Chapter ML:III

III. Linear Models

- Logistic Regression
- Overfitting
- Regularization

Logistic Regression

Binary Classification Problems

Setting:

- The feature space $X \subseteq \mathbf{R}^p$ is an inner product space.
- $C = \{0, 1\}$ is a set of two classes. Similarly: $\{-1, 1\}$, $\{\ominus, \oplus\}$, $\{\text{no}, \text{yes}\}$, etc.
- $c : X \to C$ is the (unknown) ideal classifier for X.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

Todo:

- Approximate c, which is implicitly given via D, with a logistic model function.

Logistic Regression

Binary Classification Problems

Setting:

- The feature space $X \subseteq \mathbf{R}^p$ is an inner product space.
- $C = \{0, 1\}$ is a set of two classes. Similarly: $\{-1, 1\}$, $\{\ominus, \oplus\}$, $\{\text{no}, \text{yes}\}$, etc.
- $c : X \to C$ is the (unknown) ideal classifier for X.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

Todo:

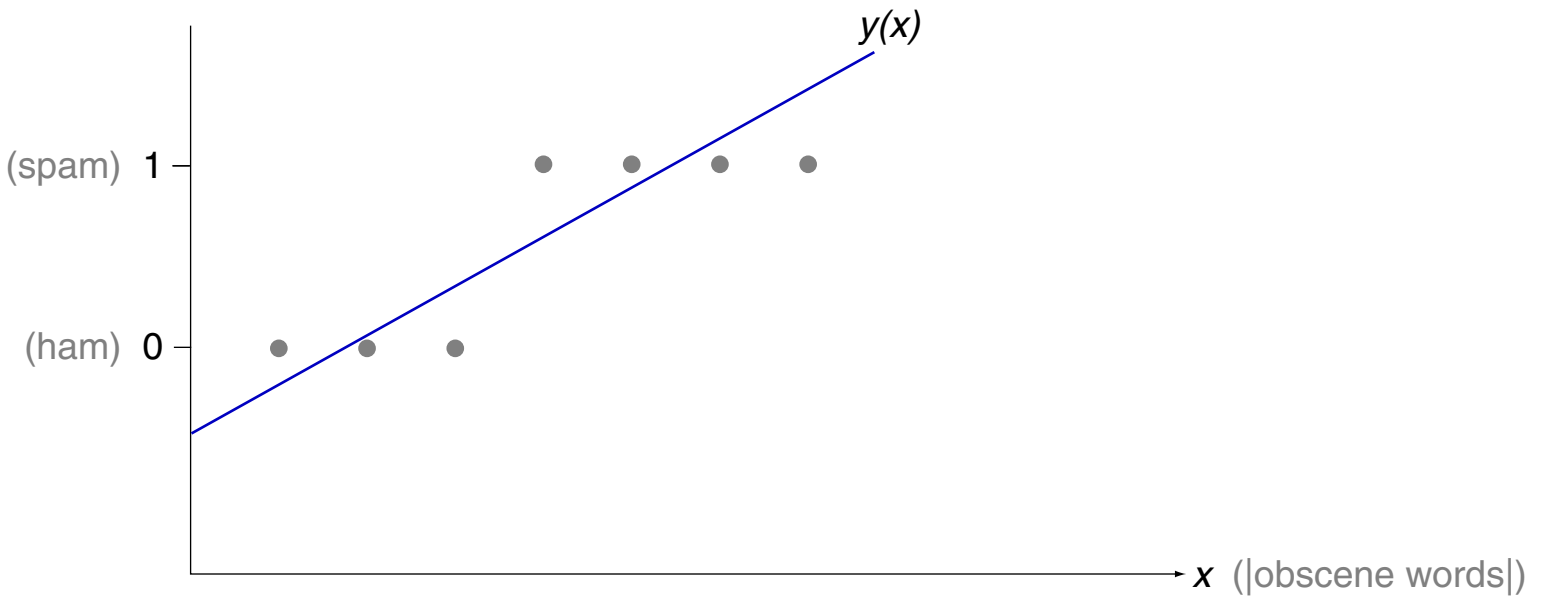
- Approximate c, which is implicitly given via D, with a logistic model function.

Examples for binary classification problems:

- Is an email spam or ham?
- Is a patient infected or healthy?
- Is a bank customer creditworthy or not?

Logistic Regression

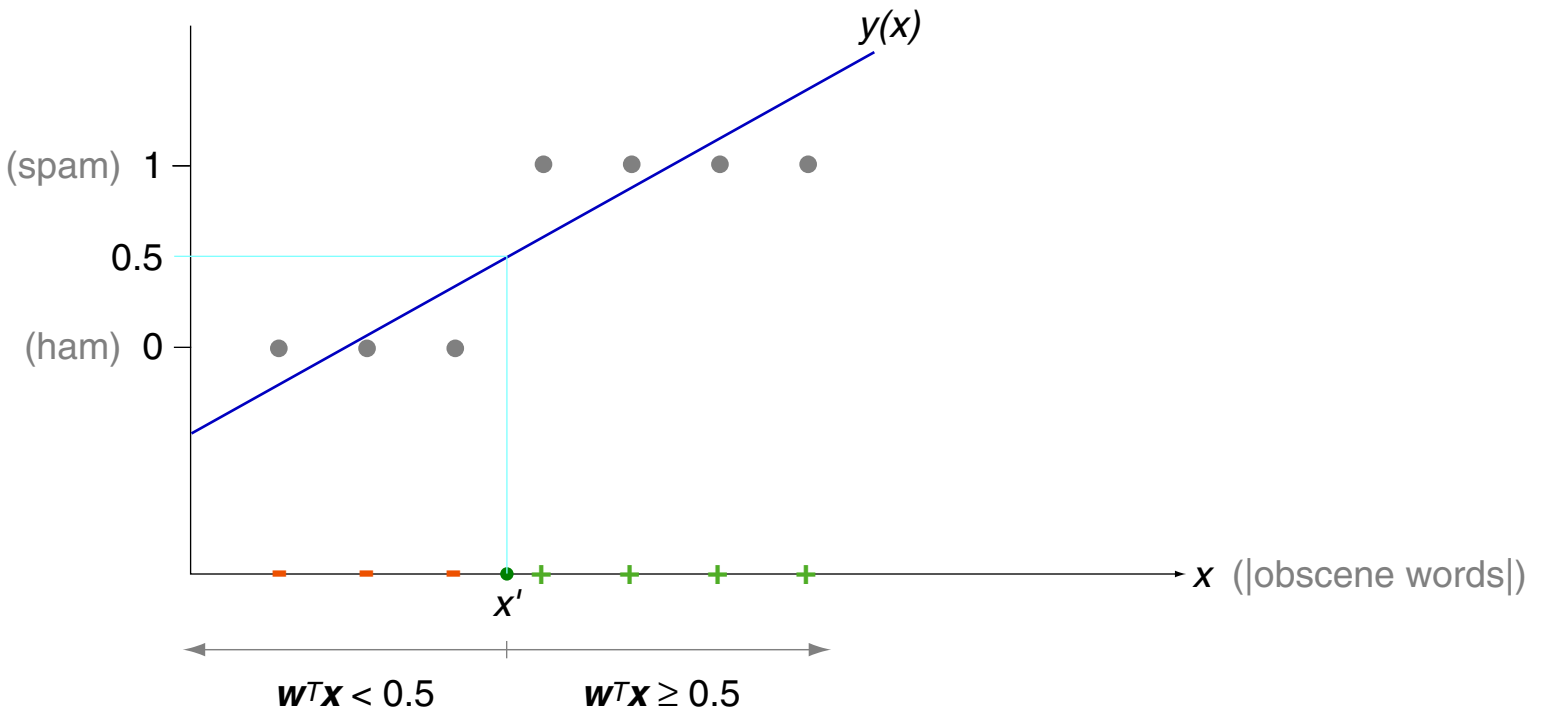
Linear Regression



- Linear regression: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

Logistic Regression

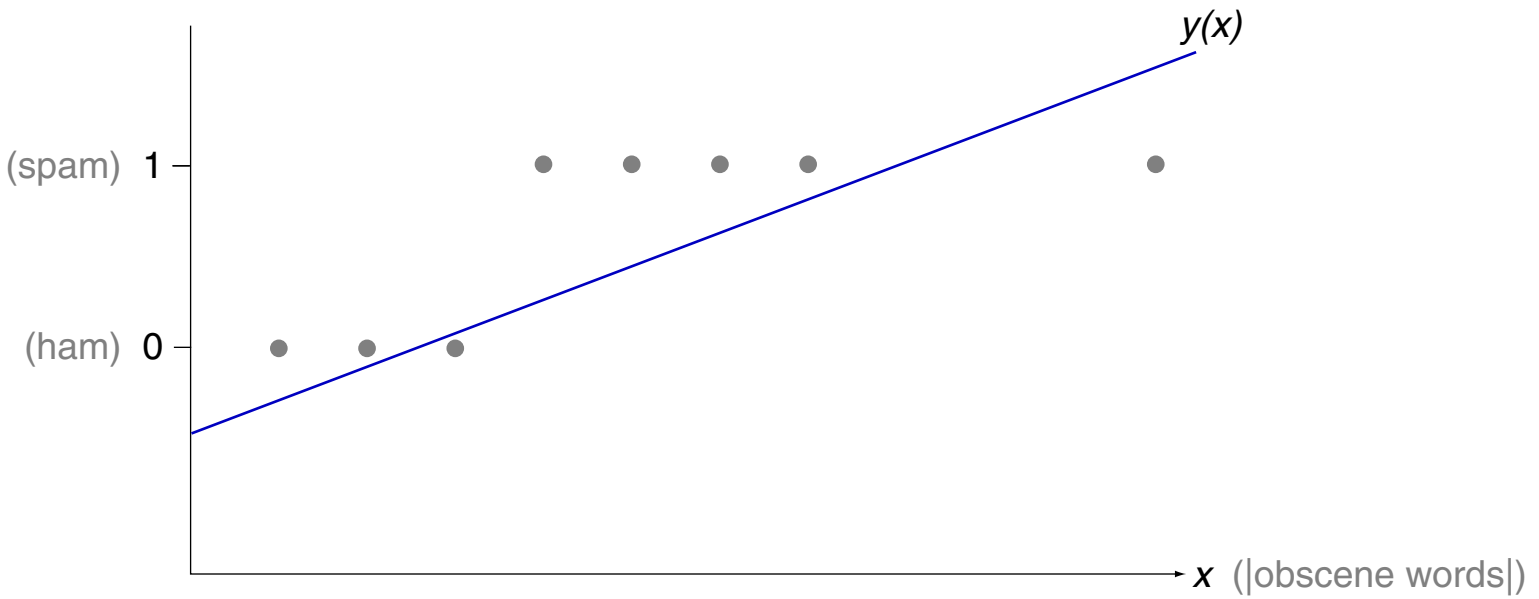
Linear Regression (continued)



- Linear regression: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
- Classification: Predict $\begin{cases} \text{“spam”}, & \text{if } \mathbf{w}^T\mathbf{x} \geq 0.5 \\ \text{“ham”}, & \text{if } \mathbf{w}^T\mathbf{x} < 0.5 \end{cases}$

Logistic Regression

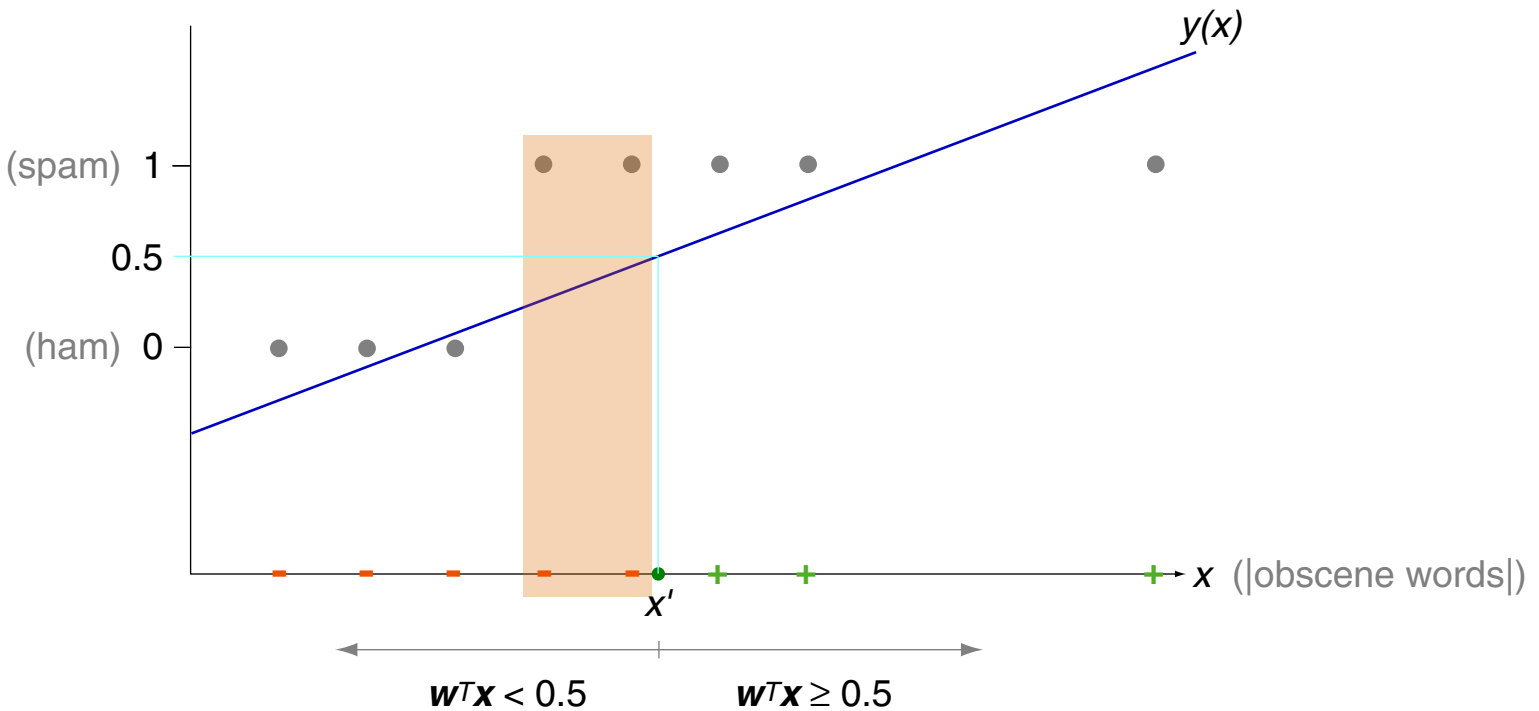
Linear Regression (continued)



- Linear regression: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
- Classification: Predict $\begin{cases} \text{“spam”}, & \text{if } \mathbf{w}^T\mathbf{x} \geq 0.5 \\ \text{“ham”}, & \text{if } \mathbf{w}^T\mathbf{x} < 0.5 \end{cases}$

Logistic Regression

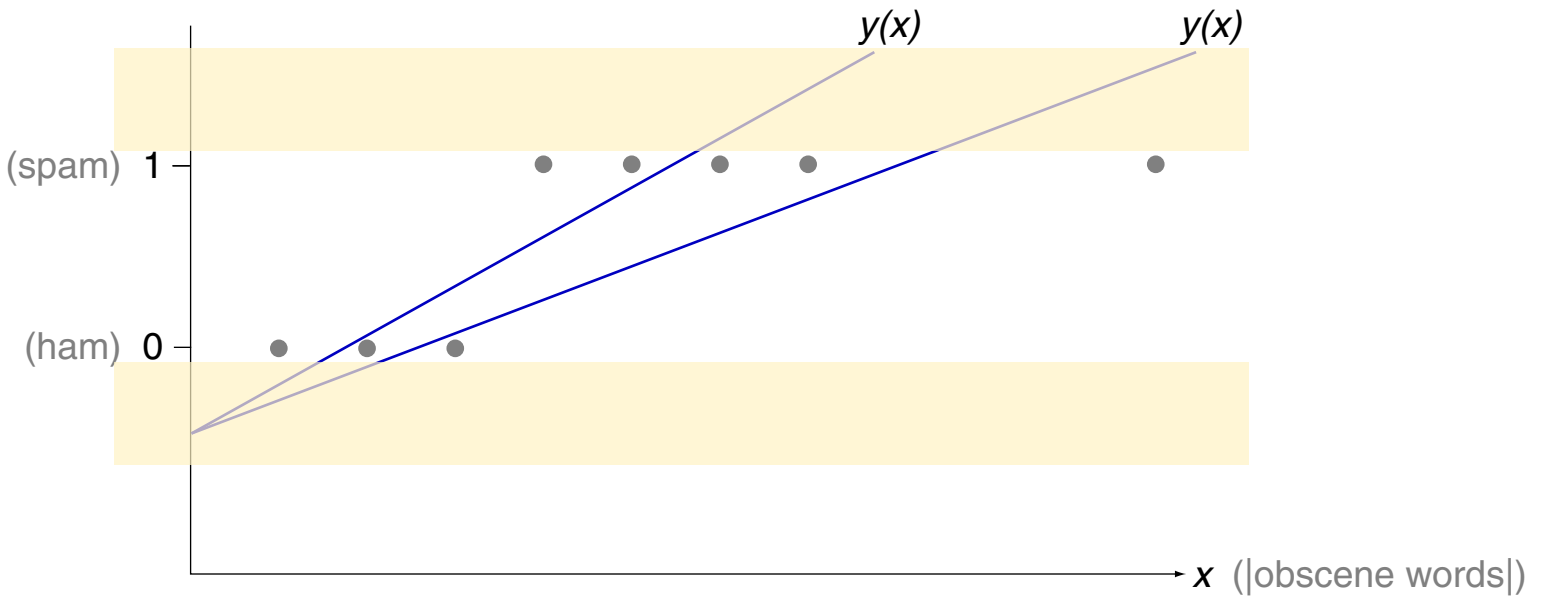
Linear Regression (continued)



- Linear regression: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
- Classification: Predict $\begin{cases} \text{“spam”}, & \text{if } \mathbf{w}^T\mathbf{x} \geq 0.5 \\ \text{“ham”}, & \text{if } \mathbf{w}^T\mathbf{x} < 0.5 \end{cases}$

Logistic Regression

Linear Regression (continued)

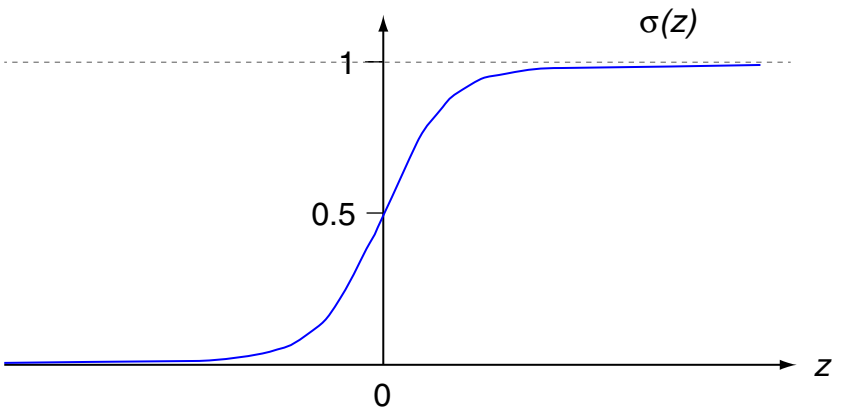


Restrict the range of $y(\mathbf{x})$ to reflect the two-class classification semantics:

$$0 \leq y(\mathbf{x}) \leq 1$$

Logistic Regression

Sigmoid (Logistic) Function

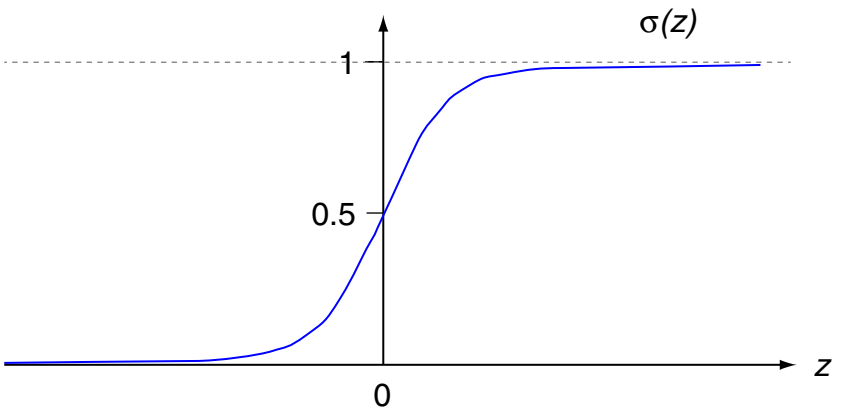


Sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Logistic Regression

Sigmoid (Logistic) Function (continued)



Linear regression

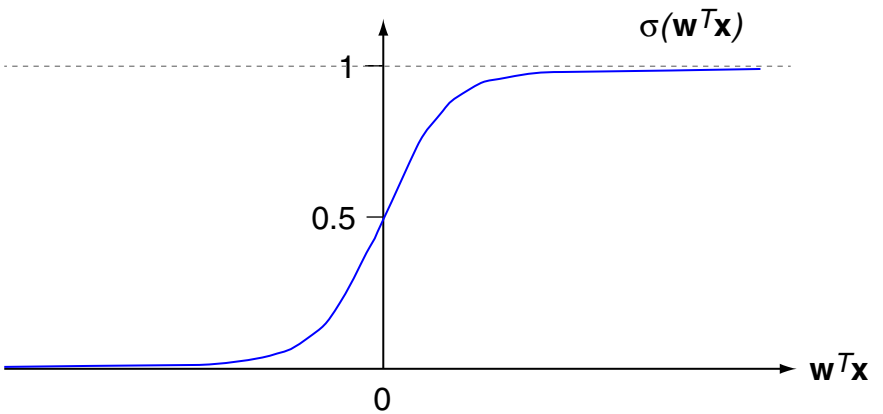
$$\mathbf{w}^T\mathbf{x}$$

Sigmoid function

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Logistic Regression

Sigmoid (Logistic) Function (continued)

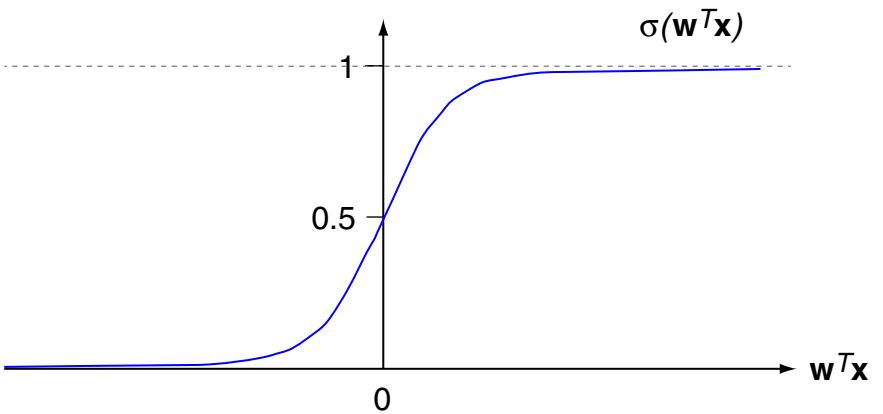


Linear regression ∘ Sigmoid function ⇝ Logistic model function

$$\mathbf{w}^T\mathbf{x} \quad \circ \quad \sigma(z) = \frac{1}{1+e^{-z}} \quad \rightsquigarrow \quad y(\mathbf{x}) \equiv \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

Logistic Regression

Sigmoid (Logistic) Function (continued)



Linear regression ∘ Sigmoid function ⇝ Logistic model function

$$\mathbf{w}^T\mathbf{x} \quad \circ \quad \sigma(z) = \frac{1}{1+e^{-z}} \quad \rightsquigarrow \quad y(\mathbf{x}) \equiv \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

$$y(\mathbf{x}) : \mathbf{R}^{p+1} \rightarrow (0;1)$$

Logistic Regression

Interpretation of the Logistic Model Function

$y(\mathbf{x})$ is the estimated probability that $c(\mathbf{x}) = 1$ given feature vector $\mathbf{x}$.

