Introduction to Machine Learning

Introduction to Artificial Intelligence with Mathematics
Lecture Notes

Ganguk Hwang

Department of Mathematical Sciences
KAIST

What is Machine Learning?

Machine Learning with a data set

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$$

- $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$: input vectors
- $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n$: target vectors
- the data set $\mathcal{D}$, called a training set, is used to tune the parameters of a (machine learning) model.
- The objective of a machine learning algorithm is to find $\mathbf{y}(\mathbf{x})$ with $\mathcal{D}$.

General Form in Machine Learning

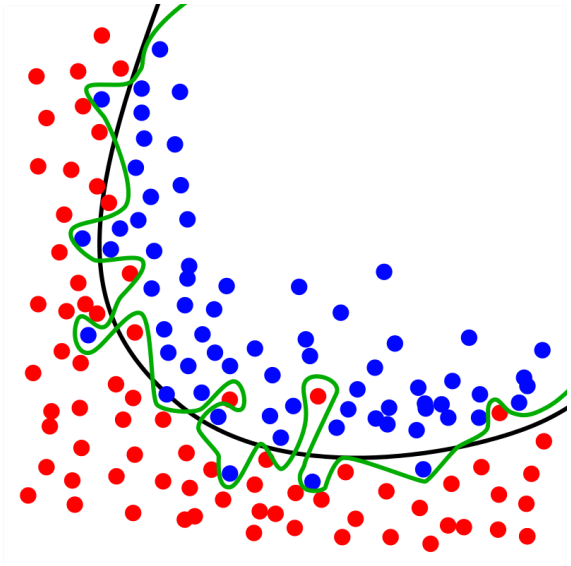
- $\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\epsilon}$
- $\mathbf{y}$ is the observation.
- f is an unknown function of $\mathbf{x} = (x_1, x_2, \cdots, x_p)$.
- ϵ is a random error and independent of $\mathbf{x}$, and zero mean.
- We want to find an approach to estimate f.
  - parametric method
  - nonparametric method

Parametric method

- For instance, assume that f is linear in $\mathbf{x}$, $f(\mathbf{x}) = \sum_{i=0}^{p} \beta_i x_i$ with $x_0 = 1$.
- The objective is to estimate the parameters $\beta_0, \beta_1, \ldots, \beta_p$.
- So the problem of estimating f is reduced to estimating a set of parameters.
- It may not fit the data well due to the model limitation.

Nonparametric method

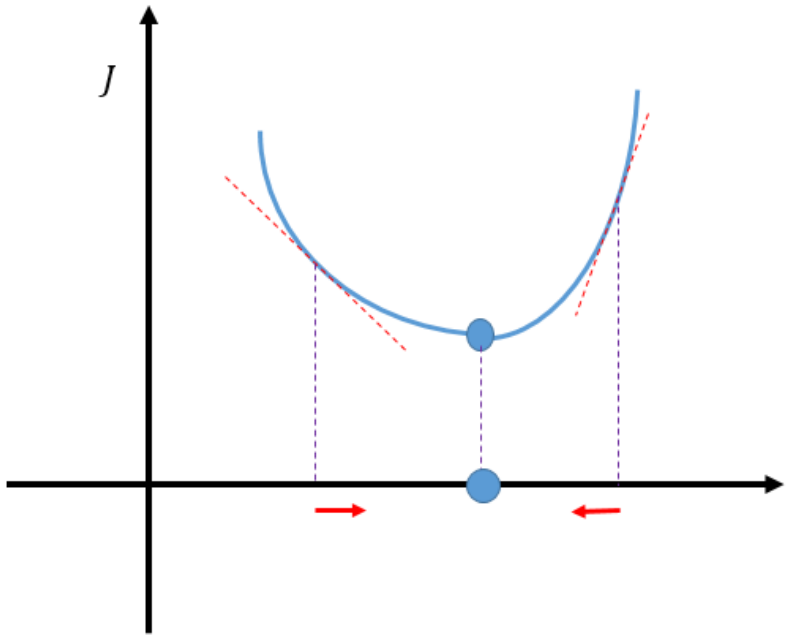
- No explicit assumption is made for f.
- It has a great potential to fit the data well.
- It needs a large number of data to learn and overfitting can occur.



source: Overfitting, Wikipedia

In machine learning we are trying to minimize the error J in the estimation of f. One good way is to use the Gradient Descent method.

