



Introduction to **Machine Learning and Data Mining** (Học máy và Khai phá dữ liệu)

Khoat Than

School of Information and Communication Technology
Hanoi University of Science and Technology

Contents

- Introduction to Machine Learning & Data Mining
- **Supervised learning**
 - **Linear regression**
- Unsupervised learning
- Practical advice

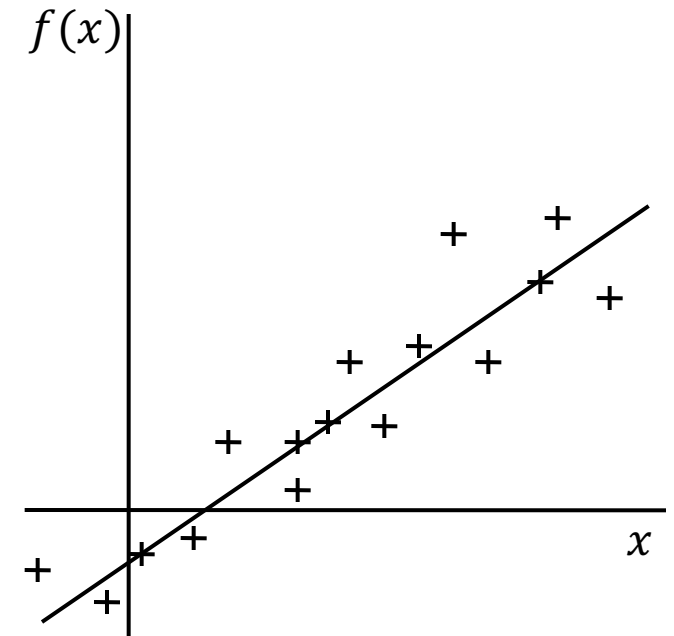
Linear regression: introduction

- **Regression problem:** learn a function $y = f(\mathbf{x})$ from a given training data $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_M, y_M)\}$ such that $y_i \cong f(\mathbf{x}_i)$ for every i
 - Each observation of \mathbf{x} is represented by a vector in an n -dimensional space, e.g., $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$. Each dimension represents an *attribute/feature/variable*.
 - Bold characters denote vectors.
- **Linear model:** if $f(\mathbf{x})$ is assumed to be of linear form
$$f(\mathbf{x}) = w_0 + w_1x_1 + \dots + w_nx_n$$
 - w_0, w_1, \dots, w_n are the regression coefficients/weights. w_0 sometimes is called “*bias*”.
- **Note:** learning a linear function is equivalent to learning the coefficient vector $\mathbf{w} = (w_0, w_1, \dots, w_n)^T$.

Linear regression: example

- What is the best function?

x	y
0.13	-0.91
1.02	-0.17
3.17	1.61
-2.76	-3.31
1.44	0.18
5.28	3.36
-1.74	-2.46
7.93	5.56
...	...



Prediction

- For each observation $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$

- The *true output*: c_x
(but unknown for future data)
- *Prediction* by our system:

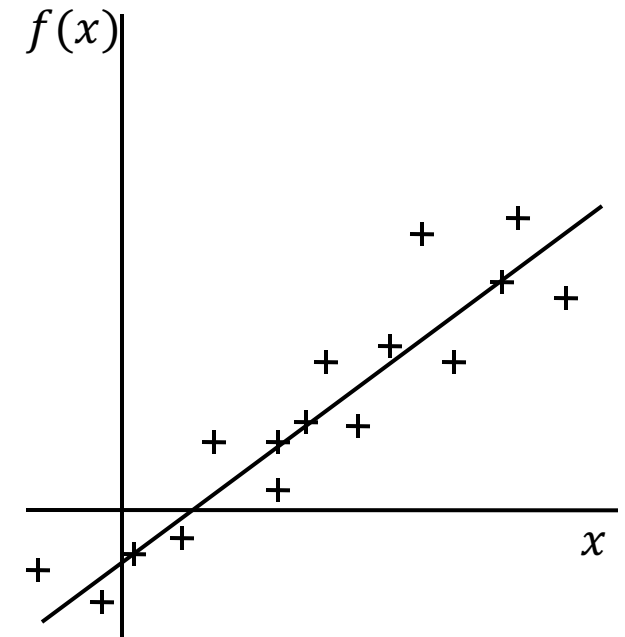
$$y_x = w_0 + w_1 x_1 + \dots + w_n x_n$$