$y(\mathbf{x}) = P(c(\mathbf{x}) = 1 \mid \mathbf{x}; \mathbf{w})$, "Probability that $c(\mathbf{x}) = 1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."

Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x})$ is the estimated probability that $c(\mathbf{x}) = 1$ given feature vector $\mathbf{x}$.

$y(\mathbf{x}) = P(c(\mathbf{x}) = 1 \mid \mathbf{x}; \mathbf{w})$, "Probability that $c(\mathbf{x}) = 1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."

Example (email spam classification):

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ |\text{obscene words}| \end{pmatrix}, \qquad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad y(\mathbf{x}_1) = 0.67$$

$\rightsquigarrow$ 67% chance that email is spam.

Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x})$ is the estimated probability that $c(\mathbf{x}) = 1$ given feature vector $\mathbf{x}$.

$y(\mathbf{x}) = P(c(\mathbf{x}) = 1 \mid \mathbf{x}; \mathbf{w})$, "Probability that $c(\mathbf{x}) = 1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."

Example (email spam classification):

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ |\text{obscene words}| \end{pmatrix}, \qquad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad y(\mathbf{x}_1) = 0.67$$

$\rightsquigarrow$ 67% chance that email is spam.

Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x})$ is the estimated probability that $c(\mathbf{x}) = 1$ given feature vector $\mathbf{x}$.

$y(\mathbf{x}) = P(c(\mathbf{x}) = 1 \mid \mathbf{x}; \mathbf{w})$, "Probability that $c(\mathbf{x}) = 1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."

Example (email spam classification):

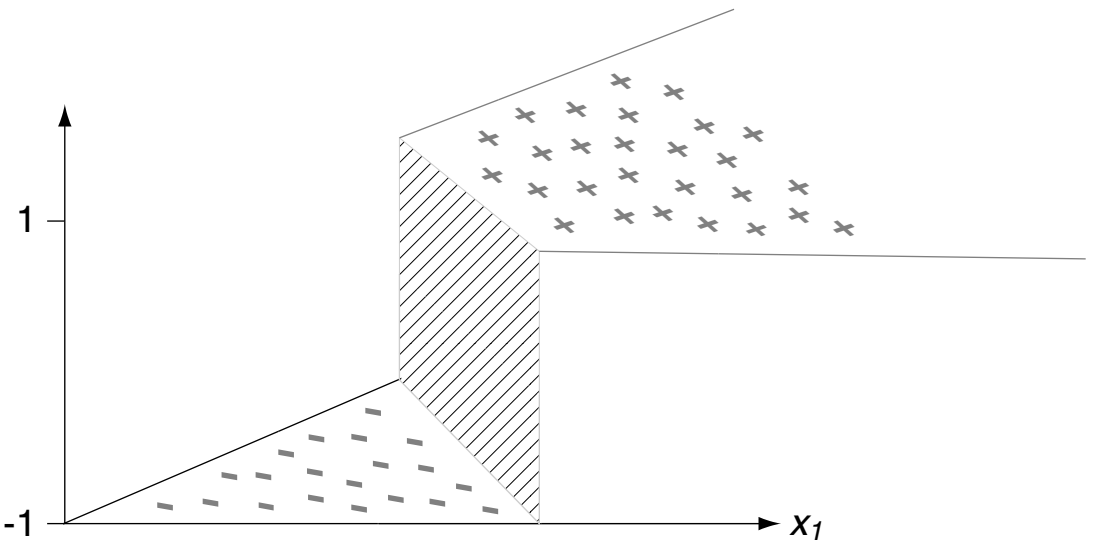
$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ |\text{obscene words}| \end{pmatrix}, \qquad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad y(\mathbf{x}_1) = 0.67$$

$\rightsquigarrow$ 67% chance that email is spam.

Classification: Predict $\begin{cases} c(\mathbf{x}) = 1, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \\ c(\mathbf{x}) = 0, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \end{cases}$

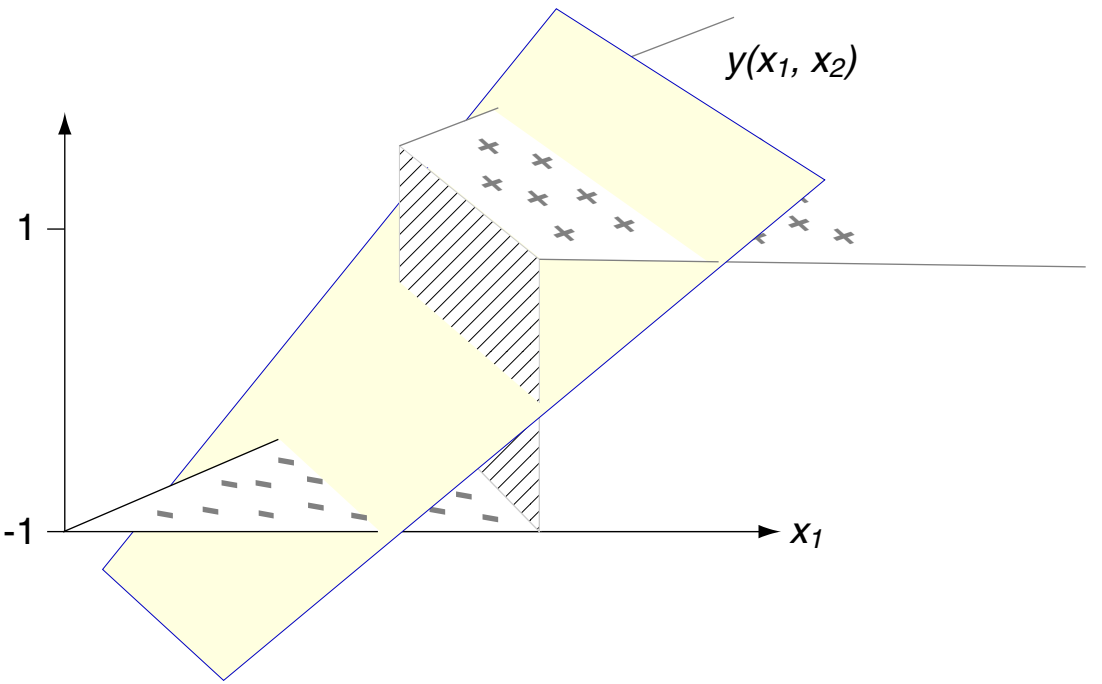
Logistic Regression

Recap: Linear Regression for Classification (illustrated for $p = 2$)



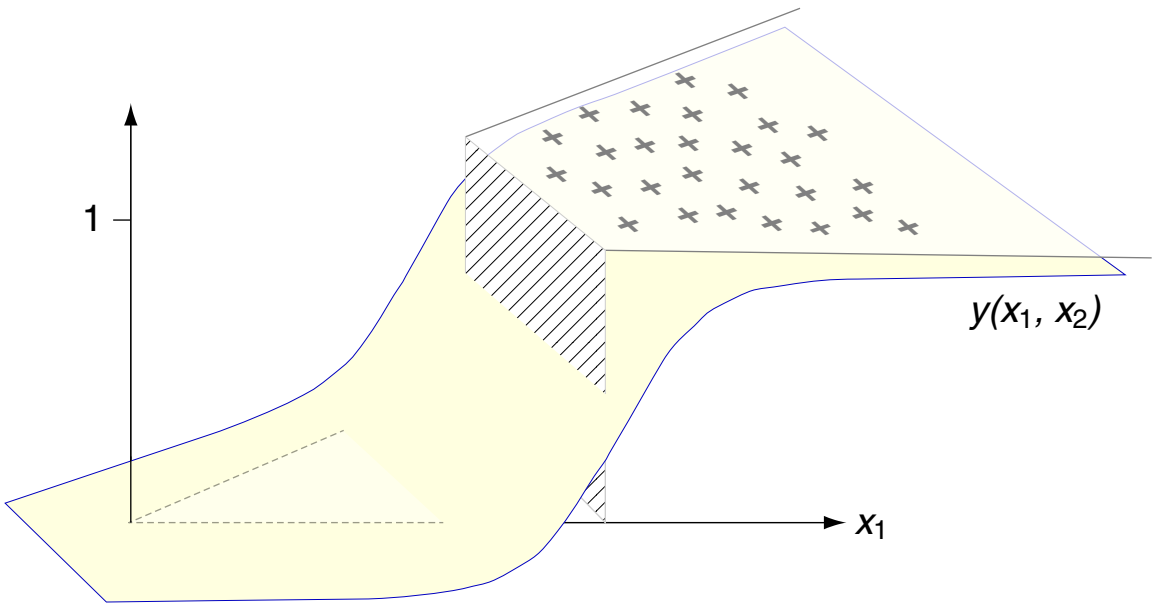
Logistic Regression

Recap: Linear Regression for Classification (illustrated for $p = 2$)



Logistic Regression

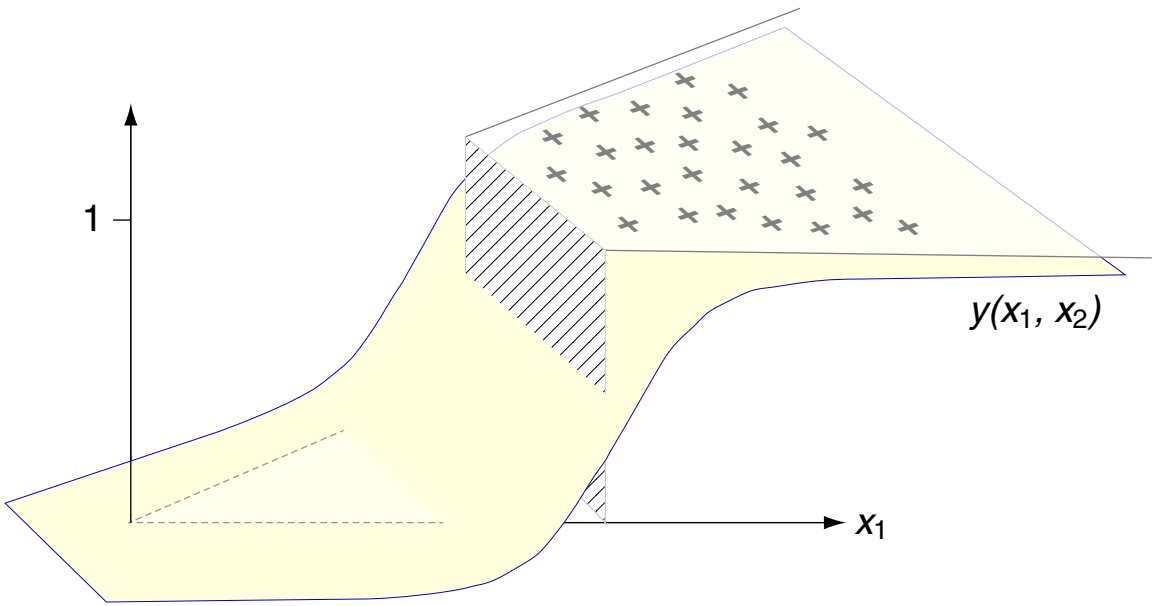
Logistic Regression for Classification (illustrated for $p = 2$)



Logistic Regression

Logistic Regression for Classification (illustrated for $p = 2$) (continued)

Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$.

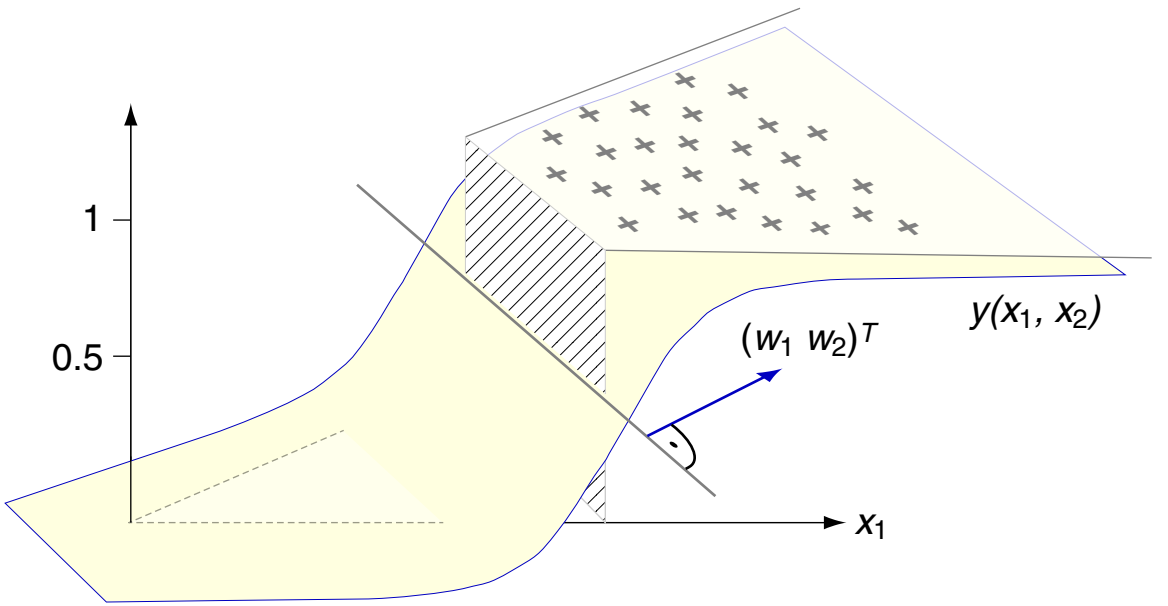


$$y(x_1, x_2) = \frac{1}{1 + e^{-(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2)}}$$

Logistic Regression

Logistic Regression for Classification (illustrated for $p = 2$) (continued)

Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.

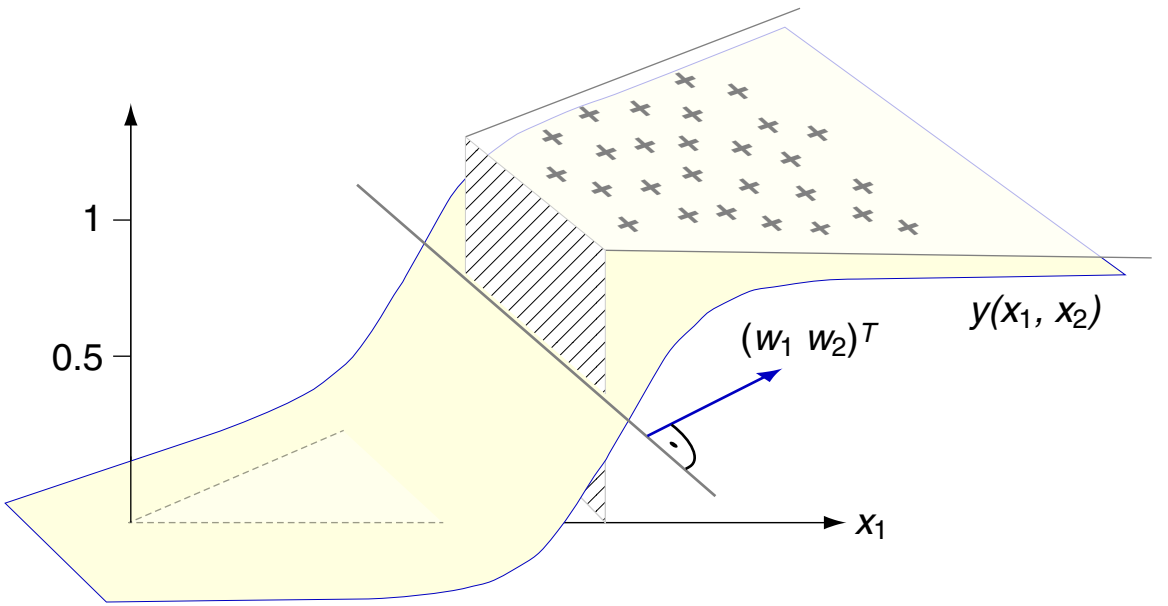


$$y(x_1, x_2) = \frac{1}{1 + e^{-(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2)}}$$

Logistic Regression

Logistic Regression for Classification (illustrated for $p = 2$) (continued)

Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.

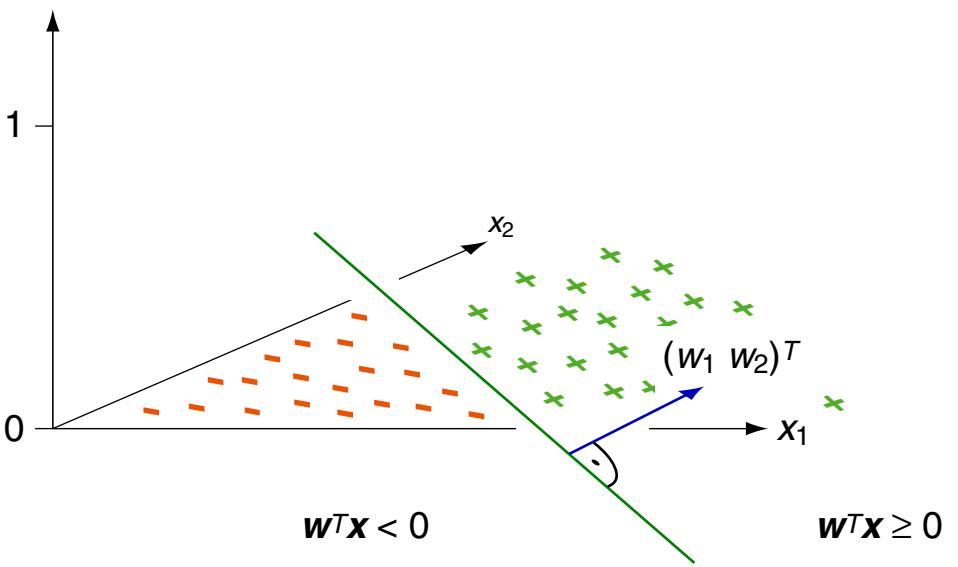


Classification: Predict $\begin{cases} c(\mathbf{x}) = 1, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \\ c(\mathbf{x}) = 0, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \end{cases}$

Logistic Regression

Logistic Regression for Classification (illustrated for $p = 2$) (continued)

Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.

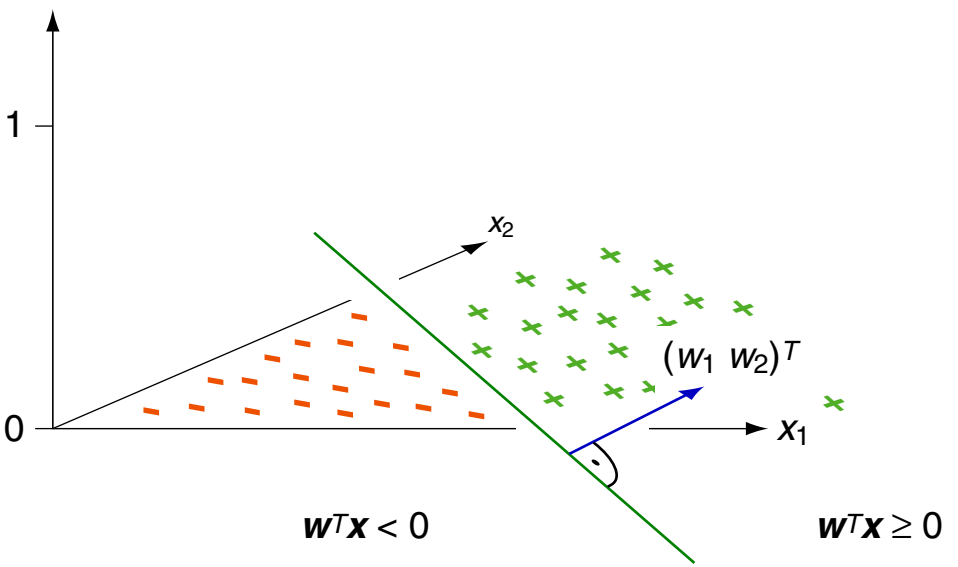


Classification: Predict $\begin{cases} c(\mathbf{x}) = 1, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \\ c(\mathbf{x}) = 0, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \end{cases}$

Logistic Regression

Logistic Regression for Classification (illustrated for $p = 2$) (continued)

Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.



Classification: Predict $\begin{cases} c(\mathbf{x}) = 1, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} \geq 0 \\ c(\mathbf{x}) = 0, \text{ if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} < 0 \end{cases}$

Remarks:

- Observe that under a logistic regression the form of a hypothesis is identical to the hypothesis in linear regression: it is a point, a straight line, a plane, or a hyperplane in the feature space. Similarly, the hypothesis spaces are the same. Hence, the expressiveness, i.e., the complexity of classification problems that can be tackled (or, the effectiveness at which classification problems can be decided) is identical for the two regression approaches.
- Linear regression and logistic regression differ in the way the respective model function parameters, $\mathbf{w}$, are computed. Put another way, the strategy by which the hypothesis space is searched is different. Since both regression methods aim at loss minimization, their different behavior results from their different loss definitions.
- In logistic regression there is no direct method (recall the normal equations in linear regression) to solve the model function $y(\mathbf{x})$ for $\mathbf{w}$.
- By now we have not specified how the parameter vector $\mathbf{w}$ is estimated under the logistic model function.

