CE9010: Introduction to Data Science

Lecture 6: Generalization and Regularization

Semester 2 2017/18

Xavier Bresson

School of Computer Science and Engineering
Data Science and AI Research Centre
Nanyang Technological University (NTU), Singapore

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

- Generalization
- The problem of over-fitting
- Addressing over-fitting
- Regularization
- Regularization by gradient descent
- Regularization by normal equation
- Conclusion

Outline

- **Generalization**
- The problem of over-fitting
- Addressing over-fitting
- Regularization
- Regularization by gradient descent
- Regularization by normal equation
- Conclusion

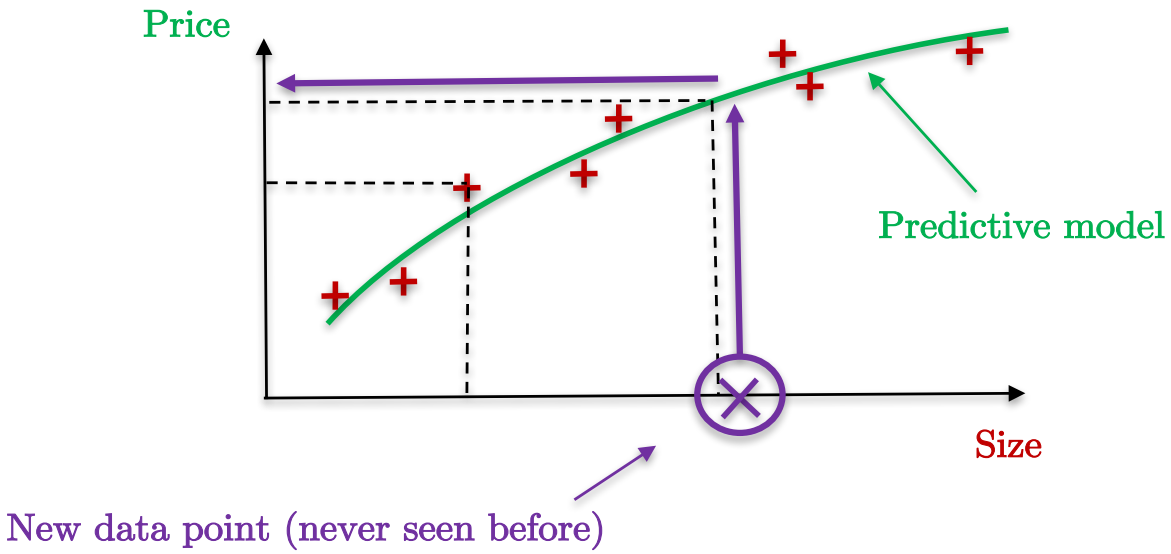
Generalization

- Generalization is the goal of all predictive models.
  - The generalization problem consists at making accurate predictions to new data points (never seen before).

- Any data science task has two stages:
  - Learning stage: Use training data to learn the parameters of a predictive model.
  - Testing stage: Once training is over, apply the predictive model to unseen data points.