Recall that the gradient of the error J, ∇J, points in the direction along which J is increasing the fastest.
So, if we wish to move in a direction in which J decreases the fastest, we should move in the direction $-\nabla J$.

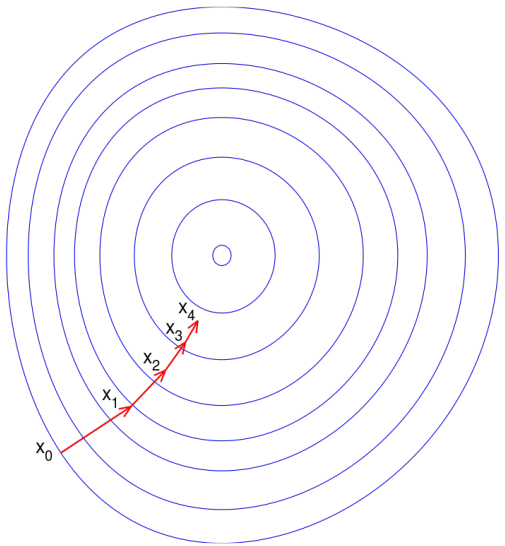


Gradient Descent Method

For instance, assume that $J = J(\beta_0, \beta_1)$. Then, two parameters β_0 and β_1 can be updated as

$$\beta_0^{(t+1)} = \beta_0^{(t)} - \alpha \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_0}$$

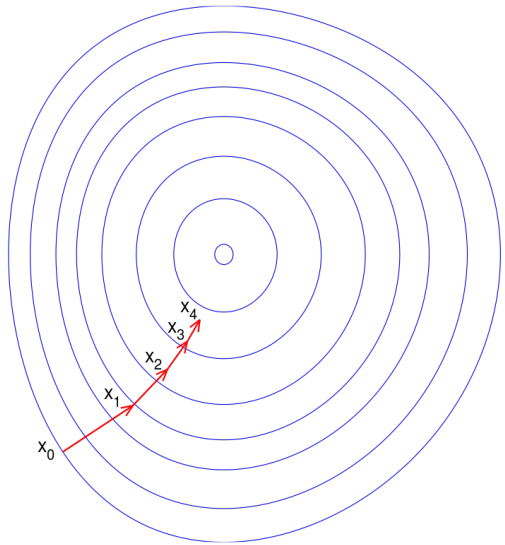
$$\beta_1^{(t+1)} = \beta_1^{(t)} - \alpha \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_1}$$



source: Gradient Descent, Wikipedia

Gradient Descent Method

- First-order optimization algorithm
- We can find the minimum of a convex function by starting at an arbitrary point and repeatedly take steps in the downward direction (the negative direction of the gradient).
- After several iterations, we will eventually reach the minimum for which we have the best fit of f.



