Machine Learning

INFQ612L, 440113450A
Spring Semester
Friday 17:00–18:40

IPS
WASEDA University

Prof. Shoji Makino

Machine Learning

Friday 17:00–18:40

1. 4/18
2. 4/25
3. 5/2
4. 5/9
5. 5/16
6. 5/23
7. 5/30
8. 6/6
9. 6/13
10. 6/20
11. 6/27

–. 7/4 No Lecture
–. 7/11 No Lecture
–. 7/18 No Lecture

At Zoom, set your name as:

Student ID, LAST_NAME, First_name

44251234, MAKINO, Shoji

At my class,

please turn on your camera

Machine Learning (8)(9)

Feature selection
and
L1 regularization

Wine Data Linear Regression

Negative impact

	Coefficients	Name
1	-0.193967	volatile acidity
6	-0.107356	total sulfur dioxide
4	-0.088183	chlorides
8	-0.063842	pH
2	-0.035553	citric acid
7	-0.033737	density
3	0.023019	residual sugar
0	0.043497	fixed acidity
5	0.045606	free sulfur dioxide
9	0.155277	sulphates
10	0.294243	alcohol

Mean squared err. 0.41676716722140794

Positive impact

Important features (volatile acidity, total sulfur dioxide)

Not so important feature?

Important features (sulphates, alcohol)

Document to Feature Vector: Bag-of-words

Document

In recent years, there has been a growing trend towards outsourcing of computational tasks with the development of cloud services. We propose two building blocks that work with FHE: a novel batch greater-than primitive, and matrix primitive for encrypted matrices

Dictionary (12,000 words)

ID	word
1168	batch
1169	bath
…	
1201	cloud
…	
1239	computation
1240	computational
…	
1172	primitive

Feature vector is characterized by whether or not it appears without considering frequency

$$\boldsymbol{x}_i = (0, 0, \ldots, 0, 1, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots)$$

Feature Selection

- Feature selection: We want to eliminate features that are unnecessary for predicting the target values and model using only useful features

12,000 dimension

$$t = \begin{pmatrix} w_1 & w_2 & \dots & w_D \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_D \end{pmatrix} = \sum_{i=1}^{D} w_i x_i$$

Bag-of-words
Most feature values are zero, so it is wasteful to calculate in all dimensions …

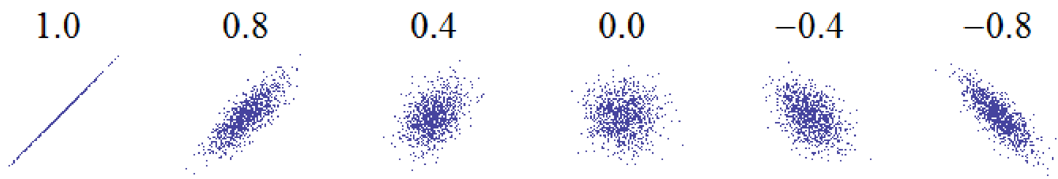
- Why feature selection?
  - Improve calculation efficiency (at the time of prediction)
  - Improved prediction accuracy of target values
    (Modeling with useless features reduces the quality of predictions)
  - Improved interpretation of learning results (which variables are effective for prediction?)

- How to implement feature selection
  - Filter method
  - Wrapper method
  - Utilization of regularization to introduce sparsity

Filter Method

- Data
  - D-dimensional feature variable (real value) $x_1, x_2, \ldots x_D$
  - Target variable (real value, for regression) t
- Score the strength of the association between each feature variable and the target variable
  - Example: Pearson correlation coefficient (Example: Correlation between s-th feature variable and target variable)

$$r_s = \frac{\sum_{i=1}^{N}(\bar{x}_s - x_{si})(\bar{t} - t_i)}{\sqrt{\sum_{i=1}^{N}(\bar{x}_s - x_{si})^2 \sum_{i=1}^{N}(\bar{t} - t_i)^2}}$$



- Filter method: Score all feature values and use only strongly related variables as features $|r_1| > \theta?, |r_2| > \theta?, \ldots, |r_D| > \theta?$
- If two or more variables affect the target value as a set, they cannot be selected by the filter method

(Naive) Rapper Method

- Data
  - D-dimensional feature variable (real value) $x_1, x_2, \ldots x_D$
  - Target variable (real value, for regression) t

- Goal: Select M features with feature index {1,2,3,⋯, D}

- All enumeration + cross-validation

  1) Divide the data into training data and test data
  2) List power sets $S \subseteq 2^{\{x_1, x_2, \ldots, x_D\}}$ of size M of {1,2,3,⋯, D}
  3) For each element of the power set:
     - In the training data, the model is trained with the features specified by the set
     - Evaluate the test error of the model
  4) Adopt a feature subset that gives the smallest test error

(Naive) Rapper Method

- Merit
  - Best feature selection in terms of test error
  - It can also handle ″when two or more variables affect the target value as a set″

- Problem
  - Subset candidates increase exponentially with feature number D
  - e.g., number of features D = 24, number of features to select M=12
  - Number of elements in the power set is ${}_{24}C_{12}$ = about 2.7 million
  - Training of 2.7 million models, test error evaluation by k−fold CV
    → It takes too much time

