Lecture 2: Linear prediction

EMAT31530/Jan 2018/Raul Santos-Rodriguez

Have a look at ...

... Russell and Norvig (Ch. 18.6)

... Hastie, Tibshirani, Friedman. The elements of statistical learning, (Ch. 3 and 4)

... Python: `http://scikit-learn.org/`

... Java: `http://www.cs.waikato.ac.nz/ml/weka/`

Outline

This lecture introduces us to the topic of supervised learning. Here the data consists of input-output pairs. Inputs are also often referred to as predictors and features; while outputs are known as variates and labels. The goal of the lecture is for you to

- Understand the supervised learning setting.
- Understand linear regression (least squares).
- Understand how to apply linear regression models to make predictions.

Supervised learning setting

We are given a training dataset of n instances of input-output pairs $\{\mathbf{x}_{1:n}, \mathbf{y}_{1:n}\}$. Each input $\mathbf{x}_i \in \mathbb{R}^{1\times d}$ is a vector with d attributes. The output (target) is $\mathbf{y}_i$. Depending on the nature of $\mathbf{y}_i$, there are different types of prediction tasks:

- Regression: y is a real number (e.g., housing prices)
- Classification: y is yes/no (binary), one of K labels (multiclass), subset of K labels (multilabel)
- Ranking: y is a permutation (e.g., ranking web pages)

Supervised learning setting

We are given a training dataset of n instances of input-output pairs $\{\mathbf{x}_{1:n}, \mathbf{y}_{1:n}\}$. Each input $\mathbf{x}_i \in \mathbb{R}^{1\times d}$ is a vector with d attributes. The output (target) is $\mathbf{y}_i$. Depending on the nature of $\mathbf{y}_i$, there are different types of prediction tasks:

- Regression: y is a real number (e.g., housing prices)
- Classification: y is yes/no (binary), one of K labels (multiclass), subset of K labels (multilabel)
- Ranking: y is a permutation (e.g., ranking web pages)

In the context of classification tasks, the model is called a classifier, and y is called a label (sometimes class, category, or tag). Note that the dichotomy of prediction tasks are not meant to be formal definitions, but rather to provide intuitions. For instance, binary classification could technically be seen as a regression problem if the labels are -1 and $+1$.

Why linear?

- Many real processes can be approximated with linear models.
- Linear regression often appears as a module of larger systems.
- Linear problems can be solved analytically.
- Linear prediction provides an introduction to many of the core concepts of machine learning.

Why linear?

Why linear?

- Many real processes can be approximated with linear models.
- Linear regression often appears as a module of larger systems.
- Linear problems can be solved analytically.
- Linear prediction provides an introduction to many of the core concepts of machine learning.

A linear predictor function is a linear function (linear combination) of a set of coefficients and explanatory variables (independent variables), whose value is used to predict the outcome of a dependent variable. Functions of this sort are standard in linear regression, where the coefficients are termed regression coefficients. However, they also occur in various types of linear classifiers (e.g. logistic regression, perceptrons, support vector machines, and linear discriminant analysis), as well as in various other models, such as principal component analysis and factor analysis. In many of these models, the coefficients are referred to as "weights" or "parameters".

Energy demand prediction

Day	Temperature outside	People inside building	Energy requirement
1	25	2	5
2	12	42	25
3	11	31	22
4	15	35	18

Given the training $\{\mathbf{x}_{1:n}, y_{1:n}\}$, we would like to learn a model of how the inputs affect the output. Given this model and a new value of the input $\mathbf{x}_{n+1}$, we can use the model to make a prediction $\hat{y}(\mathbf{x}_{n+1})$.

Day	Temperature outside	People inside building	Energy requirement
5	10	25	?

[De Freitas, 2012], `http://www.auto-grid.com/`

Medical diagnosis: prostate cancer

Goal: Predict a prostate-specific antigen (log of lpsa) from a number of clinical measures in men who are about to receive a radical prostatectomy.