source: Gradient Descent, Wikipedia

Learning rate

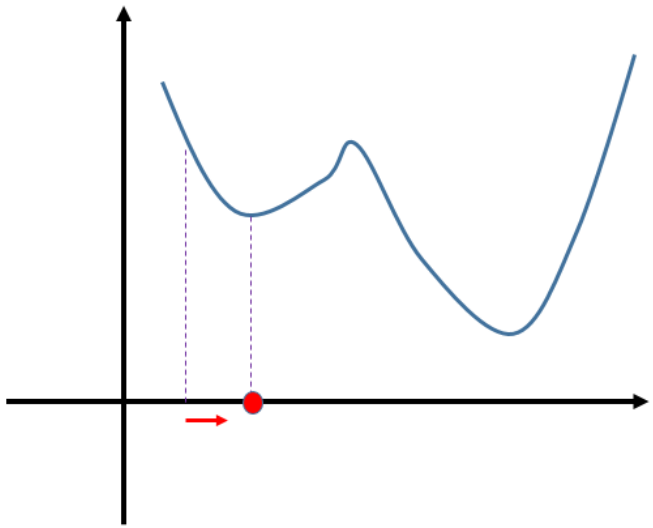
In the steps we usually use the learning rate α which determines the size of the steps we take in the downward direction.

$$\beta_0^{(t+1)} = \beta_0^{(t)} - \alpha \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_0}$$

$$\beta_1^{(t+1)} = \beta_1^{(t)} - \alpha \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_1}$$

Local Optima for nonconvex functions

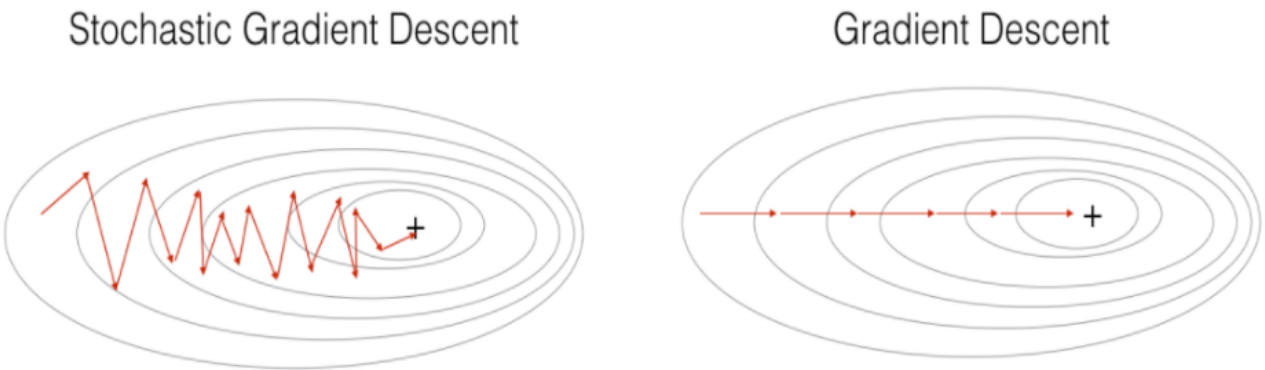
For nonconvex functions, we can reach a local optimum. So we use different starting points and get the best one among local optima.



Another popular method is the stochastic gradient method.

Mini-Batch/Stochastic Gradient Method

- Instead of taking a step using the entire training set, we sample a small batch of training data at random to determine our next step.
- Computationally more efficient and may lead to faster convergence.

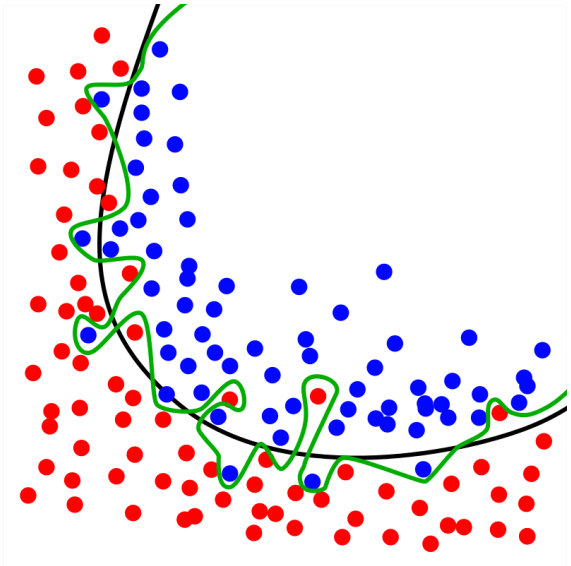


source: https://engmrk.com/mini-batch-gd/

Learning and Validation of The Model

- training set: a dataset used for learning to fit the parameters
- validation set: a dataset used to select the best model
- test set: a dataset that is independent of the training dataset, but that follows the same probability distribution as the training dataset.