Wrapper Method: Positive Greedy Search

- Goal: Set features to {1,2,3,···, D} and select useful features
  1) Divide the data into training data and test data
  2) $i = 1$, $F = \{1\}$
  3) A model is trained with the features contained in F, and the test error is evaluated with the model. Let the test error be e^{i}
  4) Set $F' = F \cup \{i + 1\}$ and model learning with the features included in F'
  5) Model training is performed using the features included in F', and the test error is evaluated using the training model. Let the test error be e^{i+1}
  6) If $e^{i+1} < e^{i}$, then $F = F'$. Set $i = i + 1$ go to 3
- A search is also used to remove the features selected backwards
- Weaknesses:
  - Best feature set in terms of test error is not selected
  - Unable to control how many features are selected at the end
  - Final result depends on how the features are arranged

Norm (Lp Norm)

- Definition of distance in vector space (simply)
- p−norm

$$\|\boldsymbol{x}\|_p = \left(\sum_{d=1}^{D} |x_d|^p \right)^{1/p}$$

- 1−norm
  - Manhattan distance

$$\|\boldsymbol{x}\|_1 = \left(\sum_{d=1}^{D} |x_d| \right)$$

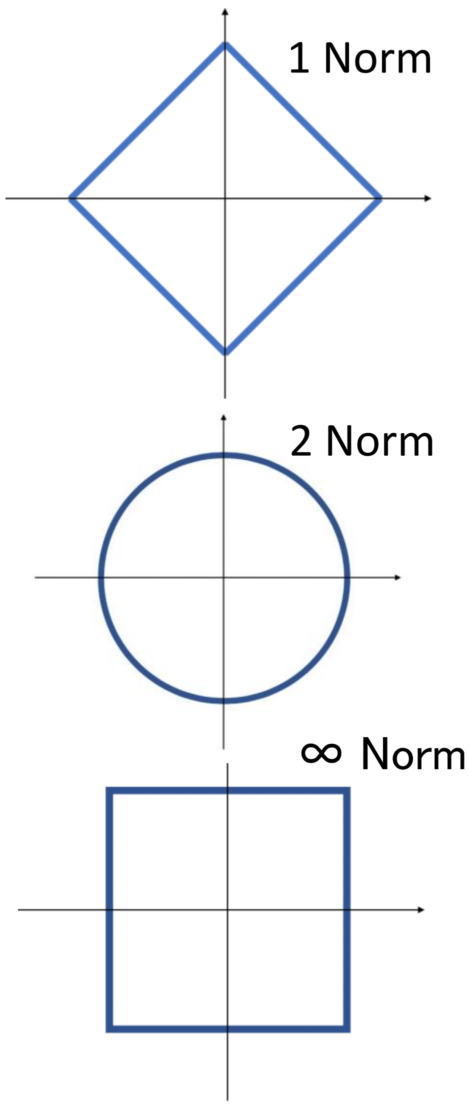
- 2−norm
  - Euclidean distance

$$\|\boldsymbol{x}\|_2 = \left(\sum_{d=1}^{D} (x_d)^2 \right)^{1/2}$$

- ∞−norm
  - Max norm

$$\|\boldsymbol{x}\|_\infty = \max_d |x_d|$$

Unit circles for each norm



Meaning of Regularization (for L2 Norm)

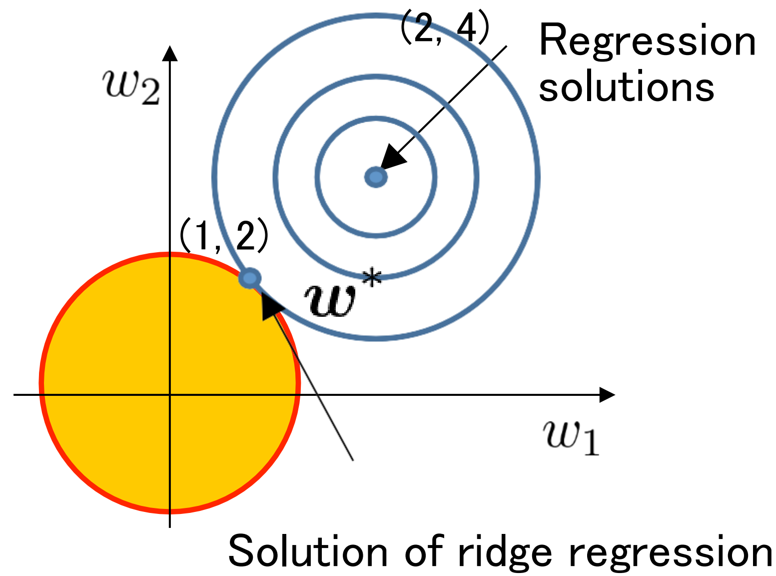
- Regression

$$E(\boldsymbol{w}) = \sum_{i=1}^{N} (t_i - \boldsymbol{w}^T \boldsymbol{x})^2$$

- Ridge regression

$$E(\boldsymbol{w}) = \sum_{i=1}^{N} (t_i - \boldsymbol{w}^T \boldsymbol{x})^2 + \lambda \underbrace{\boldsymbol{w}^T \boldsymbol{w}}_{\|\boldsymbol{w}\|_2^2}$$