For each patient the inputs are:

- Log cancer volume (lcavol)
- Log prostate weight (lweight)
- Age
- Log of the amount of benign prostatic hyperplasia (lbph)
- Seminal vesicle invasion (svi)
- Log of capsular penetration (lcp)
- Gleason score (gleason)
- Percent of Gleason scores 4 or 5 (pgg45)

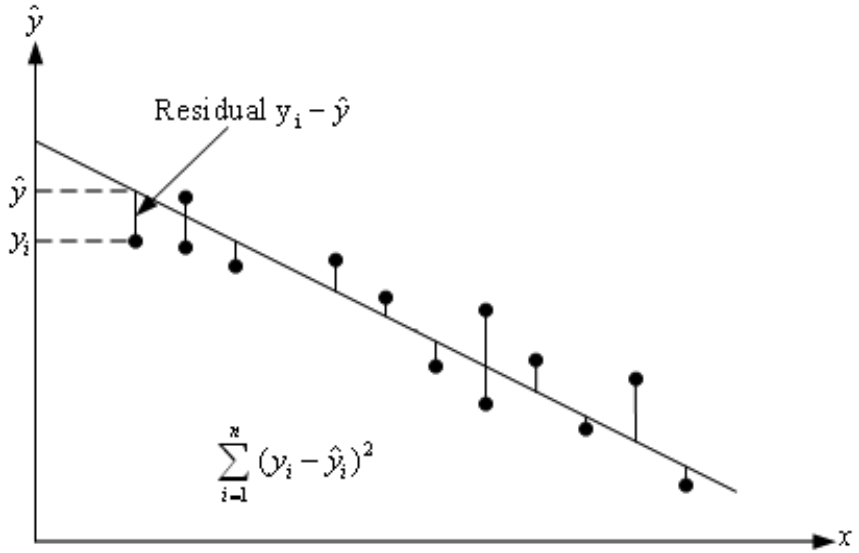
Which inputs are more important?

[Hastie, Tibshirani, Friedman]

Linear prediction: 1-d

$$\hat{y}(x_i) = \theta_0 + x_i\theta_1$$

$$J(\theta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \theta_0 - x_i\theta_1)^2$$

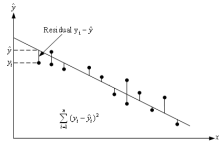


The residual is the amount by which prediction overshoots the target

Linear prediction: 1-d

$$\hat{y}(x_i) = \theta_0 + x_i\theta_1$$

$$J(\theta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \theta_0 - x_i\theta_1)^2$$



The residual is the amount by which prediction overshoots the target

A loss function $J(\theta)$ measures how unhappy you would be if you used θ to make a prediction on x when the correct output is y.

In general, the linear model is expressed as follows:

$$\hat{y}_i = \sum_{j=1}^{d} x_{ij}\theta_j,$$

where we assume that $x_{i1} = 1$ so that θ_1 is the intercept of the line with the vertical axis. θ_1 is known as the bias or offset. In matrix form, the expression

for the linear model is,

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta},$$

with $\hat{\mathbf{y}} \in \mathbb{R}^{n\times 1}$, $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{\theta} \in \mathbb{R}^{d\times 1}$. That is,

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$$

Linear prediction

In general, the linear model is expressed as follows:

$$\hat{y}_i = \sum_{j=1}^{d} x_{ij}\theta_j,$$

where we assume that $x_{i1} = 1$ so that θ_1 is the intercept of the line with the vertical axis. θ_1 is known as the bias or offset. In matrix form, the expression for the linear model is,

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta},$$

with $\hat{\mathbf{y}} \in \mathbb{R}^{n\times 1}$, $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{\theta} \in \mathbb{R}^{d\times 1}$. That is,

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$$

A weight vector (also called a parameter vector or weights) specifies the contributions of each feature vector to the prediction.

