Machine Learning in Healthcare

#C06 Regularization

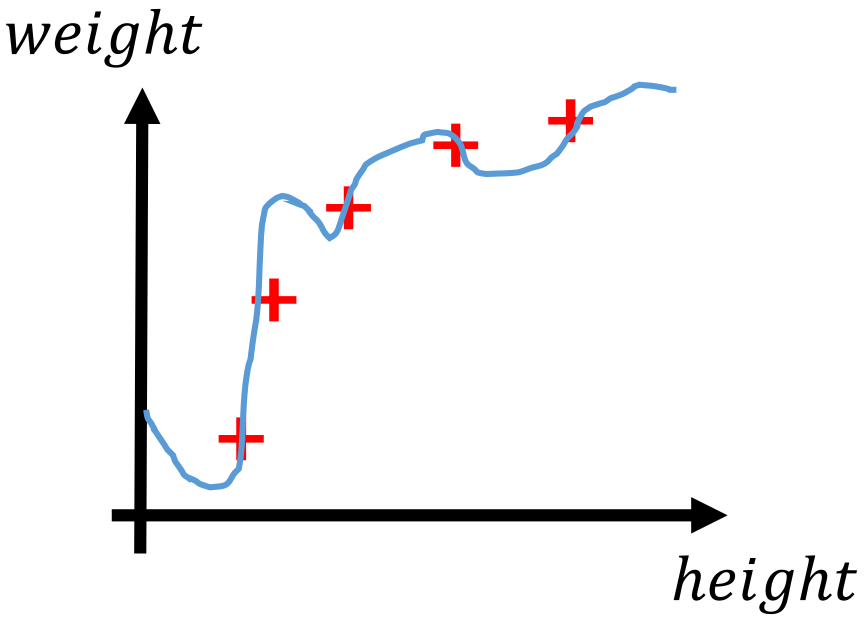
Technion-IIT, Haifa, Israel

Assist. Prof. Joachim Behar
Biomedical Engineering Faculty
Technion-IIT

AIMLab.

Introduction

- You trained a model with its $J \rightarrow 0$. You feel very proud!
- Then you go out in the real world and start making predictions. Surprise, results are not good at all! What happened?
- Very likely your model is overfitting the training examples leading to bad generalization.



$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

Introduction

Table 2. Classification performance measured by F_1. The table reports the overall and individual rhythm class performance by random forest based and XGBoost based models on the training and unseen test set.

		Recordings	Overall	N	A	O	~
Official challenge entry (Vollmer *et al* 2017)	Training set	8528	0.94	0.98	0.91	0.94	0.90
	Test set	3658	0.81	0.91	0.81	0.70	0.46
Enhanced post-challenge entry	Training set	8528	$\boxed{0.99}$	0.99	0.99	0.98	0.99
	Test set	3658	$\boxed{0.82}$	0.91	0.82	0.74	—[a]

Sodmann, Philipp, et al. "A convolutional neural network for ECG annotation as the basis for classification of cardiac rhythms." Physiological measurement 39.10 (2018): 104005.

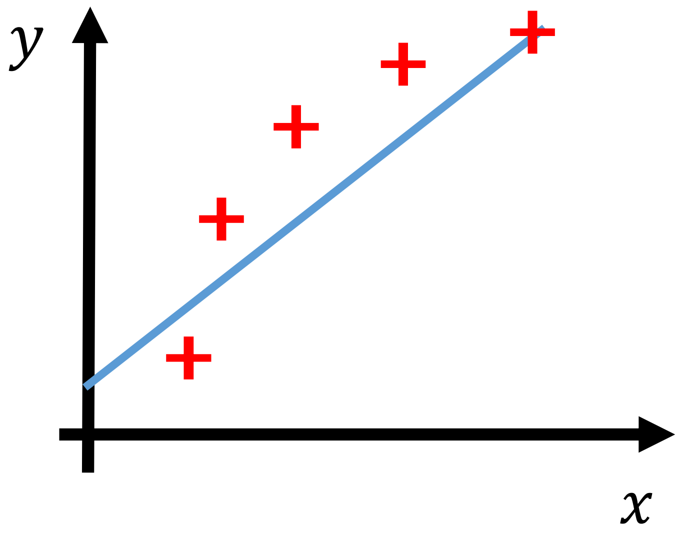
Overfitting

Overfitting

- One of the most important consideration when learning a model is how well it will generalize to new observations. This is called **generalization**.
- This is important because we train our model on a population sample dataset which has some noise.
- In other words, generalization refers to how well the concepts learned by a machine learning model will translate to new observations not seen by the model when it was trained.
- This is related to the concept of **overfitting** and **underfitting**.
- In particular, we will focus on overfitting which is a phenomenon that usually happens with complex models.

