UVA CS 4501 - 001 / 6501 – 007

Introduction to Machine Learning and Data Mining

Lecture 3: Linear Regression

Yanjun Qi / Jane

University of Virginia
Department of
Computer Science

Last Lecture Recap

- ❑ **Data Representation**
- ❑ **Linear Algebra Review**

e.g. SUPERVISED LEARNING

$$f : X \longrightarrow Y$$

- Find function to map input space X to output space Y
- So that the difference between *y* and *f(x)* of each example ***x*** is small.

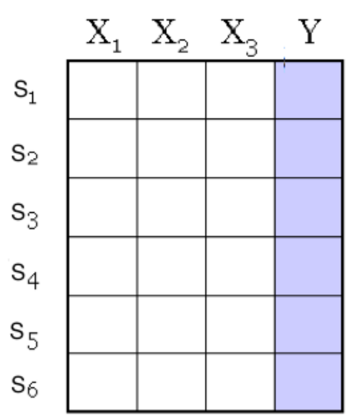
	X_1	X_2	X_3	Y
s_1				
s_2				
s_3				
s_4				
s_5				
s_6				

A Dataset

$$f : X \longrightarrow Y$$

- **Data**/*points/instances/examples/samples/records*: [rows]
- **Features**/*attributes/dimensions/independent variables/covariates/ predictors/regressors*: [columns, except the last]
- **Target**/*outcome/response/label/dependent variable*: special column to be predicted [last column]

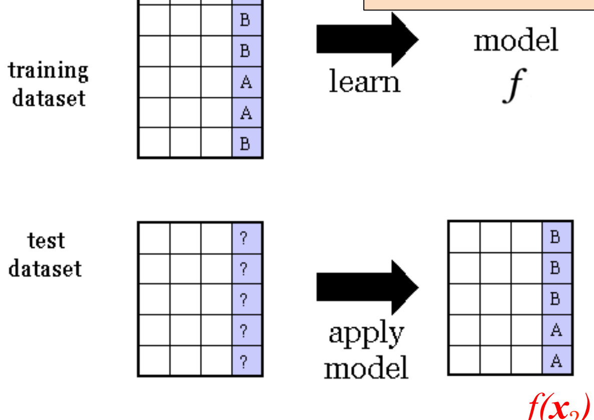
Main Types of Columns



- *Continuous*: a real number, for example, age or height
- *Discrete*: a symbol, like "Good" or "Bad"

e.g. SUPERVISED LEARNING

Training dataset consists of **input**-**output** pairs



- e.g. Here, target Y is a discrete target variable

MATRIX OPERATIONS

1) Transposition
2) Addition and Subtraction
3) Multiplication
4) Norm (of vector)
5) Matrix Inversion
6) Matrix Rank
7) Matrix calculus

Today

- ❑ Linear regression (aka **least squares**)
- ❑ Learn to derive the least squares estimate by optimization
- ❑ Evaluation with Cross-validation

For Example,
Machine learning for apartment hunting

- Now you've moved to Charlottesville !!

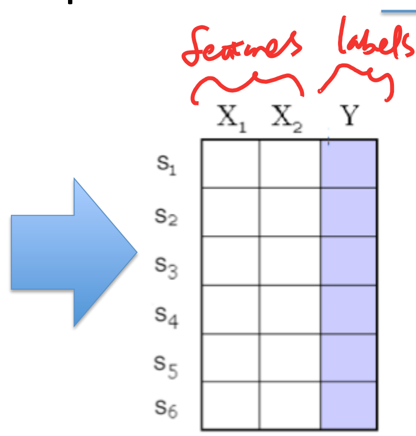
  And you want to find the **most reasonably priced** apartment satisfying your **needs:**

  square-ft., # of bedroom, distance to campus …

Living area (ft²)	# bedroom	Rent ($)
230	1	600
506	2	1000
433	2	1100
109	1	500
…		
150	1	?
270	1.5	?

For Example,
Machine learning for apartment hunting

Living area (ft²)	# bedroom	Rent ($)
230	1	600
506	2	1000
433	2	1100
109	1	500
…		
150	1	?
270	1.5	?



