\mathbf{w} = [6, 6.5, 2]^T$. Then our predictions would be

$$\hat{\mathbf{y}} = \begin{bmatrix} 25.55 \\ 34.65 \\ 12.5 \\ 41.8 \\ 24.2 \end{bmatrix}$$

Linear Prediction

Suppose someone lives 4.8km from city centre, how long would they take to get in on a Wednesday?

We can write $\mathbf{x}_{\text{new}} = [1, 4.8, 1]^T$ and then compute

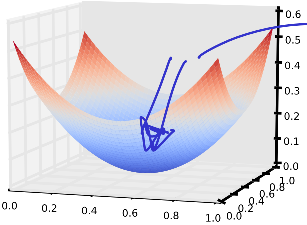
$$\hat{y}_{\text{new}} = \begin{bmatrix} 1 & 4.8 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 6.5 \\ 2 \end{bmatrix} = 39.2 \text{ minutes}$$

Interpreting regression coefficients

⚠ Using regression coefficients to explain is not straight-forward when there are associations among variables

Minimizing the Squared Error

$$L(\mathbf{w}) = \sum_{i=1}^{m} (\mathbf{x}_i^T \mathbf{w} - y_i)^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y})$$



Want to find the minimum

Finding Optimal Solutions using Calculus

$$L(\mathbf{w}) = \sum_{i=1}^{m}(\mathbf{x}_i^T\mathbf{w} - y_i)^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y})$$

$$Xw - y = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_m^T \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} - \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} x_1^T w - y_1 \\ x_2^T w - y_2 \\ \vdots \\ x_m^T w - y_m \end{pmatrix}$$

Can write n equations

$$\frac{\partial L}{\partial w_1} = \cdots = 0$$

$$\vdots$$

$$\frac{\partial L}{\partial w_n} = \cdots = 0$$

INSTEAD LEARN MATRIX DIFFERENTIATION

Differentiating Matrix Expressions

$L(\mathbf{w}) = \sum_{i=1}^{m}(\mathbf{x}_i^T\mathbf{w} - y_i)^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y})$

Rules

(i) $\nabla_{\mathbf{w}}\mathbf{c}^T\mathbf{w} = \mathbf{c}$ (Linear Form: $\sum_i c_i w_i$)

(ii) $\nabla_{\mathbf{w}}\mathbf{w}^T\mathbf{A}\mathbf{w} = \mathbf{A}\mathbf{w} + \mathbf{A}^T\mathbf{w}$ ($= 2\mathbf{A}\mathbf{w}$ for symmetric $\mathbf{A}$) (Quadratic Form $\sum_i\sum_j A_{ij} w_i w_j$)

$$L(w) = (w^TX^T - y^T)(Xw - y)$$

(SCALAR)

$$= w^TX^TXw - 2y^TXw + y^Ty$$

set $\nabla_w L = 2X^TXw - 2X^Ty = 0$

$$(X^TX)w = X^Ty$$

$$w = (X^TX)^{-1}(X^Ty)$$

Solution to Linear Regression

$L(\mathbf{w}) = \sum_{i=1}^{m} (\mathbf{x}_i^T \mathbf{w} - y_i)^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y})$

$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

What if we had one-hot encoding for days?

- 7 binary features, exactly one of them is 1
- $x_1 = 1$, $x_2 =$dist, $x_3 =$ mon?, $x_4 =$ tue?, . . . , $x_9 =$ sun?

$x_3 + x_4 + x_5 + \cdots + x_9 = 1 = x_1$

'Linear dependence'

This means: Rank $(X) < n-1$

$\therefore$ $X^T X$ is singular

One solution: DROP one feature, say SUNDAY. We will see other options next time

Linear Regression as Solving Noisy Linear Systems

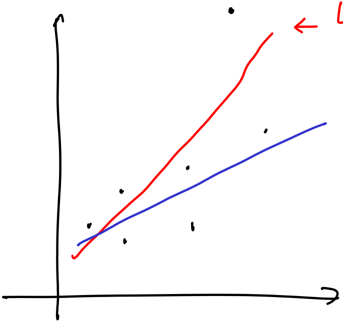
Without noise: $\mathbf{X}\mathbf{w} = \mathbf{y}$