- We often *expect* $y_x \cong c_x$.
- Prediction for a future observation $\mathbf{z} = (z_1, z_2, \dots, z_n)^T$
 - Use the learned function to make prediction

$$f(\mathbf{z}) = w_0 + w_1 z_1 + \dots + w_n z_n$$

Learning a regression function

- **Learning goal:** learn a function f^* such that its prediction in the future is the best.
 - Its generalization is the best.
- **Difficulty:** infinite number of functions
 - How can we learn?
 - Is function f better than g ?
- Use a measure
 - **Loss function** is often used to guide learning.



Loss function

■ Definition:

- The *error/loss* of the prediction for an observation $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$

$$r(\mathbf{x}) = [c_x - f(\mathbf{x})]^2 = (c_x - w_0 - w_1x_1 - \dots - w_nx_n)^2$$

- The *expected loss* of f over the whole space:

$$E = \mathbf{E}_x[r(\mathbf{x})] = \mathbf{E}_x[c_x - f(\mathbf{x})]^2$$

(\mathbf{E}_x is the expectation over \mathbf{x})

Cost, risk

- The goal of learning is to find f^* that minimizes the expected loss:

$$f^* = \arg \min_{f \in H} E_x [r(x)]$$

- H is the space of functions of linear form.

- **But**, we cannot work directly with this problem during the learning phase. (why?)

Empirical loss

- We can only observe a set of training data $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_M, y_M)\}$, and have to learn f from \mathbf{D} .

- **Empirical loss** (lỗi thực nghiệm, residual sum of squares):

$$RSS(f) = \sum_{i=1}^M (y_i - f(\mathbf{x}_i))^2 = \sum_{i=1}^M (y_i - w_0 - w_1 x_{i1} - \dots - w_n x_{in})^2$$

- $\frac{1}{M} RSS(f)$ is an approximation to $\mathbf{E}_x[r(\mathbf{x})]$.

- $\left| \frac{1}{M} RSS(f) - \mathbf{E}_x[r(\mathbf{x})] \right|$ is often known as *generalization error* of f .
(lỗi tổng quát hoá)

- Many learning algorithms base on this RSS and its variants.

Methods: ordinary least squares (OLS)

- Given \mathbf{D} , we find f^* that minimizes RSS:

$$f^* = \arg \min_{f \in H} \text{RSS}(f)$$

$$\Leftrightarrow \mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_{i=1}^M (y_i - w_0 - w_1 x_{i1} - \dots - w_n x_{in})^2 \quad (1)$$

- This method is often known as *ordinary least squares (OLS, bình phương tối thiểu)*.
- Find \mathbf{w}^* by taking the gradient of RSS and the solving the equation $\text{RSS}'=0$. We have:

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

- Where \mathbf{A} is the data matrix of size $M \times (n+1)$, whose the i^{th} row is $\mathbf{A}_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$; \mathbf{B}^{-1} is the inversion of matrix \mathbf{B} ; $\mathbf{y} = (y_1, y_2, \dots, y_M)^T$.
- Note: we assume that $\mathbf{A}^T \mathbf{A}$ is invertible (ma trận $\mathbf{A}^T \mathbf{A}$ khả nghịch).