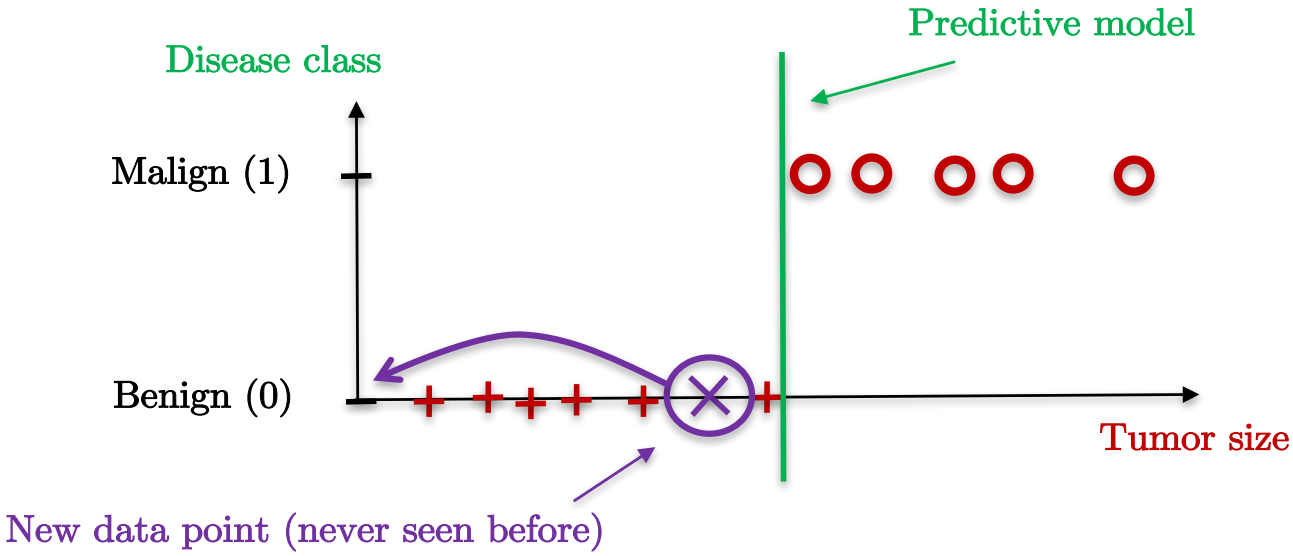
Generalization

- **Example:** Supervised regression



Quiz

- What generalization means for supervised classification?

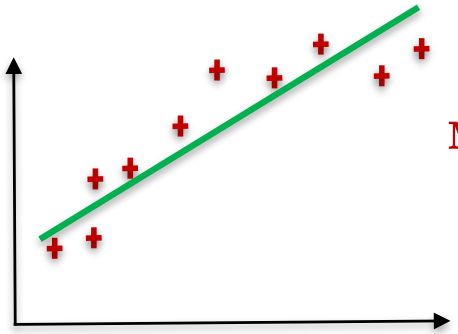


Outline

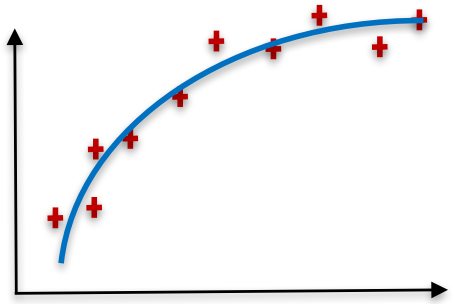
- Generalization
- **The problem of over-fitting**
- Addressing over-fitting
- Regularization
- Regularization by gradient descent
- Regularization by normal equation
- Conclusion

The problem of data fitting

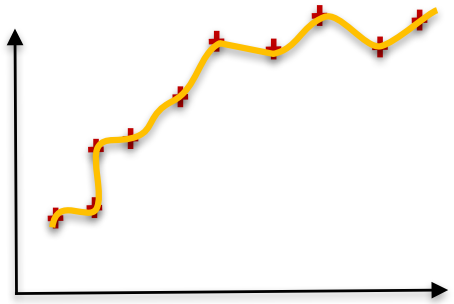
- Example: Regression task



More learning capacity ⇒



More learning capacity ⇒



Linear prediction

$f_w(x) = w_0 + w_1 x$

Quadratic prediction

$f_w(x) = w_0 + w_1 x + w_2 x^2$

Polynomial prediction

$f_w(x) = w_0 + w_1 x + \ldots$
$w_2 x^2 + \ldots + w_6 x^6$

Under-fitting
The predictive model does not fit well the training data.

The data assumption (linear data) does not reflect the true housing price (too simple model).

Right-fitting
The predictive model fits "right" the training data.

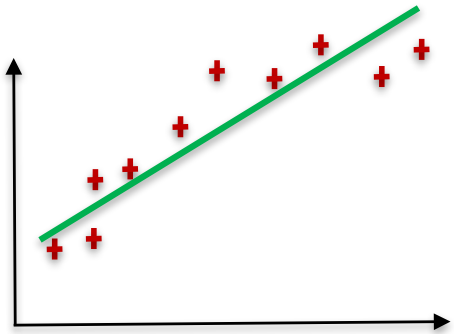
The data assumption (quadratic data) reflects the true housing price as it will predict well prices on new examples.

Over-fitting
The predictive model fits perfectly the training data.