Squared L2 norm $\|\boldsymbol{w}\|_2^2$



- Compare to a regression solution without regularization, a solution close to the origin can be found in the sense of the L2 norm
  - e.g., w* = (2, 4) → (1, 2)

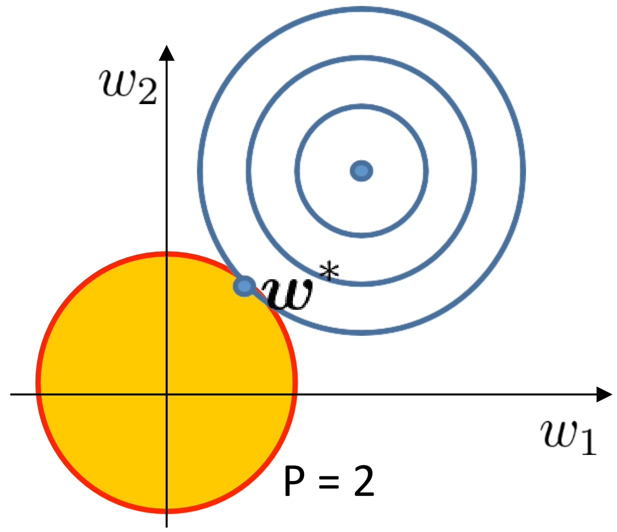
- The larger the regularization parameter λ , the bigger the penalty for larger norms
  → Solution with a smaller norm is selected rather than reducing the error
  → Attraction to the origin increases

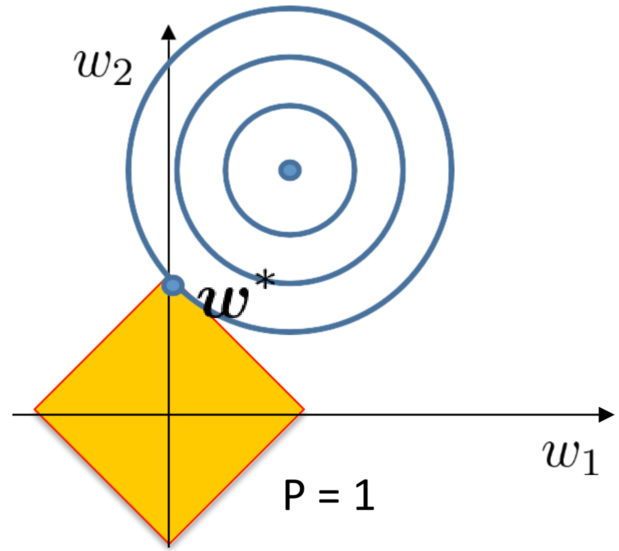
To Achieve Feature Selection by Regularization

- We want the value to be 0 (on the axis) in some dimensions of the obtained solution
  - e.g., w* = (0.1, 4.0) → (0, 3.8)
    - The first feature is not used for prediction
      → The second feature was selected as a useful feature
- What is a regularization term suitable for feature selection?

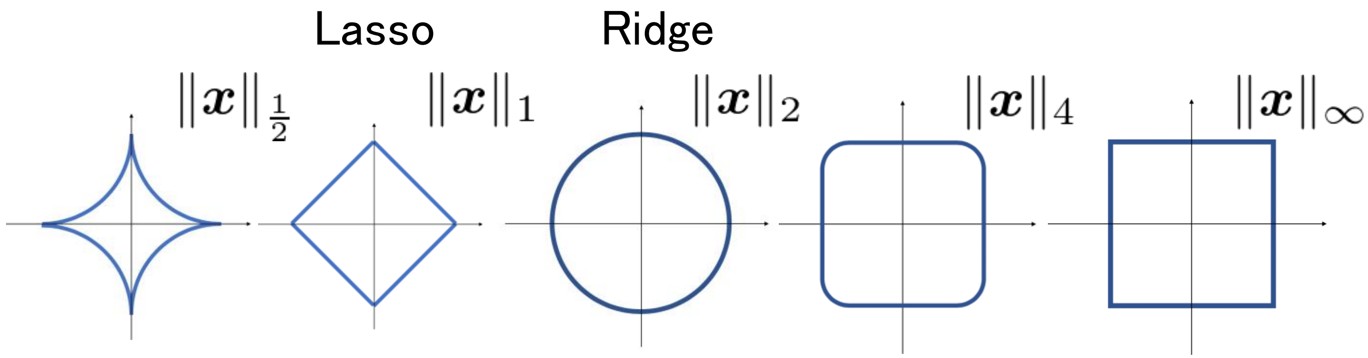
$$E(\boldsymbol{w}) = \sum_{i=1}^{N}(t_i - \boldsymbol{w}^T\boldsymbol{x})^2 + \lambda\|\boldsymbol{w}\|_p^p$$

- When p = 1, the part where the contour line of the objective function and the contour line of the regularization term meet is likely to be "on the axis"
- At this time, the coefficient becomes 0
  - Even p <1 has such a property, but it is difficult to optimize because it is not a convex function