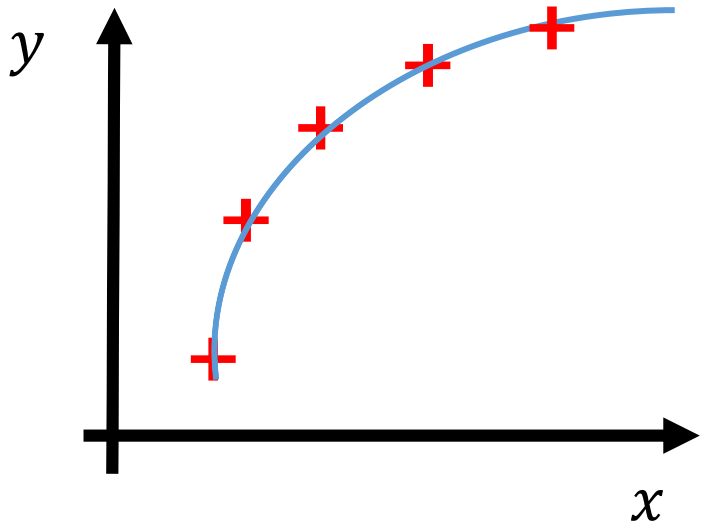
Overfitting - Regression

Overfitting: refers to a model where the learned hypothesis fits the training set very well ($J(w) \to 0$) but fails to generalize to new observations.