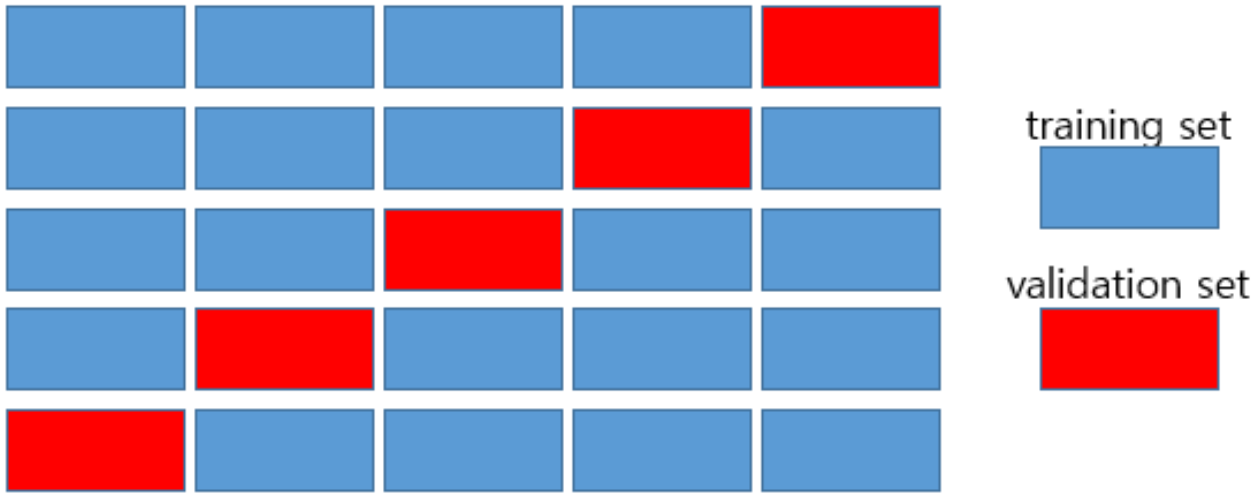
  If a model fits to the training dataset and also fits the test dataset well, minimal overfitting has taken place. A better fitting of the training dataset as opposed to the test dataset usually points to overfitting.



source: Overfitting, Wikipedia

Validation: Cross Validation

A dataset can be repeatedly split into a training dataset and a validation dataset.



Evaluation Method

The performance of machine learning algorithms can be evaluated in terms of

- Accuracy
- Confusion Matrix
- etc.

Accuracy: MSE

A good example is the Mean Square Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$$

If the MSE is small, the prediction will be very close to the true response.

Accuracy: Misclassification Rate

Misclassification Rate (MR) is given by

$$\text{MR} = \frac{\text{number of incorrect predictions}}{\text{total number of predictions}}$$

Confusion Matrix

In a binary classification (positive or negative),

		true label	
		positive	negative
prediction	positive	True Positive (TP)	False Positive (FP)
	negative	False Negative (FN)	True Negative (TN)

Error Rate

The Error Rate (ERR) is defined by the number of all incorrect predictions divided by the total number of the data. That is,

$$ERR = \frac{FP + FN}{TP + TN + FN + FP} = \frac{FP + FN}{P + N}.$$

Accuracy

The Accuracy (A) is defined by the number of all correct predictions divided by the total number of the data. From its definition, it is obvious that by $A = 1 - ERR$.

$$A = \frac{TP + TN}{P + N}$$

True Positive Rate (or Recall, Sensitivity)

The True Positive Rate (TPR) is defined by the number of correct positive predictions divided by the total number of positive data.