Lasso = Squared Error Term + L1 Regularization Term



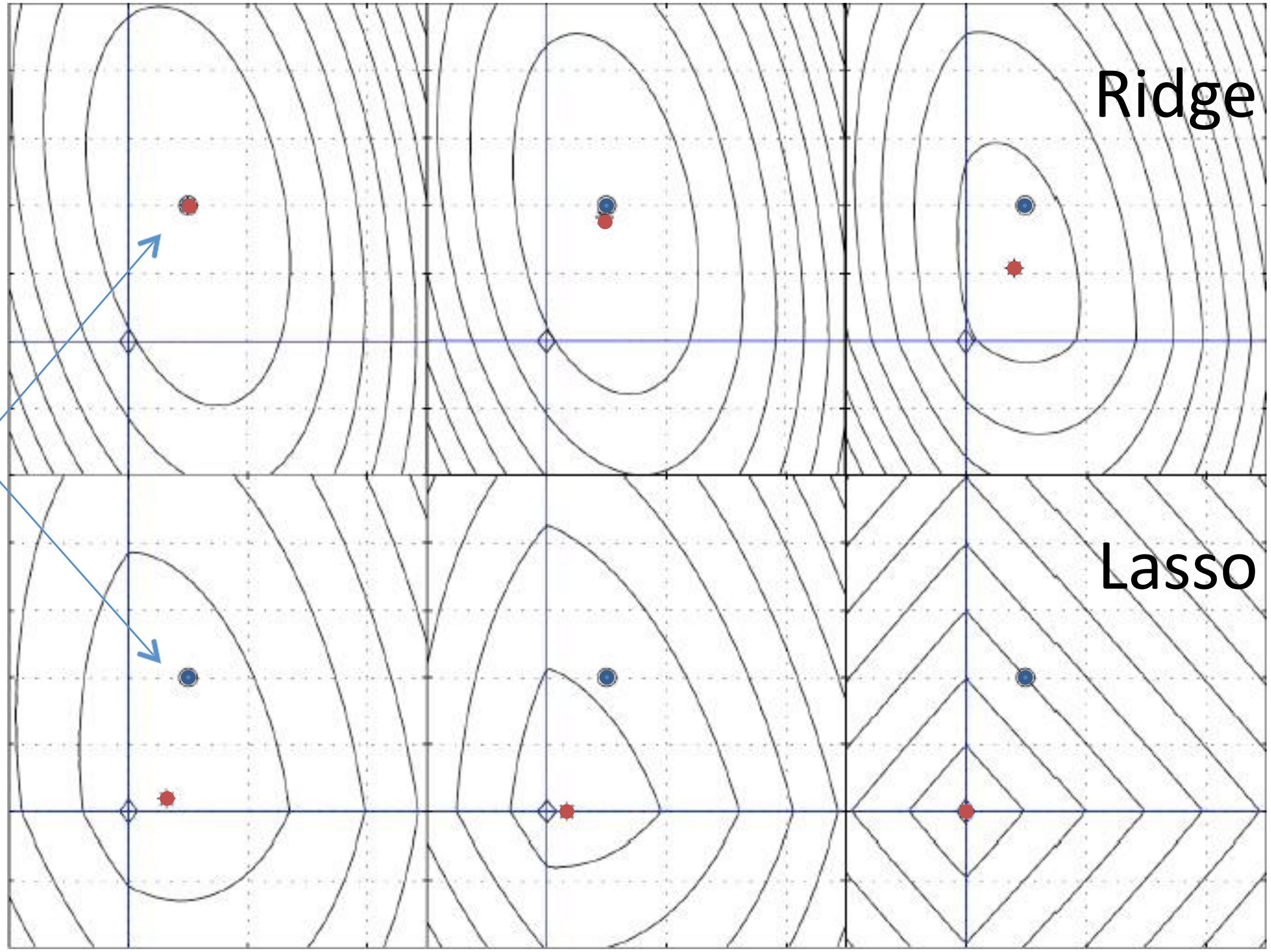
Contour lines of each regularization term

$$E(\boldsymbol{w}) = \sum_{i=1}^{N}(t_i - \boldsymbol{w}^T\boldsymbol{x})^2 + \overbrace{\lambda\sum_{i=1}^{D}|w_i|}^{\text{L1 regularization term}}$$

- L1 regularization: linear punishment for Manhattan distance from origin
- Both the error term and the L1 regularization term are convex functions
  - Optimal solution is relatively easy to find, but
  - Not differentiable near the origin → No analytical solution can be found
  - A little difficult to solve (Compared to L2) …

Why LASSO can do Feature Selection: Contour Line Change of Loss Function due to Regularization

Weak regularization ⟺ Strong regularization



Optimal solution when regularized parameter $\lambda = 0$

Regularization Path

How does the effect on the prediction of each feature (that is, the magnitude of the coefficient) change when the regularization coefficient is changed?