Methods: OLS

- Input: $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_M, y_M)\}$
- Output: \mathbf{w}^*
- Learning: compute

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

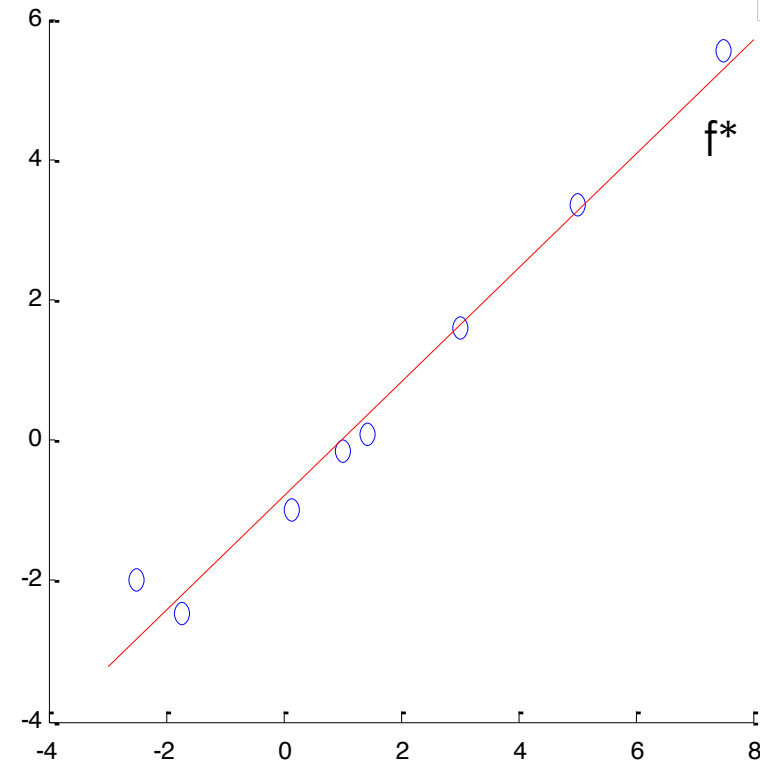
- Where \mathbf{A} is the data matrix of size $M \times (n+1)$, whose the i^{th} row is $\mathbf{A}_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$; \mathbf{B}^{-1} is the inversion of matrix \mathbf{B} ; $\mathbf{y} = (y_1, y_2, \dots, y_M)^T$.
 - Note: we assume that $\mathbf{A}^T \mathbf{A}$ is invertible.
- Prediction for a new \mathbf{x} :

$$y_x = w_0^* + w_1^* x_1 + \dots + w_n^* x_n$$

Methods: OLS example

x	y
0.13	-1
1.02	-0.17
3	1.61
-2.5	-2
1.44	0.1
5	3.36
-1.74	-2.46
7.5	5.56

$$f^*(x) = 0.81x - 0.78$$



Methods: limitations of OLS

- OLS cannot work if $\mathbf{A}^T\mathbf{A}$ is not invertible
 - If some columns (attributes/features) of \mathbf{A} are dependent, then \mathbf{A} will be singular and therefore $\mathbf{A}^T\mathbf{A}$ is not invertible.
- OLS requires considerable computation due to the need of computing a matrix inversion.
 - Intractable for the very high dimensional problems.
- OLS very likely tends to overfitting, because the learning phase just focuses on minimizing errors on the training data.

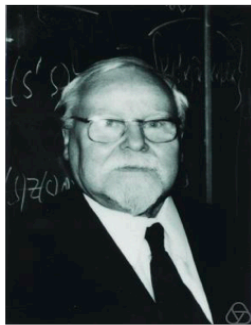
Methods: Ridge regression (1)

- Given $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_M, y_M)\}$, we solve for:

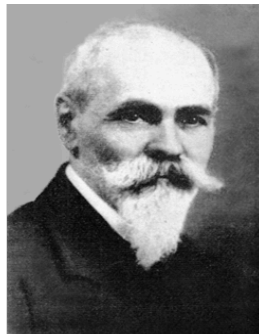
$$f^* = \arg \min_{f \in H} RSS(f) + \lambda \|\mathbf{w}\|_2^2$$

$$\Leftrightarrow \mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_{i=1}^M (y_i - \mathbf{A}_i \mathbf{w})^2 + \lambda \sum_{j=0}^n w_j^2 \quad (2)$$

- Where $\mathbf{A}_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$ is composed from \mathbf{x}_i ; and λ is a regularization constant ($\lambda > 0$). $\|\mathbf{w}\|_2$ is the L^2 norm.