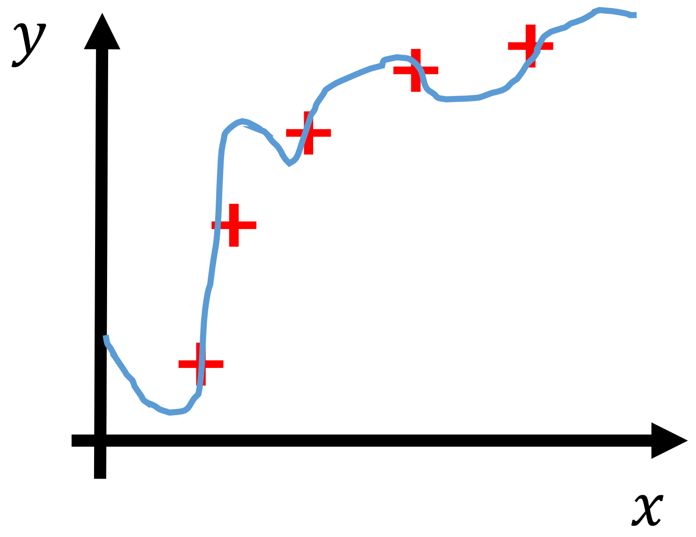


$$y = w_0 + w_1 x$$

- Underfitting
- High bias



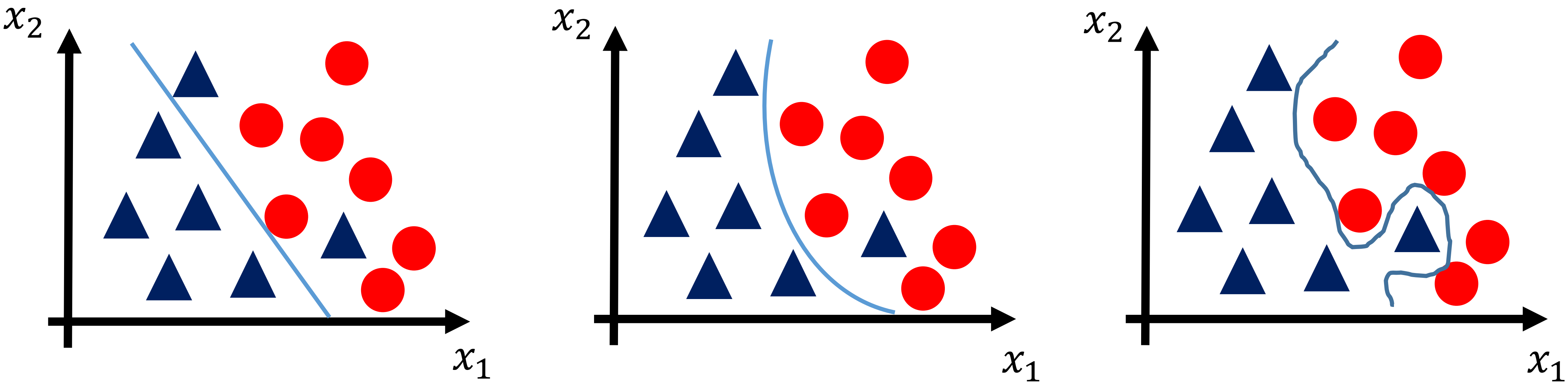
$$y = w_0 + w_1 x + w_2 x^2$$



$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

- Overfitting
- High variance

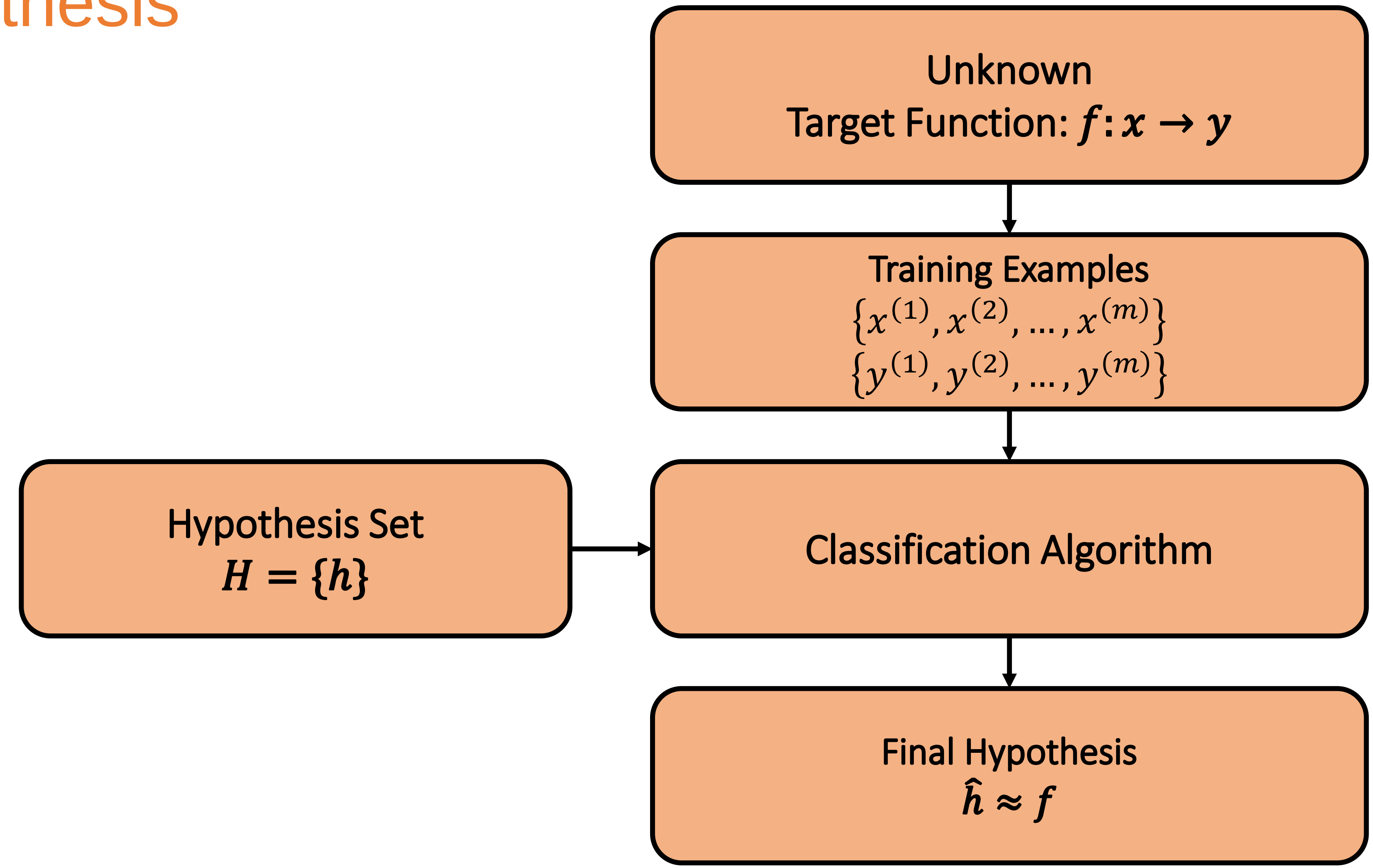
Overfitting – Classification



- Underfitting
- High bias

- Overfitting
- High variance

Hypothesis


Unknown
Target Function: $\boldsymbol{f}: \boldsymbol{x} \rightarrow \boldsymbol{y}$
Training Examples
$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
$\{y^{(1)}, y^{(2)}, \dots, y^{(m)}\}$
Hypothesis Set
$\boldsymbol{H} = \{\boldsymbol{h}\}$
Classification Algorithm
Final Hypothesis
$\hat{\boldsymbol{h}} \approx \boldsymbol{f}$