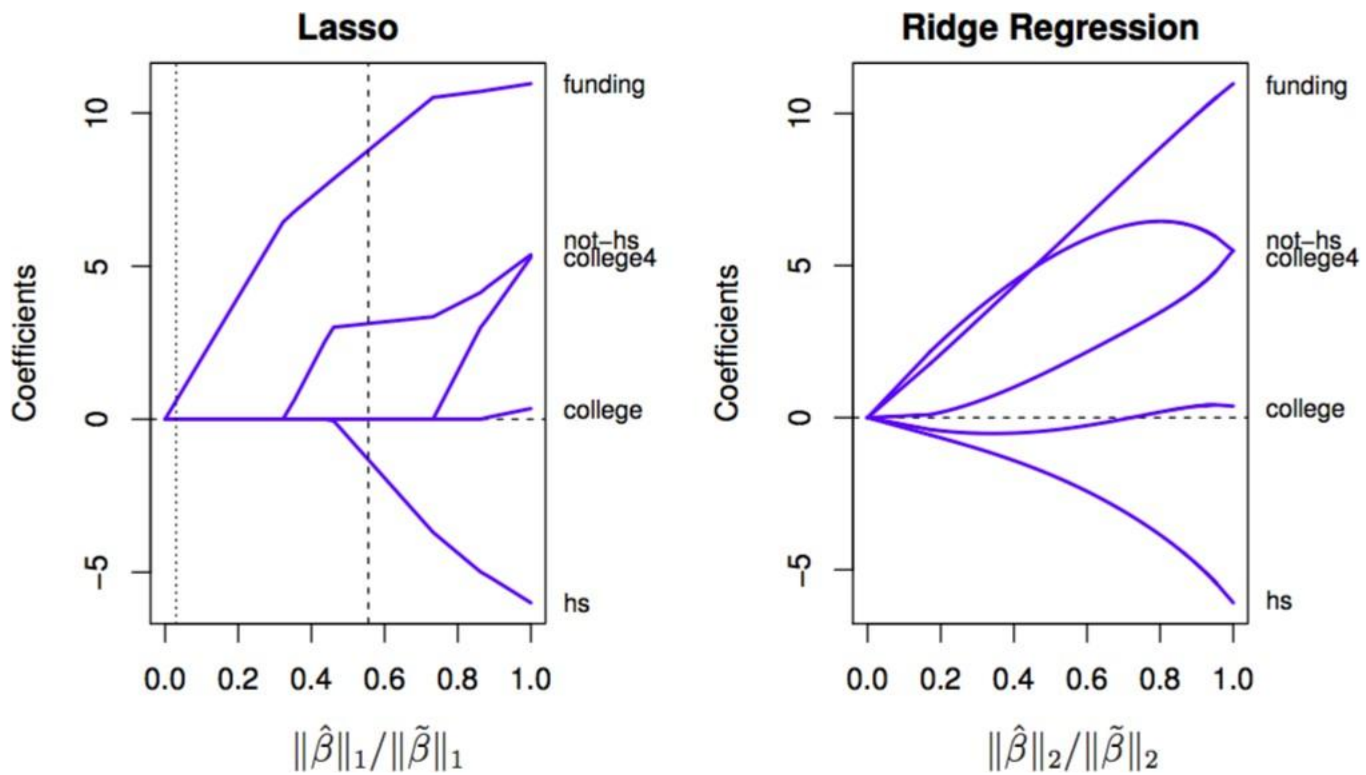
Table 2.1 *Crime data: Crime rate and five predictors, for $N = 50$ U.S. cities.*

city	funding	hs	not-hs	college	college4	crime rate
1	40	74	11	31	20	478
2	32	72	11	43	18	494
3	57	70	18	16	16	643
4	31	71	11	25	19	341
5	67	72	9	29	24	773
⋮	⋮	⋮	⋮	⋮		
50	66	67	26	18	16	940

Feature value:

Annual police budget, high school graduation rate over 25 years old, high school graduation rate 16-19 years old, college graduation rate 18-24 years old, 4-year college graduation rate over 25 years old

Target value: Crime rate



Feature Selection Summary

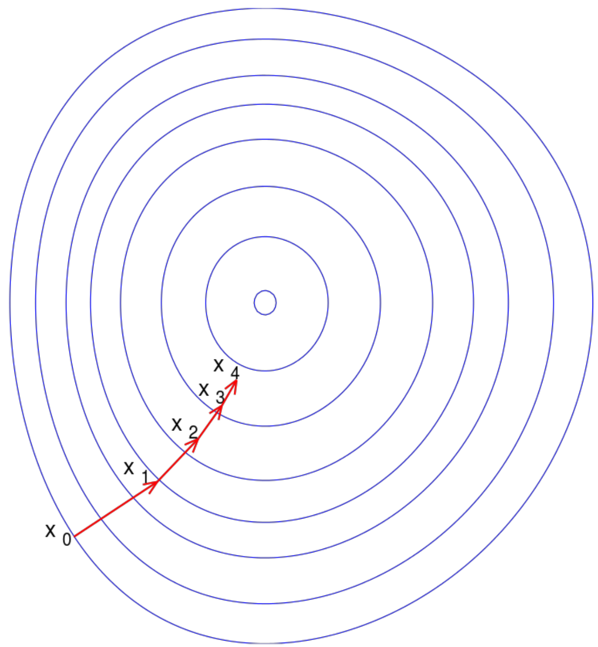
- Too many features lead to poor prediction accuracy and increased calculation time
- Feature selection is to narrow down such features
- There are two major feature selection methods
- Filter method / wrapper method
    - Enumerate the combinations of various features,
    - Exhaustive / experimental selection of good combinations in terms of test error
- L1 regularization
    - Introduces regularization that penalizes the number of features to be used, and makes it possible to control the number of features
    - Use k-fold CV to tune regularization parameter λ based on test error to ensure that the appropriate combination of features is used in the model as a result