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P}$$

Precision (or Positive Predictive Value)

The Precision (PREC) is defined by the number of correct positive predictions divided by the total number of positive predictions.

$$PREC = \frac{TP}{TP + FP}$$

True Negative Rate (or Specificity)

The True Negative Rate (TNR) is defined by the number of correct negative predictions divided by the total number of negative data.

$$TNR = \frac{TN}{TN + FP} = \frac{TN}{N}$$

False Positive Rate

The False positive rate (FPR) is defined by the number of incorrect positive predictions divided by the total number of negative data. From its definition, $FPR = 1 - TNR$.

$$FPR = \frac{FP}{TN + FP} = 1 - TNR$$

Note that

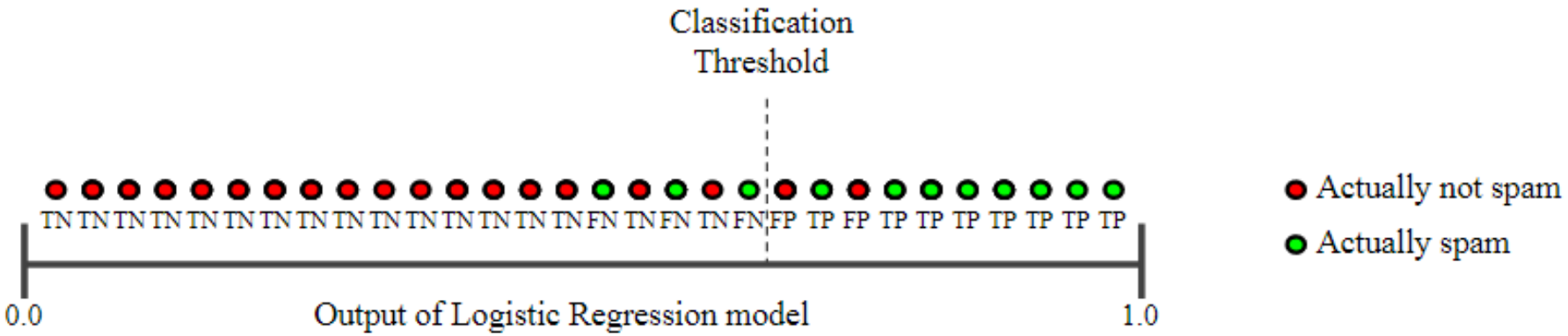
- Precision: fraction of data that are truely positive in the set of data that are predicted as positive
- Recall: fraction of data that are predicted as positive in the set of data that are truely positive.

We have a trade-off between precision and recall.

- If we use a wider net (using a relaxed threshold for classification), we can detect more positive cases (i.e., higher recall), but we have more false alarms (i.e., lower precision).
- For instance, if we classify everything as positive, we have 1.0 recall but a bad precision because there are many FPs.
- If we adjust the threshold more strict so as to get a good precision, we classify more truely positive data as negative, resulting in lower recall.