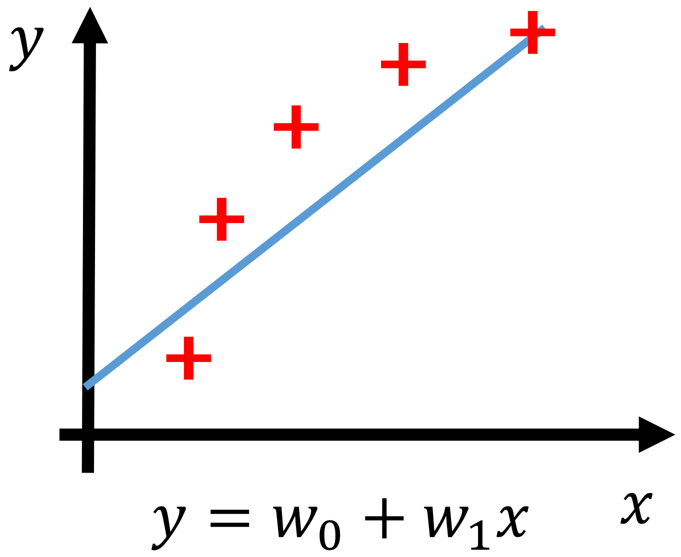
Bias-Variance

Bias-Variance

- The prediction error ($\mathcal{E}$) of a model can be divided into:
  - $\mathcal{E} = \mathcal{E}_b + \mathcal{E}_v + \mathcal{E}_i$
  - $\mathcal{E}_b$: Bias error
  - $\mathcal{E}_v$: Variance error
  - $\mathcal{E}_i$: Irreducible error.
- The irreducible error is the one that we cannot fix whatever model we use because of the way the problem is framed.

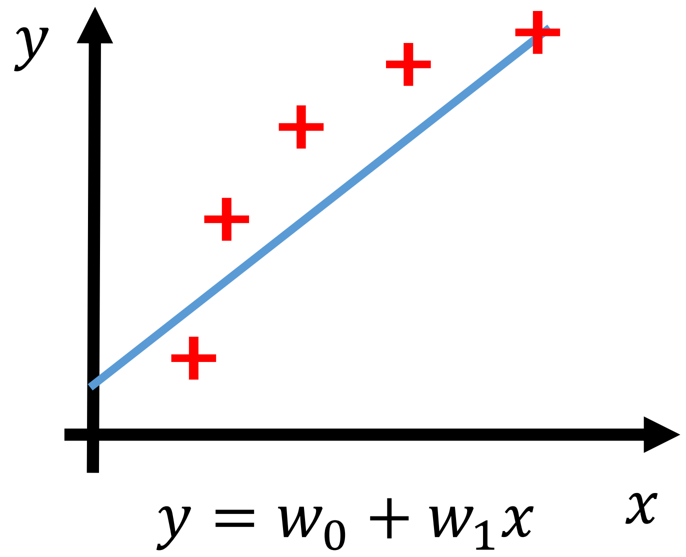
Bias

- The bias error comes from erroneous assumptions in the learning algorithm. Often these assumptions are made to use a simple model.
  - Low bias: suggests good or too complex hypothesis representation.
  - High bias: suggests the need for a more flexible hypothesis representation.
- A high bias may cause the algorithm to miss the relationship between features and the target output and lead to **underfitting**.



