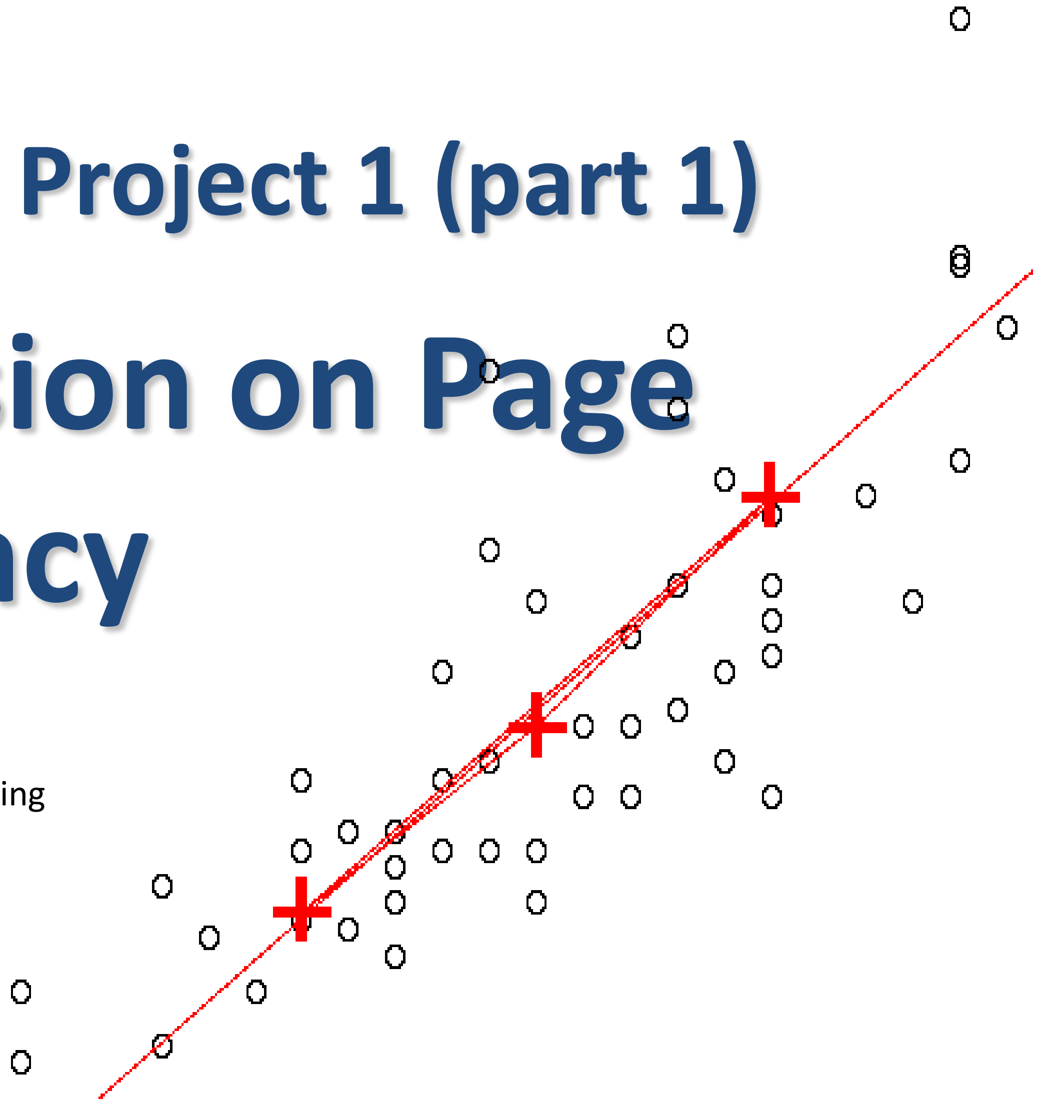
CSE 4/574 Project 1 (part 1)

Regression on Page Relevancy

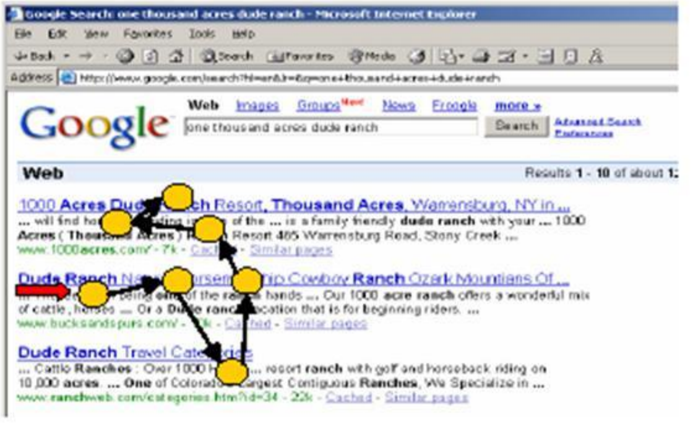


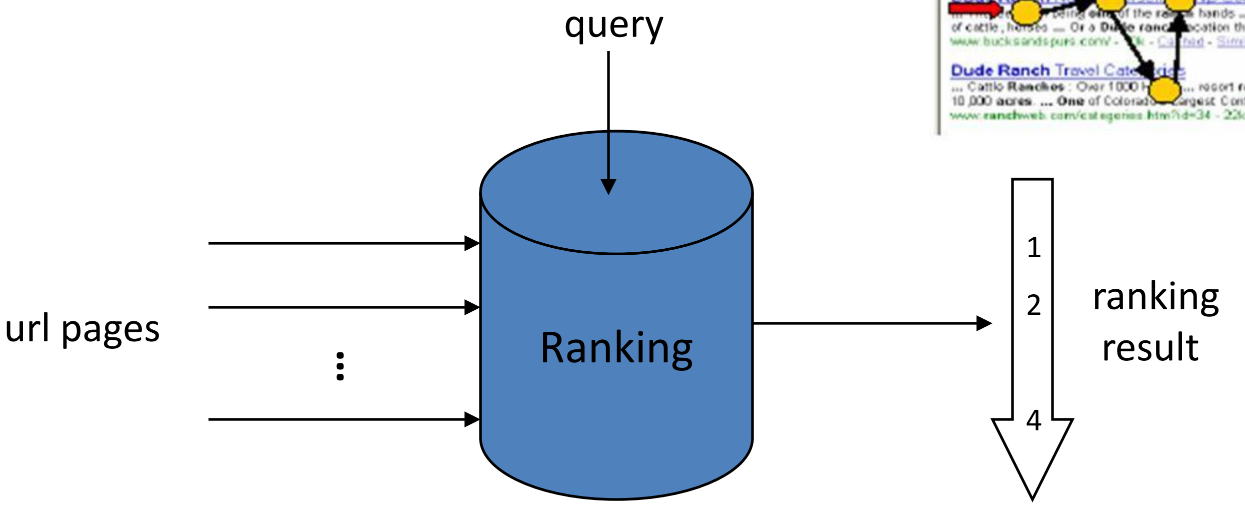
CSE4/574 Machine Learning
TA: Yu Liu
yl73@buffalo.edu

Web search ranking

Goal: given queries and a documents/urls, estimate the Web search results (relevance) of the pages to the queries.