Tikhonov,
smoothing an ill-
posed problem



Zaremba, model
complexity
minimization



Bayes: priors
over parameters



Andrew Ng: need no
maths, but it prevents
overfitting!

Methods: Ridge regression (2)

- Problem (2) is equivalent to the following:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_{i=1}^M (y_i - \mathbf{A}_i \mathbf{w})^2 \quad (3)$$

Subject to $\sum_{j=0}^n w_j^2 \leq t$

- for some constant t .
- The **regularization/penalty** term: $\lambda \|\mathbf{w}\|_2^2$
 - Limits the magnitude/size of \mathbf{w}^* (i.e., reduces the search space for \mathbf{f}^*).
 - Helps us to trade off between *the fitting of f on \mathbf{D}* and *its generalization* on future observations.

Methods: Ridge regression (3)

- We solve for \mathbf{w}^* by taking the gradient of the objective function in (2), and then zeroing it. Therefore we obtain:

$$\mathbf{w}^* = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I}_{n+1})^{-1} \mathbf{A}^T \mathbf{y}$$

- Where \mathbf{A} is the data matrix of size $M \times (n+1)$, whose the i^{th} row is $\mathbf{A}_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$; \mathbf{B}^{-1} is the inversion of matrix \mathbf{B} ; $\mathbf{y} = (y_1, y_2, \dots, y_M)^T$; \mathbf{I}_{n+1} is the identity matrix of size $n+1$.
- Compared with OLS, Ridge can
 - Avoid the cases of singularity, unlike OLS. Hence Ridge always works.
 - Reduce overfitting.
 - But error in the training data might be greater than OLS.
- **Note:** *the predictiveness of Ridge depends heavily on the choice of the hyperparameter λ .*

Methods: Ridge regression (4)

- Input: $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_M, y_M)\}$ and $\lambda > 0$
- Output: \mathbf{w}^*
- Learning: compute

$$\mathbf{w}^* = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I}_{n+1})^{-1} \mathbf{A}^T \mathbf{y}$$

- Prediction for a new \mathbf{x} :

$$y_x = w_0^* + w_1^* x_1 + \dots + w_n^* x_n$$

- **Note:** to avoid some negative effects of the magnitude of y on covariates \mathbf{x} , one should remove w_0 from the penalty term in (2). In this case, the solution of \mathbf{w}^* should be modified slightly.