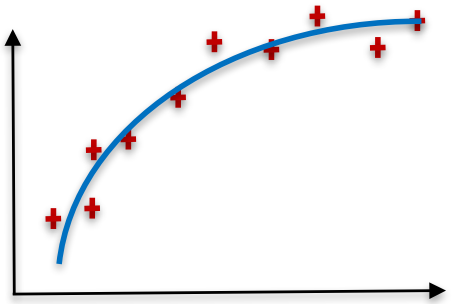
The data assumption (polynominal data) does not reflect the true housing price, (too complex model).

Bias and variance

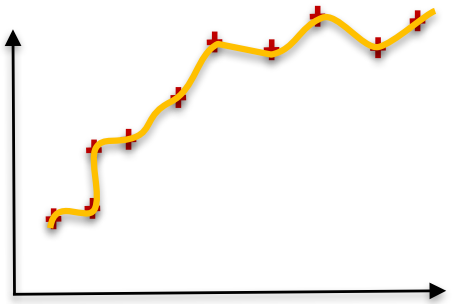
- **Model bias:** Variations between predictive value and training value
- **Model variance:** Variations of predictive function



More learning capacity ⇒



More learning capacity ⇒



Linear prediction

$f_w(x) = w_0 + w_1 x$

Quadratic prediction

$f_w(x) = w_0 + w_1 x + w_2 x^2$

Polynomial prediction

$f_w(x) = w_0 + w_1 x + ... \; w_2 x^2 + ... + w_6 x^6$

Under-fitting
High bias
Small variance

Unlikely to generalize

Right-fitting
Good balance between bias and variance

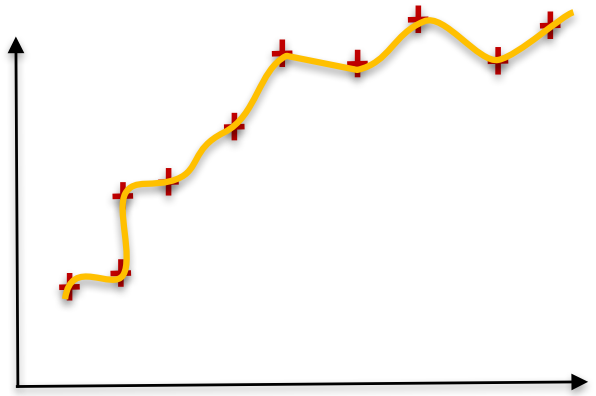
Good generalization

Over-fitting
Small bias
High variance

Unlikely to generalize

The problem of over-fitting

- Over-fitting is the most common fitting problem (e.g. neural networks) because predictive models have usually large capacity to learn (high number of weight parameters).

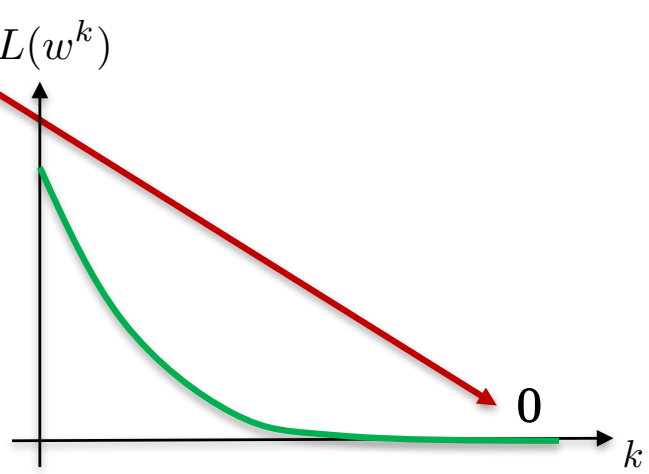


- Problem: The predictive function *f* fits almost perfectly all training data:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}\left(f_w(x_i) - y_i\right)^2 \approx 0$$

but fails to generalize to new data examples.