Linear SUPERVISED LEARNING

$$f : X \longrightarrow Y$$

e.g.

$$\hat{y} = f(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2$$

Features:

Living area, distance to campus, # bedroom …

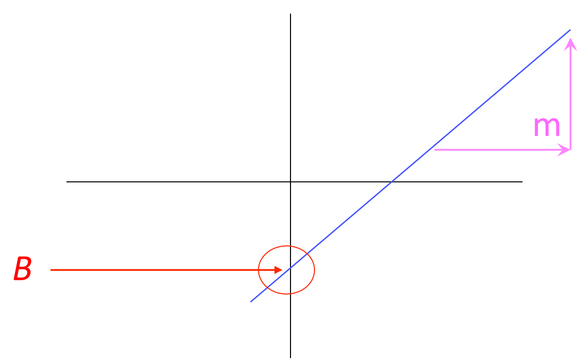
Target:

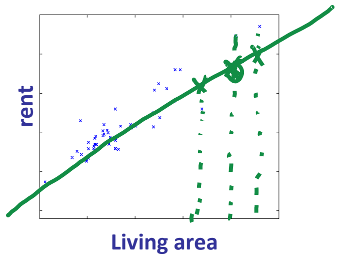
Rent

Remember this: "Linear"?

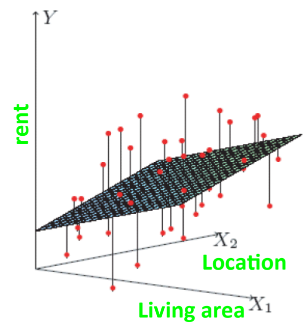
- *Y=mX+B?*

A slope of 2 (i.e. m=2) means that every 1-unit change in X yields a 2-unit change in Y.





1D case ($\mathcal{X} = \mathbb{R}$): a line



$\mathcal{X} = \mathbb{R}^2$: a plane

A new representation

- Assume that each sample **x** is a column vector,
  - Here we assume a vacuous "feature" X^0=1 (this is the intercept term), and define the feature vector to be:

    $\mathbf{x}^T=[x^0, x^1, x^2, \ldots x^{p-1}]$
  - the parameter vector θ is also a column vector

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{p-1} \end{bmatrix} \Rightarrow \hat{y} = f(x) = \mathbf{x}^T \theta$$

Training / learning problem

- We can represent the whole Training set:

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n^T & -- \end{bmatrix} = \begin{bmatrix} x_1^0 & x_1^1 & \dots & x_1^{p-1} \\ x_2^0 & x_2^1 & \dots & x_2^{p-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_n^0 & x_n^1 & \dots & x_n^{p-1} \end{bmatrix}$$

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

- Predicted output for each training sample:

$$\begin{bmatrix} f(\mathbf{x}_1^T) \\ f(\mathbf{x}_2^T) \\ \vdots \\ f(\mathbf{x}_n^T) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T\theta \\ \mathbf{x}_2^T\theta \\ \vdots \\ \mathbf{x}_n^T\theta \end{bmatrix} = X\theta$$

training / learning goal

- Using matrix form, we get the following general representation of the linear function:

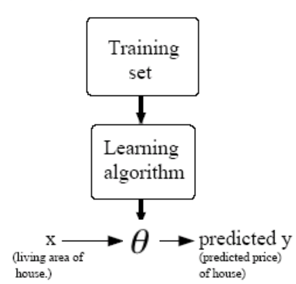
$$\hat{\mathbf{Y}} = X\theta$$

nx1 nxp px1

- Our goal is to pick the optimal θ that minimize the following cost function:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(f(\vec{x}_i) - y_i)^2$$

Our goal:



Today

- ❑ Linear regression (aka **least squares**)
- ❑ Learn to derive the least squares estimate by optimization
- ❑ Evaluation with Cross-validation

Method I: normal equations

- Write the cost function in matrix form:

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

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n^T & -- \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

To minimize $J(\theta)$, take derivative and set to zero:

$$\Rightarrow X^T X\theta = X^T\vec{y}$$

The normal equations

$$\Downarrow$$

$$\theta^* = (X^T X)^{-1} X^T \vec{y}$$

Review: Special Uses for Matrix Multiplication

- Dot (or Inner) Product of two Vectors <x, y>

which is the sum of products of elements in similar positions for the two vectors

<x, y> = <y, x>

Where <x, y> = $x^T y \in \mathbb{R} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ x_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i.$