Logistic Regression

Loss Computation: Linear Regression

- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x})), c(\mathbf{x})) = \begin{cases} 0 & \text{if sign}(y(\mathbf{x})) = c(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $L_2(y(\mathbf{x}), c(\mathbf{x})) = (y(\mathbf{x}) - c(\mathbf{x}))^2$

Logistic Regression

Loss Computation: Linear Regression (continued)

- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x})), c(\mathbf{x})) = \begin{cases} 0 & \text{if sign}(y(\mathbf{x})) = c(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $L_2(y(\mathbf{x}), c(\mathbf{x})) = (y(\mathbf{x}) - c(\mathbf{x}))^2$

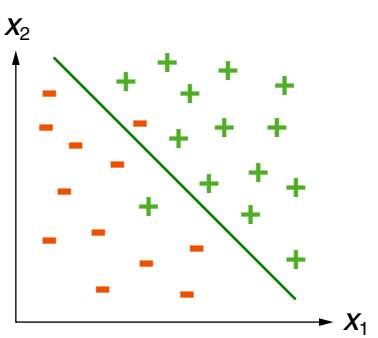
Logistic Regression

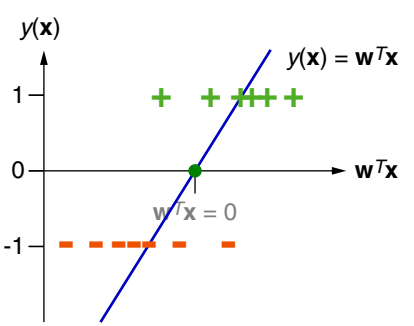
Loss Computation: Linear Regression (continued)

- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x})), c(\mathbf{x})) = \begin{cases} 0 & \text{if sign}(y(\mathbf{x})) = c(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $L_2(y(\mathbf{x}), c(\mathbf{x})) = (y(\mathbf{x}) - c(\mathbf{x}))^2$

Illustration:

Input space: $y(\mathbf{x})$ over hyperplane distance:





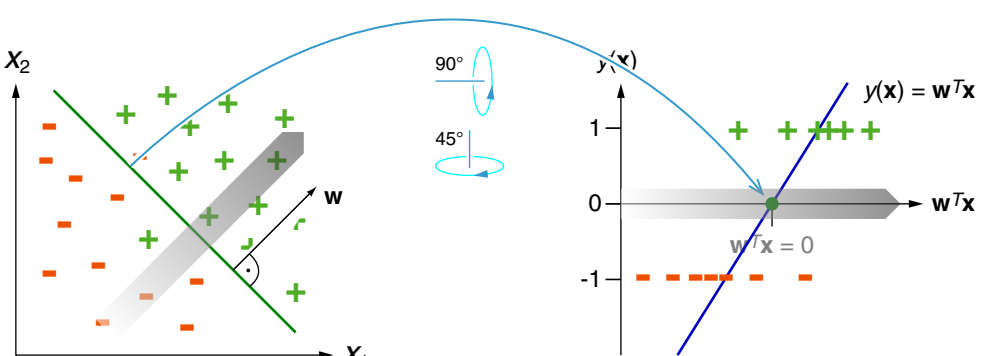
Logistic Regression

Loss Computation: Linear Regression (continued)

- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x})), c(\mathbf{x})) = \begin{cases} 0 & \text{if sign}(y(\mathbf{x})) = c(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $L_2(y(\mathbf{x}), c(\mathbf{x})) = (y(\mathbf{x}) - c(\mathbf{x}))^2$

Illustration:

Input space: $y(\mathbf{x})$ over hyperplane distance:



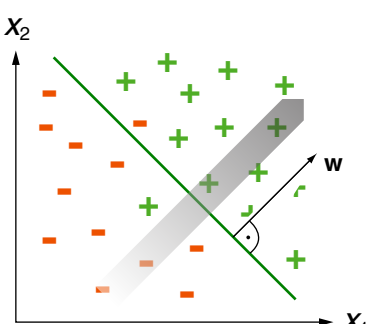
Logistic Regression

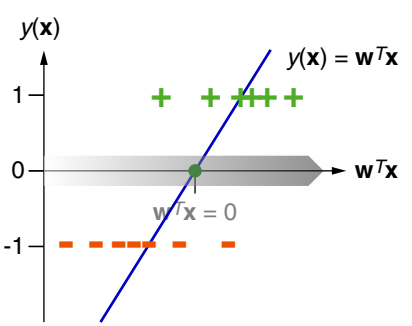
Loss Computation: Linear Regression (continued)

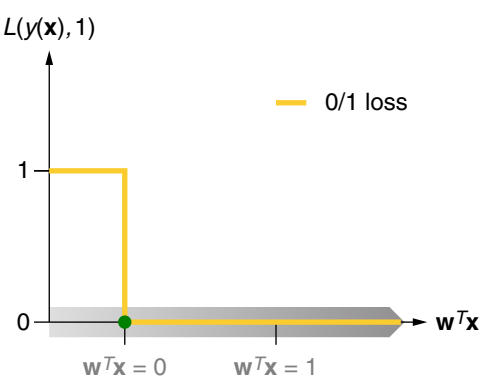
- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x})), c(\mathbf{x})) = \begin{cases} 0 & \text{if sign}(y(\mathbf{x})) = c(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $L_2(y(\mathbf{x}), c(\mathbf{x})) = (y(\mathbf{x}) - c(\mathbf{x}))^2$

Illustration:

Input space: $y(\mathbf{x})$ over hyperplane distance: Loss over hyperplane distance:







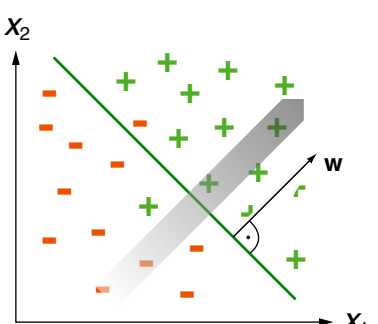
Logistic Regression

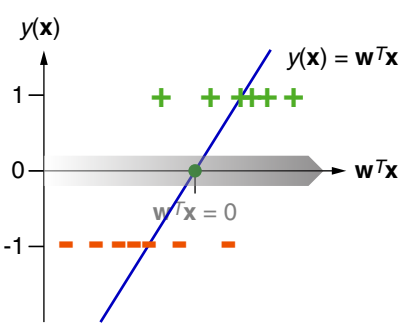
Loss Computation: Linear Regression (continued)

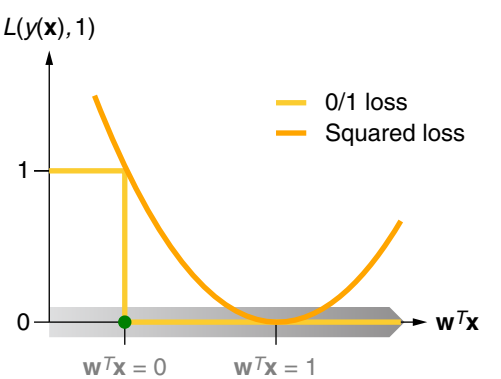
- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x})), c(\mathbf{x})) = \begin{cases} 0 & \text{if sign}(y(\mathbf{x})) = c(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $L_2(y(\mathbf{x}), c(\mathbf{x})) = (y(\mathbf{x}) - c(\mathbf{x}))^2$

Illustration:

Input space: $y(\mathbf{x})$ over hyperplane distance: Loss over hyperplane distance:







Remarks:

- We distinguish between the pointwise loss L, which is computed for a single $\mathbf{x}$, and the global loss $\mathcal{L}$, which accumulates the pointwise loss of all $\mathbf{x} \in X$.
- Various loss functions for classification and regression problems have been devised.
- The 0/1 loss computes the misclassification error. Recall in this regard the definition of the true misclassification rate.
- The squared-loss computes the squared residual. Recall in this regard the residual sum of squares.
- As before, I denotes the indicator function.

Logistic Regression

Loss Computation: Logistic Regression

- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x}) - 0.5), c(\mathbf{x}))$
  - Logistic loss. $L_\sigma(y(\mathbf{x}), c(\mathbf{x})) = \begin{cases} -\log(y(\mathbf{x})) & \text{if } c(\mathbf{x}) = 1 \\ -\log(1 - y(\mathbf{x})) & \text{otherwise} \end{cases}$

Logistic Regression

Loss Computation: Logistic Regression (continued)

- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x}) - 0.5), c(\mathbf{x}))$
  - Logistic loss. $L_\sigma(y(\mathbf{x}), c(\mathbf{x})) = \begin{cases} -\log(y(\mathbf{x})) & \text{if } c(\mathbf{x}) = 1 \\ -\log(1 - y(\mathbf{x})) & \text{otherwise} \end{cases}$

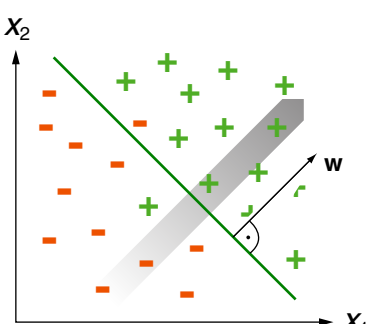
Logistic Regression

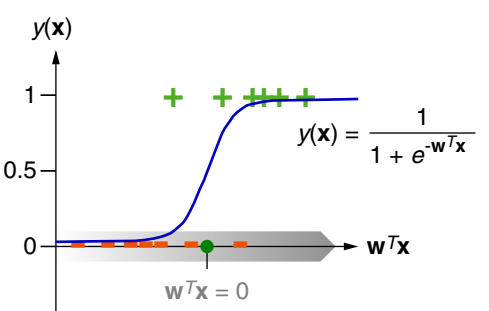
Loss Computation: Logistic Regression (continued)

- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x}) - 0.5), c(\mathbf{x}))$
  - Logistic loss. $L_\sigma(y(\mathbf{x}), c(\mathbf{x})) = \begin{cases} -\log(y(\mathbf{x})) & \text{if } c(\mathbf{x}) = 1 \\ -\log(1 - y(\mathbf{x})) & \text{otherwise} \end{cases}$

Illustration:

Input space: $y(\mathbf{x})$ over hyperplane distance:





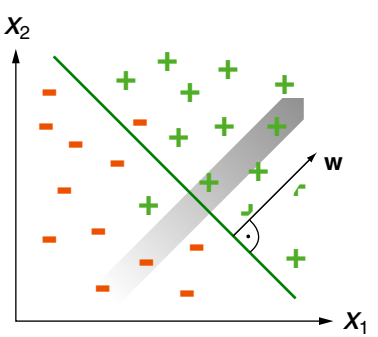
Logistic Regression

Loss Computation: Logistic Regression (continued)

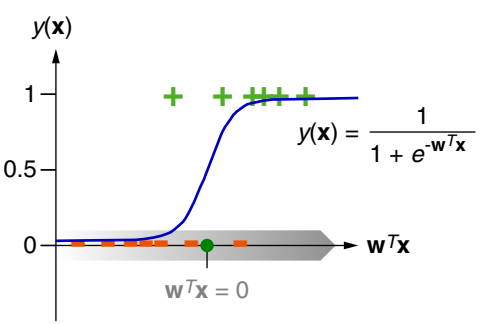
- The pointwise loss, $L(y(\mathbf{x}), c(\mathbf{x}))$, quantifies the penalty charge introduced by some $\mathbf{x}$. The loss depends on a hypothesis y and the true class, $c(\mathbf{x})$, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:
  - 0/1 loss. $L_{0/1}(y(\mathbf{x}), c(\mathbf{x})) = I(\text{sign}(y(\mathbf{x}) - 0.5), c(\mathbf{x}))$
  - Logistic loss. $L_\sigma(y(\mathbf{x}), c(\mathbf{x})) = \begin{cases} -\log(y(\mathbf{x})) & \text{if } c(\mathbf{x}) = 1 \\ -\log(1 - y(\mathbf{x})) & \text{otherwise} \end{cases}$

Illustration:

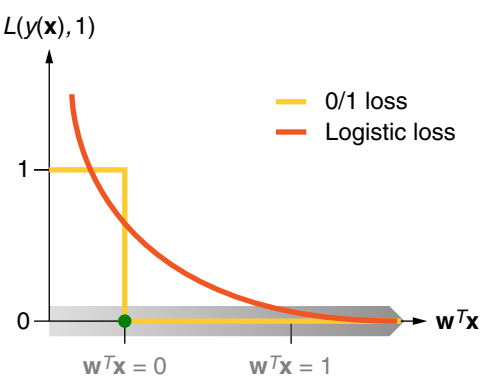
Input space:



$y(\mathbf{x})$ over hyperplane distance:



Loss over hyperplane distance:



Remarks:

- The logistic loss can be rewritten: $L_\sigma = -c(\mathbf{x}) \cdot \log(y(\mathbf{x})) - (1 - c(\mathbf{x})) \cdot \log(1 - y(\mathbf{x}))$
- Observe that loss functions are employed at two places:
  1. For the fitting (the actual regression / optimization / hyperplane search) of the data, where the optimum position of the hyperplane is determined (= a particular hypothesis is selected), and,
  2. for the evaluation of a hypothesis' effectiveness, where the portion of correctly and misclassified examples is analyzed.
- Typically, fitting (optimization) and effectiveness evaluation are done with different loss functions. E.g., logistic regression uses L_σ and $L_{0/1}$ for fitting and evaluation respectively. However, linear regression (not classification) uses RSS (the L_2 loss) for both fitting and evaluation. The basic perceptron learning algorithm uses the misclassification information (the $L_{0/1}$ loss) for both fitting and evaluation.

Chapter ML:III (continued)

III. Linear Models

- Logistic Regression
- Loss Computation
- Overfitting
- Regularization

Overfitting

Definition 9 (Overfitting)

Let D be a set of examples and let H be a hypothesis space. The hypothesis $h \in H$ is considered to overfit D if an $h' \in H$ with the following property exists:

$$\mathit{Err}(h, D) < \mathit{Err}(h', D) \quad \text{and} \quad \mathit{Err}^*(h) > \mathit{Err}^*(h'),$$

where $\mathit{Err}^*(h)$ denotes the true misclassification rate of h, while $\mathit{Err}(h, D)$ denotes the error of h on the example set D.

Overfitting

Definition 9 (Overfitting)

Let D be a set of examples and let H be a hypothesis space. The hypothesis $h \in H$ is considered to overfit D if an $h' \in H$ with the following property exists:

$$\mathit{Err}(h, D) < \mathit{Err}(h', D) \quad \text{and} \quad \mathit{Err}^*(h) > \mathit{Err}^*(h'),$$

where $\mathit{Err}^*(h)$ denotes the true misclassification rate of h, while $\mathit{Err}(h, D)$ denotes the error of h on the example set D.

Reasons for overfitting are often rooted in the example set D :

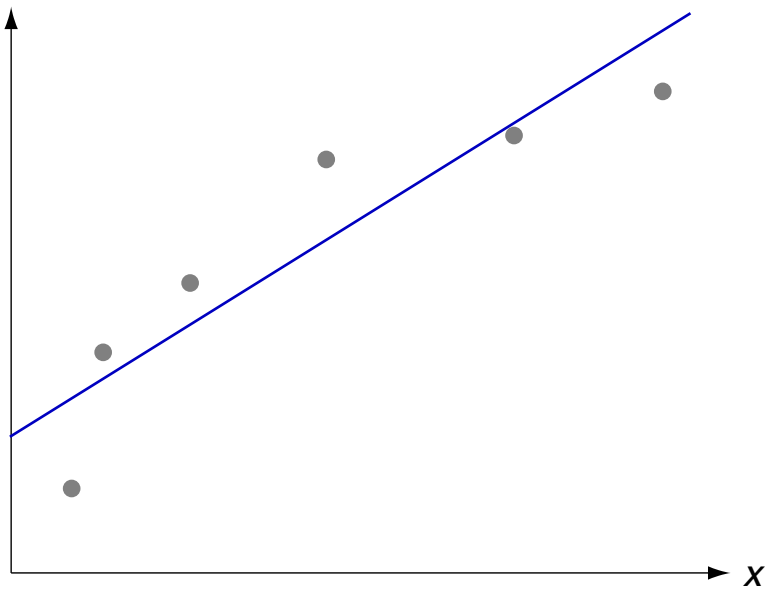
- D is noisy and we “learn noise”
- D is biased and hence not representative
- D is too small and hence pretends unrealistic data properties

Overfitting

Example: Linear Regression

Overfitting

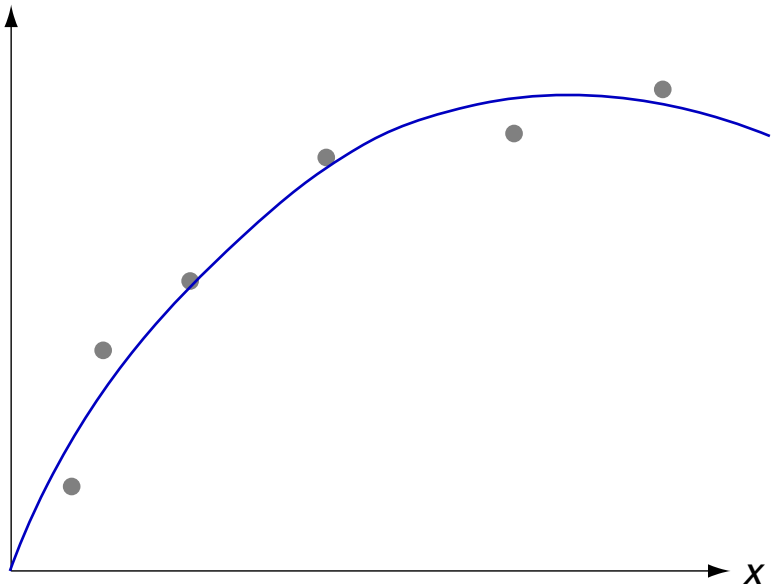
Example: Linear Regression



$$y(x) = w_0 + w_1 \cdot x$$

Overfitting

Example: Linear Regression

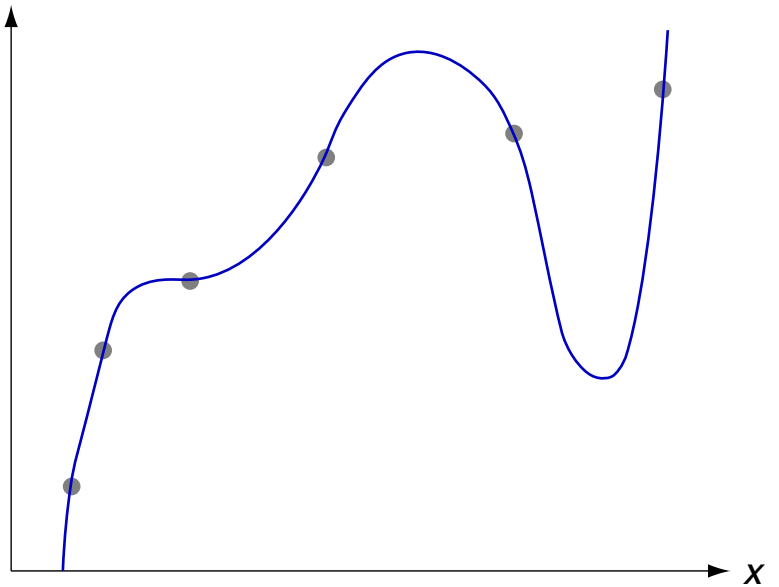


$$y(x) = w_0 + w_1 \cdot x + w_2 \cdot x^2$$

(basis expansion)

Overfitting

Example: Linear Regression



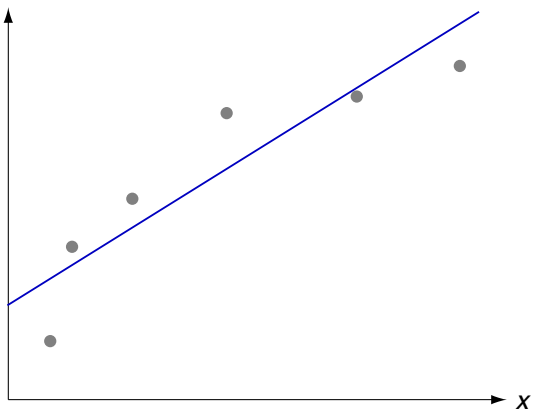
$$y(x) = w_0 + \sum_{i=1}^{6} w_i \cdot x^i$$

(basis expansion)

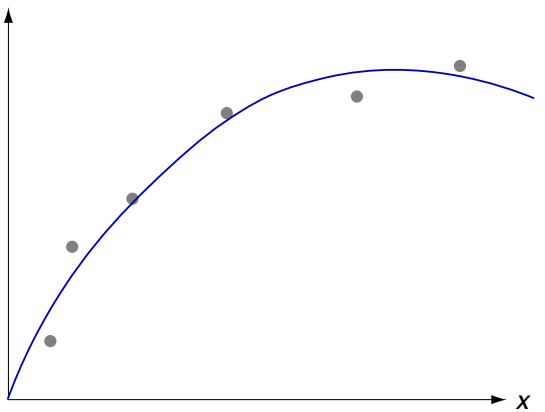
Overfitting

Example: Linear Regression

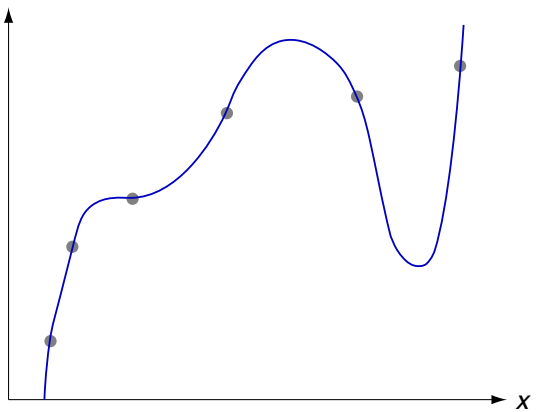
Given three polynomial model functions $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ of degrees 1, 2, and 6, and a training set of D_{tr}, select the function that best fits the data:



$\text{RSS}(\mathbf{w}) \gg 0$



$\text{RSS}(\mathbf{w}) > 0$

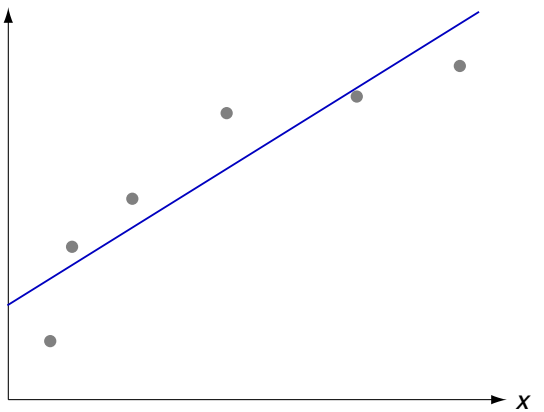


$\text{RSS}(\mathbf{w}) = 0$

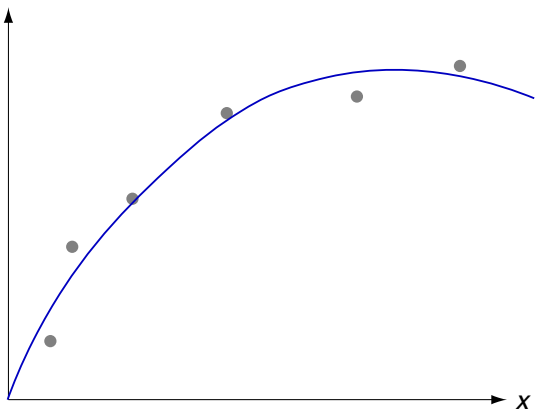
Overfitting

Example: Linear Regression

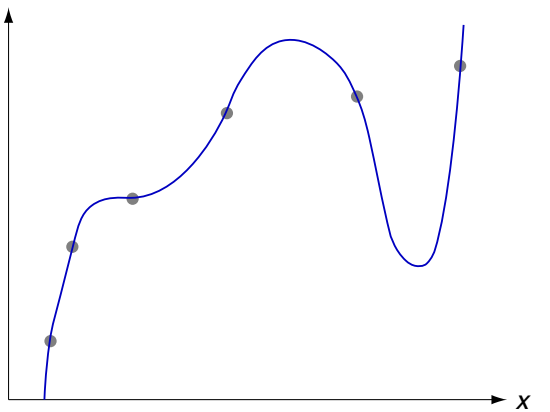
Given three polynomial model functions $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ of degrees 1, 2, and 6, and a training set of D_{tr}, select the function that best fits the data:



$\text{RSS}(\mathbf{w}) \gg 0$



$\text{RSS}(\mathbf{w}) > 0$



$\text{RSS}(\mathbf{w}) = 0$

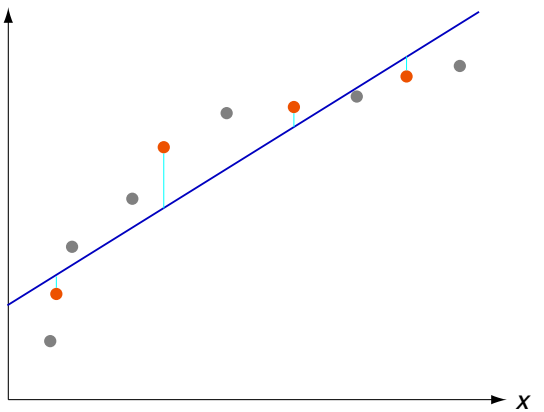
Questions:

- Which model function (hypothesis) is the best choice?
- How to choose among the many hypothesis spaces available?