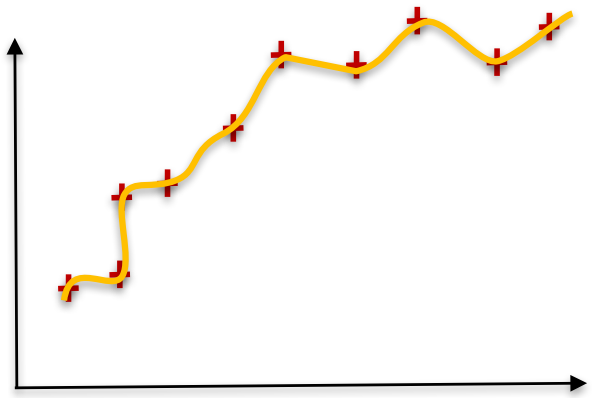
- Dynamic of the gradient descent:



The problem of over-fitting

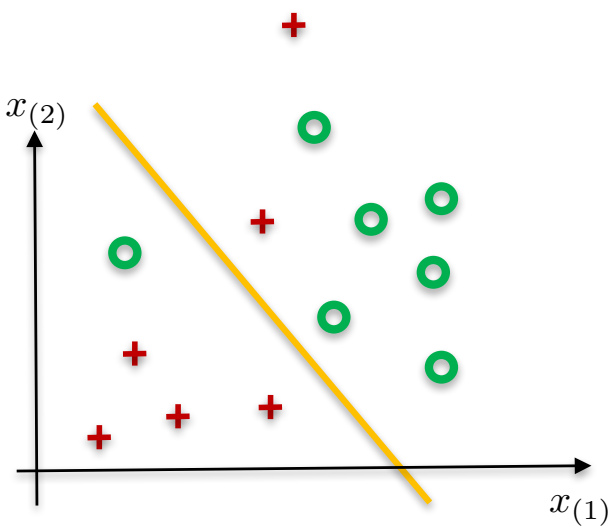
- Problem (different view): The space of prediction is too large ⇒ Non-meaningful predictions will be done.

- Curse of dimensionality: The over-fitting problem is more important when data have too many variables because they lie in high-dimensional spaces.



Quiz

- What over-fitting means for the classification task?

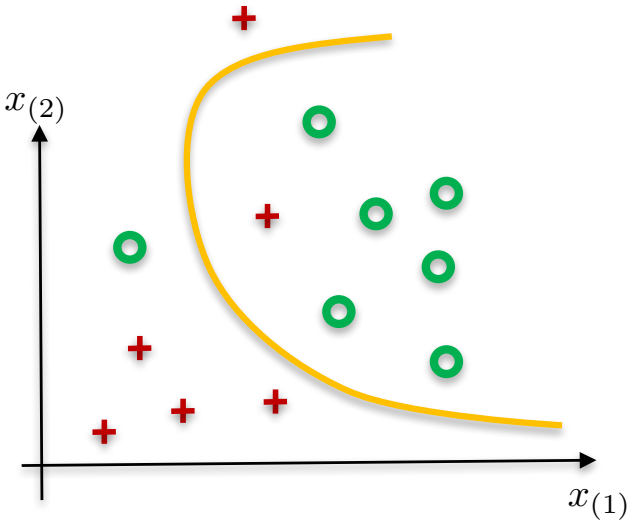


$p_w(x) = \sigma(w_0 + w_1 x_{(1)} + w_2 x_{(2)})$

Linear function

Under-fitting
High bias
Small variance

Unlikely to generalize

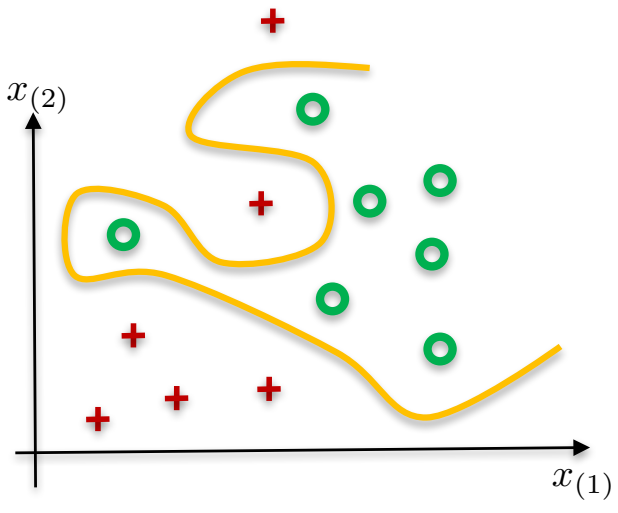


$p_w(x) = \sigma(w_0 + w_1 x_{(1)} + ... + w_4 x_{(2)}^2)$

Quadratic function

Right-fitting
Good balance between
bias and variance

Good generalization



$p_w(x) = \sigma(w_0 + w_1 x_{(1)} + ... + w_6 x_{(2)}^6)$

Polynomial function

Over-fitting
Small bias
High variance

Unlikely to generalize

Outline

- Generalization
- The problem of over-fitting
- **Addressing over-fitting**
- Regularization
- Regularization by gradient descent
- Regularization by normal equation
- Conclusion

How to address over-fitting?