Gradient Descent Method (Steepest Descent Method)

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \nabla E(\mathbf{w}^{(t)})$$
$$t \leftarrow t+1 \text{ until convergence}$$

- Calculate the gradient of *w* of the total training error of all training samples

$$\nabla E(\boldsymbol{w}^{(t)}) = \nabla \left(\sum_{i=1}^{N} [(\boldsymbol{w}^{(t)})^T \boldsymbol{x}_n - t_n]^2 \right)$$



- Gradient can be calculated relatively accurately, but the amount of calculation per update is *O (N)* for the number of samples *N*
- Not suitable when the number of samples is very large

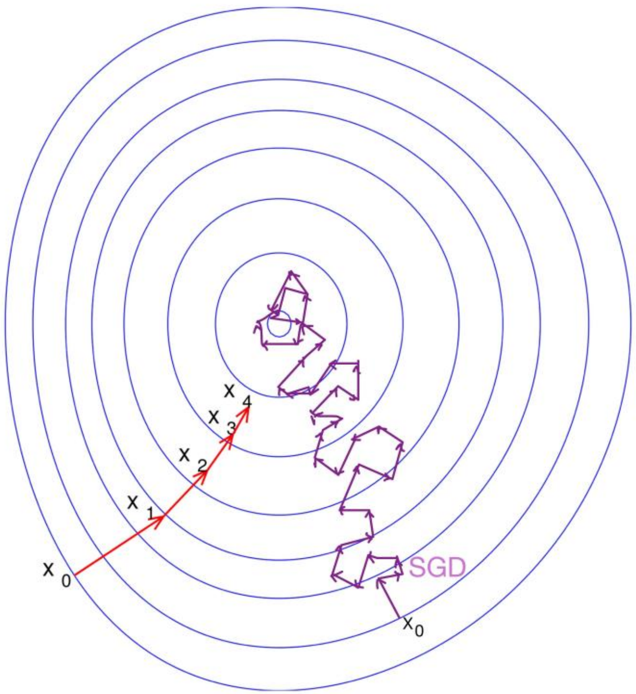
Stochastic Gradient Descent (SGD)

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \nabla E(\mathbf{w}^{(t)})$$
$$t \leftarrow t + 1 \text{ until convergence}$$

- Calculate the gradient for the training error of one randomly selected training sample

$$\nabla E(\boldsymbol{w}^{(t)}) = \nabla \left((\boldsymbol{w}^{(t)})^T \boldsymbol{x}_n - t_n)^2 \right)$$

- Although it is approximate in the sense of the gradient of *E(w)*, the amount of calculation per update is *O(1)* for the number of samples *N*
- There is a deviation from the gradient for all data, and the optimization process is unstable (training error does not decrease monotonically)



Mini Batch

$$\begin{gathered}\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \nabla E(\mathbf{w}^{(t)}) \\ t \leftarrow t+1 \text{ until convergence}\end{gathered}$$

- Mini batch D_t
  - A set of randomly selected relatively small number of data from all data
- Gradient calculation for prediction error of randomly selected mini-batch

$$\nabla E(\boldsymbol{w}^{(t)}) = \nabla \left(\sum_{(\boldsymbol{x},t) \in D_t}^{N} [(\boldsymbol{w}^{(t)})^T \boldsymbol{x} - t]^2 \right)$$