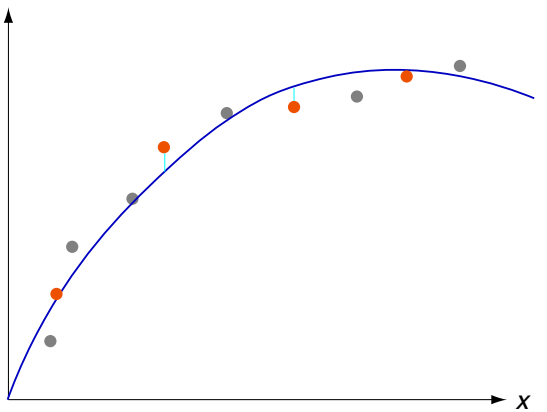
Overfitting

Example: Linear Regression

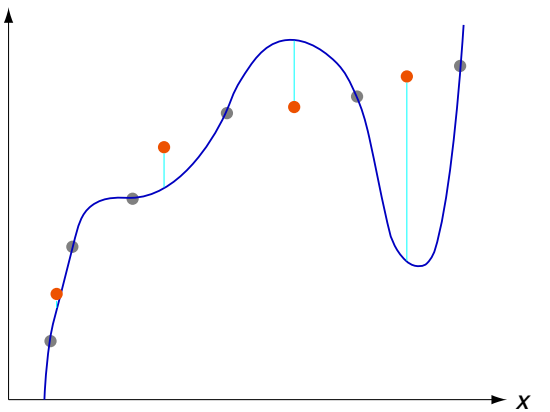
Given three polynomial model functions $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ of degrees 1, 2, and 6, and a training set of D_{tr}, select the function that best fits the data:



$\text{RSS}(\mathbf{w}) \gg 0$



$\text{RSS}(\mathbf{w}) > 0$



$\text{RSS}(\mathbf{w}) \gg 0$

Let D_{ts} be a set of test examples.

If $D = D_{tr} \cup D_{ts}$ is representative of the real-world population in X, the quadratic model function $y(x) = w_0 + w_1 \cdot x + w_2 \cdot x^2$ is the closest match.

Overfitting

Estimation

Let $D_{tr} \subset D$ be the training set. Then $\mathit{Err}^*(h)$ can be estimated with a test set $D_{ts} \subset D$ where $D_{ts} \cap D_{tr} = \emptyset$ [holdout estimation]. The hypothesis $h \in H$ is considered to overfit D if an $h' \in H$ with the following property exists:

$$\mathit{Err}(h, D_{tr}) < \mathit{Err}(h', D_{tr}) \quad \text{and} \quad \mathit{Err}(h, D_{ts}) > \mathit{Err}(h', D_{ts})$$

Overfitting

Mitigation Strategies

How to detect overfitting:

- Visual inspection
  Apply projection or embedding for dimensionalities $p > 3$.
- Validation
  Given a validation set, the difference $Err_{val}(y) - Err_{tr}(y)$ is too large.

Overfitting

Mitigation Strategies

How to detect overfitting:

- Visual inspection
  Apply projection or embedding for dimensionalities $p > 3$.
- Validation
  Given a validation set, the difference $\mathit{Err}_{\mathit{val}}(y) - \mathit{Err}_{\mathit{tr}}(y)$ is too large.

How to prevent overfitting:

- Early stopping through model selection
  Let m be the number of training steps, and $y_{\pi_1}, \ldots, y_{\pi_m}$ the models obtained after each step.
  Indicator of overfitting: The difference $\mathit{Err}_{\mathit{val}}(y_{\pi_i}) - \mathit{Err}_{\mathit{tr}}(y_{\pi_i})$ increases with i.
- Increasing quantity or quality of the training data D
  Quantity: More data averages out noise.
  Quality: Omitting poor examples enables a better fit.
- Manually enforcing a higher bias by using a less complex hypothesis space.
- Regularization
  Automatic adjustment of the loss function to penalize model complexity.

Regularization

Motivation

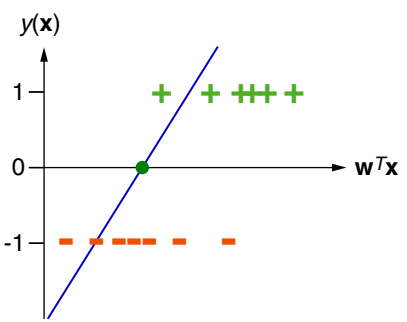
Given the choice, can a learning algorithm favor a less-than-optimal hypothesis?

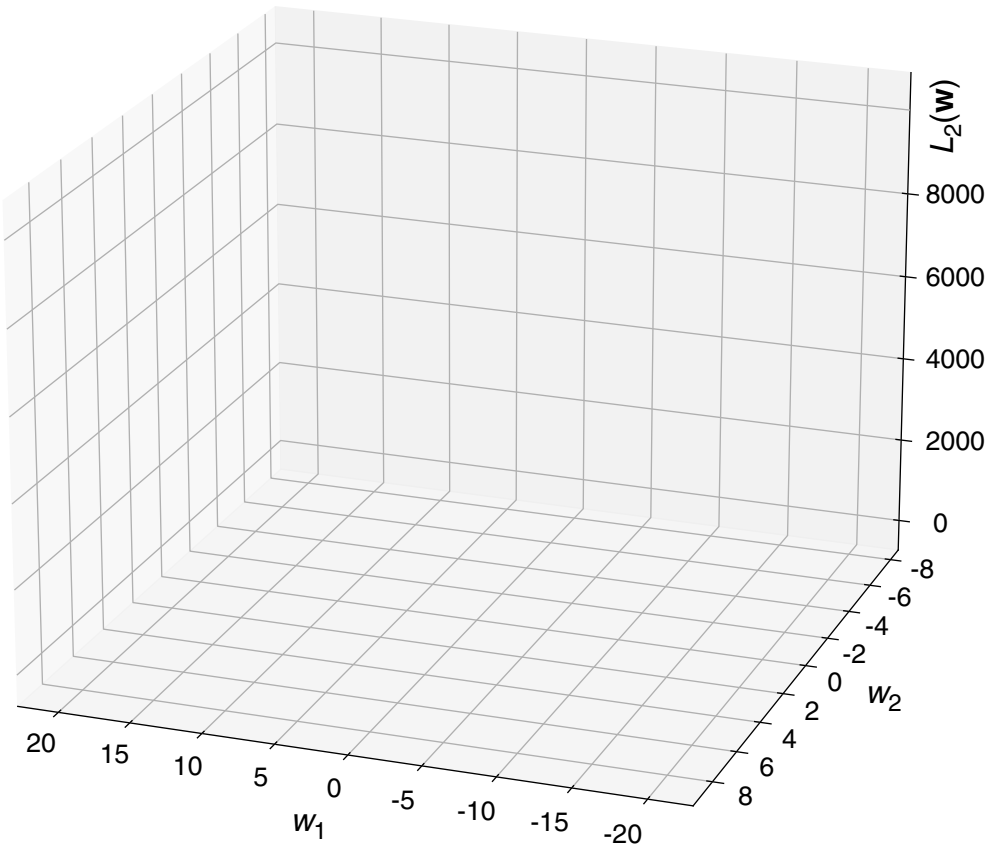
Regularization

Motivation

Given the choice, can a learning algorithm favor a less-than-optimal hypothesis?

Example: Linear Regression



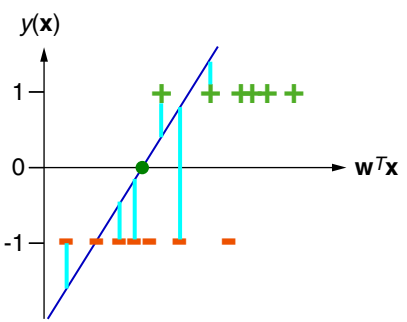


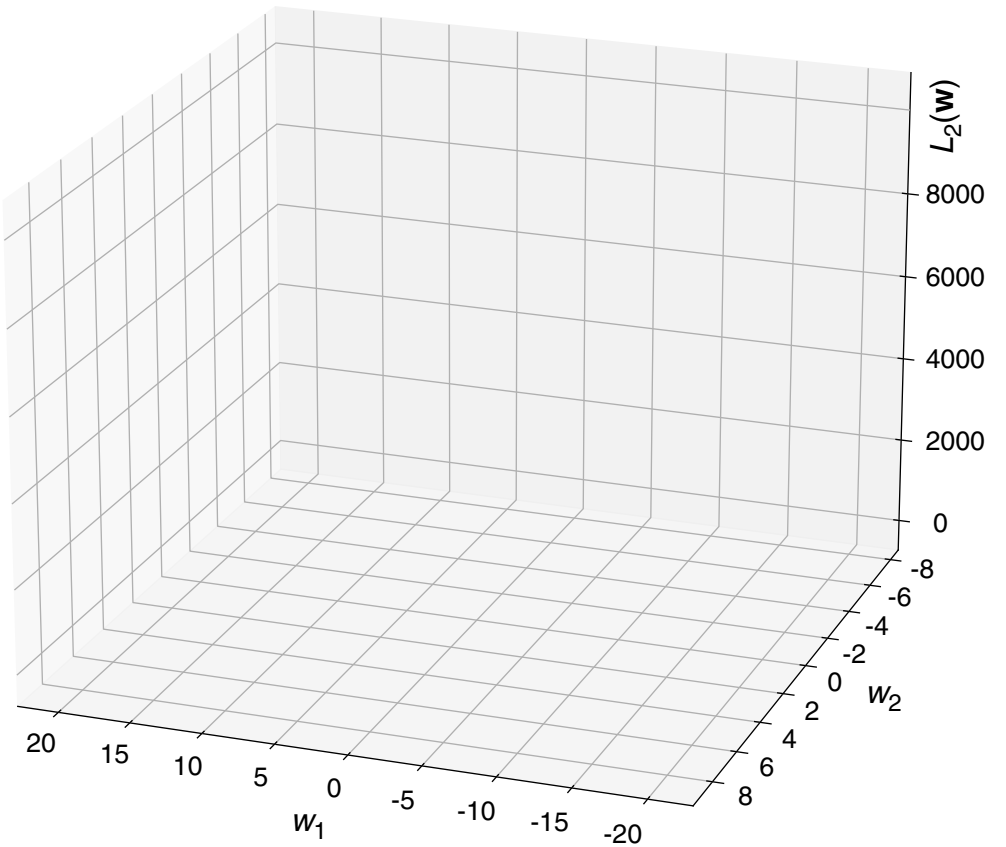
Regularization

Motivation

Given the choice, can a learning algorithm favor a less-than-optimal hypothesis?

Example: Linear Regression



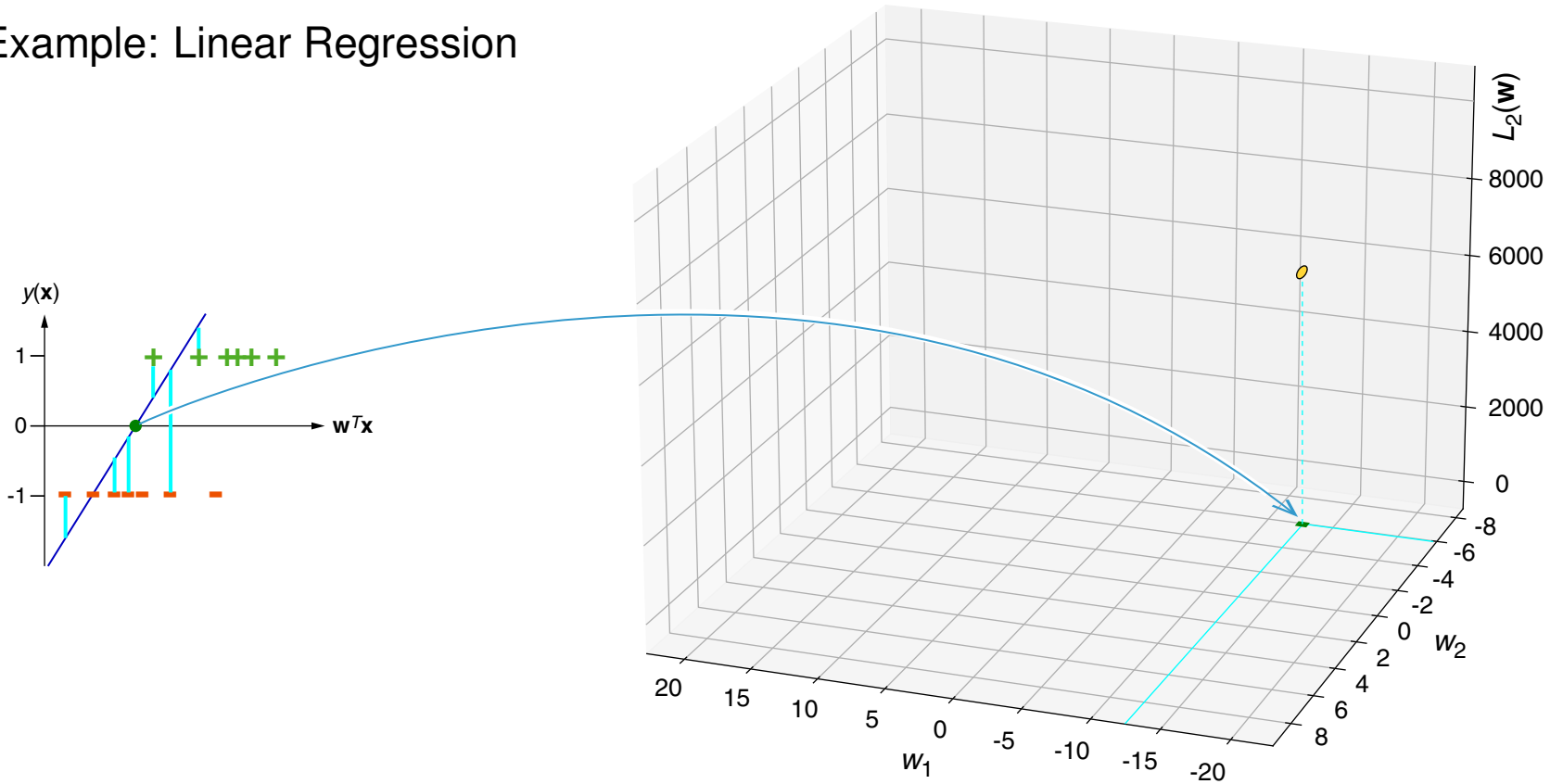


Regularization

Motivation

Given the choice, can a learning algorithm favor a less-than-optimal hypothesis?

Example: Linear Regression

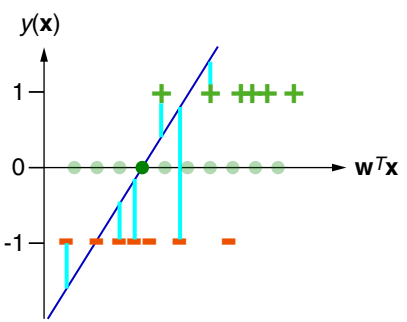


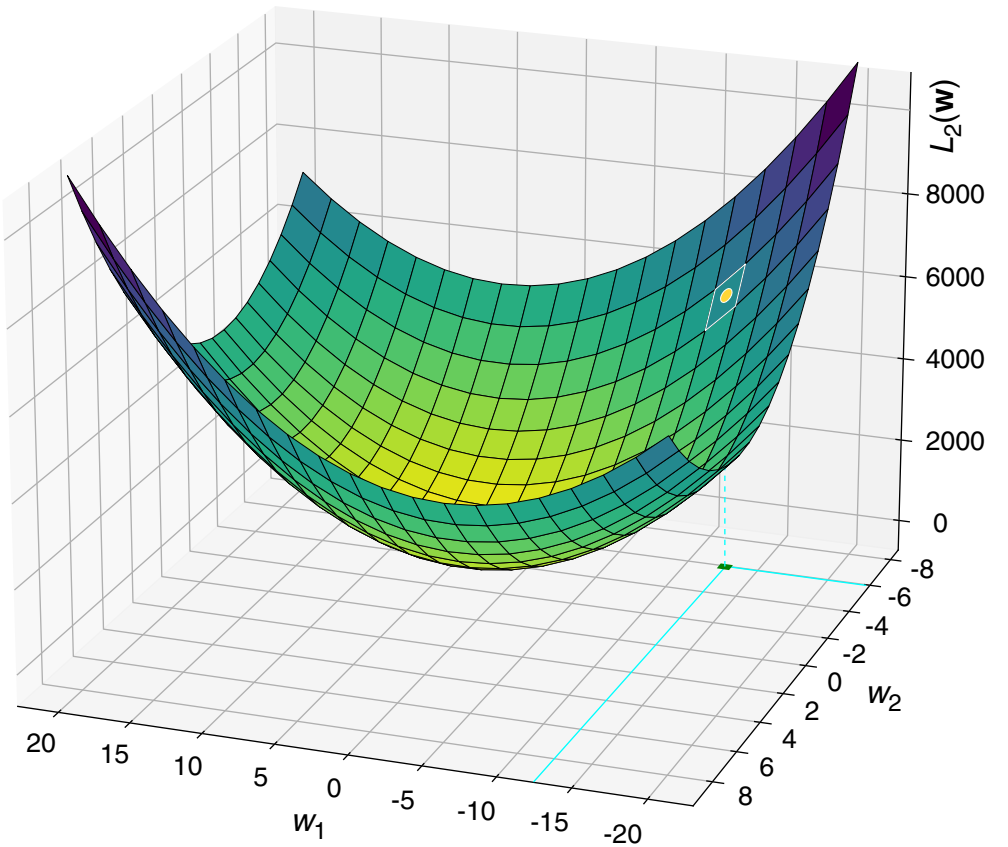
Regularization

Motivation

Given the choice, can a learning algorithm favor a less-than-optimal hypothesis?

Example: Linear Regression



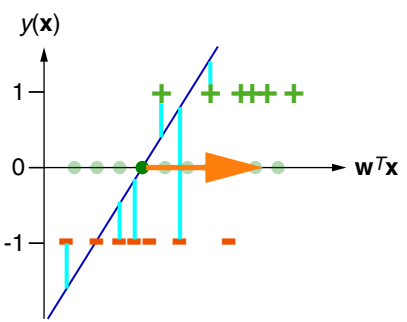


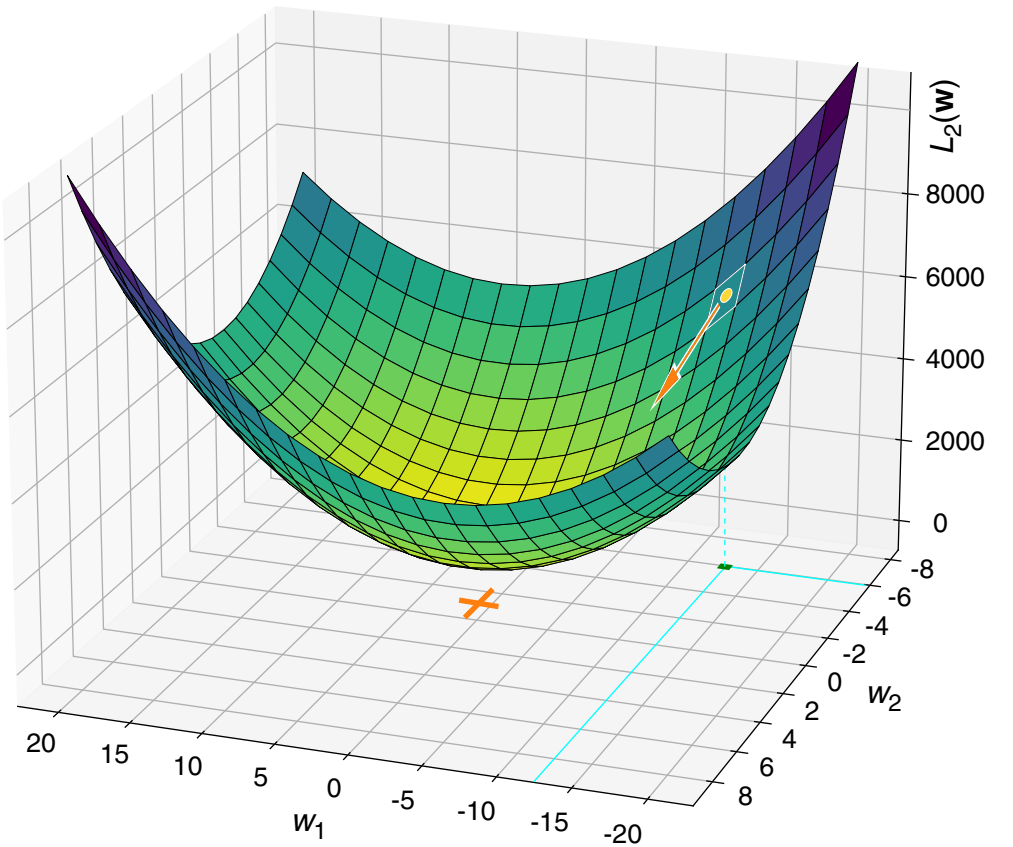
Regularization

Motivation

Given the choice, can a learning algorithm favor a less-than-optimal hypothesis?

Example: Linear Regression





Answer: No.

Regularization

Motivation

Change the optimal hypothesis $h^* \in H$ by augmenting the loss function L.

Regularization

Motivation

Change the optimal hypothesis $h^* \in H$ by augmenting the loss function L.

Criteria regarding the desired h^*:

- Generalizability to unseen data.
- Simplicity (Occam's razor)
- Stability wrt. to changes in D.
- Smoothness of the model function.

Desired mathematical properties:

- Differentiability
- Convexity
- Well-posedness of the inverse problem.

Regularization

Motivation

Change the optimal hypothesis $h^* \in H$ by augmenting the loss function L.

Criteria regarding the desired h^*:

- Generalizability to unseen data.
- Simplicity (Occam's razor)
- Stability wrt. to changes in D.
- Smoothness of the model function.

Desired mathematical properties:

- Differentiability
- Convexity
- Well-posedness of the inverse problem.

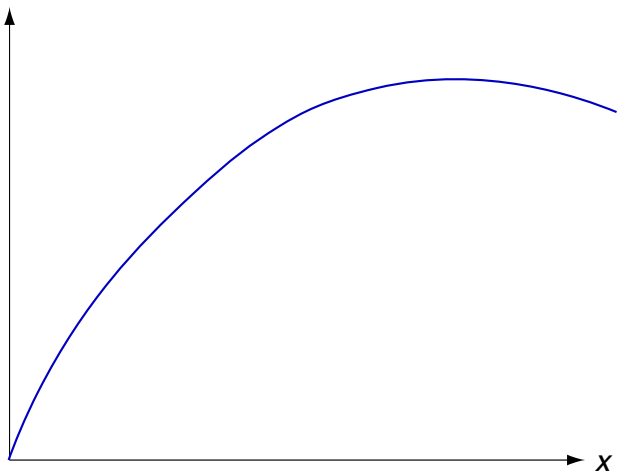
Regularization is the theory of augmenting loss functions.