Ranking the pages via a relevance function.



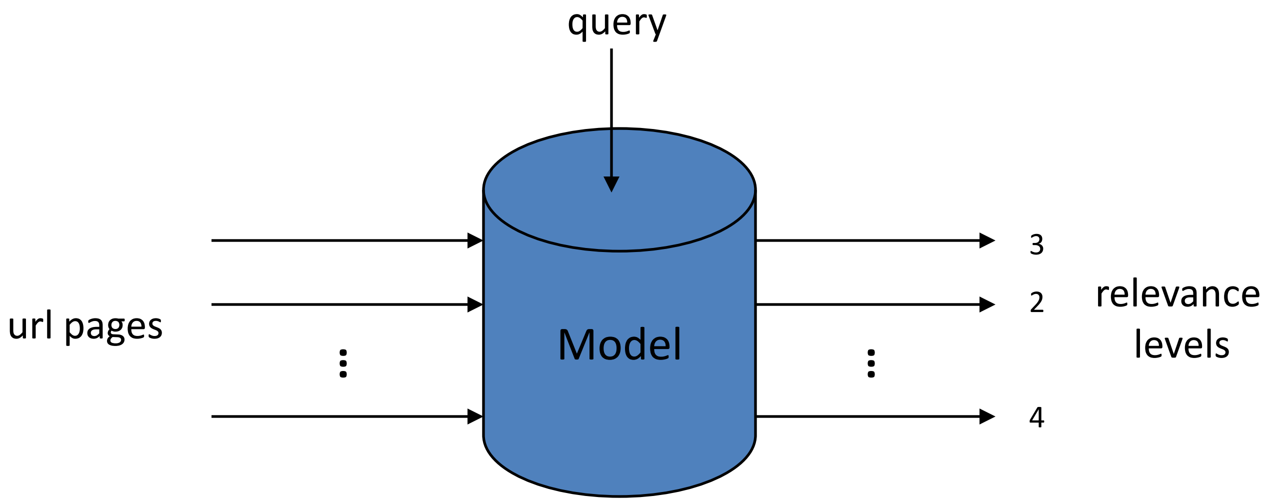


Regression on Page Relevancy

Not Ranking!!

Goal: Train a regression model based on query-url pair datasets , then predict the page relevancy labels for new coming queries.

Binary / multiple levels of relevance (Bad, Fair, Good, Excellent, Perfect, ...)



Datasets

Large scale real world learning to rank (LTR) datasets that has been released:

	Queries	Doc.	Rel.	Feat.	Year
Letor3.0 – Gov	575	568k	2	64	2008
Letor3.0 – Ohsumed	106	16k	3	45	2008
Letor4.0	2476	85k	3	46	2009
Yandex	20267	213k	5	245	2009
Yahoo	36251	883k	5	700	2010

Letor4.0 Dataset

LETOR is a package of benchmark data sets for research on Learning To Rank released by Microsoft Research Asia.

- The latest version, 4.0, can be found at
http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4dataset.aspx
(It contains 8 datasets for four ranking settings derived from the two query sets and the Gov2 web page collection.)

- For this project, one dataset of MQ2008 is used (supervised ranking):

”Querylevelnorm.txt” (15211 urls/samples in total)

Letor4.0 Dataset

Sample rows from the MQ2008 dataset:

Querylevelnorm.txt

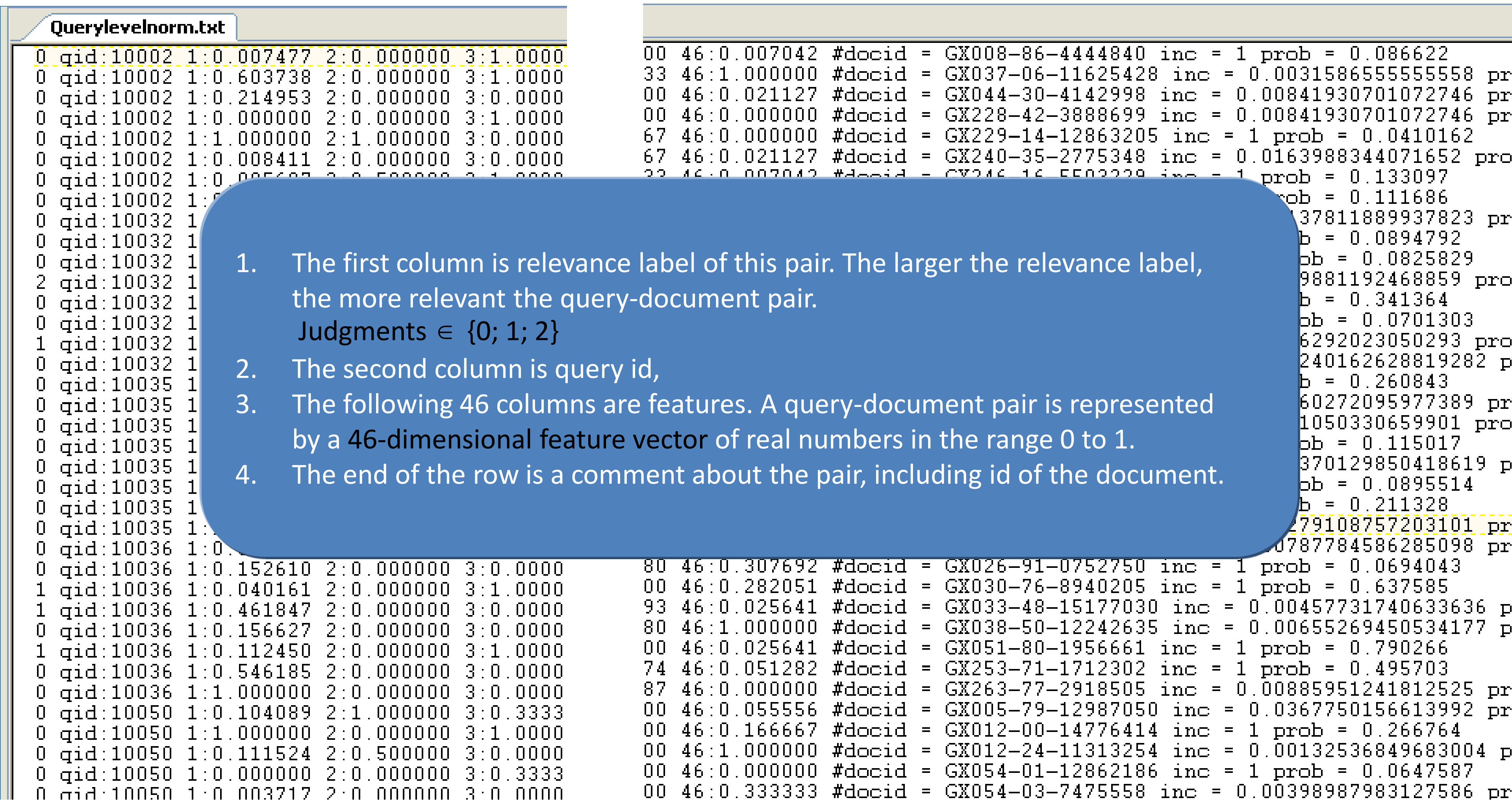
```
0 qid:10002 1:0.007477 2:0.000000 3:1.0000
0 qid:10002 1:0.603738 2:0.000000 3:1.0000
0 qid:10002 1:0.214953 2:0.000000 3:0.0000
0 qid:10002 1:0.000000 2:0.000000 3:1.0000
0 qid:10002 1:1.000000 2:1.000000 3:0.0000
0 qid:10002 1:0.008411 2:0.000000 3:0.0000
0 qid:10002 1:0.005607 2:0.500000 3:1.0000
0 qid:10002 1:0.259813 2:1.000000 3:0.0000
0 qid:10032 1:0.021201 2:0.000000 3:1.0000
0 qid:10032 1:0.000000 2:0.000000 3:0.0000
0 qid:10032 1:0.007067 2:0.000000 3:0.6666
2 qid:10032 1:0.056537 2:0.000000 3:0.6666
0 qid:10032 1:0.279152 2:0.000000 3:0.0000
0 qid:10032 1:0.130742 2:0.000000 3:0.3333
1 qid:10032 1:0.593640 2:1.000000 3:0.0000
0 qid:10032 1:1.000000 2:0.000000 3:0.0000
0 qid:10035 1:0.643564 2:0.000000 3:0.4285
0 qid:10035 1:0.039604 2:0.000000 3:0.0000
0 qid:10035 1:0.891089 2:1.000000 3:1.0000
0 qid:10035 1:0.000000 2:0.000000 3:0.4285
0 qid:10035 1:0.356436 2:0.750000 3:0.4285
0 qid:10035 1:0.039604 2:0.000000 3:0.0000
0 qid:10035 1:0.326733 2:0.000000 3:0.0000
0 qid:10035 1:1.000000 2:0.000000 3:0.0000
0 qid:10036 1:0.000000 2:0.000000 3:0.0000
0 qid:10036 1:0.152610 2:0.000000 3:0.0000
1 qid:10036 1:0.040161 2:0.000000 3:1.0000
1 qid:10036 1:0.461847 2:0.000000 3:0.0000
0 qid:10036 1:0.156627 2:0.000000 3:0.0000
1 qid:10036 1:0.112450 2:0.000000 3:1.0000
0 qid:10036 1:0.546185 2:0.000000 3:0.0000
0 qid:10036 1:1.000000 2:0.000000 3:0.0000
0 qid:10050 1:0.104089 2:1.000000 3:0.3333
0 qid:10050 1:1.000000 2:0.000000 3:1.0000
0 qid:10050 1:0.111524 2:0.500000 3:0.0000
0 qid:10050 1:0.000000 2:0.000000 3:0.3333
0 qid:10050 1:0.003717 2:0.000000 3:0.0000
```

• • •

```
00 46:0.007042 #docid = GX008-86-4444840 inc = 1 prob = 0.086622
33 46:1.000000 #docid = GX037-06-11625428 inc = 0.0031586555555558 pr
00 46:0.021127 #docid = GX044-30-4142998 inc = 0.00841930701072746 pr
00 46:0.000000 #docid = GX228-42-3888699 inc = 0.00841930701072746 pr
67 46:0.000000 #docid = GX229-14-12863205 inc = 1 prob = 0.0410162
67 46:0.021127 #docid = GX240-35-2775348 inc = 0.0163988344071652 pro
33 46:0.007042 #docid = GX246-16-5503229 inc = 1 prob = 0.133097
33 46:0.035211 #docid = GX255-50-7550514 inc = 1 prob = 0.111686
00 46:0.153846 #docid = GX010-65-7921994 inc = 0.00137811889937823 pr
00 46:0.461538 #docid = GX024-71-0000000 inc = 1 prob = 0.0894792
00 46:0.000000 #docid = GX029-17-16711721 inc = 1 prob = 0.0825829
00 46:0.076923 #docid = GX029-35-5894638 inc = 0.0119881192468859 pro
00 46:1.000000 #docid = GX030-77-6315042 inc = 1 prob = 0.341364
00 46:1.000000 #docid = GX140-98-13566007 inc = 1 prob = 0.0701303
00 46:0.000000 #docid = GX256-43-0740276 inc = 0.0136292023050293 pro
00 46:0.000000 #docid = GX266-75-11189217 inc = 0.00240162628819282 p
00 46:0.750000 #docid = GX026-92-0492427 inc = 1 prob = 0.260843
00 46:0.166667 #docid = GX031-29-0590777 inc = 0.00960272095977389 pr
00 46:0.166667 #docid = GX046-28-2590531 inc = 0.0121050330659901 pro
00 46:0.000000 #docid = GX058-84-15460908 inc = 1 prob = 0.115017
00 46:1.000000 #docid = GX072-27-16566993 inc = 0.00370129850418619 p
00 46:0.000000 #docid = GX187-61-14052950 inc = 1 prob = 0.0895514
00 46:0.000000 #docid = GX259-93-1304063 inc = 1 prob = 0.211328
00 46:0.000000 #docid = GX271-73-0262448 inc = 0.00279108757203101 pr
00 46:0.205128 #docid = GX004-58-2379388 inc = 0.00787784586285098 pr
80 46:0.307692 #docid = GX026-91-0752750 inc = 1 prob = 0.0694043
00 46:0.282051 #docid = GX030-76-8940205 inc = 1 prob = 0.637585
93 46:0.025641 #docid = GX033-48-15177030 inc = 0.00457731740633636 p
80 46:1.000000 #docid = GX038-50-12242635 inc = 0.00655269450534177 p
00 46:0.025641 #docid = GX051-80-1956661 inc = 1 prob = 0.790266
74 46:0.051282 #docid = GX253-71-1712302 inc = 1 prob = 0.495703
87 46:0.000000 #docid = GX263-77-2918505 inc = 0.00885951241812525 pr
00 46:0.055556 #docid = GX005-79-12987050 inc = 0.0367750156613992 pr
00 46:0.166667 #docid = GX012-00-14776414 inc = 1 prob = 0.266764
00 46:1.000000 #docid = GX012-24-11313254 inc = 0.00132536849683004 p
00 46:0.000000 #docid = GX054-01-12862186 inc = 1 prob = 0.0647587
00 46:0.333333 #docid = GX054-03-7475558 inc = 0.00398987983127586 pr
```