Review: Special Uses for Matrix Multiplication

- Sum the Squared Elements of a Vector
  - Premultiply a column vector **a** by its transpose
    - If

$$\mathbf{a} = \begin{bmatrix} 5 \\ 2 \\ 8 \end{bmatrix}$$

then premultiplication by a row vector $\mathbf{a}^T$

$$a^T = \begin{bmatrix} 5 & 2 & 8 \end{bmatrix}$$

will yield the sum of the squared values of elements for **a**, i.e.

$$a^T\mathbf{a} = \begin{bmatrix} 5 & 2 & 8 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 8 \end{bmatrix} = 5^2 + 2^2 + 8^2 = 93$$

Review: Matrix Calculus: Types of Matrix Derivatives

	Scalar	Vector	Matrix
Scalar	$\frac{dy}{dx}$	$\frac{d\mathbf{y}}{dx} = \left[\frac{\partial y_i}{\partial x}\right]$	$\frac{d\mathbf{Y}}{dx} = \left[\frac{\partial y_{ij}}{\partial x}\right]$
Vector	$\frac{dy}{d\mathbf{x}} = \left[\frac{\partial y}{\partial x_j}\right]$	$\frac{d\mathbf{y}}{d\mathbf{x}} = \left[\frac{\partial y_i}{\partial x_j}\right]$	
Matrix	$\frac{dy}{d\mathbf{X}} = \left[\frac{\partial y}{\partial x_{ji}}\right]$		

By Thomas Minka. Old and New Matrix Algebra Useful for Statistics

Details for slide [18] :

$$J(\theta) = \sum_{i=1}^{n} (x_i^T \theta - y_i)^2$$

$$= (X\theta - y)^T (X\theta - y)$$

$n \times p \quad p \times 1 \quad n \times 1$

Since $w^T w = \|w\|_2^2 = \sum_{i=1}^{n} w_i^2$

$w \Downarrow X\theta$

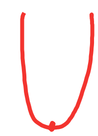
$$J(\theta) = (X\theta - y)^T (X\theta - y)$$
$$= ((X\theta)^T - y^T)(X\theta - y)$$
$$= (\theta^T X^T - y^T)(X\theta - y)$$
$$= \theta^T X^T X \theta - \theta^T X^T y - y^T X \theta - y^T y$$

Since $\theta^T X^T y = y^T X \theta$

$\langle X\theta, y \rangle \quad \langle y, X\theta \rangle$

$$= \theta^T X^T X \theta - 2\theta^T X^T y - y^T y$$

$\Rightarrow J(\theta)$ quadratic func of θ; if 1-d,



$\frac{\partial J(\theta)}{\partial \theta} = 0$

See handout 4.1 + 4.3 $\Rightarrow$ matrix calculus, partial deri $\Rightarrow$ Gradient

$\nabla_\theta (\theta^T X^T X \theta) = 2X^T X \theta$ (P24)

$\nabla_\theta (-2\theta^T X^T y) = -2X^T Y$ (P24)

$\nabla_\theta (-y^T y) = 0$

$\Rightarrow \nabla_\theta J(\theta) = 2X^T X\theta - 2X^T Y \overset{\text{Set to}}{=\!=} 0$