Linear regression finds $\hat{\mathbf{y}}$ that makes system $\mathbf{X}\mathbf{w} = \hat{\mathbf{y}}$ feasible

And $\hat{\mathbf{y}}$ is closest to $\mathbf{y}$ in Euclidean distance

Other Loss Functions

$$L(\mathbf{w}) = \sum_{i=1}^{m} |\mathbf{x}_i^T \mathbf{w} - y_i|$$



← Least squares fit is bad with outliers

Using sum of dist, instead of squared dist, less susceptible

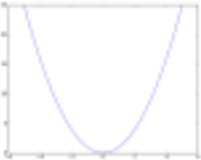
No closed form solution

We will see how to solve using linear programming

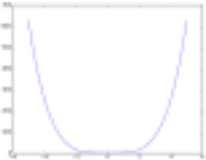
Loss Functions

- Consider $\ell : \mathbb{R}^+ \to \mathbb{R}^+$ satisfying
  1. $\ell(0) = 0$
  2. ℓ is non-decreasing

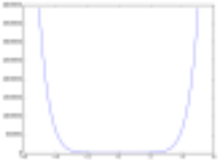
- $$L(\mathbf{w}) = \sum_{i=1}^{m} \ell(|\mathbf{x}_i^T \cdot \mathbf{w} - y_i|)$$



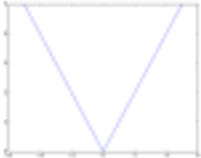
$\ell(z) = |z|^2$



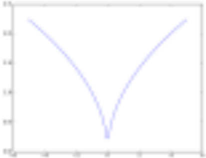
$\ell(z) = |z|^4$



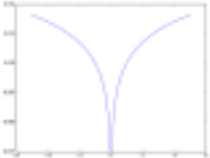
$\ell(z) = |z|^{10}$



$\ell(z) = |z|$

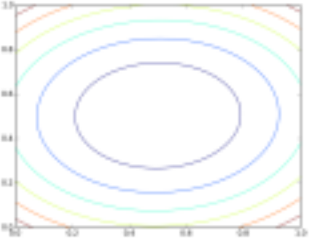


$\ell(z) = \sqrt{|z|}$

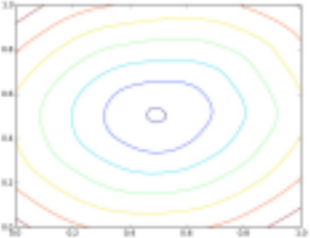


$\ell(z) = |z|^{0.01}$

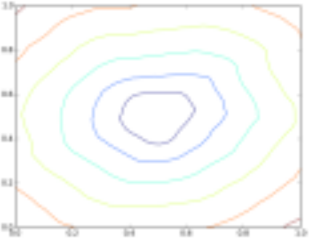
Optimization with different loss functions



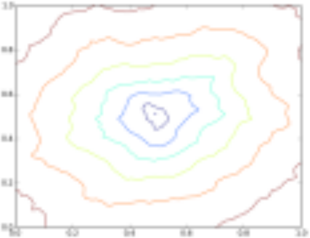
$$L(\mathbf{w}) = \sum |\hat{y}_i - y_i|^2$$



$$L(\mathbf{w}) = \sum |\hat{y}_i - y_i|$$



$$L(\mathbf{w}) = \sum \sqrt{|\hat{y}_i - y_i|}$$



$$L(\mathbf{w}) = \sum |\hat{y}_i - y_i|^{0.1}$$

Probabilistic Modelling

- Linear Model: $y = \mathbf{x}^T \mathbf{w}^* +$ *noise* (for some $\mathbf{w}^*$)
- $\mathbb{E}[y \mid \mathbf{x}, \mathbf{w}^*] = \mathbf{x}^T \mathbf{w}^*$
- Data $\langle (\mathbf{x}_i, y_i) \rangle_{i=1}^m$
- Unbiased estimator $\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$