Letor4.0 Dataset

Sample rows from the MQ2008 dataset:



1. The first column is relevance label of this pair. The larger the relevance label, the more relevant the query-document pair.
   Judgments ∈ {0; 1; 2}
2. The second column is query id,
3. The following 46 columns are features. A query-document pair is represented by a 46-dimensional feature vector of real numbers in the range 0 to 1.
4. The end of the row is a comment about the pair, including id of the document.

Features

Given a query and a document, construct a feature vector (normalized between 0 and 1)

Column in Output	Description
1	TF(Term frequency) of body
2	TF of anchor
3	TF of title
4	TF of URL
5	TF of whole document
6	IDF(Inverse document frequency) of body
7	IDF of anchor
8	IDF of title
9	IDF of URL
10	IDF of whole document
11	TF*IDF of body
12	TF*IDF of anchor
13	TF*IDF of title
14	TF*IDF of URL
15	TF*IDF of whole document
16	DL(Document length) of body
17	DL of anchor
18	DL of title
19	DL of URL
20	DL of whole document
21	BM25 of body
22	LMIR.ABS of body
23	LMIR.DIR of body
24	LMIR.JM of body
25	BM25 of anchor
26	LMIR.ABS of anchor
27	LMIR.DIR of anchor
28	LMIR.JM of anchor
29	BM25 of title
30	LMIR.ABS of title
31	LMIR.DIR of title
32	LMIR.JM of title
33	BM25 of URL
34	LMIR.ABS of URL
35	LMIR.DIR of URL
36	LMIR.JM of URL
37	BM25 of whole document
38	LMIR.ABS of whole document
39	LMIR.DIR of whole document
40	LMIR.JM of whole document
41	PageRank
42	Inlink number
43	Outlink number
44	Number of slash in URL
45	Length of URL
46	Number of child page

Import Data Set

- Matlab function: fopen, textscan, strfind, etc.

Read by line

File -> Import Data…
>> line_string = importedData{1}   % imported data is nx1 cell

Name	Value	Class
Querylevelnorm	<15211x1 cell>	cell

or

>> fid   = fopen(‘dataset.txt’);
>> data = textscan(fid, ‘%[^\n]’);   % read by lines, data is 1x1 cell
>> line_string = data{1}{1};

Name	Value	Class
data	<1x1 cell>	cell
data_in_cell	<15211x1 cell>	cell

Example of line in string   [1x604 char]

```
>> line_string = 0 qid:10002 1:0.007477 2:0.000000 3:1.000000 4:0.000000 5:0.007470 6:0.000000 7:0.000000 8:0.000000 9
```

Process Data Set (i)

Process the original data into a matrix containing relevance labels (the first column) and feature vectors. This input matrix (training data) will be feed into your regression model.