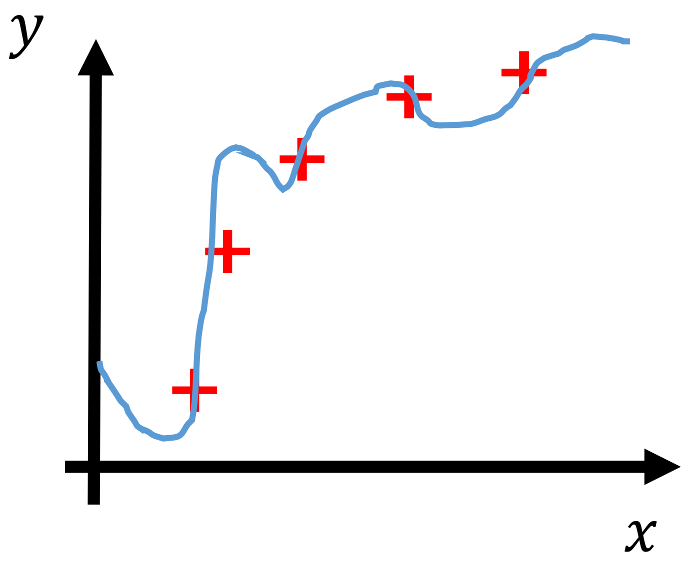
Bias

- Low-bias ML algorithms:
  - Decision Trees,
  - k-Nearest Neighbors,
  - Support Vector Machines.
- High-bias ML algorithms:
  - Linear Regression,
  - Linear Discriminant Analysis,
  - Logistic Regression.



Variance

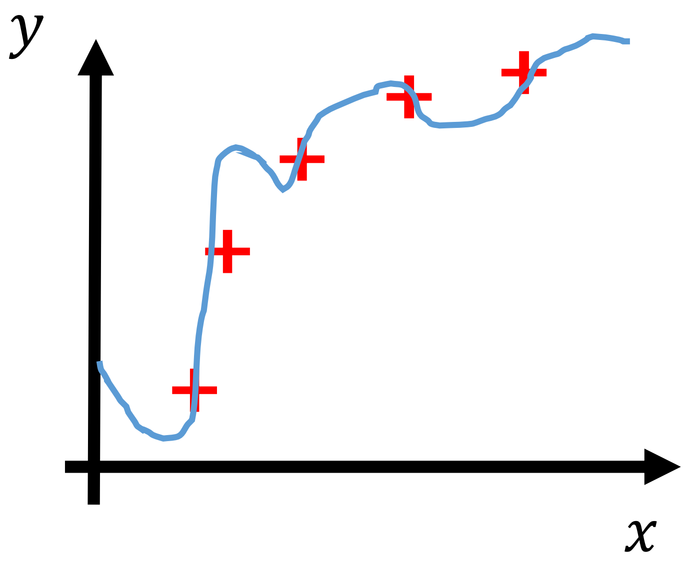
- The variance reflects how much the target function will change if different training data was used.
  - Low variance: suggests that changing the training dataset will lead to small changes to the estimate of the target function.
  - High variance: suggests that changing the training dataset will lead to large changes to the estimate of the target function.
- High variance can cause to model the noise in the training set which will lead to **overfitting**.
- Nonparametric machine learning algorithms have more flexibility and generally a higher variance.