$\Rightarrow X^T X \theta = X^T Y$

$\Rightarrow \theta = (X^T X)^{-1} X^T Y$

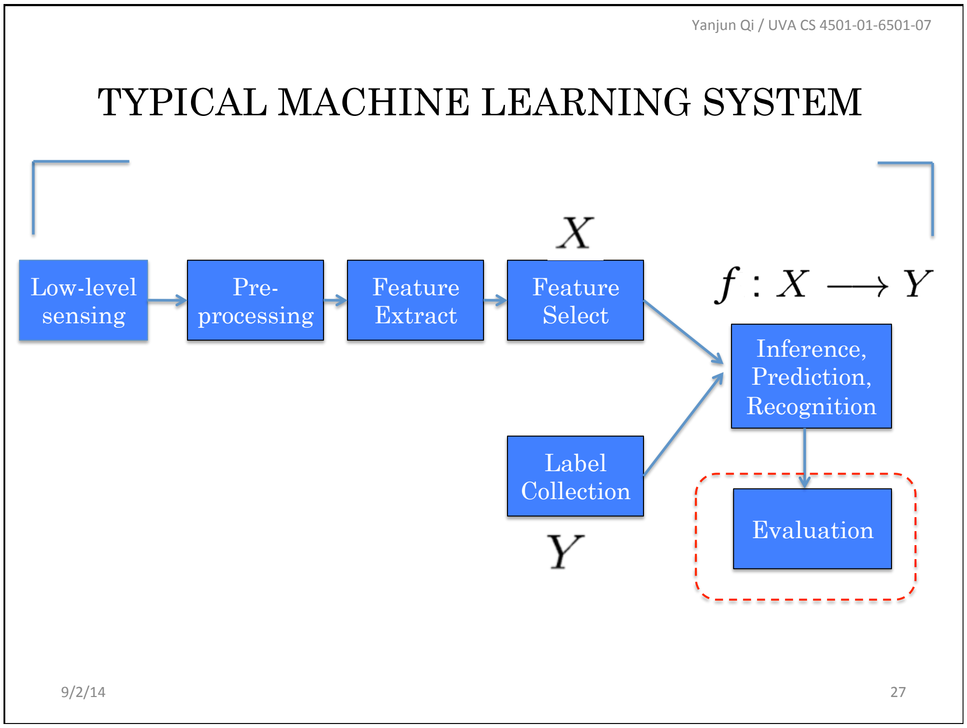
$(X^T X)^{-1}$ ⇓ under certain condition

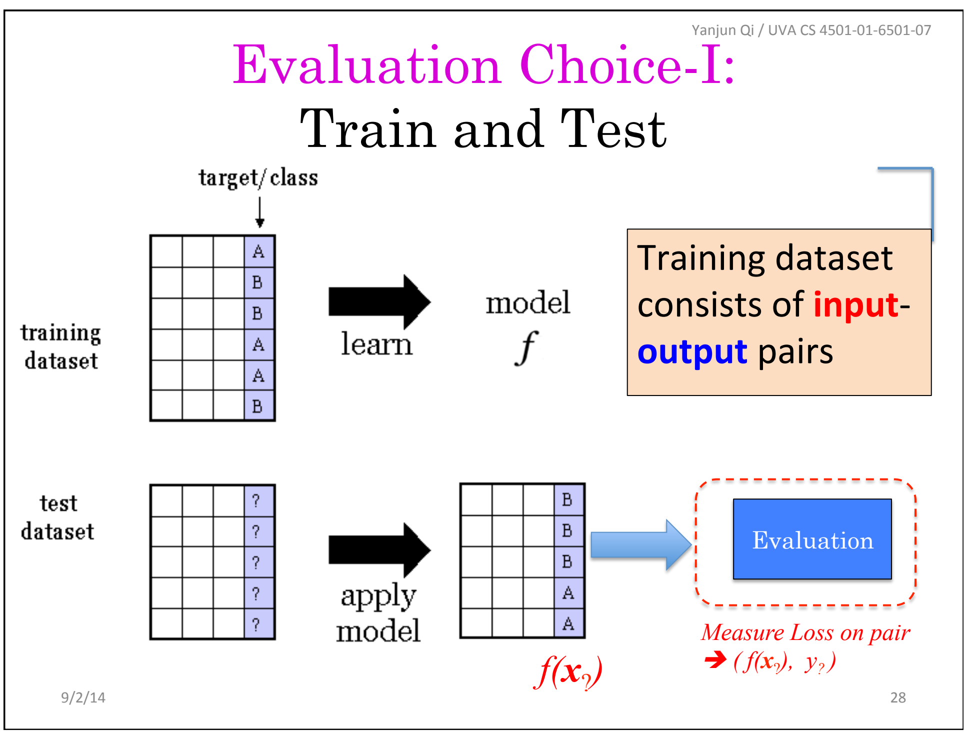
Comments on the normal equation

- In most situations of practical interest, the number of data points N is larger than the dimensionality p of the input space and the matrix $\mathbf{X}$ is of full column rank. If this condition holds, then it is easy to verify that $X^T X$ is necessarily invertible.
- The assumption that $X^T X$ is invertible implies that it is positive definite, thus the critical point we have found is a minimum.
- What if $\mathbf{X}$ has less than full column rank? → regularization (later).