Energy demand prediction

Day	Temperature outside	People inside building	Energy requirement
1	25	2	5
2	12	42	25
3	11	31	22
4	15	35	18

For out energy prediction example, we would form the following matrices with $n = 4$ and $d = 3$:

$$\mathbf{y} = \begin{bmatrix} 5 \\ 25 \\ 22 \\ 18 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & 25 & 2 \\ 1 & 12 & 42 \\ 1 & 11 & 31 \\ 1 & 15 & 35 \end{bmatrix}, \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Suppose that $\hat{\boldsymbol{\theta}} = \begin{bmatrix} 1 & 0 & 0.5 \end{bmatrix}^T$. Then, by multiplying $\mathbf{X}$ times $\hat{\boldsymbol{\theta}}$, we would get the following predictions on the training set:

$$\hat{\mathbf{y}} = \begin{bmatrix} 2 \\ 22 \\ 16.5 \\ 18.5 \end{bmatrix} = \begin{bmatrix} 1 & 25 & 2 \\ 1 & 12 & 42 \\ 1 & 11 & 31 \\ 1 & 15 & 35 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0.5 \end{bmatrix}$$

Energy demand prediction

Day	Temperature outside	People inside building	Energy requirement
1	25	2	5
2	12	42	25
3	11	31	22
4	15	35	18

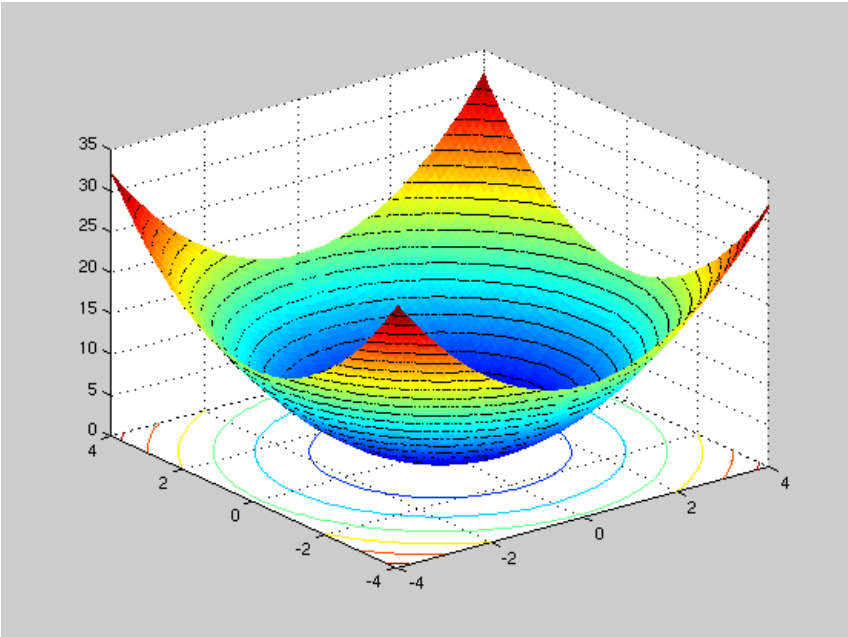
For out energy prediction example, we would form the following matrices with $n = 4$ and $d = 3$:

$$\mathbf{y} = \begin{bmatrix} 5 \\ 25 \\ 22 \\ 18 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & 25 & 2 \\ 1 & 12 & 42 \\ 1 & 11 & 31 \\ 1 & 15 & 35 \end{bmatrix}, \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Likewise, for a point that we have never seen before, say $\mathbf{x}_{n+1} = \begin{bmatrix} 12 & 20 \end{bmatrix}$, we generate the following prediction:

$$\hat{y}(\mathbf{x}_{n+1}) = \begin{bmatrix} 1 & 12 & 20 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0.5 \end{bmatrix} = 1 + 0 + 10 = 11$$

Optimisation



Our aim is to minimise the quadratic cost (loss function) between the output labels and the model predictions

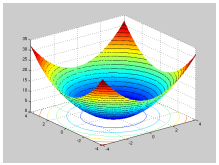
$$J(\boldsymbol{\theta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) =$$

$$= ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2 = \sum_{i=1}^{n}(y_i - \mathbf{x}_i\boldsymbol{\theta})^2$$

We have to solve

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Optimisation



Our aim is to minimise the quadratic cost (loss function) between the output labels and the model predictions

$$J(\boldsymbol{\theta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) =$$

$$= ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2 = \sum_{i=1}^{n}(y_i - \mathbf{x}_i\boldsymbol{\theta})^2$$

We have to solve

$$\boxed{\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})}$$

We see that the loss function is convex, as defined in this slide; this is true for every linear regression problem with an L_2 loss function, and implies that there are no local minima.