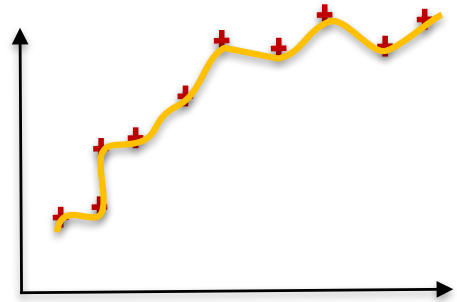
- Generalization needs to fix the issue of over-fitting.
- They are two options:
  - Reduce the number of data features: Arbitrary select less features or use feature selection model to automatically select the most meaningful features.
    - Two problems:
      - We may lose good information about data.
      - Feature selection models are not perfect.
  - Regularization: Adapt the importance of the features depending on the data and the task at hand.
    - Advantages:
      - Keep all features.
      - Learning stage will compute the weight values.
      - Good for high-dim data

Outline

- Generalization
- The problem of over-fitting
- Addressing over-fitting
- **Regularization**
- Regularization by gradient descent
- Regularization by normal equation
- Conclusion

Intuition

- Idea: Starting from high-capacity (cubic) function, we enforce the weight values w_3, w_4 to be small $\Rightarrow$ The high-capacity function becomes a simpler, smoother (quadratic) function:



Cubic prediction

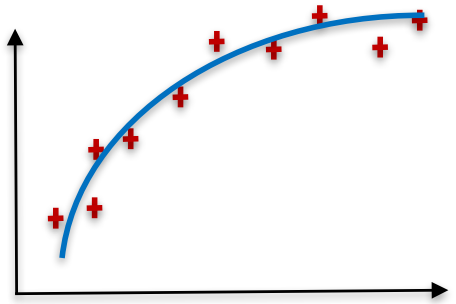
$$f_w(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

Over-fitting

Less learning capacity

$\Rightarrow$

$w_3, w_4 \approx 0$



Quadratic prediction

$$f_w(x) = w_0 + w_1 x + w_2 x^2$$

Right-fitting

- How to make the weights small?

Penalization

- We can enforce the weights to be small by penalizing their values:

$$\min_w \quad \frac{1}{n}\sum_{i=1}^{n}\left(f_w(x_i)-y_i\right)^2 + 1000.w_3^2 + 1000.w_4^2$$

Arbitrary large number

Minimizing this loss function will force w_3, w_4 to have a small value, that is $w_3, w_4 \approx 0$.

$$f_w(x) = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4 \quad \Rightarrow \quad w_3, w_4 \approx 0 \quad f_w(x) = w_0 + w_1x + w_2x^2$$

Regularization

- Idea: Start with a high-capacity predictive function and learn smoother, lower-capacity function by optimizing a (regularized) loss that penalizes the values of the parameter weights of the predictive function:

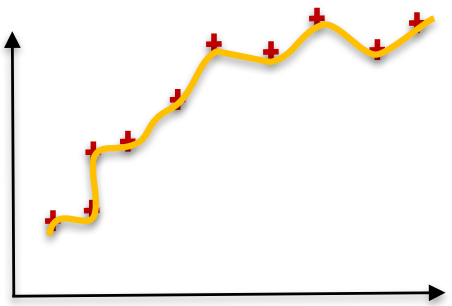
Regularization parameter

$$\min_w \quad L(w) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(f_w(x_i) - y_i\right)^2}_{\text{Fitting loss between predictive function and training data}} + \underbrace{\frac{\lambda}{d}\sum_{j=1}^{d} w_j^2}_{\text{Penalty/ regularization loss (enforce small values) a.k.a. L2 loss}}$$