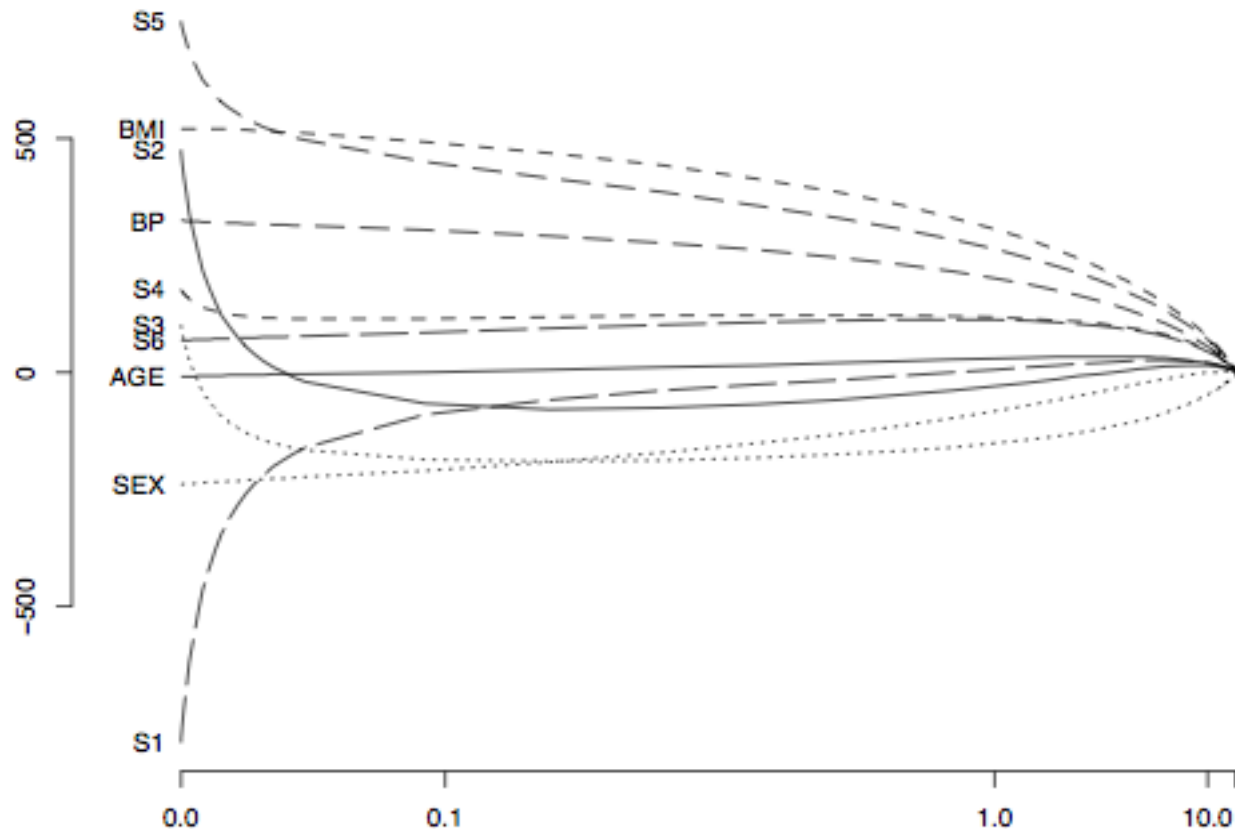
An example of using Ridge and OLS

- The training set **D** contains 67 observations on prostate cancer, each was represented with 8 attributes. Ridge and OLS were learned from **D**, and then predicted 30 new observations.

w	Ordinary Least Squares	Ridge
0	2.465	2.452
lcavol	0.680	0.420
lweight	0.263	0.238
age	-0.141	-0.046
lbph	0.210	0.162
svi	0.305	0.227
lcp	-0.288	0.000
gleason	-0.021	0.040
pgg45	0.267	0.133
Test RSS	0.521	0.492

Effects of λ in Ridge regression

- $\mathbf{W}^* = (w_0, S1, S2, S3, S4, S5, S6, \text{AGE}, \text{SEX}, \text{BMI}, \text{BP})$ changes as the regularization constant λ changes.



LASSO

- Ridge regression use L^2 norm for regularization:

$$w^* = \arg \min_{\mathbf{w}} \sum_{i=1}^M (y_i - \mathbf{A}_i \mathbf{w})^2, \text{ subject to } \sum_{j=0}^n w_j^2 \leq t \quad (3)$$