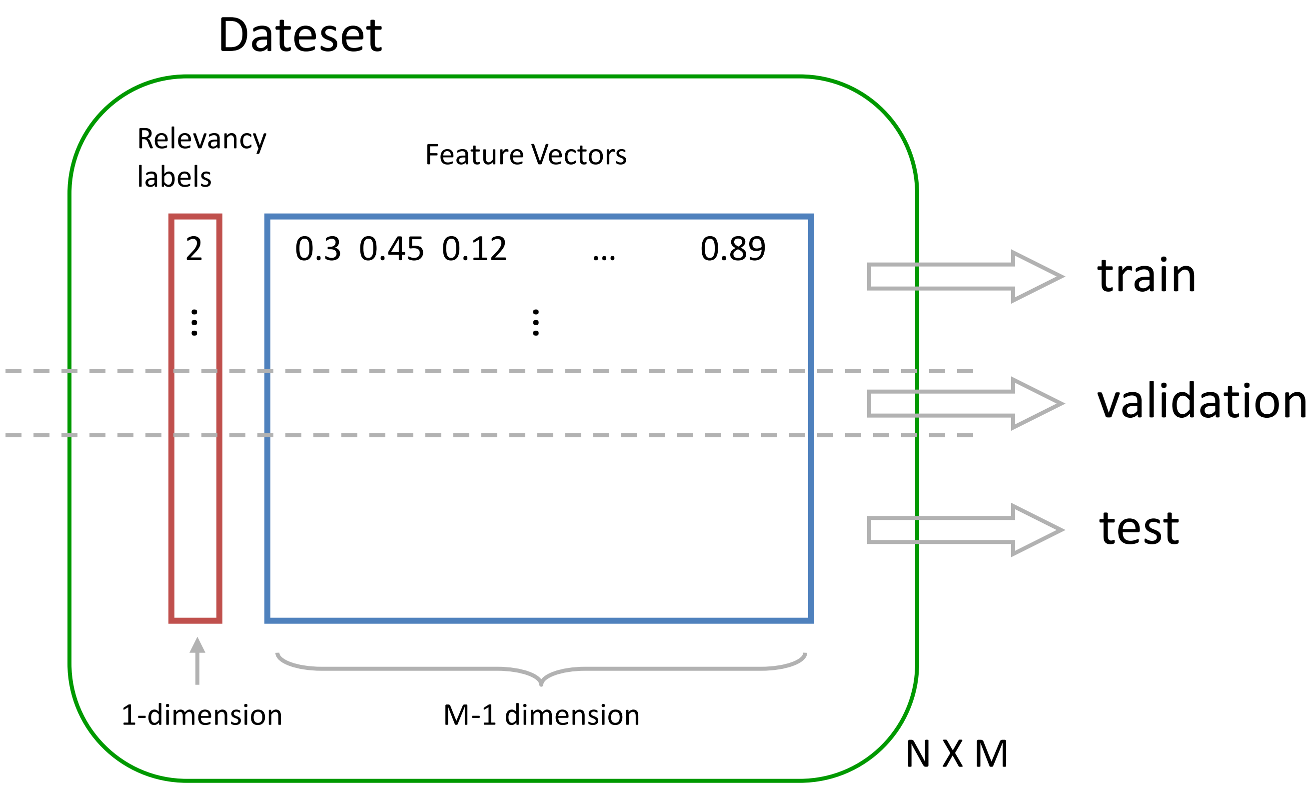
- LETOR 4.0

2 qid:10002 1:0.007477 2:0.000000 3:1.000000 4:0.000000 5:0.007470 … 46:0.007042 #docid = GX008-86-4444840 inc = 1 prob = 0.086622

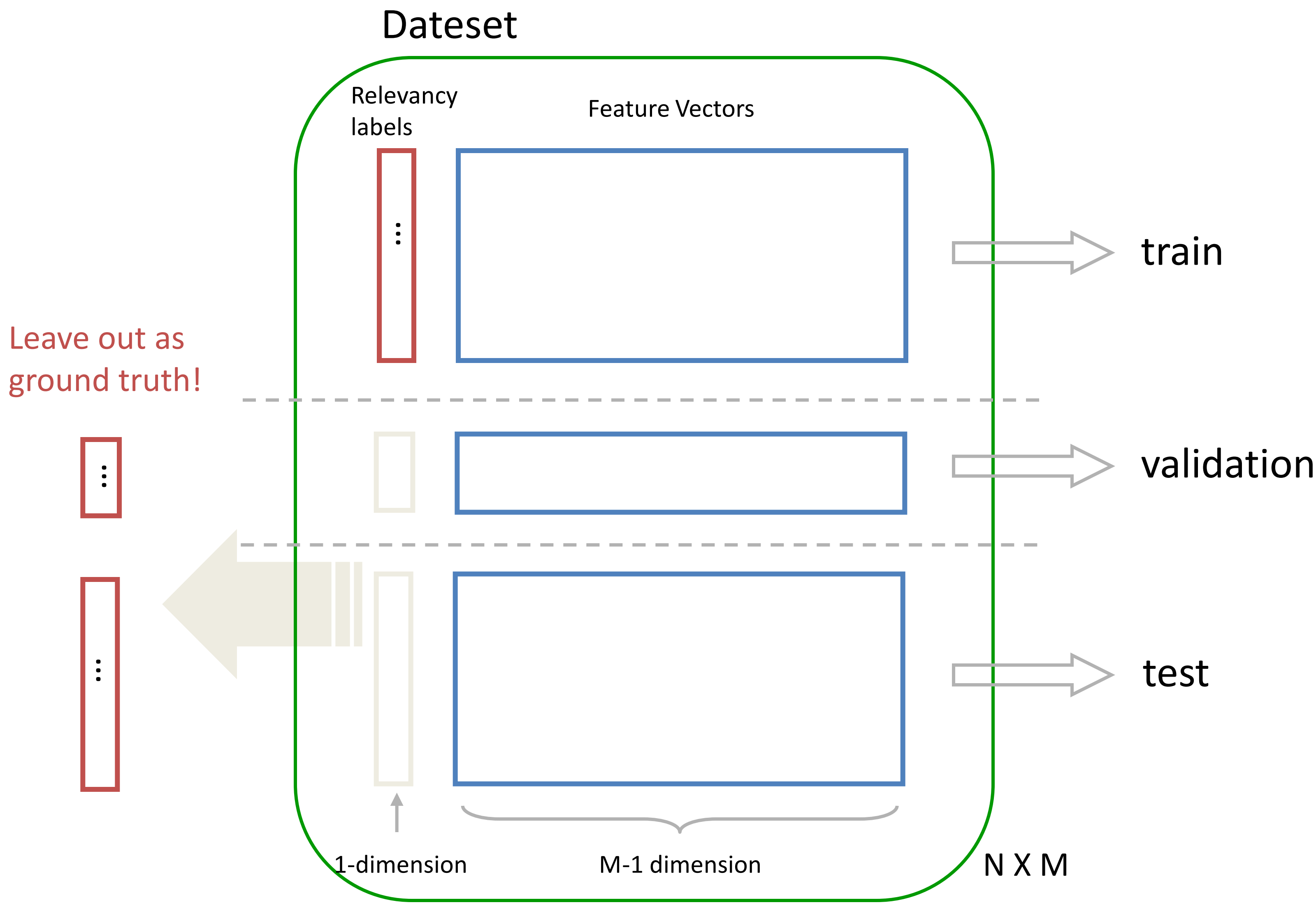
2 ~~qid:10002~~ ~~1:~~0.007477 ~~2:~~0.000000 ~~3:~~1.000000 ~~4:~~0.000000 ~~5:~~0.007470 … ~~46:~~0.007042 ~~#docid = GX008-86-4444840 inc = 1 prob = 0.086622~~

Process Data Set (ii)

For LETOR 4.0, you need partition the data set into three subsets.

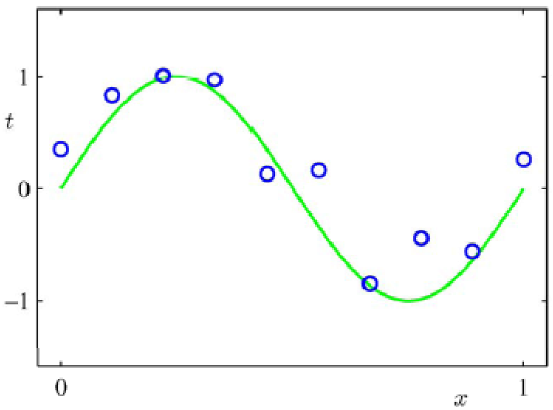


Train/Validation/Test Sets



Linear Regression

Problem: We want a general way of obtaining a linear model (model is linear in the parameters) that fitted to observed data.