$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

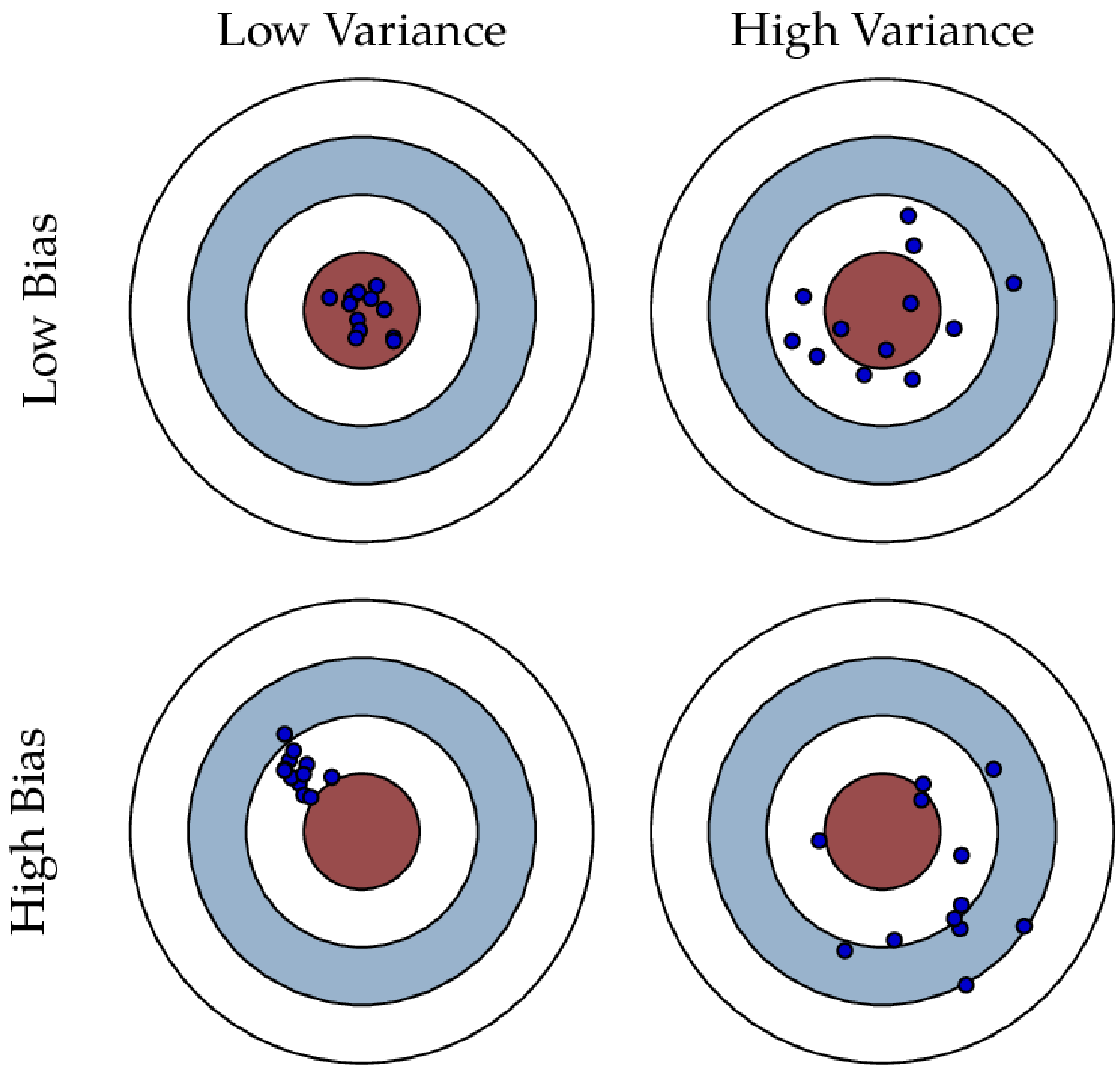
Variance

- Low variance ML algorithms:
  - Linear Regression,
  - Linear Discriminant Analysis,
  - Logistic Regression.
- High variance ML algorithms:
  - Decision Trees,
  - k-Nearest Neighbors,
  - Support Vector Machines.



$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

Bias-Variance



Bias-Variance Tradeoff

- In training a classifier we want a **low bias** and a **low variance**.
- Parametric or linear machine learning algorithms will often have a high bias but a low variance.
- Non-parametric or non-linear machine learning algorithms will often have a low bias but a high variance.
- In training any classifier we will need to find a **tradeoff between bias and variance**. This is not an easy task because:
  - Increasing the bias will lead to a lower variance.
  - Increasing the variance will lead to a lower bias.

Bias-variance tradeoff.

Addressing overfitting

- How can we address overfitting?
  - Visualize and adjust your model
  - But does not very help when we have many features
  - What else can we do?
    - Reduce the number of features (manually or using some algorithm).
    - Increase training data set.
    - Ensemble prediction from final models.
    - **Regularization**: keep all the features but reduce $||w_j||$.