Spam or not? $PRE = \frac{8}{8+2} = 0.8, REC = \frac{8}{8+3} = 0.73$

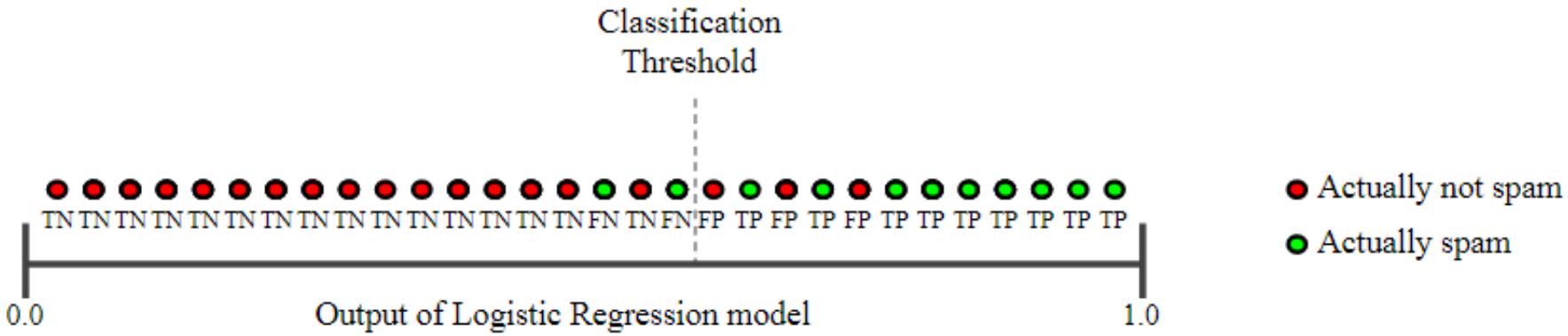
		true label	
		positive	negative
prediction	positive	True Positive (TP): 8	False Positive (FP): 2
	negative	False Negative (FN): 3	True Negative (TN): 17



source: https://developers.google.com/machine-learning/crash-course/classification/video-lecture

Spam or not? $PRE = \frac{9}{9+3} = 0.75, REC = \frac{9}{9+2} = 0.82$

		true label	
		positive	negative
prediction	positive	True Positive (TP): 9	False Positive (FP): 3
	negative	False Negative (FN): 2	True Negative (TN): 16



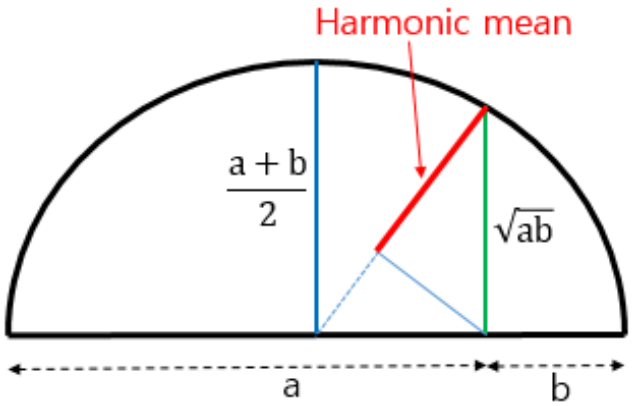
source: https://developers.google.com/machine-learning/crash-course/classification/video-lecture

F_1 Score

The F_1 Score combines precision and recall into a single metric, and is defined by the harmonic mean of precision (PRE) and recall (REC). That is,

$$F_1 = \frac{2 \times PRE \times REC}{PRE + REC} = \frac{2 \times TP}{(2 \times TP + FP + FN)}$$

Note that the harmonic mean of two numbers tends to be closer to the smaller of two numbers, so F_1 is high only when both precision and recall are high.