Today

- ❑ Linear regression (aka **least squares**)
- ❑ Learn to derive the least squares estimate by optimization
- ❑ Evaluation with Train/Test OR k-folds Cross-validation

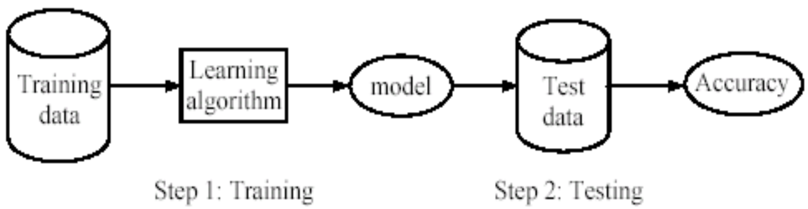

Yanjun Qi / UVA CS 4501-01-6501-07
TYPICAL MACHINE LEARNING SYSTEM
X
$f : X \longrightarrow Y$
Low-level sensing
Pre-processing
Feature Extract
Feature Select
Inference, Prediction, Recognition
Label Collection
Y
Evaluation
9/2/14
27


Yanjun Qi / UVA CS 4501-01-6501-07
Evaluation Choice-I:
Train and Test
target/class
training dataset
A
B
B
A
A
B
learn
model
f
Training dataset consists of input-output pairs
test dataset
?
?
?
?
?
apply model
B
B
B
A
A
$f(x_?)$
Evaluation
Measure Loss on pair
→ $(f(x_?),\ y_?)$
9/2/14
28

Evaluation Choice-I:

e.g. for supervised classification

- ✓ Training (Learning): Learn a model using the training data
- ✓ Testing: Test the model using unseen test data to assess the model accuracy



$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$

Evaluation Choice-II:

Cross Validation

- Problem: don't have enough data to set aside a test set
- Solution: Each data point is used both as train and test
- Common types:
  - -K-fold cross-validation (e.g. K=5, K=10)
  - -2-fold cross-validation
  - -Leave-one-out cross-validation

K-fold Cross Validation

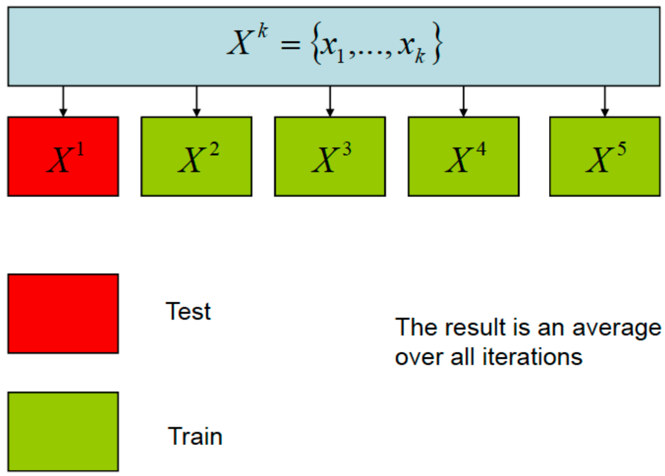
- Basic idea:
  - -Split the whole data to N pieces;
  - -N-1 pieces for fit model; 1 for test;
  - -Cycle through all N cases;
  - -K=10 “folds” a common rule of thumb.
- The advantage:
  - - all pieces are used for both training and validation;
  - - each observation is used for validation exactly once.

e.g. 10 fold Cross Validation

- Divide data into 10 equal pieces
- 9 pieces as training set, the rest 1 as test set
- Collect the scores from the diagonal

model	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	train	train	train	train	train	train	train	train	train	test
2	train	train	train	train	train	train	train	train	test	train
3	train	train	train	train	train	train	train	test	train	train
4	train	train	train	train	train	train	test	train	train	train
5	train	train	train	train	train	test	train	train	train	train
6	train	train	train	train	test	train	train	train	train	train
7	train	train	train	test	train	train	train	train	train	train
8	train	train	test	train	train	train	train	train	train	train
9	train	test	train	train	train	train	train	train	train	train
10	test	train	train	train	train	train	train	train	train	train

e.g. 5 fold Cross Validation



The result is an average over all iterations

Today Recap

- ❑ Linear regression (aka **least squares**)
- ❑ Learn to derive the least squares estimate by optimization
- ❑ Evaluation with Train/Test OR k-folds Cross-validation

References

- Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides

- ❑ http://www.cs.cmu.edu/~zkolter/course/15-884/linalg-review.pdf (please read)
- ❑ http://www.cs.cmu.edu/~aarti/Class/10701/recitation/LinearAlgebra_Matlab_Rev iew.ppt
- ❑ Prof. Alexander Gray's slides