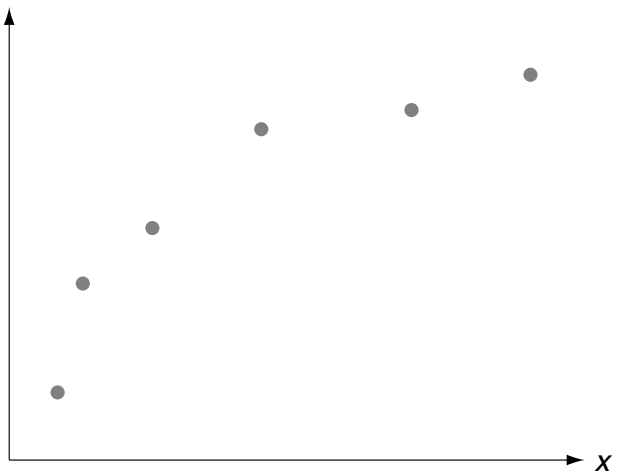
Remarks:

- The term “regularization” derives from “regular”, a synonym for describing “smooth” model functions. [stackexchange]
- The origins of regularization are found in inverse problem theory and solving ill-posed problems.
- Regularization is only useful in settings where the set of examples D is much smaller than the population of real-world objects O. From the examples, machine learning infers a hypothesis h that is supposed to generalize to the entire population. Based on the assumption, that D is a representative sample of the population, by choosing a simple, stable, and smooth hypothesis h, the risk of making errors on unseen objects is minimized.
- In situations where D covers (nearly) the entire population O, minimizing the loss L takes precedence over simplicity, stability, and smoothness of the hypothesis h. One cannot expect the (natural) laws governing O to be simple and smooth.

Regularization

Machine Learning as an Inverse Problem



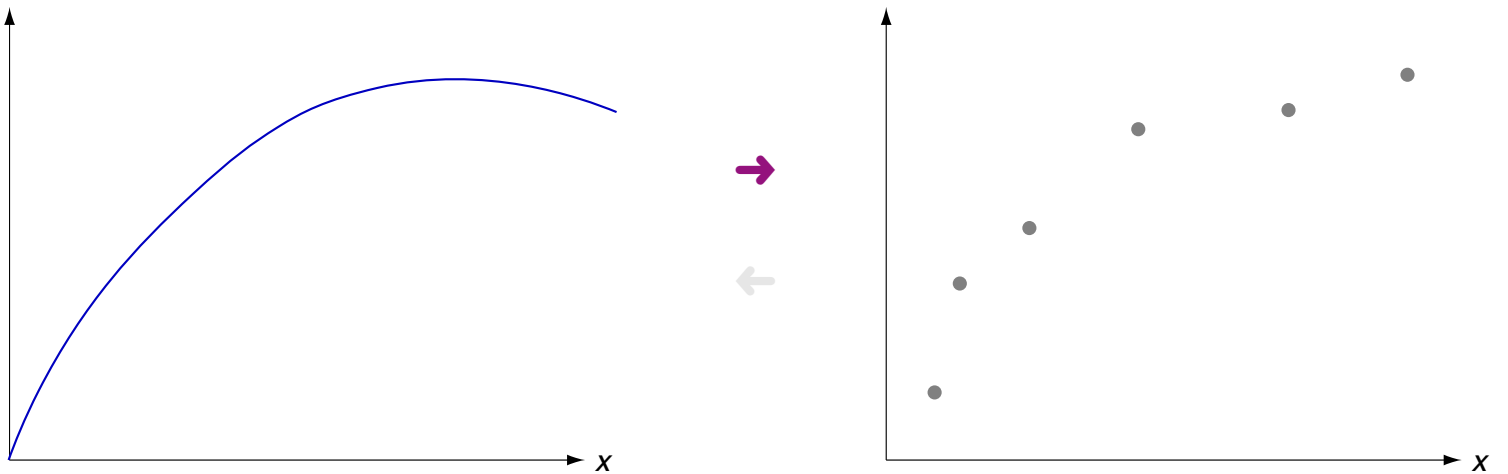


The scientific method to study physical systems:

- ➜ Forward modeling. Proposition of a theory (discovery of a model). Suggestion of model parameters to predict observable system parameters.
- ➜ Inverse modeling. Use observable system parameters to identify the parameters of the model representing the system.
  - Minimization of the set of model parameters that fully characterize the system.

Regularization

Machine Learning as an Inverse Problem

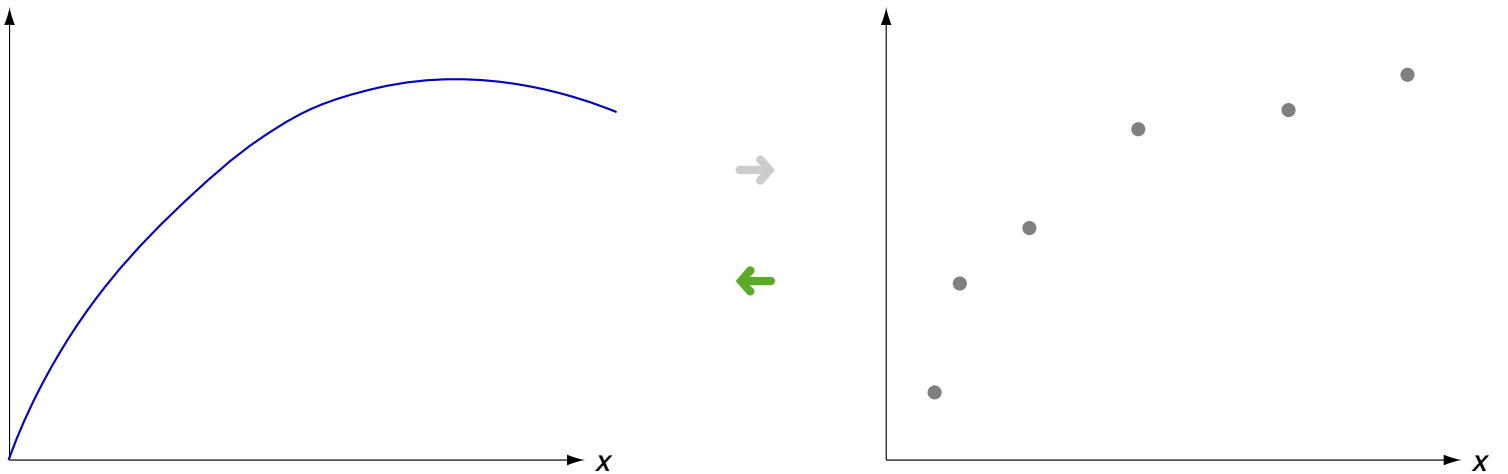


The scientific method to study physical systems:

- ➜ Forward modeling. Proposition of a theory (discovery of a model). Suggestion of model parameters to predict observable system parameters.
- ➜ Inverse modeling. Use observable system parameters to identify the parameters of the model representing the system.
- ❑ Minimization of the set of model parameters that fully characterize the system.

Regularization

Machine Learning as an Inverse Problem

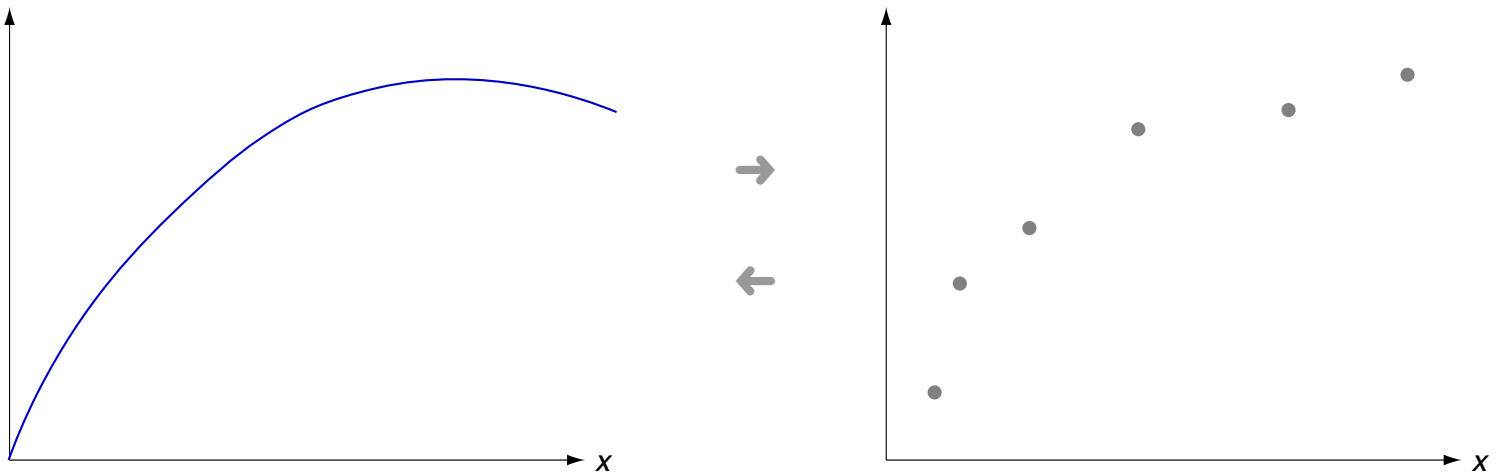


The scientific method to study physical systems:

- ➜ Forward modeling. Proposition of a theory (discovery of a model). Suggestion of model parameters to predict observable system parameters.
- ← Inverse modeling. Use observable system parameters to identify the parameters of the model representing the system.
- ❑ Minimization of the set of model parameters that fully characterize the system.

Regularization

Machine Learning as an Inverse Problem



The scientific method to study physical systems:

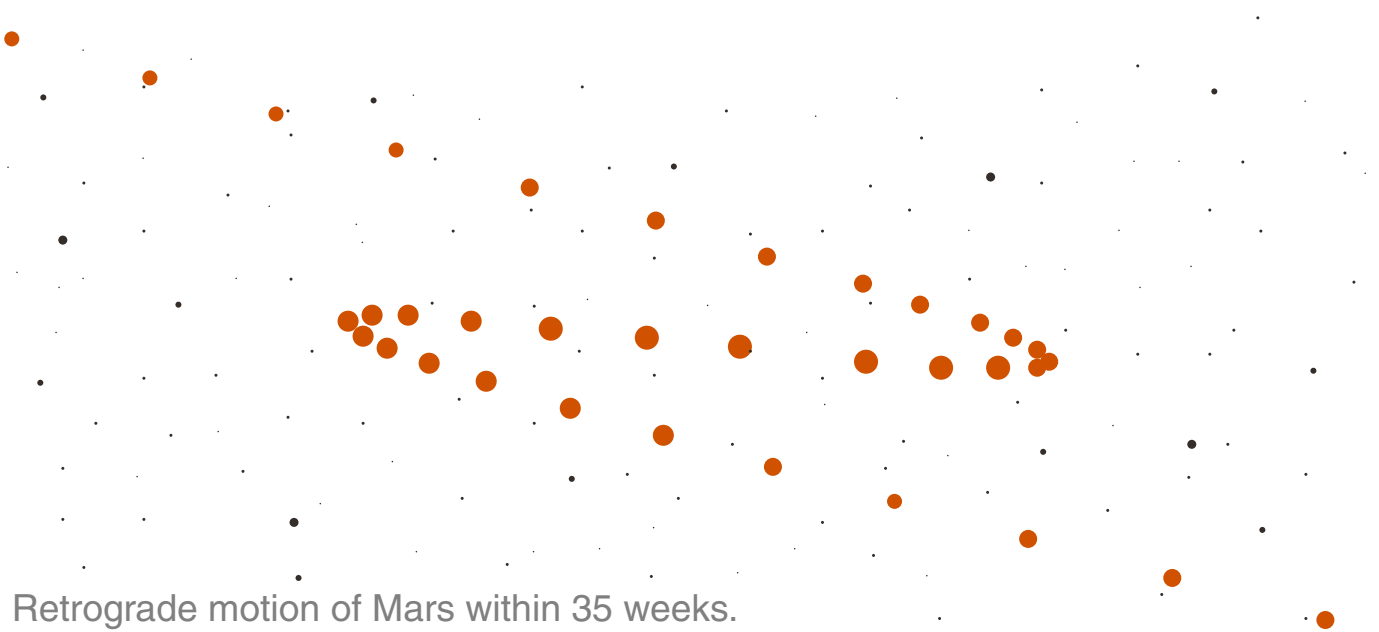
- ➜ Forward modeling. Proposition of a theory (discovery of a model). Suggestion of model parameters to predict observable system parameters.
- ➜ Inverse modeling. Use observable system parameters to identify the parameters of the model representing the system.
  - ❑ Minimization of the set of model parameters that fully characterize the system.

Regularization

Machine Learning as an Inverse Problem

Historical inverse problems [Google scholar] :

- Modeling the motions of the planets from observations of the night sky.
  Aristotle → Copernicus → Kepler → Newton → Einstein [NASA 2009]
- Modeling the inner composition of the Earth from observations at its surface.
- Modeling and predicting the weather based on past observations.



Retrograde motion of Mars within 35 weeks.

Regularization

Machine Learning as an Inverse Problem

Historical inverse problems [Google scholar] :

- Modeling the motions of the planets from observations of the night sky.
  Aristotle → Copernicus → Kepler → Newton → Einstein [NASA 2009]
- Modeling the inner composition of the Earth from observations at its surface.
- Modeling and predicting the weather based on past observations.

Practical considerations:

- Observable parameters may be insufficient to explain the system's behavior.
- Measurements contain errors dependent on the accuracy of the instrument.
- Inverse problems are often ill-posed due to discretization.

Regularization

Well-Posed vs. Ill-Posed Problems

Definition 10 (Well-posed Problem [Hadamard 1902]**)**

A mathematical problem is called well-posed if

1. a solution exists,
2. the solution is unique,
3. the solution's behavior changes continuously with the initial conditions.

Otherwise, the problem is called ill-posed.

Regularization

Well-Posed vs. Ill-Posed Problems

Definition 10 (Well-posed Problem [Hadamard 1902]**)**

A mathematical problem is called well-posed if

1. a solution exists,
2. the solution is unique,
3. the solution's behavior changes continuously with the initial conditions.

Otherwise, the problem is called ill-posed.

Linear regression:

- Existence and uniqueness of a solution depends on the rank of X.

➜ Use the pseudoinverse $(X^T X)^{-1} X^T$.

- The solution does not depend continuously on X.
- Introducing noise or removing data may strongly affect the solution.

➜ A possible way to find a solution is regularization. [Tikhonov 1977]

Remarks:

- According to Hadamard's philosophy, ill-posed problems are actually ill-posed, in the sense that the underlying model is wrong.
- Hadamard thought that ill-posed problems are a pure mathematical phenomenon and that all real-life problems are well-posed. However, in the second half of the century a number of very important real-life problems were found to be ill-posed. In particular, ill-posed problems arise when one tries to reverse the cause-effect relations: to find unknown causes from known consequences. Even if the cause-effect relationship forms a one-to-one mapping, the problem of inverting it can be ill-posed. [Vapnik 2000]
- Supervised learning has been shown to be an inverse problem. [de Vito 2005]

Regularization

Definition

Let $L(\mathbf{w})$ denote a loss function used to optimize the parameters $\mathbf{w}$ of a model function $y(\mathbf{x})$. Regularization introduces a trade-off between model complexity and inductive bias:

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w}),$$

where $\lambda \geq 0$ controls the impact of the regularization term $R(\mathbf{w}) \geq 0$. $\mathcal{L}$ is called "objective function".

Regularization

Definition

Let $L(\mathbf{w})$ denote a loss function used to optimize the parameters $\mathbf{w}$ of a model function $y(\mathbf{x})$. Regularization introduces a trade-off between model complexity and inductive bias:

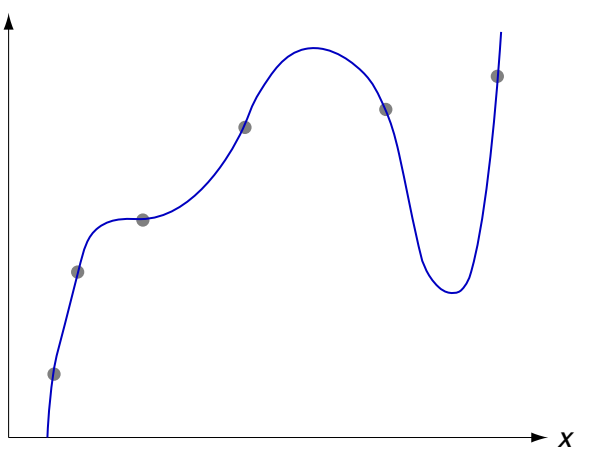
$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w}),$$

where $\lambda \geq 0$ controls the impact of the regularization term $R(\mathbf{w}) \geq 0$. $\mathcal{L}$ is called "objective function".

Example:

- $y(x) = w_0 + \sum_{i=1}^{6} w_i \cdot x^i; \qquad L(\mathbf{w}) = \text{RSS}(\mathbf{w})$
- $\lambda = 0$
- $R(\mathbf{w}) = 0$

$\rightsquigarrow \quad \hat{\mathbf{w}} = (-0.7,\ 14.1,\ -63.4,\ 114.8,\ -39.8,\ -84.1,\ 59.9)^T$



Regularization

Definition

Let $L(\mathbf{w})$ denote a loss function used to optimize the parameters $\mathbf{w}$ of a model function $y(\mathbf{x})$. Regularization introduces a trade-off between model complexity and inductive bias:

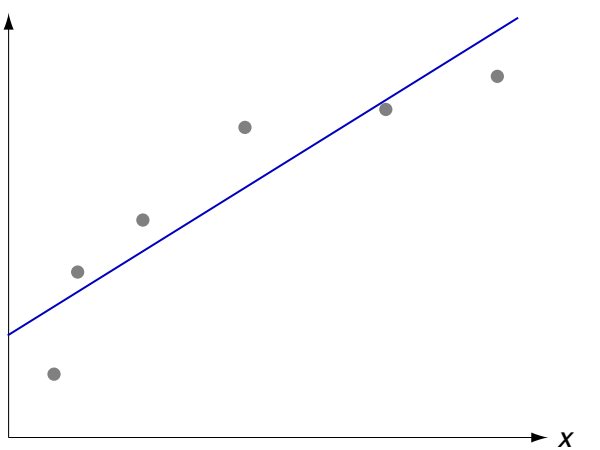
$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w}),$$

where $\lambda \geq 0$ controls the impact of the regularization term $R(\mathbf{w}) \geq 0$. $\mathcal{L}$ is called "objective function".

Example:

- $y(x) = w_0 + \sum_{i=1}^{6} w_i \cdot x^i; \qquad L(\mathbf{w}) = \text{RSS}(\mathbf{w})$
- $\lambda = 1$
- $R(\mathbf{w}) = 0 \cdot |w_1| + 100 \cdot |w_2| \;+\; \ldots \;+\; 100 \cdot |w_6|$

$\rightsquigarrow \quad \hat{\mathbf{w}} = (-0.2,\ 0.6,\ 0,\ 0,\ 0,\ 0,\ 0)^T$



Regularization

Definition

Let $L(\mathbf{w})$ denote a loss function used to optimize the parameters $\mathbf{w}$ of a model function $y(\mathbf{x})$. Regularization introduces a trade-off between model complexity and inductive bias:

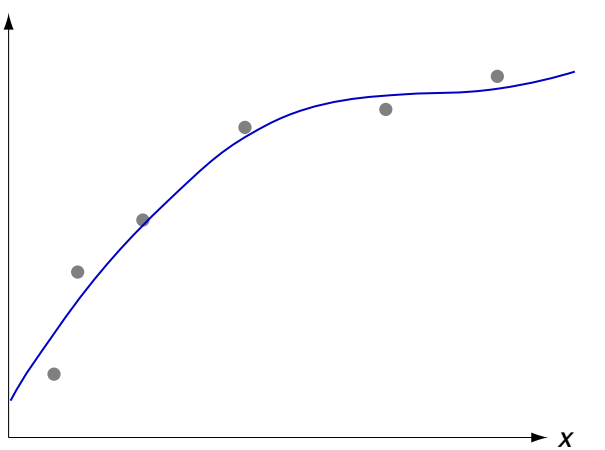
$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w}),$$

where $\lambda \geq 0$ controls the impact of the regularization term $R(\mathbf{w}) \geq 0$. $\mathcal{L}$ is called "objective function".

Example:

- $y(x) = w_0 + \sum_{i=1}^{6} w_i \cdot x^i; \qquad L(\mathbf{w}) = \text{RSS}(\mathbf{w})$
- $\lambda = 1/1000$
- $R(\mathbf{w}) = 0 \cdot |w_1| + 100 \cdot |w_2| \;+\; \ldots \;+\; 100 \cdot |w_6|$

$\rightsquigarrow \quad \hat{\mathbf{w}} = (-0.1,\ 0.6,\ 2.1,\ -1.0,\ -3.1,\ -0.5,\ 2.5)^T$



Regularization

Definition

Let $L(\mathbf{w})$ denote a loss function used to optimize the parameters $\mathbf{w}$ of a model function $y(\mathbf{x})$. Regularization introduces a trade-off between model complexity and inductive bias:

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w}),$$

where $\lambda \geq 0$ controls the impact of the regularization term $R(\mathbf{w}) \geq 0$. $\mathcal{L}$ is called "objective function".

Observations:

- Model complexity (partially) depends on the magnitude of the weights $\mathbf{w}$.
- Minimizing $L(\mathbf{w})$ places no bounds on the weights $\mathbf{w}$.
- Regularization introduces a "counterweight" $\lambda \cdot R(\mathbf{w})$ that grows with $\mathbf{w}$.
- Except λ, no additional hyperparameters are introduced.

Regularization

Definition

Let $L(\mathbf{w})$ denote a loss function used to optimize the parameters $\mathbf{w}$ of a model function $y(\mathbf{x})$. Regularization introduces a trade-off between model complexity and inductive bias:

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w}),$$

where $\lambda \geq 0$ controls the impact of the regularization term $R(\mathbf{w}) \geq 0$. $\mathcal{L}$ is called "objective function".

Regularization using the vector norm:

- Lasso regression. $R_{||\vec{\mathbf{w}}||_1}(\mathbf{w}) \;=\; \sum_{i=1}^{p} |w_i|$
- Ridge regression. $R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w}) \;=\; \sum_{i=1}^{p} w_i^2 \;=\; \vec{\mathbf{w}}^T \vec{\mathbf{w}}$

Remarks:

- "Lasso" is an acronym for "least absolute shrinkage and selection operator" and "ridge" a metaphor describing planes shaped like a hill range the highest point of which is not easily discernible.
- Ridge regression predates lasso regression. It is also known as weight decay in machine learning, and with multiple independent discoveries, it is variously known as the Tikhonov-Miller method, the Phillips-Twomey method, the constrained linear inversion method, and the method of linear regularization. [Wikipedia]
- $||\cdot||_k$ denotes the vector norm operator:

  $$||\mathbf{x}||_k \equiv \left(\sum_{i=1}^{p} |x_i|^k\right)^{1/k},$$

  where $k \in [1, \infty)$ and p is the dimensionality of vector $\mathbf{x}$.

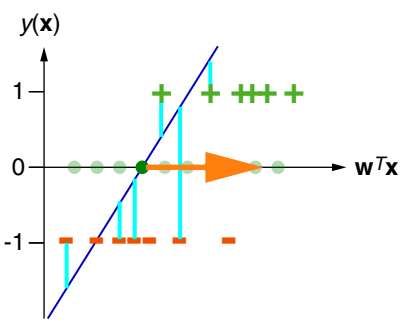
  By convention, $||\cdot||$ (omitting the subscript) typically refers to the Euclidean norm ($k = 2$).
- The regularization term constrains the magnitude of the direction vector of the hyperplane, progressively reducing the hyperplane's steepness as λ increases. The intercept w_0 is adjusted accordingly through minimization of $\mathcal{L}(\mathbf{w})$ but must not be part of the regularization term itself, which would lead to an incorrect solution.
  To denote the difference, we write $\mathbf{w} \equiv (w_0, w_1, \ldots, w_p)^T$ to refer to the entire parameter vector (the actual hypothesis), and $\vec{\mathbf{w}} \equiv (w_1, \ldots, w_p)^T$ for the direction vector excluding w_0.