Regularized loss
Trade-off between perfect fit and smooth predictive function

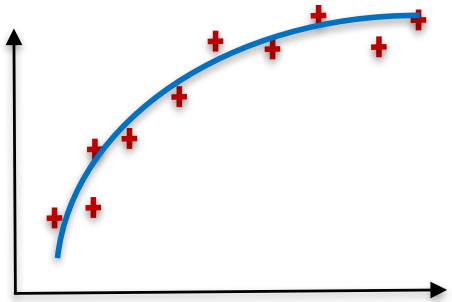
Quiz

- The regularization parameter λ controls the generalization/over-fitting issue. It is an hyper-parameter that can be estimated by cross-validation (later discussed).
- What is the predictive function for a large λ value?



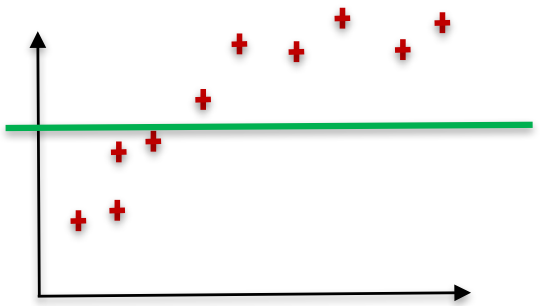
$\lambda = 0$

Over-fitting



$\lambda = 1000$

Right-fitting



$\lambda = 1,000,000$

Under-fitting

$$\min_w \quad L(w) = \frac{1}{n}\sum_{i=1}^{n}\Big(f_w(x_i) - y_i\Big)^2 + \frac{\lambda}{d}(w_1^2 + w_2^2 + w_3^2 + w_4^2)$$

$$\lambda = 10^6 \Rightarrow w_1 = w_2 = w_3 = w_4 = 0$$

$$f_w(x) = w_0 + w_1 x + \ldots + w_d x^d \qquad \Rightarrow \qquad f_w(x) = w_0$$

Outline

- Generalization
- The problem of over-fitting
- Addressing over-fitting
- Regularization
- **Regularization by gradient descent**
- Regularization by normal equation
- Conclusion

Regularized loss

- **Regularized regression loss:**

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}\left(f_w(x_i) - y_i\right)^2 + \frac{\lambda}{d}\sum_{j=1}^{d} w_j^2$$

$$f_w(x) = w^T x$$

- **Regularized classification loss:**

$$L(w) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log p_w(x_i) + (1 - y_i)\log(1 - p_w(x_i)) + \frac{\lambda}{d}\sum_{j=1}^{d} w_j^2$$

$$p_w(x) = \sigma(w^T x)$$

Optimization by gradient descent

- **Optimization:** $\min_w L(w)$
- **Gradient descent:** $w_j \leftarrow w_j - \tau \frac{\partial}{\partial w_j} L(w)$
- **Gradient:**

$$\frac{\partial}{\partial w_j} L(w) = \frac{1}{n} \sum_{i=1}^{n} \Big(f_w(x_i) - y_i \Big) x_{i(j)} + \frac{\lambda}{d} \frac{\partial}{\partial w_j} \Big(\sum_{j=1}^{d} w_j^2 \Big)$$

$$\frac{1}{n} \sum_{i=1}^{n} \Big(f_w(x_i) - y_i \Big) x_{i(j)} + \frac{2\lambda}{d} w_j$$

Decreasing effect

- Gradient descent:

$$w_j \leftarrow w_j - \tau \frac{\partial}{\partial w_j} L(w)$$

$$w_j \leftarrow \underbrace{(1 - \frac{2\tau\lambda}{d})}_{\text{Shrinking effect}} w_j - \tau \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i)) x_{i(j)}$$

Value smaller than 1 (e.g. 0.9)
Decrease the value of w_j towards 0.