General set up:
Given a set of training examples ($\mathbf{x}_n$, t_n), n =1, …N
Goal: learn a function y(x) to minimize some loss function (error function): E(y,t)

Linear Basis function Model:

$$y(\mathbf{x},\mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \boxed{\phi_j(x)} = \phi(\mathbf{x})\mathbf{w}$$

- Typically, $\phi_0(x) = 1$, so that w_0 acts as a bias parameter.
- In the simplest case, we use linear basis functions : $\phi_j(x) = x_j$.

Linear Regression

a single data

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad \boldsymbol{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix} \quad \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix} \quad \Phi(\mathbf{x}) = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \cdots & \phi_{M-1}(x_2) \\ & & \ddots & \\ \phi_0(x_N) & \phi_1(x_N) & \cdots & \phi_{M-1}(x_N) \end{pmatrix}$$

a basis function

N x M design matrix

Estimation:

$$\mathbf{y}(\mathbf{x}, \boldsymbol{w}) = \Phi\, \boldsymbol{w}$$

Squared Error function:

$$\boldsymbol{E}(\mathbf{y}, \mathbf{t}) = (\Phi\, \boldsymbol{w} - \mathbf{t})^{\mathrm{T}} (\Phi\, \boldsymbol{w} - \mathbf{t})$$

Minimize error:

$$\mathbf{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}}\ \boldsymbol{E}(\mathbf{y}, \mathbf{t})$$

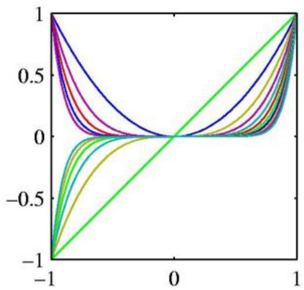
Least squares solution:

$$\nabla_{\boldsymbol{w}} \boldsymbol{E} = \Phi^{\mathrm{T}} (\Phi\, \boldsymbol{w} - \mathbf{t}) = 0$$

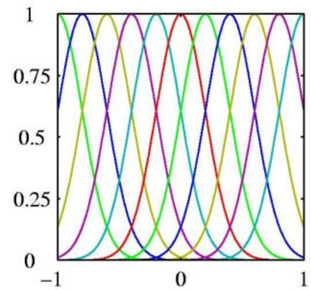
$$\mathbf{w}^* = (\Phi^{\mathrm{T}} \Phi)^{-1} \Phi^{\mathrm{T}} \mathbf{t}$$

Linear Basis Function Models

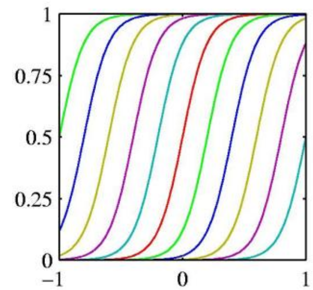
$$y(\boldsymbol{x},\boldsymbol{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\boldsymbol{x}) = \phi(\boldsymbol{x})\boldsymbol{w}$$



Polynomial



Gaussian



Sigmoid

$$\phi_j(\boldsymbol{x}) = x^j$$

$$\phi_j(\boldsymbol{x}) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

$$\phi_j(\boldsymbol{x}) = \sigma\left(\frac{x-\mu_j}{s}\right)$$

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$

Linear Regression for Project

Project Goal: To predict the value of one or more continuous target variables t given the value of a D-dimensional vector x of input variables.

$$\boldsymbol{x} = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^D \\ x_2^1 & x_2^2 & & x_2^D \\ & \ddots & & \\ x_n^1 & x_n^2 & \dots & x_n^D \end{pmatrix} \qquad \boldsymbol{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix}$$

One dimensional:
D = 1 (already encountered)

Find $\boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_? \end{pmatrix}$

Linear Regression for Project

Polynomial Basis Function (not required) $\phi_j(\boldsymbol{x}) = x^j$

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{j=1}^{M\text{-}1} \sum_{i=1}^{D} w_{(i,j)} \phi_j(x_i)$$

Different orders of polynomial

Sum over D dimension

$$\Phi(\mathbf{x}) = \begin{pmatrix} 1, x_1^1, x_1^2, \ldots, x_1^D, (x_1^1)^2, (x_1^2)^2, \ldots, (x_1^D)^2, \ldots, (x_1^1)^{M-1}, (x_1^2)^{M-1}, \ldots, (x_1^D)^{M-1} \\ \vdots \\ \vdots \\ 1, x_N^1, x_N^2, \ldots, x_N^D, (x_N^1)^2, (x_N^2)^2, \ldots, (x_N^D)^2, \ldots, (x_N^1)^{M-1}, (x_N^2)^{M-1}, \ldots, (x_N^D)^{M-1} \end{pmatrix}$$

N x ((M-1)xD + 1) matrix

w: (M-1)xD+1 dimension weight vector

Linear Regression for Project

Gaussian Basis Function

$$\phi_j(\boldsymbol{x}) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

$$y(\boldsymbol{x},\boldsymbol{w}) = w_0 + \sum_{j=1}^{M\text{-}1}\sum_{i=1}^{D} w_{(i,j)}\phi_j(x_i)$$

Different Gaussian parameter settings

Sum over D dimension

$$\Phi(\mathbf{x}) = \begin{pmatrix} 1, \phi_1(x_1^1), \phi_1(x_1^2), \ldots, \phi_1(x_1^D), \phi_2(x_1^1), \phi_2(x_1^2), \ldots, \phi_2(x_1^D), \ldots \phi_{M-1}(x_1^1), \phi_{M-1}(x_1^2), \ldots, \phi_{M-1}(x_1^D) \\ \vdots \\ \vdots \\ 1, \phi_1(x_N^1), \phi_1(x_N^2), \ldots, \phi_1(x_N^D), \phi_2(x_N^1), \phi_2(x_N^2), \ldots, \phi_2(x_N^D), \ldots \phi_{M-1}(x_N^1), \phi_{M-1}(x_N^2), \ldots, \phi_{M-1}(x_N^D) \end{pmatrix}$$