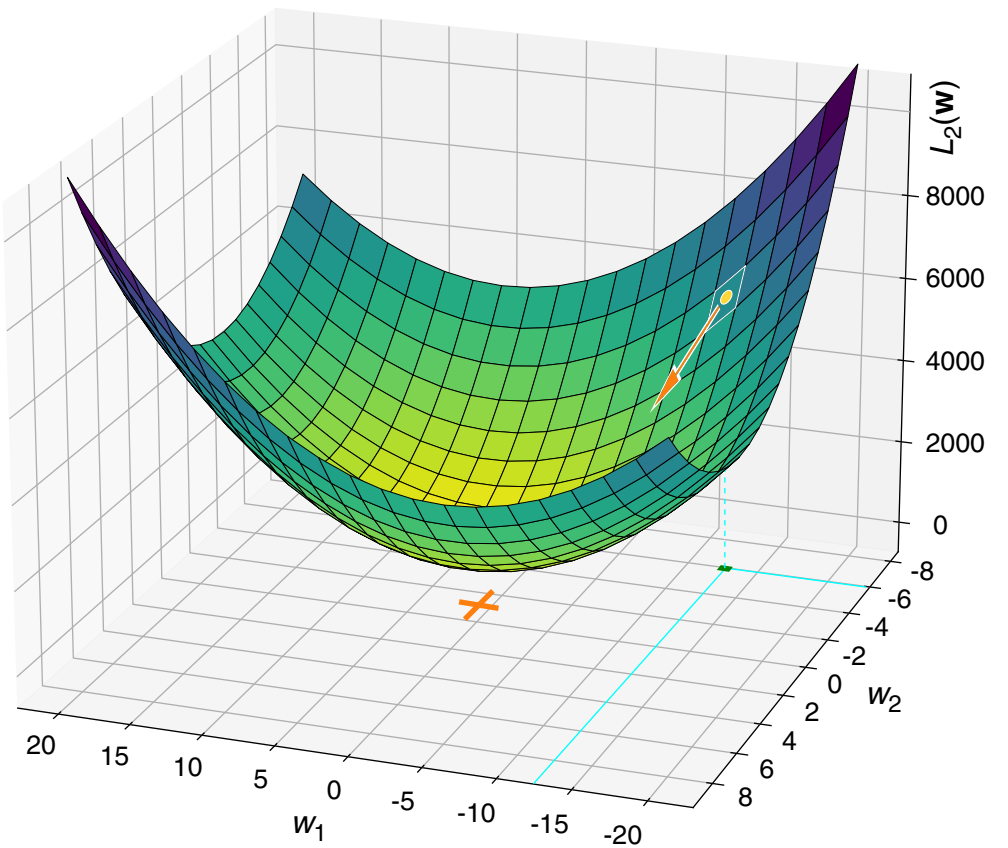
Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w})$$

Example:



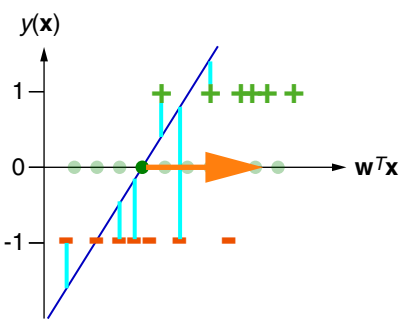


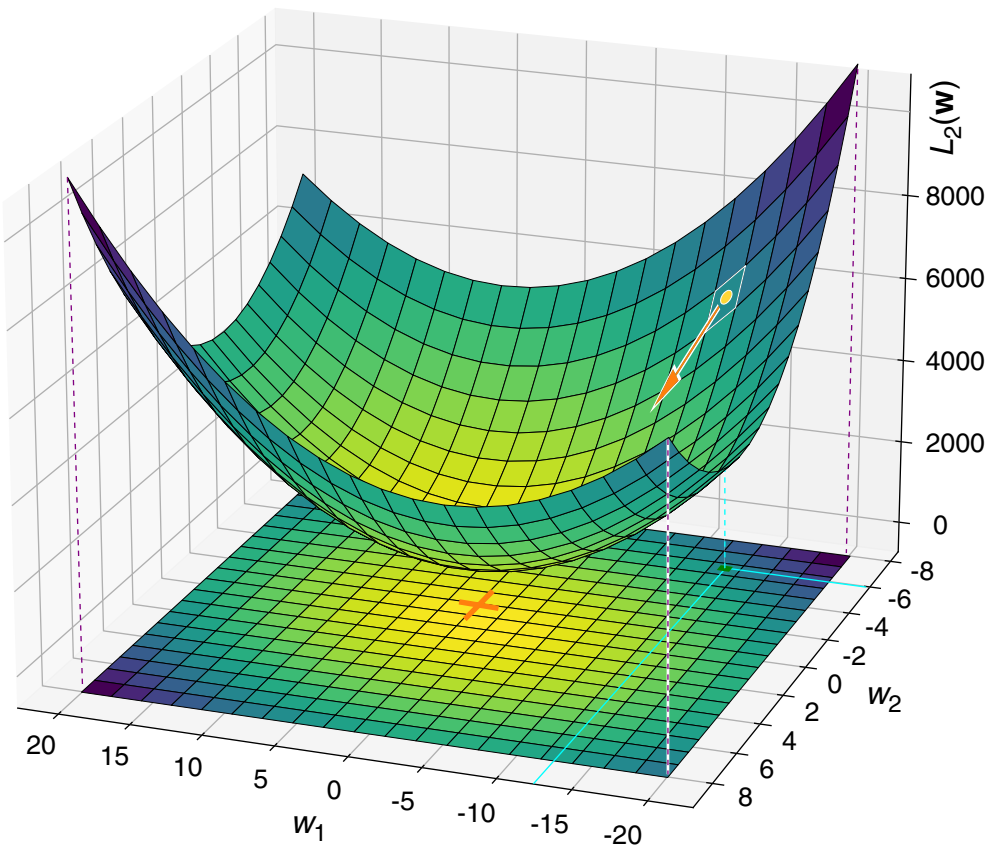
Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w})$$

Example:



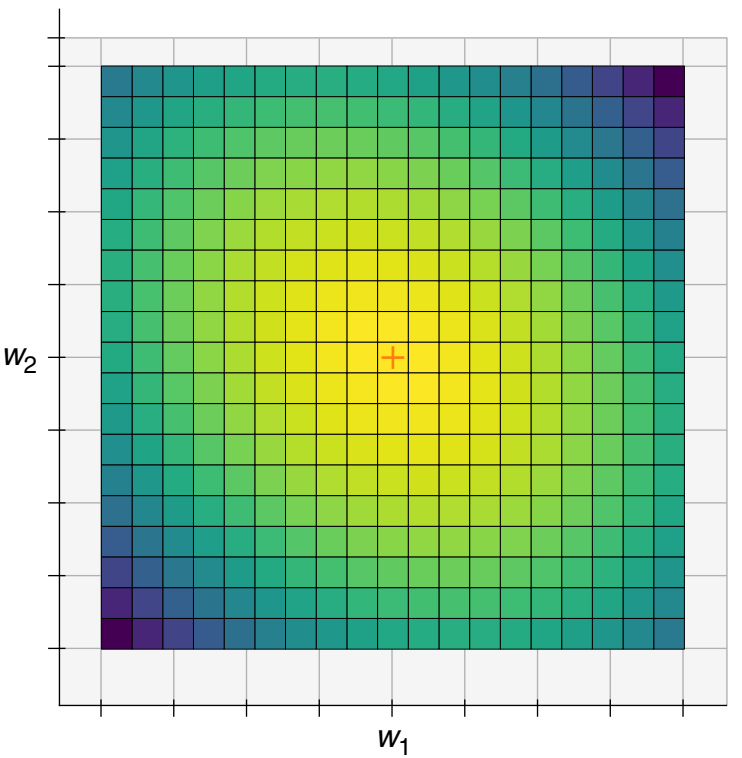


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w})$$

Example:

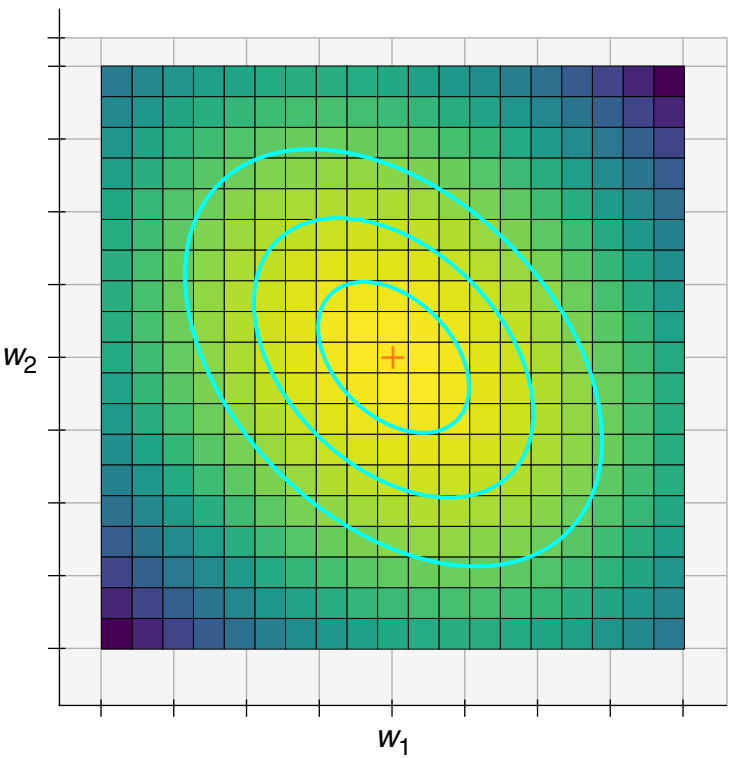


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w})$$

Example:

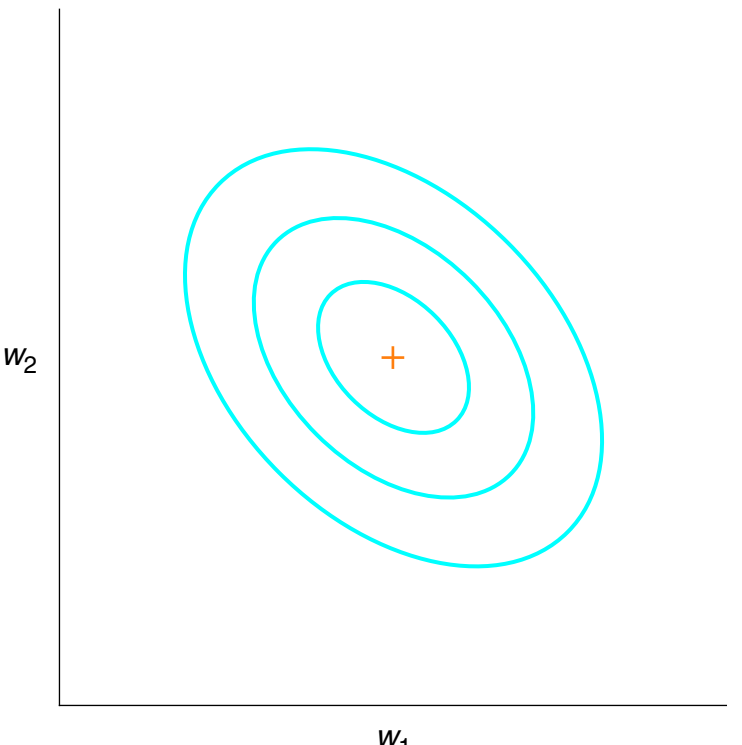


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w})$$

Example:

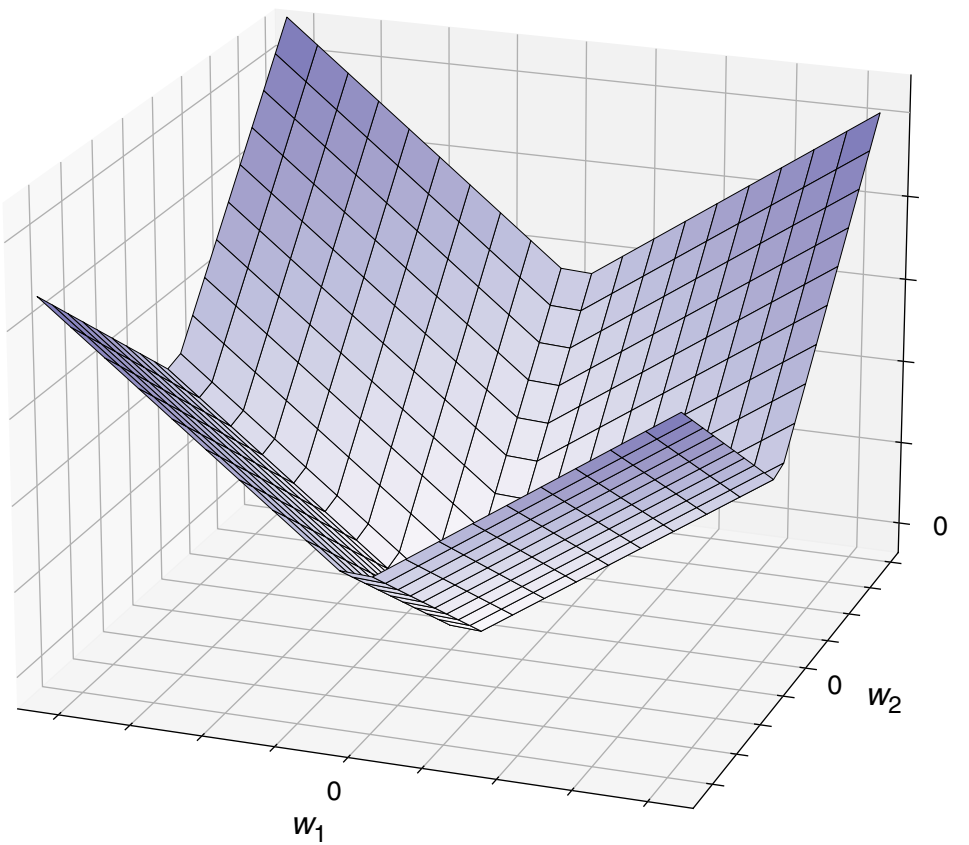


Regularization

Illustration: Lasso Regression

$$R_{||\vec{\mathbf{w}}||_1}(\mathbf{w}) \quad = \quad \sum_{i=1}^{p} |w_i|$$

Example:

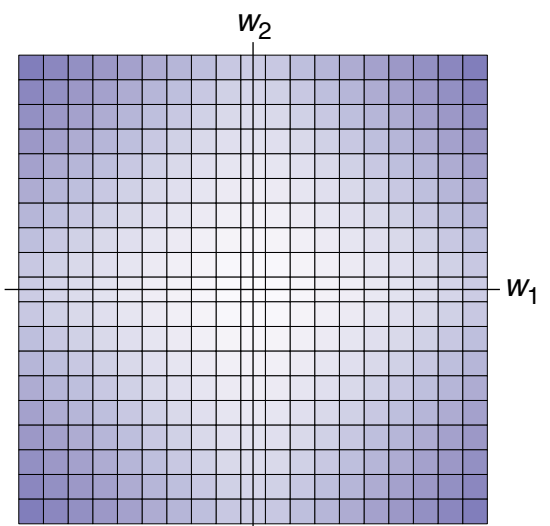


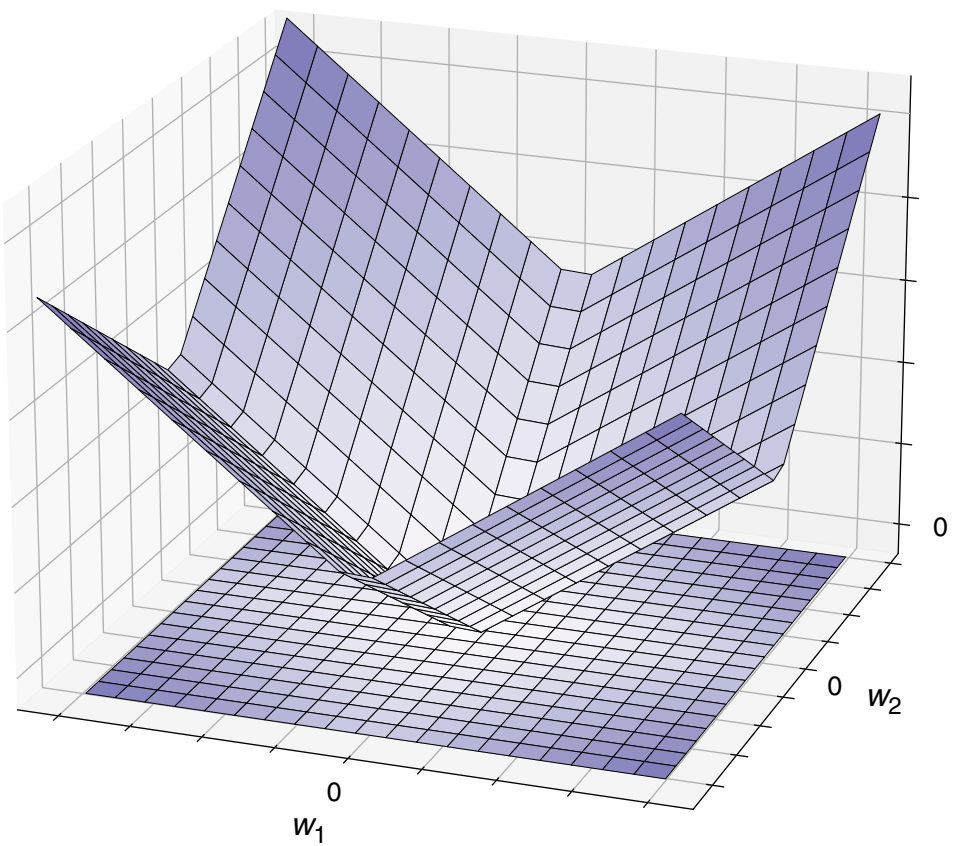
Regularization

Illustration: Lasso Regression

$$R_{||\vec{\mathbf{w}}||_1}(\mathbf{w}) \quad = \quad \sum_{i=1}^{p} |w_i|$$

Example:



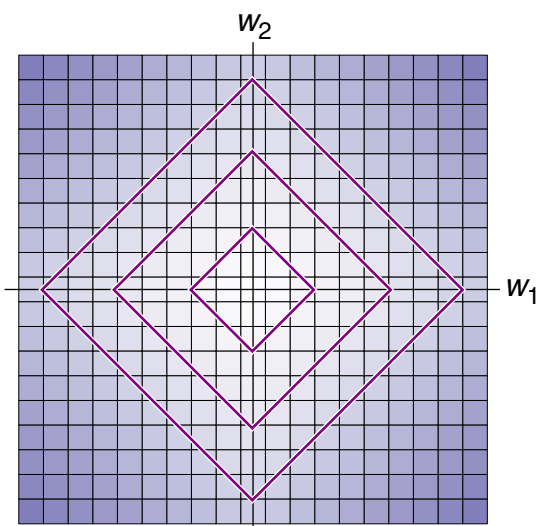


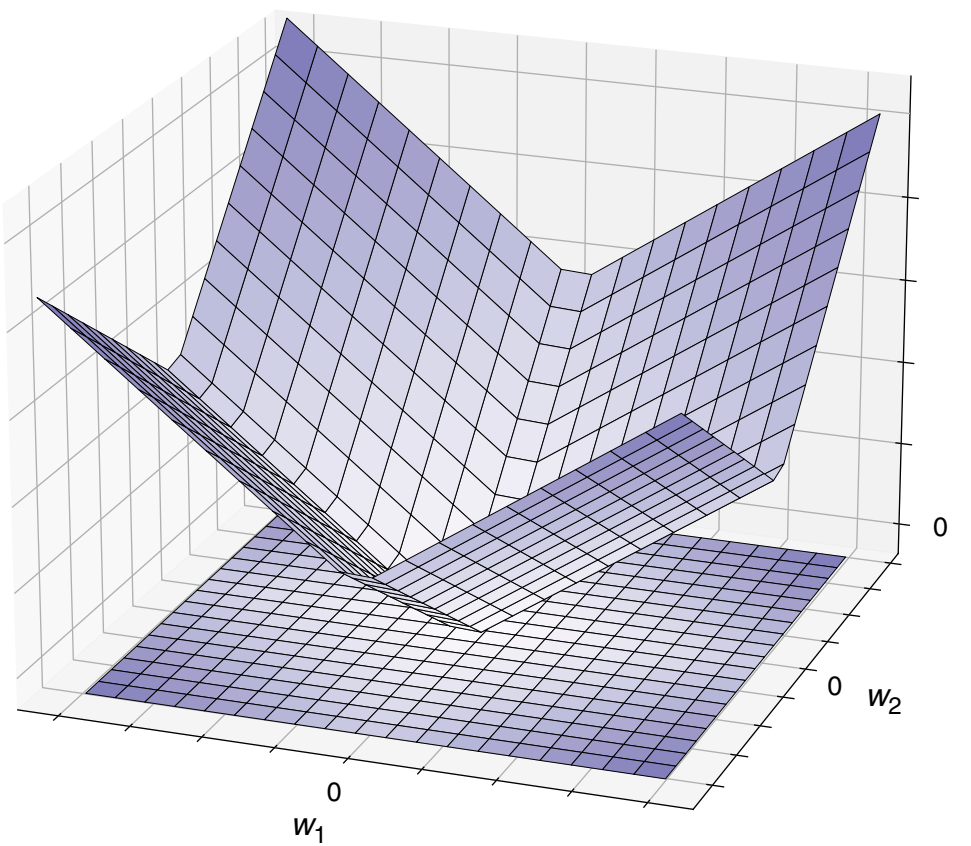
Regularization

Illustration: Lasso Regression

$$R_{||\vec{\mathbf{w}}||_1}(\mathbf{w}) \quad = \quad \sum_{i=1}^{p} |w_i|$$

Example:



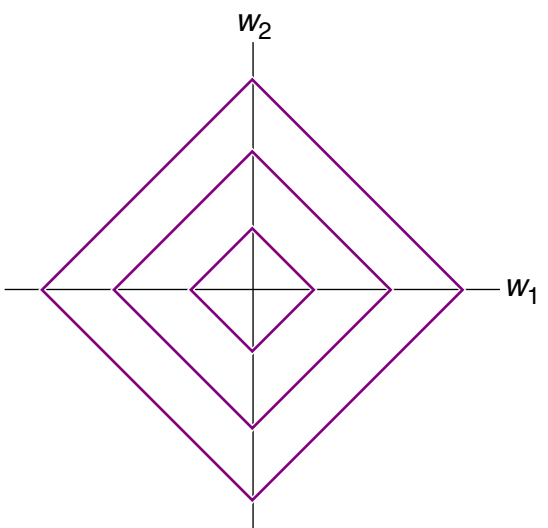


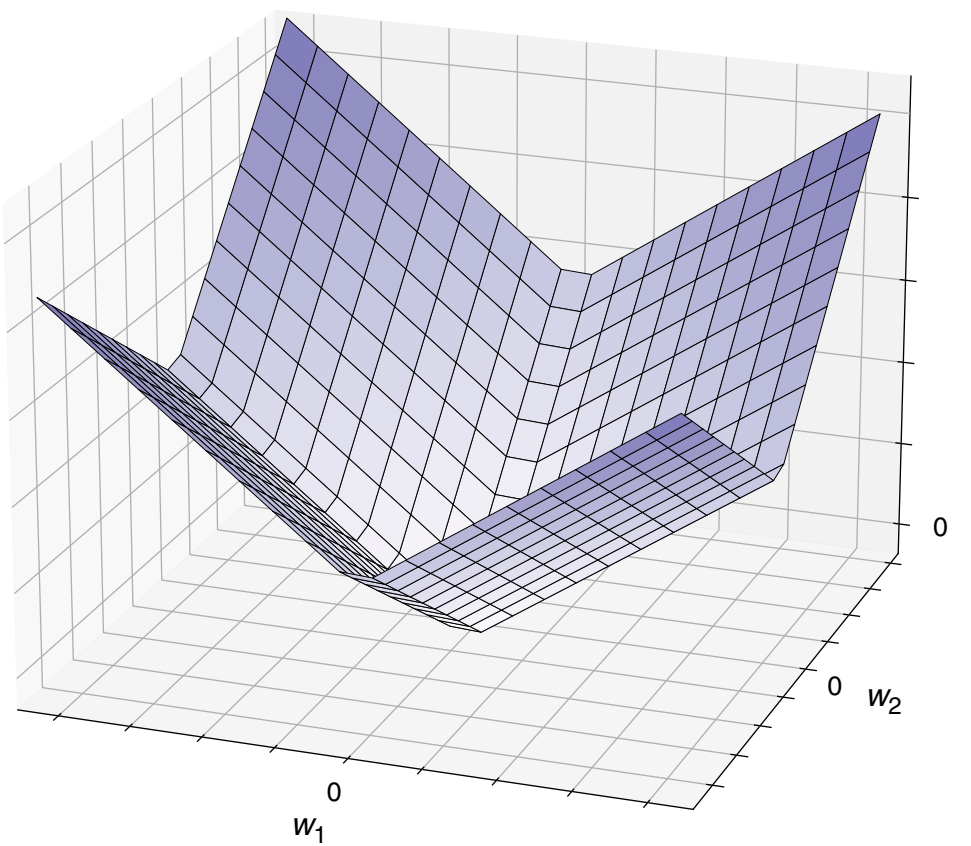
Regularization

Illustration: Lasso Regression

$$R_{||\vec{\mathbf{w}}||_1}(\mathbf{w}) \quad = \quad \sum_{i=1}^{p} |w_i|$$

Example:



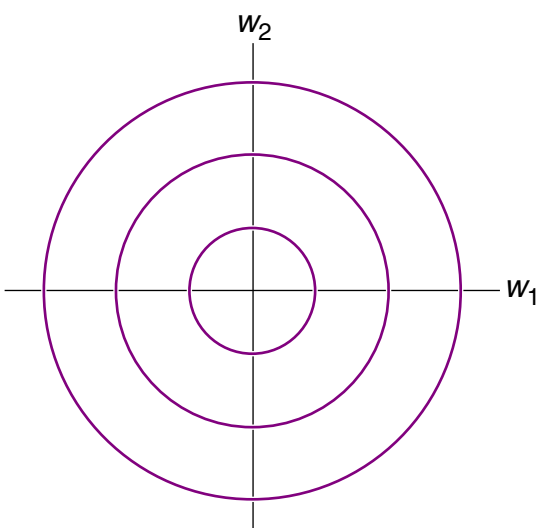


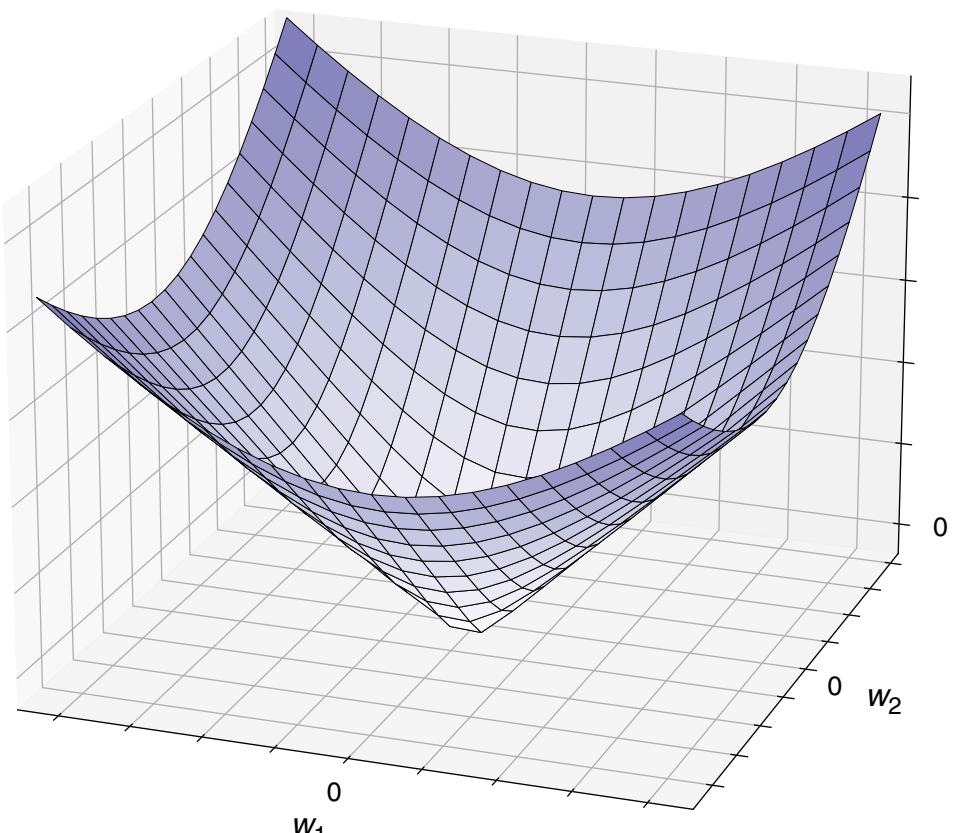
Regularization

Illustration: Ridge Regression

$$R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w}) \quad = \quad \sum_{i=1}^{p} w_i^2 \quad = \quad \vec{\mathbf{w}}^T \vec{\mathbf{w}}$$

Example:



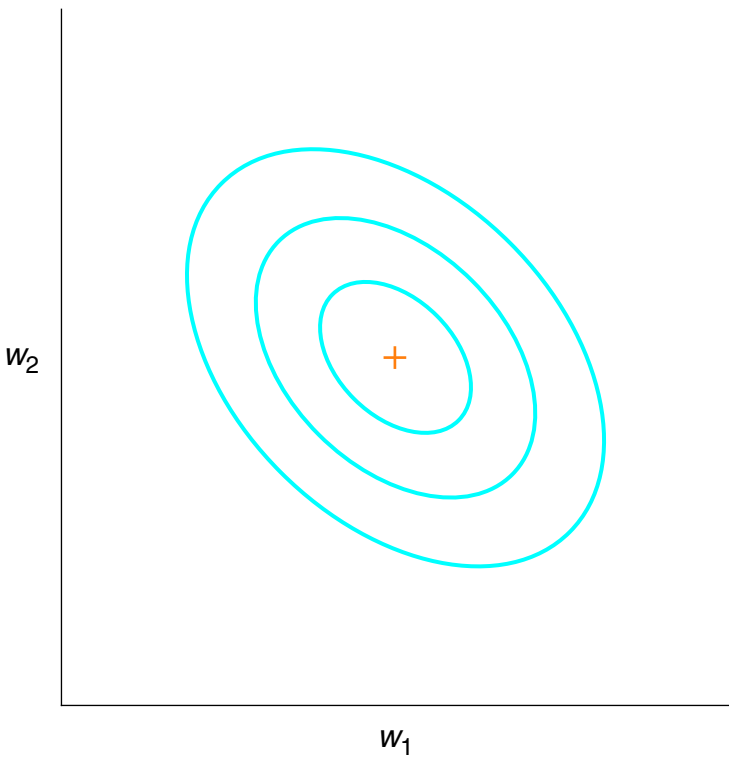


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R(\mathbf{w})$$

Example:

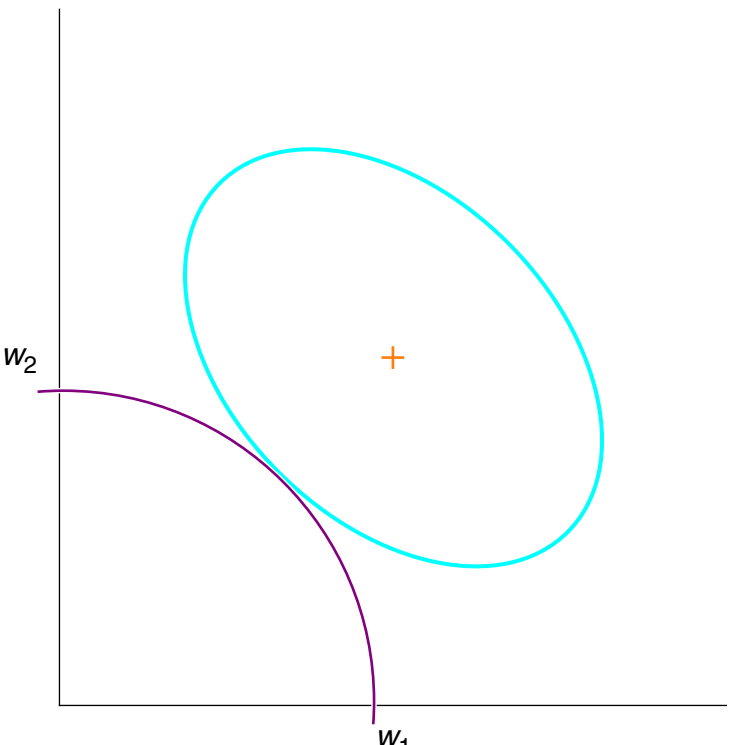


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

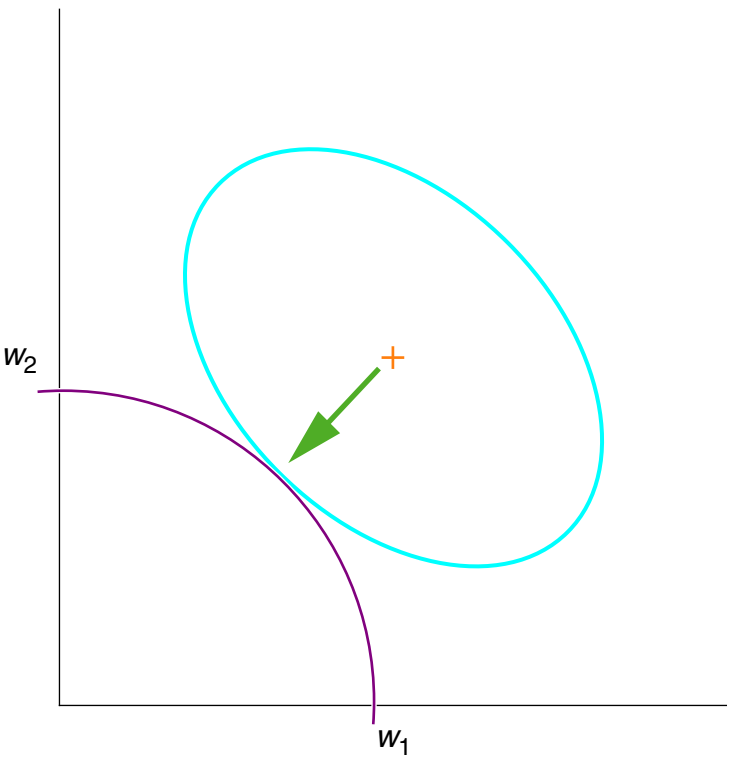


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

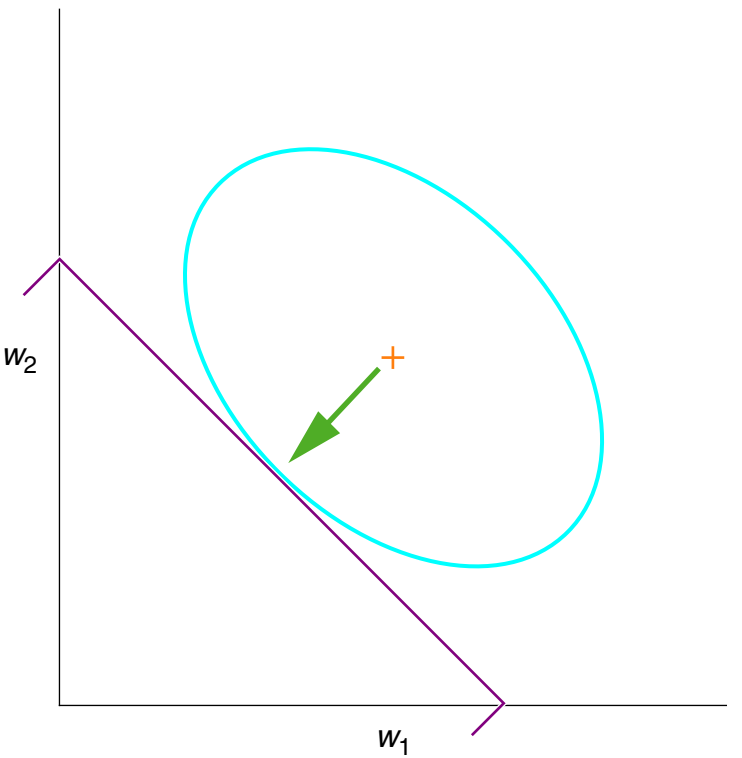


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda \cdot R_{||\vec{\mathbf{w}}||_1}(\mathbf{w})$$

Example:

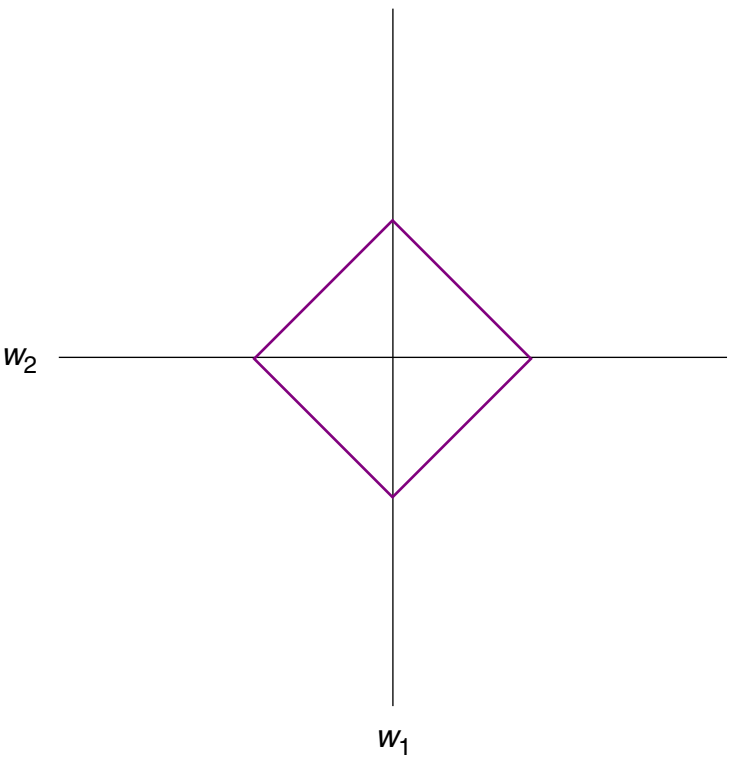


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda \cdot R_{||\vec{\mathbf{w}}||_1}(\mathbf{w})$$

Example:

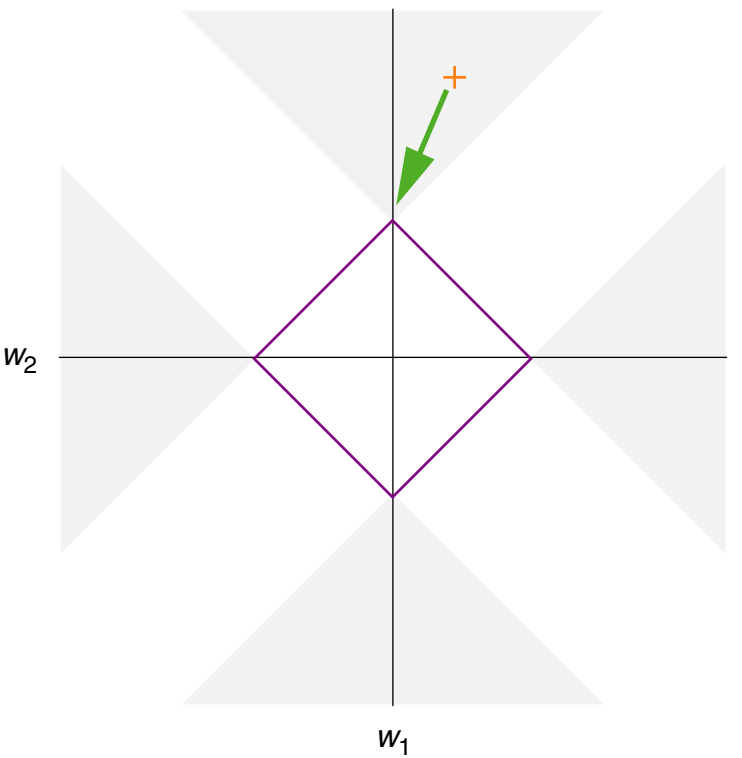


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda \cdot R_{||\vec{\mathbf{w}}||_1}(\mathbf{w})$$

Example:

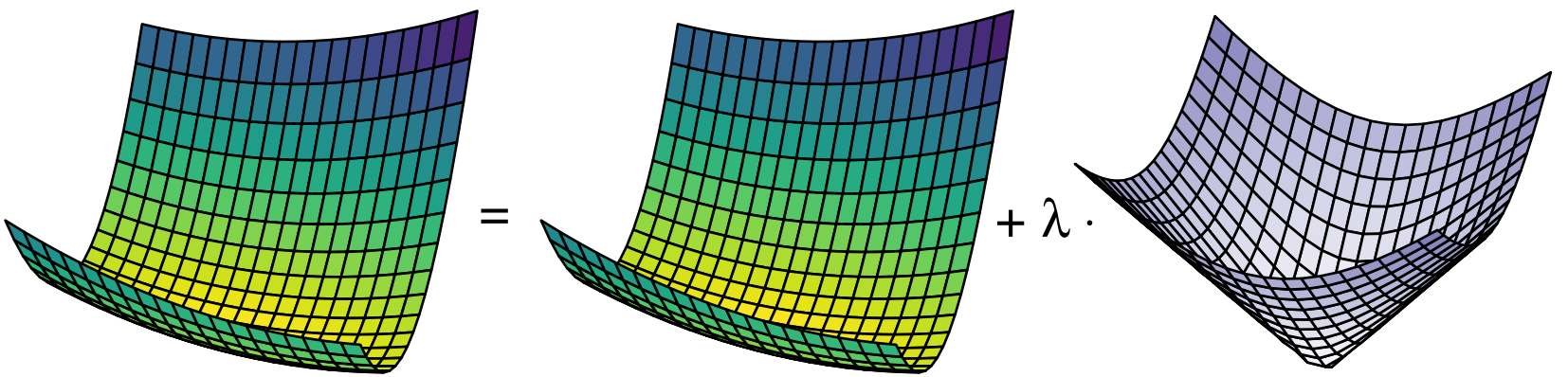


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

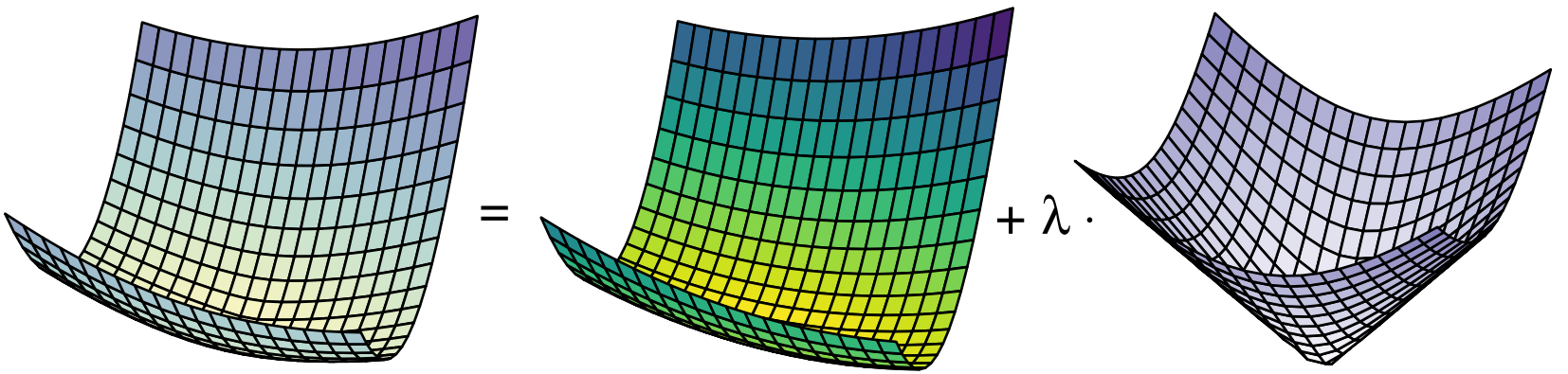


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

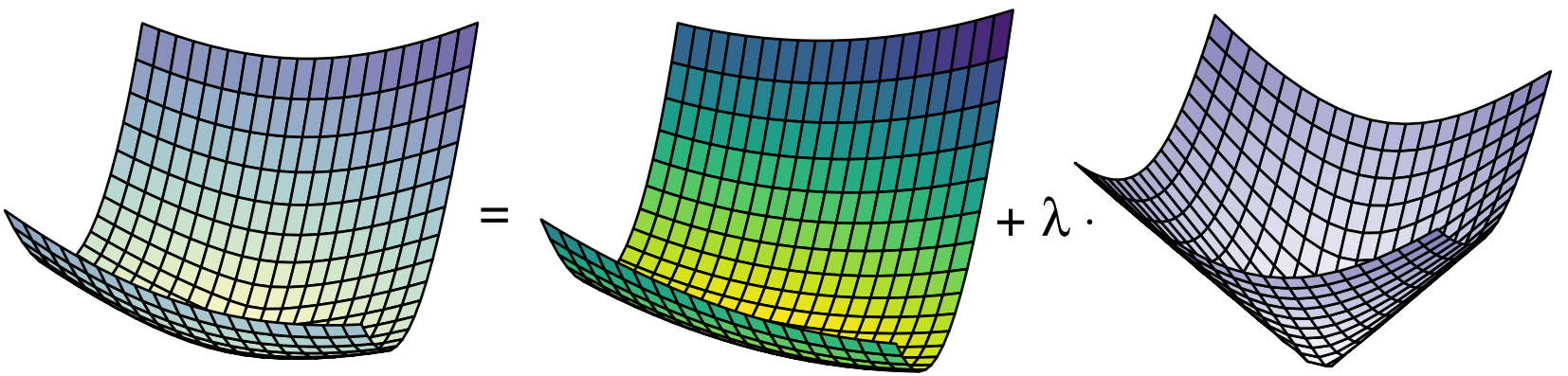


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

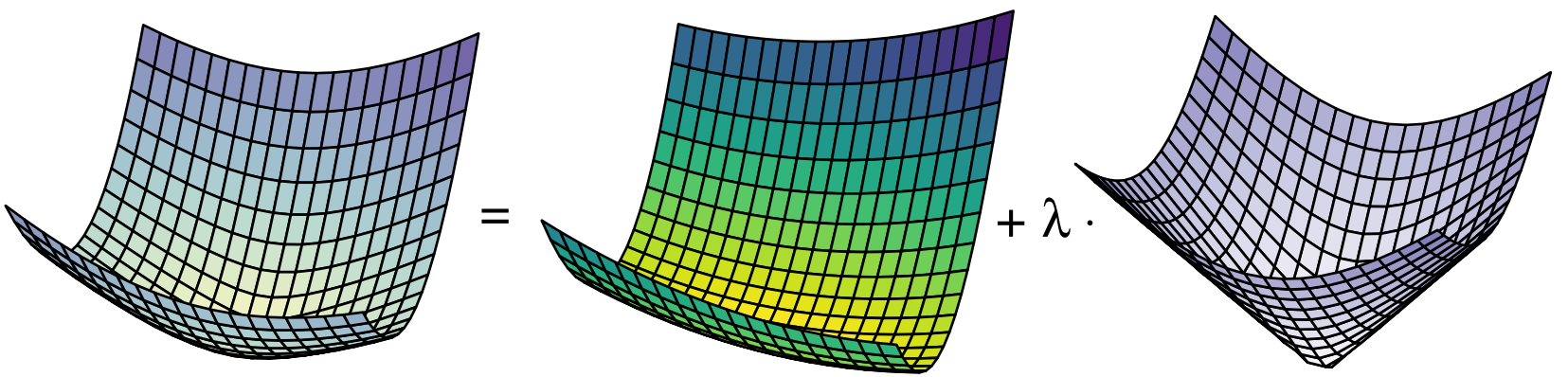


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

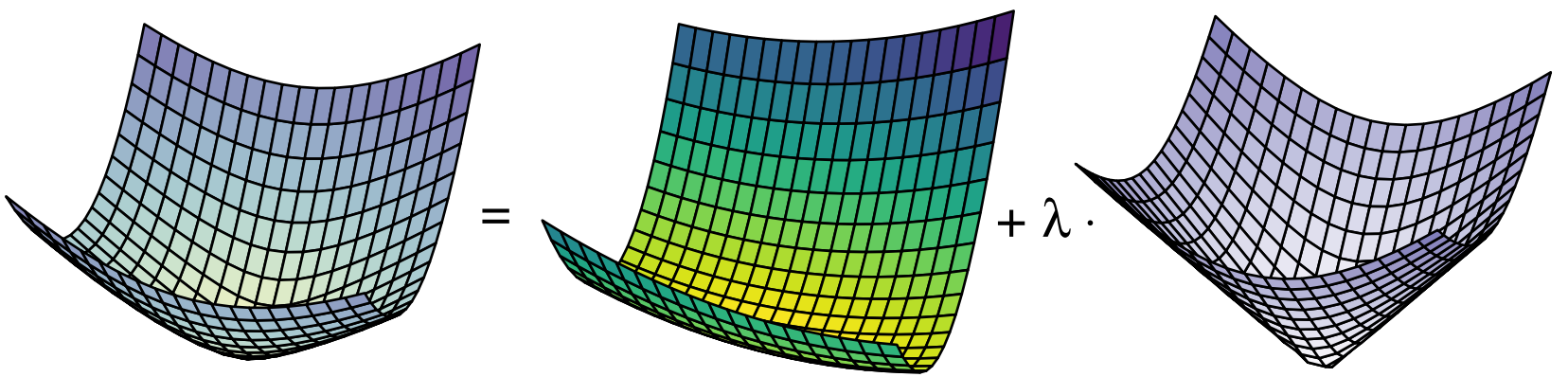


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

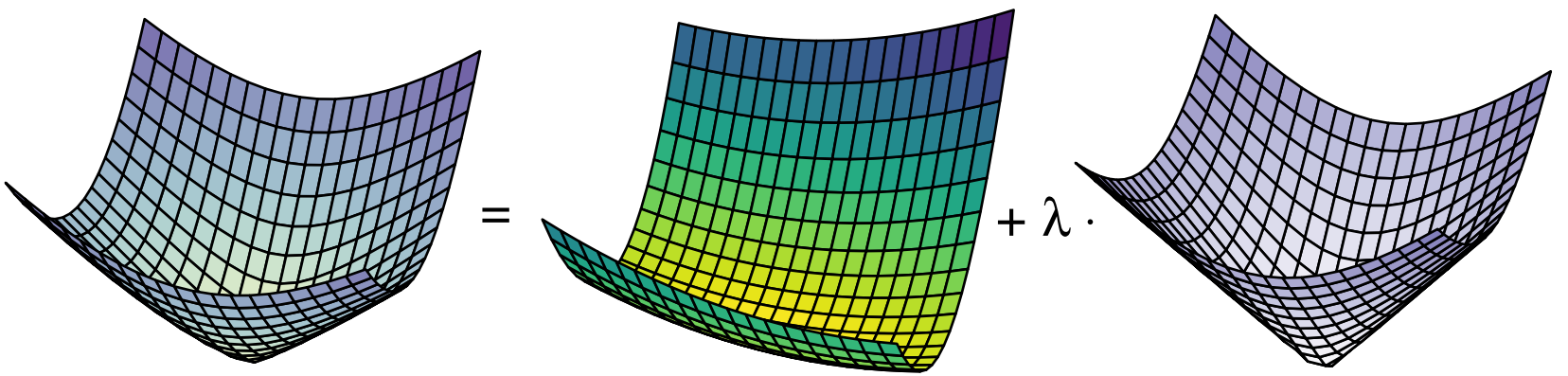


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

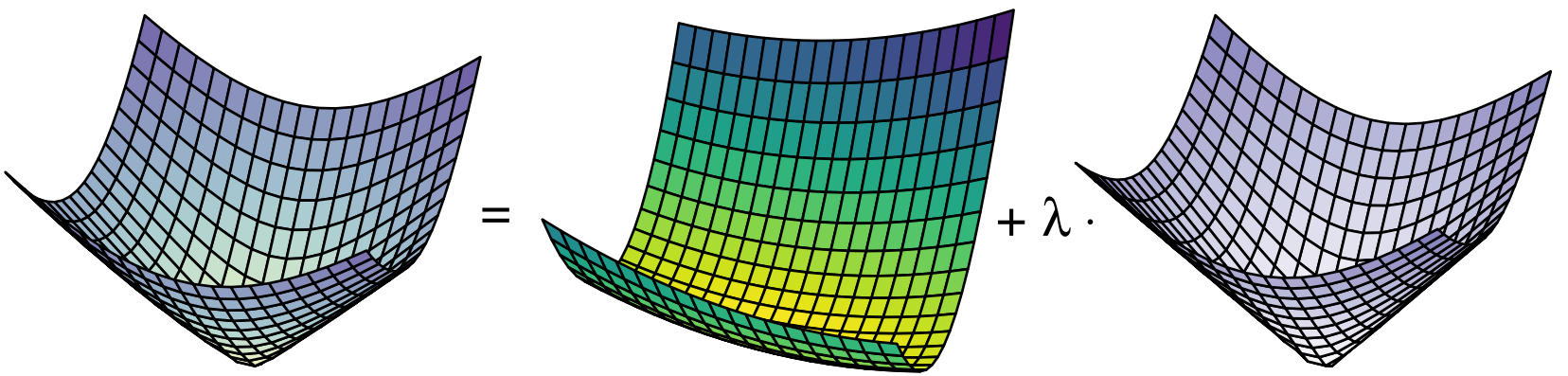


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

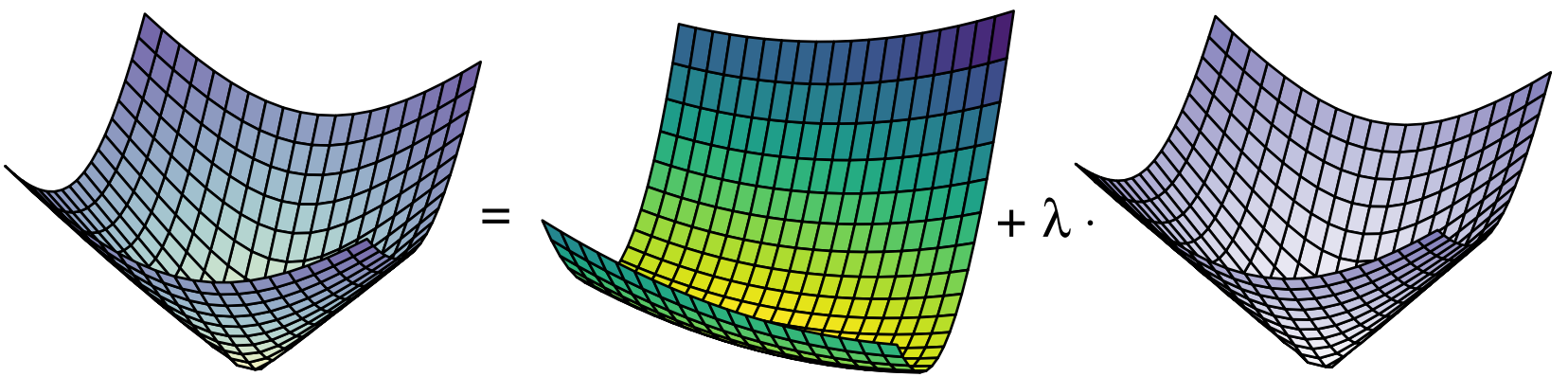


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:

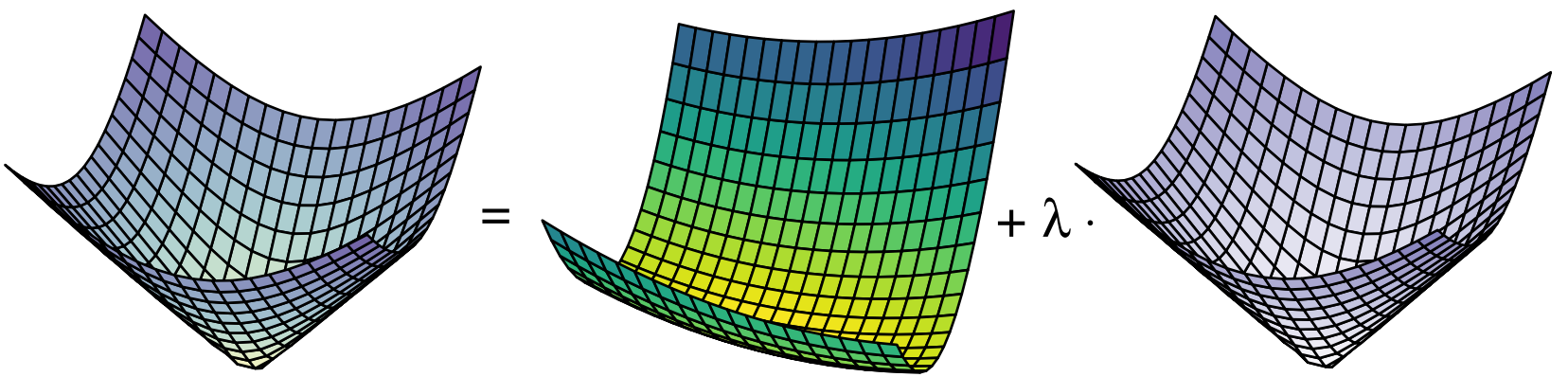


Regularization

Illustration

$$\mathcal{L}(\mathbf{w}) \;=\; L(\mathbf{w}) \;+\; \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w})$$

Example:



Remarks:

- The exemplified contour line plots of the parameter space (hypothesis space) show two-dimensional projections of the three-dimensional convex loss function (e.g., RSS) for a given set of example data, as well as the two regularization functions lasso $R_{||\mathbf{w}||_1}$ and ridge $R_{||\mathbf{w}||_2^2}$, respectively, whose shapes do not depend on the data. A contour line is a curve along which the respective function has a constant value.
- The exemplified loss function is minimal at the cross. Without regularization (e.g., $\lambda = 0$), the weights associated with the minimum would be the result of a linear regression. By adding the regularization term $\lambda \cdot R(\mathbf{w})$, with $\lambda > 0$, the joint minimum of the two functions is found closer to the origin of the parameter space than the minimum of the loss function.
- The choice of λ determines how much closer the joint minimum is to the origin of the parameter space; the higher, the closer, and thus the smaller the parameters $\mathbf{w}$.
- For a given λ, the new minimum is found where a contour line of the loss function tangents that of the regularization function.
- The choice of the regularization function (e.g., lasso $R_{||\vec{\mathbf{w}}||_1}$ or ridge $R_{||\vec{\mathbf{w}}||_2^2}$) determines the trajectory the minimum takes towards the origin as a function of λ. [stackexchange]

Remarks: (continued)

- A key difference between lasso and ridge regression is that, with lasso regression, parameters can be reduced to zero, eliminating the corresponding feature from the model function. With ridge regression, a parameter will be reduced to zero if, and only if, the minimum of the loss function is found on that parameter's axis.
- Lasso regression divides the parameter space into regions; when the loss function's minimum is found in the ones shaded gray, with increasing λ, the parameter value of the closest axis is eventually reduced to zero.
- Using lasso regression renders the associated inverse problem ill-posed, since it's solutions may not be unique. Moreover, the optimal parameters cannot be computed via a direct method.

Regularization

Regularized Linear Regression [Linear Regression]

- Given $\mathbf{x}$, predict a real-valued output under a linear model function:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot x_j$$

- Vector notation with $x_0 = 1$ and $\mathbf{w} = (w_0, w_1, \ldots, w_p)^T$:

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

- Given $\mathbf{x}_1, \ldots, \mathbf{x}_n$, assess goodness of fit of the objective function:

$$\mathcal{L}(\mathbf{w}) \ = \text{RSS}(\mathbf{w}) + \lambda \cdot R_{||\vec{\mathbf{w}}||_2^2}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda \cdot \vec{\mathbf{w}}^T \vec{\mathbf{w}} \tag{1}$$

- Estimate $\mathbf{w}$ by minimizing the residual sum of squares:

$$\hat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmin}} \ \mathcal{L}(\mathbf{w}) \tag{2}$$

Regularization

Regularized Linear Regression

- Let X denote the $n \times (p+1)$ matrix, where row i is $(1 \;\; \mathbf{x}_i^T)$, $\mathbf{x}_i \in D$.

  Let $\mathbf{y}$ denote the n-vector of outputs in the training set D.

$\rightsquigarrow \quad \mathcal{L}(\mathbf{w}) = (\mathbf{y} - X\mathbf{w})^T(\mathbf{y} - X\mathbf{w}) + \lambda \cdot \vec{\mathbf{w}}^T\vec{\mathbf{w}}$

Regularization

Regularized Linear Regression

- Let X denote the $n \times (p+1)$ matrix, where row i is $(1 \;\; \mathbf{x}_i^T)$, $\mathbf{x}_i \in D$.

  Let $\mathbf{y}$ denote the n-vector of outputs in the training set D.

$\rightsquigarrow \quad \mathcal{L}(\mathbf{w}) = (\mathbf{y} - X\mathbf{w})^T(\mathbf{y} - X\mathbf{w}) + \lambda \cdot \vec{\mathbf{w}}^T\vec{\mathbf{w}}$

- Minimizing $\mathcal{L}(\mathbf{w})$ by a direct method:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} &= -2X^T(\mathbf{y} - X\mathbf{w}) + 2\lambda \cdot \begin{pmatrix} 0 \\ \vec{\mathbf{w}} \end{pmatrix} = 0 \\
X^T(\mathbf{y} - X\mathbf{w}) - \lambda \cdot \begin{pmatrix} 0 \\ \vec{\mathbf{w}} \end{pmatrix} &= 0 \\
\Leftrightarrow \quad (X^TX + \lambda \cdot \mathrm{diag}(0, 1, \ldots, 1))\mathbf{w} &= X^T\mathbf{y} \\
\Leftrightarrow \quad \mathbf{w} &= (X^TX + \mathrm{diag}(0, \lambda, \ldots, \lambda))^{-1}X^T\mathbf{y}
\end{aligned}$$

Regularization

Regularized Linear Regression

- Let X denote the $n \times (p+1)$ matrix, where row i is $(1 \;\; \mathbf{x}_i^T)$, $\mathbf{x}_i \in D$.

  Let $\mathbf{y}$ denote the n-vector of outputs in the training set D.

$\rightsquigarrow \quad \mathcal{L}(\mathbf{w}) = (\mathbf{y} - X\mathbf{w})^T(\mathbf{y} - X\mathbf{w}) + \lambda \cdot \vec{\mathbf{w}}^T\vec{\mathbf{w}}$

- Minimizing $\mathcal{L}(\mathbf{w})$ by a direct method:

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = -2X^T(\mathbf{y} - X\mathbf{w}) + 2\lambda \cdot \begin{pmatrix} 0 \\ \vec{\mathbf{w}} \end{pmatrix} = 0$$

$$X^T(\mathbf{y} - X\mathbf{w}) - \lambda \cdot \begin{pmatrix} 0 \\ \vec{\mathbf{w}} \end{pmatrix} = 0$$

$$\Leftrightarrow \quad (X^TX + \lambda \cdot \mathrm{diag}(0, 1, \ldots, 1))\mathbf{w} = X^T\mathbf{y} \qquad \text{Normal eqns.}$$

$$\Leftrightarrow \quad \mathbf{w} = \underbrace{(X^TX + \mathrm{diag}(0, \lambda, \ldots, \lambda))}^{-1}X^T\mathbf{y} \qquad \text{If } \lambda > 0.$$

Conditioning the moment matrix X^TX [Wikipedia 1, 2, 3]

Regularization

Regularized Linear Regression

- Let X denote the $n \times (p+1)$ matrix, where row i is $(1 \;\; \mathbf{x}_i^T)$, $\mathbf{x}_i \in D$.

  Let $\mathbf{y}$ denote the n-vector of outputs in the training set D.

$\rightsquigarrow \quad \mathcal{L}(\mathbf{w}) = (\mathbf{y} - X\mathbf{w})^T(\mathbf{y} - X\mathbf{w}) + \lambda \cdot \vec{\mathbf{w}}^T\vec{\mathbf{w}}$

- Minimizing $\mathcal{L}(\mathbf{w})$ by a direct method:

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = -2X^T(\mathbf{y} - X\mathbf{w}) + 2\lambda \cdot \begin{pmatrix} 0 \\ \vec{\mathbf{w}} \end{pmatrix} = 0$$

$$X^T(\mathbf{y} - X\mathbf{w}) - \lambda \cdot \begin{pmatrix} 0 \\ \vec{\mathbf{w}} \end{pmatrix} = 0$$

$$\Leftrightarrow \quad (X^TX + \lambda \cdot \mathrm{diag}(0, 1, \ldots, 1))\mathbf{w} = X^T\mathbf{y} \qquad \text{Normal eqns.}$$

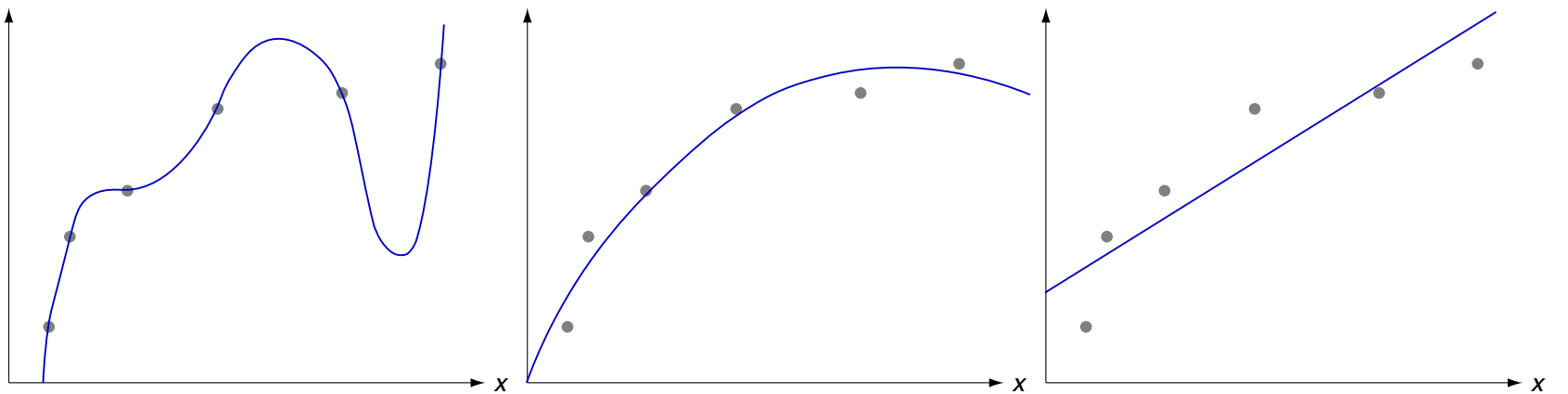
$$\Leftrightarrow \quad \hat{\mathbf{w}} \equiv \mathbf{w} = \underbrace{(X^TX + \mathrm{diag}(0, \lambda, \ldots, \lambda))}^{-1}X^T\mathbf{y} \qquad \text{If } \lambda > 0.$$

Conditioning the moment matrix X^TX [Wikipedia 1, 2, 3]

$$\hat{y}(\mathbf{x}_i) = \hat{\mathbf{w}}^T \mathbf{x}_i$$ Regression function with least squares estimator $\hat{\mathbf{w}}$.

Regularization

Regularized Linear Regression: Hyperparameter λ



$\lambda = 0$ ——— $\lambda = 0.2$ ———▷ $\lambda = 0.7$

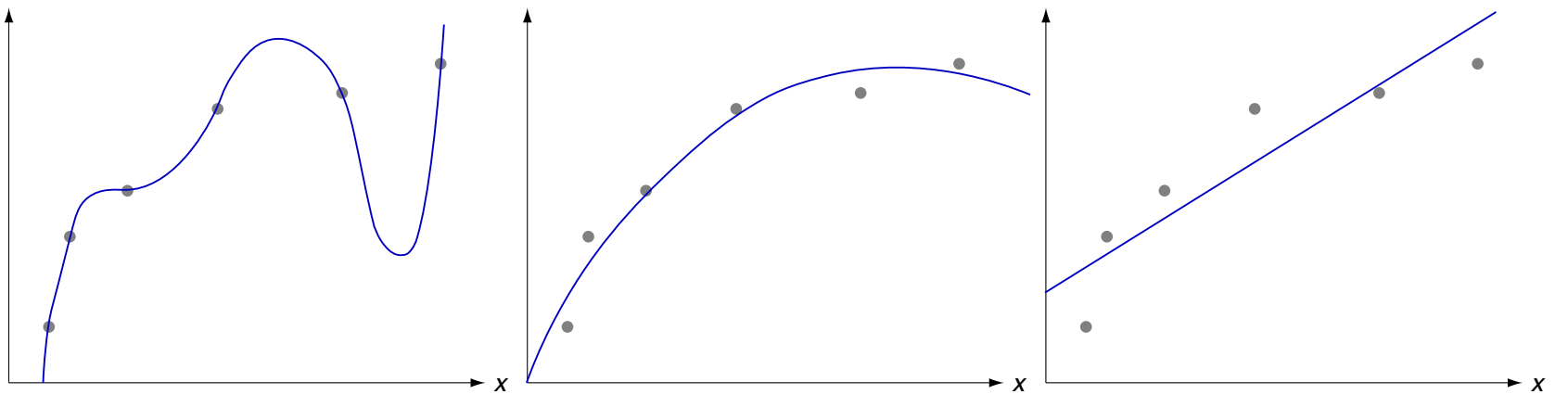
The regularization parameter λ penalizes high weights w_i:

$$y(x) = w_0 + \sum_{i=1}^{p} w_i \cdot x^i = \mathbf{w}^T \mathbf{x}$$

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda \cdot \vec{\mathbf{w}}^T \vec{\mathbf{w}}$$

Regularization

Regularized Linear Regression: Hyperparameter λ



$$\lambda = 0 \longrightarrow \lambda = 0.2 \longrightarrow \lambda = 0.7$$

"No black-box procedures for choosing the regularization parameter λ are available, and most likely will never exist." [Hansen and Hanke 1993]

Heuristics:

- Model selection for a choice of $\lambda_1, \ldots, \lambda_m$. [stackoverflow]
- More advanced heuristics have been proposed. [Bauer and Lukas 2011]