Quiz

- How fast is the decreasing property?

$$w_j \leftarrow \underbrace{(1 - \frac{2\tau\lambda}{d})}_{\text{Shrinking effect}} w_j - \tau \frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i)) x_{i(j)}$$

Value smaller than 1 (e.g. 0.9)
Decrease the value of w_j towards 0 exponentially fast:

$$\begin{aligned} w^{k=0} &= w_0 \\ w^{k=1} &= 0.9 w^{k=0} = 0.9 w_0 \\ w^{k=2} &= 0.9 w^{k=1} = 0.9^2 w_0 \\ &\vdots \\ w^{k} &= 0.9 w^{k-1} = 0.9^k w_0 \end{aligned}$$

$0.9^{\#\text{iter}}$, examples: 0.9^{10}=0.34 and 0.9^{100}=0.00001

Outline

- Generalization
- The problem of over-fitting
- Addressing over-fitting
- Regularization
- Regularization by gradient descent
- **Regularization by normal equation**
- Conclusion

Normal equation

- Solution of MSE loss with linear predictive function:

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} \left(f_w(x_i) - y_i \right)^2$$

$$\Downarrow$$

$$\min_w \; L(w) \quad \Leftrightarrow \quad \frac{\partial}{\partial w} L(w) = 0 \quad \Rightarrow \quad w = (X^T X)^{-1} X^T y$$

One line of code

$$\underset{(d+1)\text{ x }1}{w} = \begin{bmatrix} w_0 \\ \vdots \\ w_d \end{bmatrix} \qquad \underset{(d+1)\text{ x }1}{x_i} = \begin{bmatrix} x_{i(0)} \\ \vdots \\ x_{i(d)} \end{bmatrix} \qquad \underset{\text{n x (d+1)}}{X} = \begin{bmatrix} —x_1^T— \\ \vdots \\ —x_n^T— \end{bmatrix} \qquad \underset{\text{n x 1}}{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

with $x_{i(0)} = 1$

Data matrix

Regularization with normal equation

- Regularized regression loss:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}\Big(f_w(x_i) - y_i\Big)^2 + \frac{\lambda}{d}\sum_{j=1}^{d} w_j^2$$

$\Downarrow$ Matrix-vector representation

$$L(w) = \frac{1}{n}\Big(Xw - y\Big)^T\Big(Xw - y\Big) + \frac{\lambda}{d}w^T w$$

Regularization with normal equation

- **Gradient:**

$$\begin{aligned}
\frac{\partial}{\partial w}L(w) &= \frac{\partial}{\partial w}\Big[\frac{1}{n}\Big(Xw-y\Big)^T\Big(Xw-y\Big)+\frac{\lambda}{d}w^Tw\Big] \\
&= \frac{1}{n}\frac{\partial}{\partial w}\Big[\Big(w^TX^T-y^T\Big)\Big(Xw-y\Big)\Big]+\frac{\lambda}{d}\frac{\partial}{\partial w}\Big[w^Tw\Big] \\
&= \frac{2}{n}X^T\Big(Xw-y\Big)+\frac{2\lambda}{d}w \\
&= \frac{2}{n}\Big((X^TX+\frac{\lambda}{d}I)w-X^Ty\Big) \\
&= 0 \Rightarrow w=(X^TX+\frac{\lambda}{d}I)^{-1}X^Ty
\end{aligned}$$

Identity matrix $Iw = w$

One line of code

(Bonus) Normal equation for non-invertible matrix

- NE do not admit a solution when n<d, i.e. when the number of training data n is smaller than the data dimensionality d.

  Example genetics: 100K genes and 1K training data.

- This is called an over-parametrized linear system of equations. The matrix X^TX is not invertible.

- Examples:

$$a_{11}w_1 + a_{12}w_2 = b_1$$
$$a_{21}w_1 + a_{22}w_2 = b_2$$

2 unknowns w_1, w_2
and 2 equations
✓

$$a_{11}w_1 + a_{12}w_2 = b_1$$

2 unknowns w_1, w_2
but 1 equation
✘
(Infinite number of solutions)

- Solution is to simply regularize: $w = (X^TX + \frac{\lambda}{d}I)^{-1}X^Ty$

Outline

- Generalization
- The problem of over-fitting
- Addressing over-fitting
- Regularization
- Regularization by gradient descent
- Regularization by normal equation
- **Conclusion**

Conclusion

- Generalization is the ultimate goal of learning techniques, and one obstacle is over-fitting.

- Over-fitting can be reduced by
  - Selecting of small number of features (domain expertise, limited)
  - Regularization (L2 loss, dropout for neural networks)
  - Data augmentation (can be challenging to produce more data)

- Selection of regularization hyper-parameter can be time-consuming (cross-validation).

Questions?