- Replacing L^2 by L^1 norm will result in LASSO:

$$w^* = \arg \min_{\mathbf{w}} \sum_{i=1}^M (y_i - \mathbf{A}_i \mathbf{w})^2$$

Subject to $\sum_{j=0}^n |w_j| \leq t$

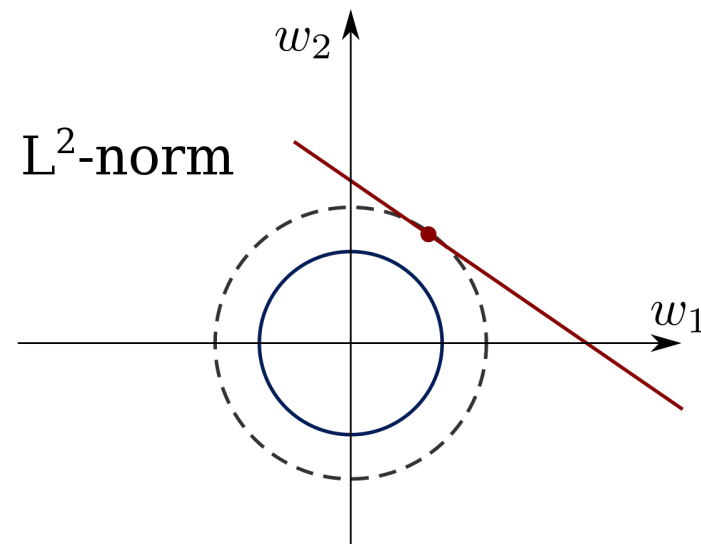
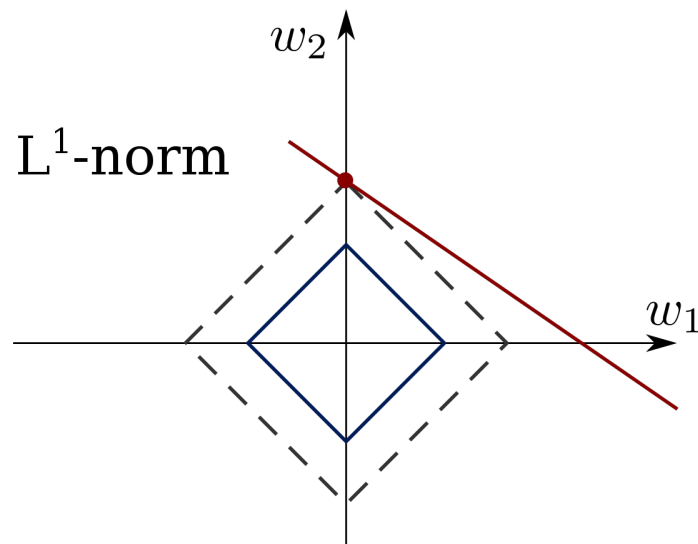
- Equivalently:

$$w^* = \arg \min_{\mathbf{w}} \sum_{i=1}^M (y_i - \mathbf{A}_i \mathbf{w})^2 + \lambda \|\mathbf{w}\|_1 \quad (4)$$

- This problem is non-differentiable \rightarrow the training algorithm should be more complex than Ridge.

LASSO: regularization role

- The regularization types lead to different domains for \mathbf{w} .
- LASSO often produces **sparse** solutions, i.e., many components of \mathbf{w} are zero.
 - Shrinkage and selection at the same time



OLS, Ridge, and LASSO

- The training set **D** contains 67 observations on prostate cancer, each was represented with 8 attributes. OLS, Ridge, and LASSO were trained from **D**, and then predicted 30 new observations.

w	Ordinary Least Squares	Ridge	LASSO
0	2.465	2.452	2.468
lcavol	0.680	0.420	0.533
lweight	0.263	0.238	0.169
age	-0.141	-0.046	
lbph	0.210	0.162	0.002
svi	0.305	0.227	0.094
lcp	-0.288	0.000	
gleason	-0.021	0.040	
pgg45	0.267	0.133	
Test RSS	0.521	0.492	0.479

Some weights are 0
 → some attributes may not be important

References

- Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.
- Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the lasso". *Journal of the Royal Statistical Society. Series B (methodological)*. Wiley. 58 (1): 267–88.

Exercises

- Derive the solution of (1) and (2) in details.
- Derive the solution of (2) when removing w_0 from the penalty term.