Regularization

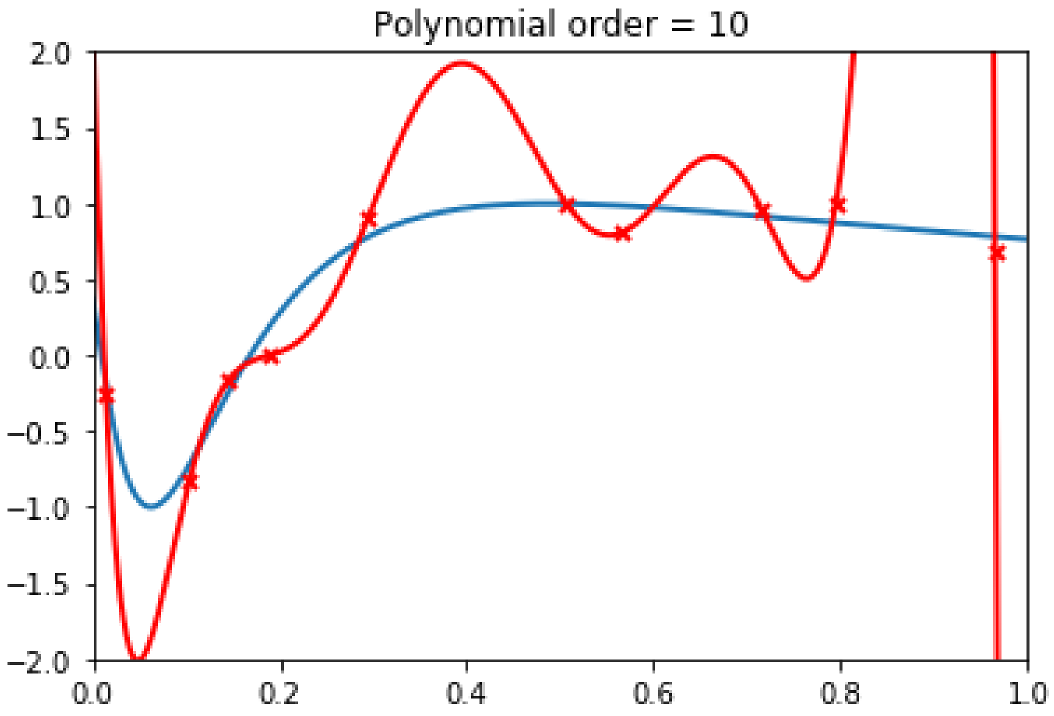
Regularization- General

- We seek to control the magnitude of the w_j
- Small values are preferable because it will lead to a simpler hypothesis representation.
- A simpler hypothesis representation is less prone to overfitting.
- $J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_w\left(x^{(i)}\right) - y^{(i)})^2 + \lambda \sum_{i=1}^{n_x} w_j^2\right]$
  - Blue: the regularization term
  - λ: Regularization parameter. It controls the tradeoff between good fitting and keeping the w_j small i.e. a more simple hypothesis representation.
  - $\lambda \rightarrow 0$: no regularization.
  - $\lambda \rightarrow \infty$: underfitting ($h_w(x) = w_0$).
  - Thus the λ parameter should be chosen carefully.

Regularized Linear Regression

Regularized Linear Regression

- Why regularization? We want to avoid overfitting.
- We saw two ways to find the solution to the linear regression problem:
  - Using gradient descent.
  - Using the normal equation.
- How do we regularize?
- Intuition:
  - We introduce a penalization term $E(w)$
  - $J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - w^T \cdot x^{(i)})^2 + E(w)$
  - We want this term to "push away" the Value of w from the original overfitted optimal value.



Regularized Linear Regression

- Sum-of-square error for regularization (Ridge Regression):
  - $J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - w^T \cdot x^{(i)})^2 + \frac{\lambda}{2} w^T \cdot w$
- Closed form solution (prove it!):
  - $w = (\lambda \cdot I + X^T X)^{-1} X^T y$
- Gradient descent:
  - $w_j := w_j - \alpha \left[\frac{1}{m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m} w_j\right]$
- Assuming a sum-of-square regularization term we obtained a closed form solution.
- In statistics this provides an example of parameters shrinkage method because the weights are dragged to be small.
- What about the more general case where the regularization term is not sum-of-square?

Regularized Linear Regression

- More general expression:
  - $J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - w^T x^{(i)})^2 + \frac{\lambda}{2}\sum_{j=1}^{n_x}|w_j|^q,\ q \in \mathbb{N}$
  - If $q = 2$ this is known as **Ridge Regression**. It makes use of the $L2$ norm.
  - If $q = 1$ this is known as **Lasso Regression**. It makes use of the $L1$ norm.
  - In the case of Lasso, if λ is sufficiently large then some coefficients w are driven to zero.
    - $w_j \coloneqq w_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m} sign(w_j)\right]$
    - $w_j \coloneqq w_j - \lambda\, sign\left(w_j\right) - \cdots$
    - So if $w_j > 0$ then the correction term will drag $w_j \to 0$
    - So if $w_j < 0$ then the correction term will drag $w_j \to 0$.
  - So in practice **Lasso Regression** tends to zeros some coefficients. It does some form of feature selection.

Regularized Logistic Regression

- Graphical interpretation:

Figure 3.4 Plot of the contours of the unregularized error function (blue) along with the constraint region (3.30) for the quadratic regularizer $q = 2$ on the left and the lasso regularizer $q = 1$ on the right, in which the optimum value for the parameter vector $\mathbf{w}$ is denoted by $\mathbf{w}^{\star}$. The lasso gives a sparse solution in which $w_1^{\star} = 0$.