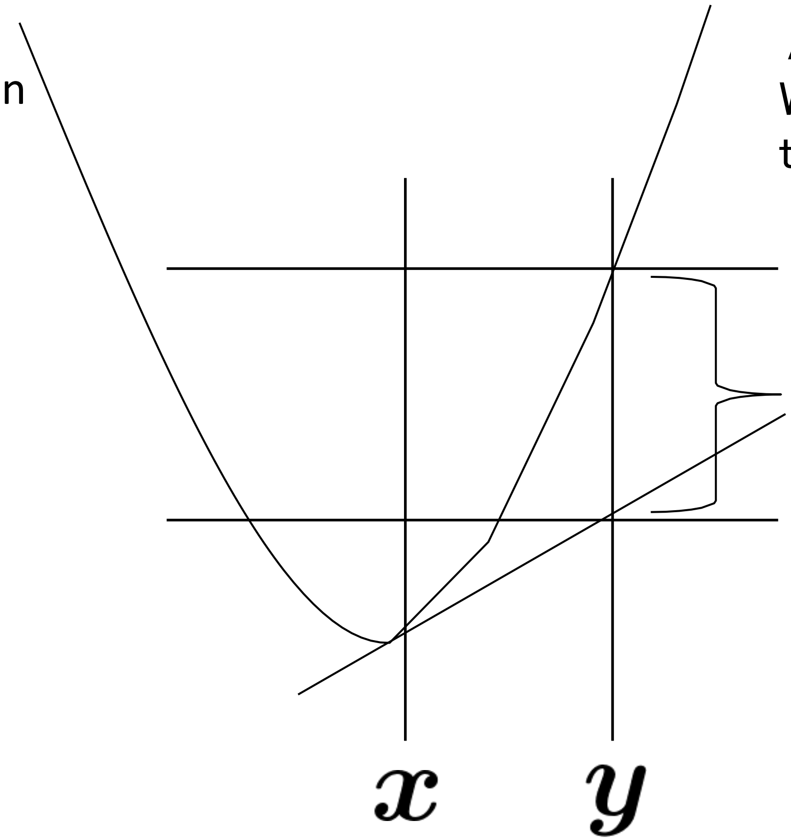
- Computational complexity per update is proportional to mini-batch size
- Gradients are not as unstable as SGD as they are calculated for mini-batch
- Effective use of parallel computer resources
  - Calculate the gradient independently for each batch

Convex Function Characterization
μ Strong Convex

- If the following is satisfied, f: RD → R is μ strong convex

$$f(\boldsymbol{y}) > f(\boldsymbol{x}) + (\boldsymbol{y}-\boldsymbol{x})^T \nabla f(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$
$$f(\boldsymbol{y}) > f(\boldsymbol{x}) + (\boldsymbol{y}-\boldsymbol{x})^T \nabla f(\boldsymbol{x})$$

cf. Convex function



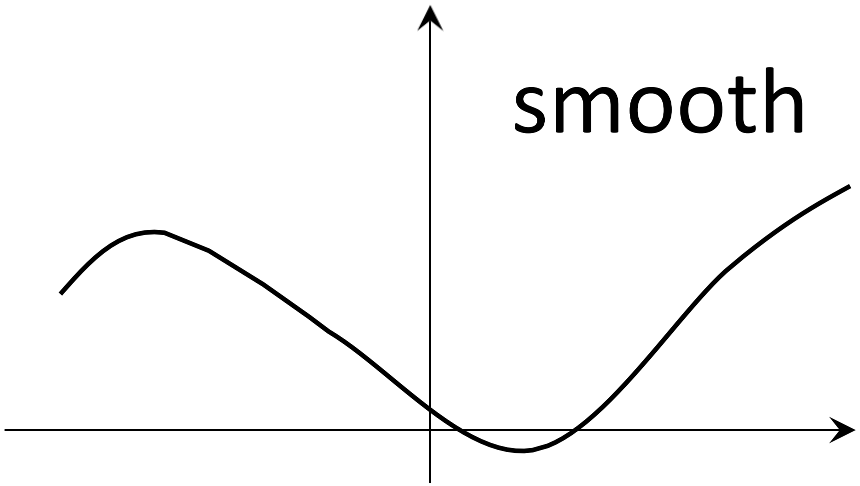
μ represents the strength of convexity
Whether there is still convexity even after subtracting the quadratic function

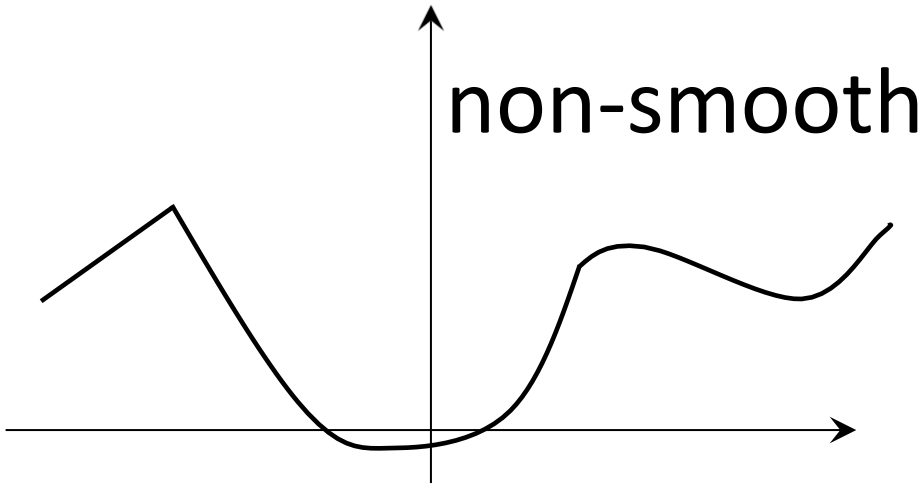
Convex Function Characterization: L Smooth

- If the following is satisfied, f: $R^D \rightarrow R$ is L smooth ($L > 0$)
  - Gradient change is smaller than a constant multiple of input change

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 < L\|\boldsymbol{x} - \boldsymbol{y}\|_2$$