Optimization

$$J(\theta) = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \sum_{i=1}^{n}(y_i - \mathbf{x}_i\boldsymbol{\theta})^2$$

We will need the following results from matrix differentiation: $\frac{\partial \mathbf{A}\boldsymbol{\theta}}{\partial \boldsymbol{\theta}} = \mathbf{A}^T$ and $\frac{\partial \boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta}}{\partial \boldsymbol{\theta}} = 2\mathbf{A}^T\boldsymbol{\theta}$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} =$$

$$= \frac{\partial (\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0 = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$$

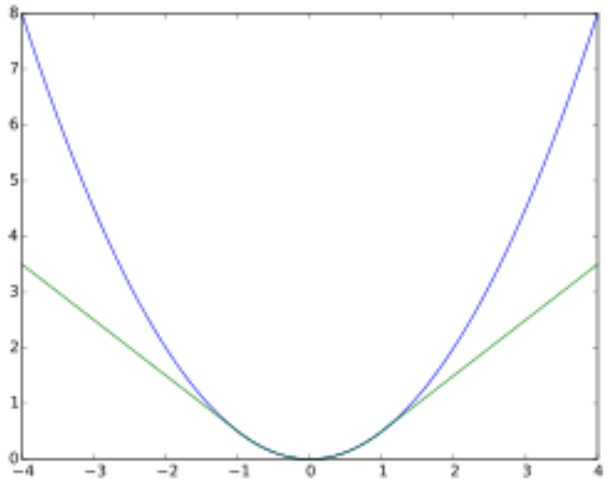
$$\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} = \mathbf{X}^T\mathbf{y}$$

$$\boxed{\hat{\boldsymbol{\theta}}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

Regression loss functions

Squared loss $J(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i \boldsymbol{\theta})^2$

Absolute deviation $J(\boldsymbol{\theta}) = \sum_{i=1}^{n} |y_i - \mathbf{x}_i \boldsymbol{\theta}|$

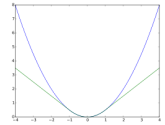


Any other options?

Regression loss functions

Squared loss $J(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i \boldsymbol{\theta})^2$

Absolute deviation $J(\boldsymbol{\theta}) = \sum_{i=1}^{n} |y_i - \mathbf{x}_i \boldsymbol{\theta}|$



Any other options?

We will need to face the fact that the equations defining loss functions will often have no closed-form solution. Instead, we will face a general optimisation search problem in a continuous parameter/weight space. Such problems usually addressed by a hill-climbing algorithm that follows the gradient of the function to be optimised. In this case, because we are trying to minimise the loss, we will use gradient descent. We choose any starting point in weight space - here, in two dimensions, a point in the (θ_0, θ_1) plane - and then move to a neighbouring point that is downhill, repeating until we converge on the minimum possible loss:

$\boldsymbol{\theta} \leftarrow$ any point in the parameter space

loop until convergence

for each θ_i in $\boldsymbol{\theta}$ do

$$\theta_i \leftarrow \theta_i - \alpha \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_i}$$

The parameter α, which we is usually called the learning rate when we are trying to minimise loss in a learning problem. It can be a fixed constant, or it can decay over time as the learning process proceeds.

Summary

We've looked at:

- Supervised learning
- Linear models
- Loss functions
- Convex optimisation

For the weekend:

- What happens if $\mathbf{X}^T\mathbf{X}$ is singular?
- Proof that, assuming normality, the maximum likelihood estimates of the weights are just the minimal least squares estimates!
- `http://www.informationweek.com/mobile/mobile-devices/google-taps-machine-learning-to-make-smartphones-smarter/d/d-id/1324090`
- `http://www.theguardian.com/culture/gallery/2015/jan/08/the-top-20-artificial-intelligence-films-in-pictures`

Next lecture

We will discuss the concepts of generalisation, regularisation and cross-validation.