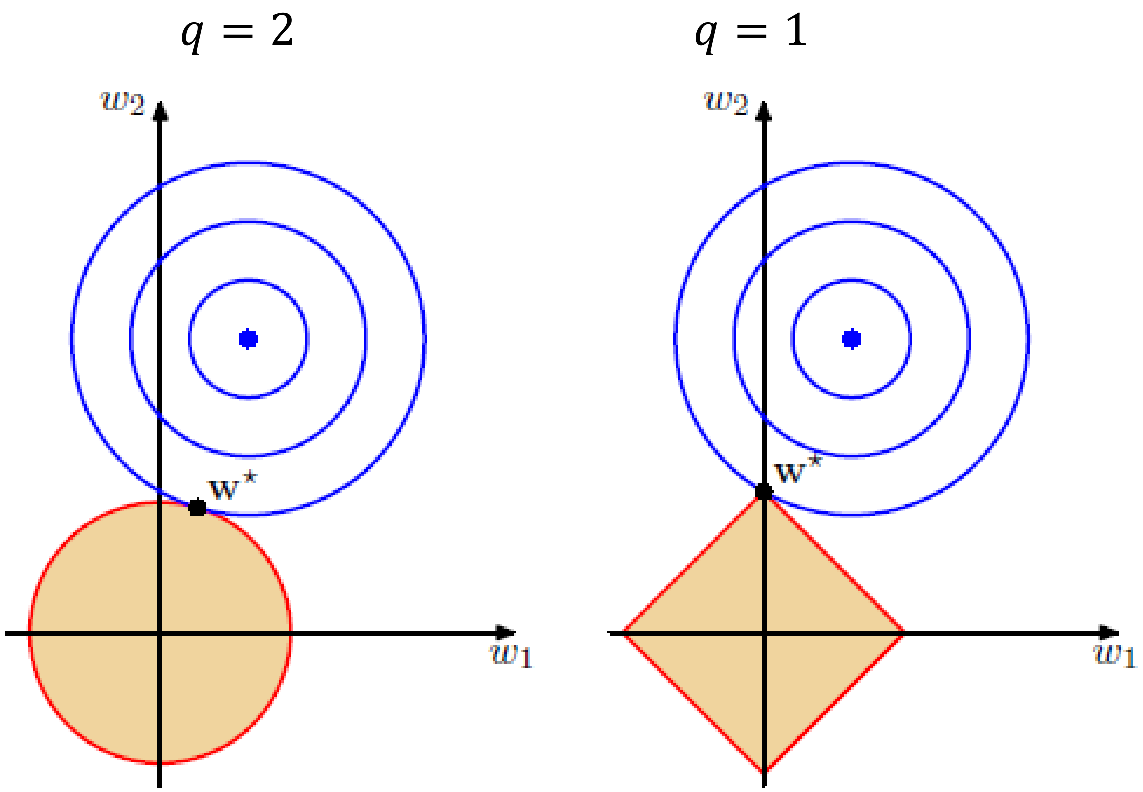


Image: From Bishop's Pattern recognition and machine learning, Figure 3.4

Regularized Logistic Regression

Regularized Logistic Regression

- Cost function for LR:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}log\left(h_w\left(x^{(i)}\right)\right) - (1 - y^{(i)})\log(1 - h_w\left(x^{(i)}\right))\right]$
- If we add the regularization term:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}log\left(h_w\left(x^{(i)}\right)\right) - (1 - y^{(i)})\log(1\right.$

Take Home

- Underfitting and overfitting are not desirable effects and reflect some limitations on our choice made of the hypothesis function. This is related to the **tradeoff** between **bias and variance**.
- **Bias** is the reflection of the hypothesis function complexity.
- **Variance** reflects how the model generalizes to new observations.
- We want a model with low bias and low variance.
- However, when increasing the bias we decrease the variance and when increasing the variance we decrease the bias. So we need to find a **tradeoff**.
- **Regularization**. In particular, **Ridge regression** ($q = 2$), **Lasso regression** ($q = 1$).
- **Lasso** has a nice property of cancelling some weights thus enabling some **sparsity** which is a form of feature selection while keeping the cost function **convex**.

References

[1] Machine Learning Mastery:
https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/
[2] Pattern recognition and Machine Learning. Christopher M. Bishop. 2006 Springer Science.
[3] Coursera, Andrew Ng. Regularization.