  - Gradients of adjacent points are close to each other





Convergence of Gradient Descent

Theorem: $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} E(\boldsymbol{w})$ Also, let the function *E(w)* be a convex function that can be differentiated. If the function *E(w)* is L smooth (L> 0) and the step size is η <1/L, then:

$$E(\boldsymbol{w}^{(t)}) - E(\boldsymbol{w}^*) < \frac{2L\|\boldsymbol{w}^{(0)} - \boldsymbol{w}^*\|_2^2}{t+4}$$

- Convergent speed is *O (1/t)* for the number of steps *t*
- *O (1/ε)* updates are required to reduce the error to ε or less
  - (Roughly speaking) If an error of about ε = 0.01 is achieved by updating *T* times, an error of about ε = 0.001 can be achieved by updating about *10T* times

Convergence of Gradient Descent (Strong Convex)

Theorem: $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} E(\boldsymbol{w})$ The function *E(w)* is a differentiable convex function. If the function *E(w)* is μ strong convex (μ > 0) and L smooth L (> 0), and η = 1/L, then the following holds:

$$E(\boldsymbol{w}^{(t)}) - E(\boldsymbol{w}^*) < \left(1 - \frac{\mu}{L}\right)^t [E(\boldsymbol{w}^{(0)}) - E(\boldsymbol{w}^*)]$$

- Convergent speed is $O(c^t)$ for the number of steps t
- To reduce the error to ε or less, $O(\log(1/\varepsilon))$ updates are sufficient
  - (Roughly speaking) If an error of about ε = 0.01 is achieved by updating about *2T* times, an error of about ε = 0.001 can be achieved by updating about *3T* times
- Strong convex converges much faster (first-order convergence)
  - Regression and ridge regression are strong convex and L smooth
  - Logistic regression is L-smooth but not strong convex (will come out later)
  - Lasso, SVM is not L-smooth (will come out later)

Step Size Parameter (Learning Rate)

- Learning rate selection
  - If it is too small, convergence will be slow
  - If it is too large, it will converge quickly, but it will vibrate around the optimum solution
  - If it is very large, it will diverge
- Learning rate scheduling
  - Decrease the step size as the number of updates increases
  - Decrease the step size as the amount of updates decreases
  - Adaptively change the learning rate for each parameter
- Difficult to theoretically determine an appropriate learning rate
- Need to tune for each problem

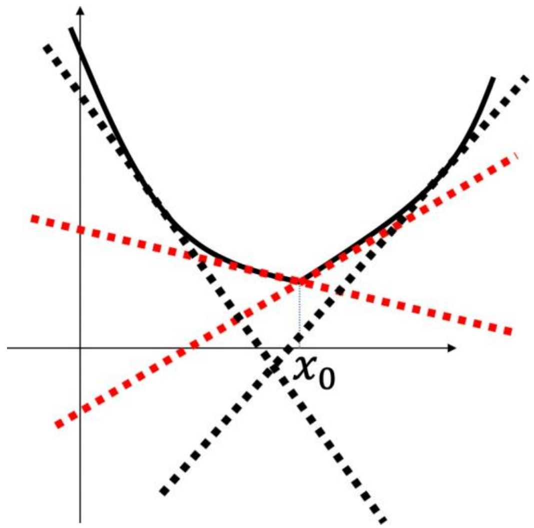
Sub-Gradient

- How to optimize an objective function that contains non-differentiable terms?

L1 regularization term is non-differentiable

$$E(\boldsymbol{w}) = \sum_{i=1}^{N}(t_i - \boldsymbol{w}^T\boldsymbol{x})^2 + \boxed{\lambda \sum_{i=1}^{D} |w_i|}$$

- Subgradient $\partial f(\mathbf{w}_0) = \{\mathbf{g} \in \mathbf{R}^D | \forall \mathbf{w}, f(\mathbf{w}) - f(\mathbf{w}_0) \geq (\mathbf{w} - \mathbf{w}_0)^T \mathbf{g}\}$
  - Example $f(\mathbf{w}) = |w_1| + 2|w_2|$



Sub-derivative in $\mathbf{w}_0 = (1, 0)^T$

$$\partial f(\mathbf{w}_0) = \left\{ \begin{pmatrix} 1 \\ 2z \end{pmatrix} \middle| z \in [1, -1] \right\}$$

Sub-Gradient Descent

- Replace the gradient in the gradient descent with a sub–gradient

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\partial E(\mathbf{w}^{(t)})$$

- cf. Gradient Descent

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta\nabla E(\mathbf{w}^{(t)})$$

Exercises 5: Derivation of ridge regression

4.4　リッジ回帰の最急降下法と確率的最急降下法

$$\boldsymbol{x}_i = \begin{pmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{iD} \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_N^T \end{pmatrix}, \boldsymbol{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}, \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{pmatrix}$$

とする。事例群 $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^N$ を使ってリッジ回帰による線形モデルを求めることを考える。

1. Derive the update formula of the gradient descent method of ridge regression
2. Derive the update formula of the stochastic gradient descent method of ridge regression

Machine Learning (8)(9)

Feature selection

and

L1 regularization

Machine Learning

INFQ612L, 440113450A

Spring Semester

Friday 17:00-18:40

IPS

WASEDA University

Prof. Shoji Makino