N x ((M-1)xD + 1) matrix

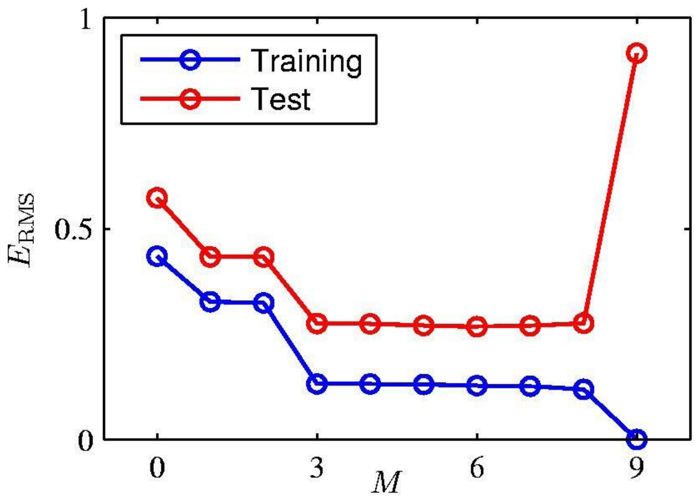
w: (M-1)xD+1 dimension weight vector

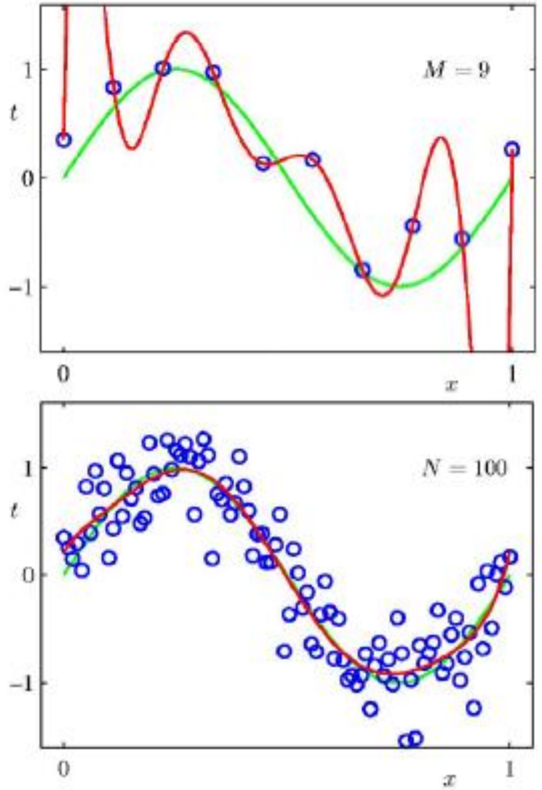
Sigmoid basis function: similar to Gaussian

Overfitting Issue

What can we do to curb overfitting?

- Use less complex model
- Use more training examples
- Regularization





Regularized Least Square

Add regularization term to error function to control over-fitting:

$$\boldsymbol{E}(\mathbf{w}) = \underbrace{\boldsymbol{E}_D(\mathbf{w})}_{\text{Data dependent term}} + \lambda \underbrace{\boldsymbol{E}_W(\mathbf{w})}_{\text{Regularization term}}$$

Squared Error function:

$$\boldsymbol{E}(\mathbf{w}) = (\boldsymbol{\Phi}\,\mathbf{w} - \mathbf{t})^{\mathrm{T}}(\boldsymbol{\Phi}\,\mathbf{w} - \mathbf{t}) + \frac{1}{2}\lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}$$

encourage small weight values!

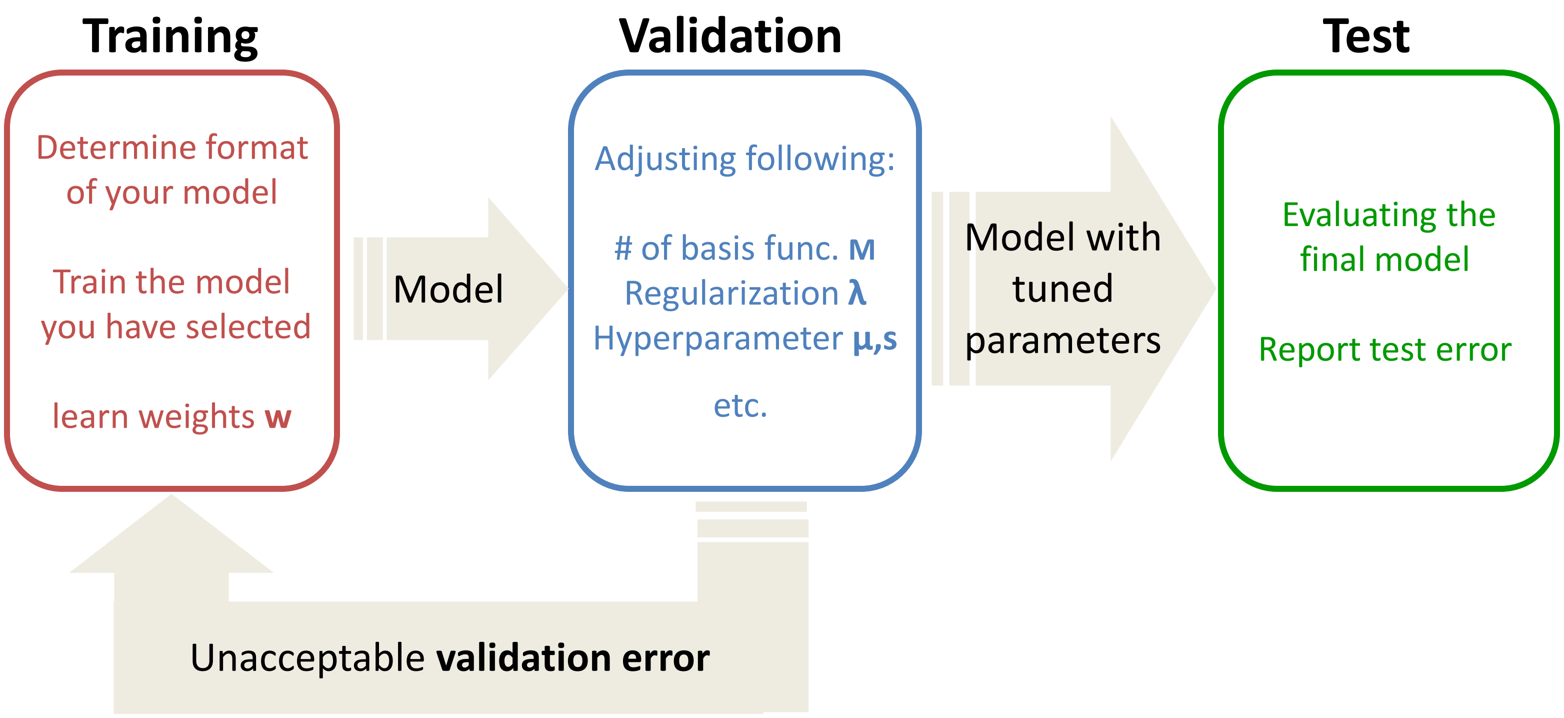
Minimize error:

$$\mathbf{w}^* = \operatorname*{argmin}_{w} \boldsymbol{E}(\mathbf{w})$$

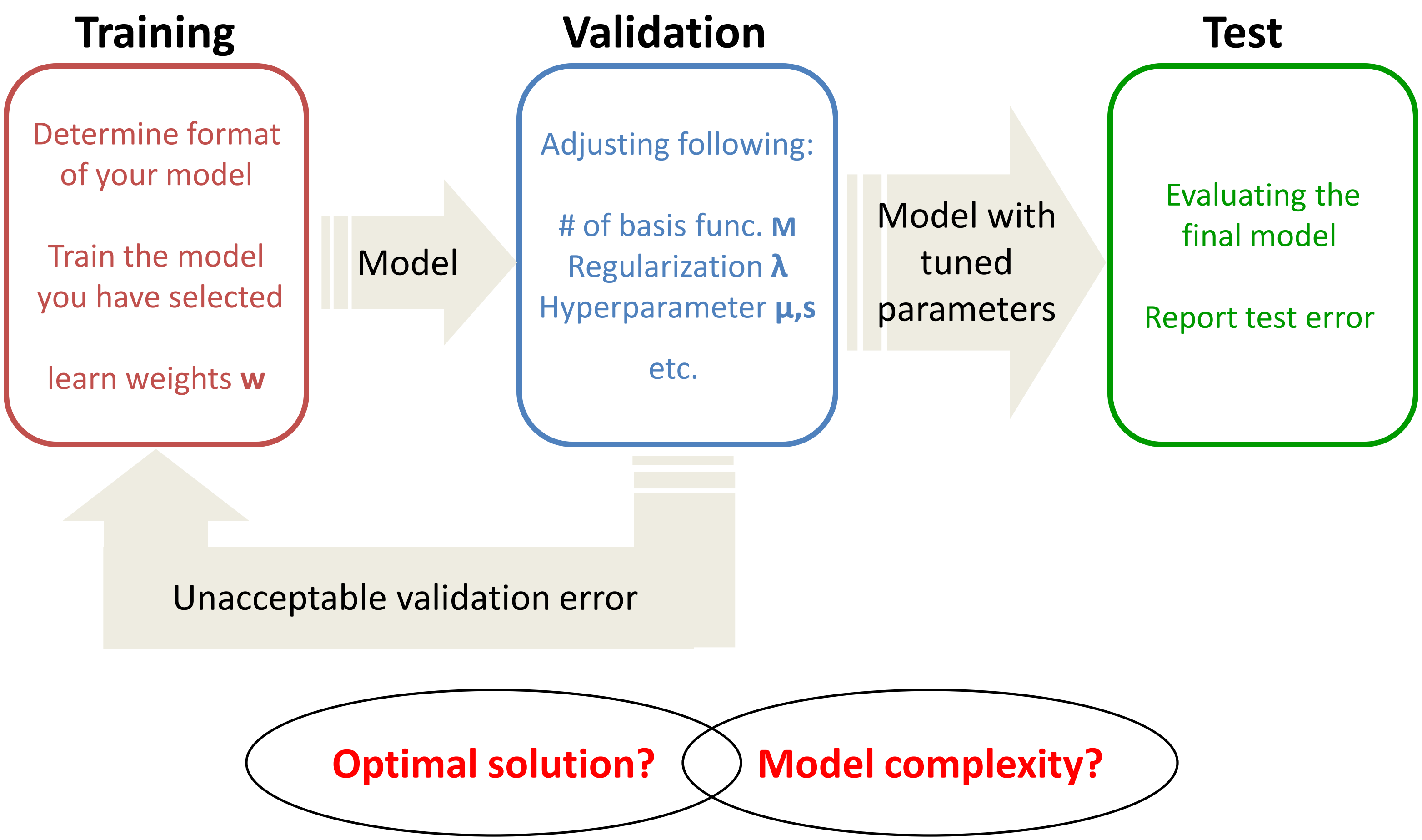
Regularized Least squares solution:

$$\nabla_w \boldsymbol{E} = \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{\Phi}\,\mathbf{w} - \mathbf{t}) + \lambda \mathbf{w} \quad \Rightarrow \quad \mathbf{w}^* = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda \mathbf{I})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

Experimental Phases



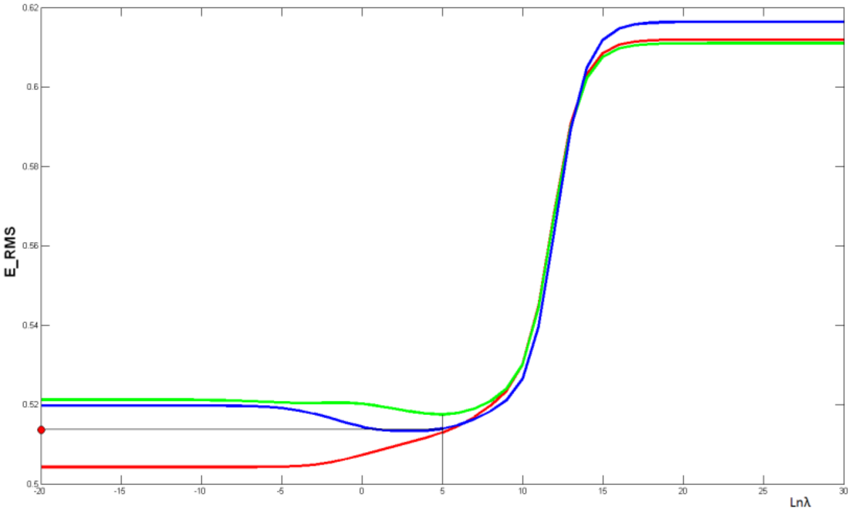
Experimental Phases



Evaluation Metrics

Express results as Root Mean Square Error: E_{RMS}

$$E_{RMS}(\mathbf{w}) = \sqrt{\frac{2E_D(\mathbf{w})}{N}}$$



N: number of data in data set

$E_D(w)$: sum of square error function (data-dependent error)

Project Report

- Explain the problem and how you choose your model.

- Elaborate your validating process.

  - The intuitive choice of parameters)

  There are no limitation on setting parameters and there could be infinity choices. You can define some range or choose some specific values.

  - Description of how you went about avoiding overfitting.

- Generate graphs showing how error changes with the adjusting of parameters.

- Report final result and evaluating model performance.