Generalization of The Model

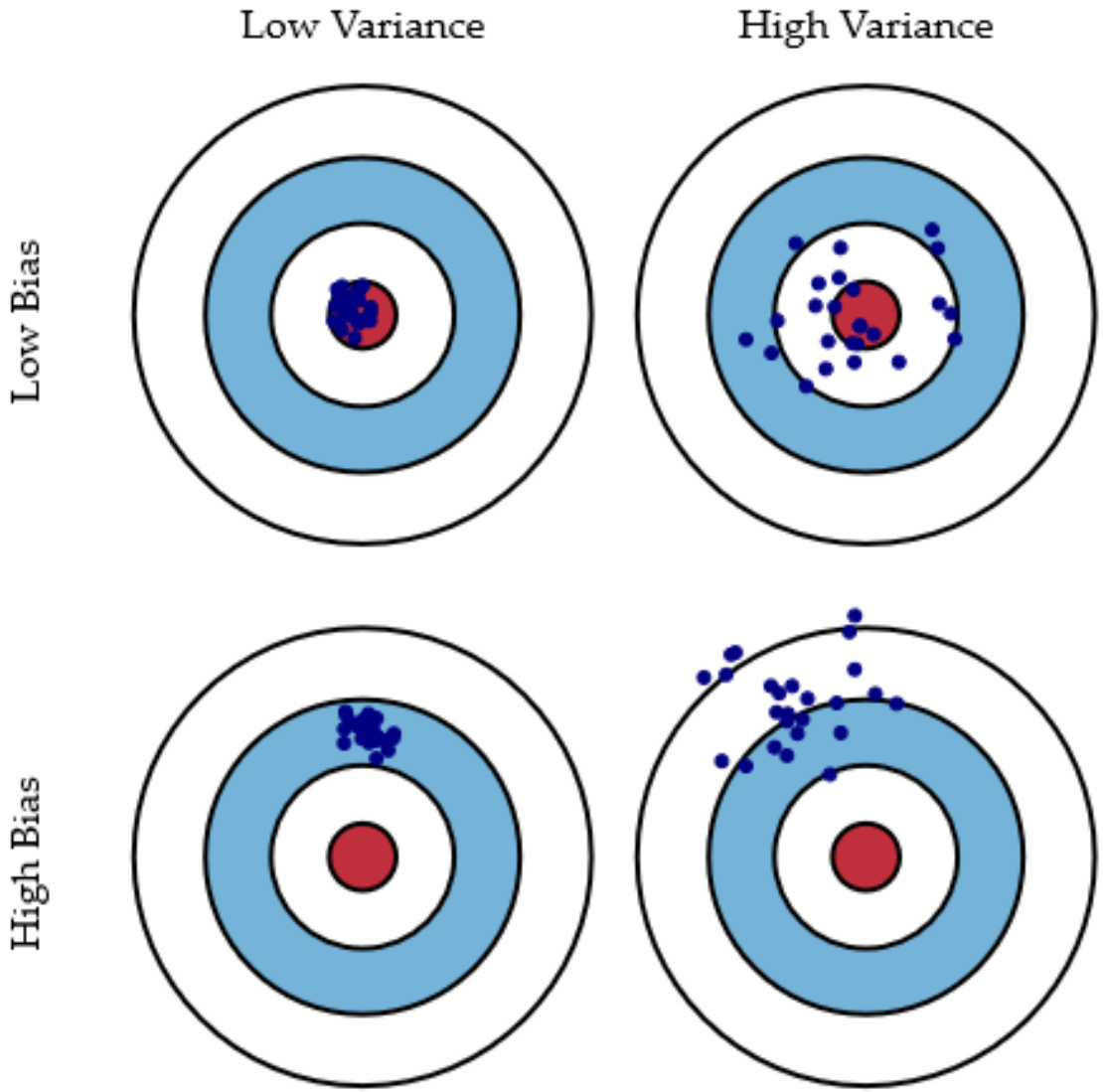
- Generalization refers to how well the models learned can be applied to other datasets.
- A good machine learning model allows us to make good predictions on new datasets.
- Related issues: overfitting and underfitting (regularization and more features)

Bias and Variance Trade-off

Observe that

$$\begin{aligned}
&E_{\mathcal{D}}[(y(\mathbf{x};\mathcal{D}) - f(\mathbf{x}))^2] \\
&\quad = E_{\mathcal{D}}[(y(\mathbf{x};\mathcal{D}) - E_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] + E_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - f(\mathbf{x}))^2] \\
&\quad = E_{\mathcal{D}}[(y(\mathbf{x};\mathcal{D}) - E_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})])^2] + E_{\mathcal{D}}[(E_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - f(\mathbf{x}))^2].
\end{aligned}$$

The first term $E_{\mathcal{D}}[(y(\mathbf{x};\mathcal{D}) - E_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})])^2]$ is the variance and the second term $E_{\mathcal{D}}[(E_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - f(\mathbf{x}))^2]$ is the squared bias. So the expected squared error can be decomposed into two terms - the variance and the squared bias.



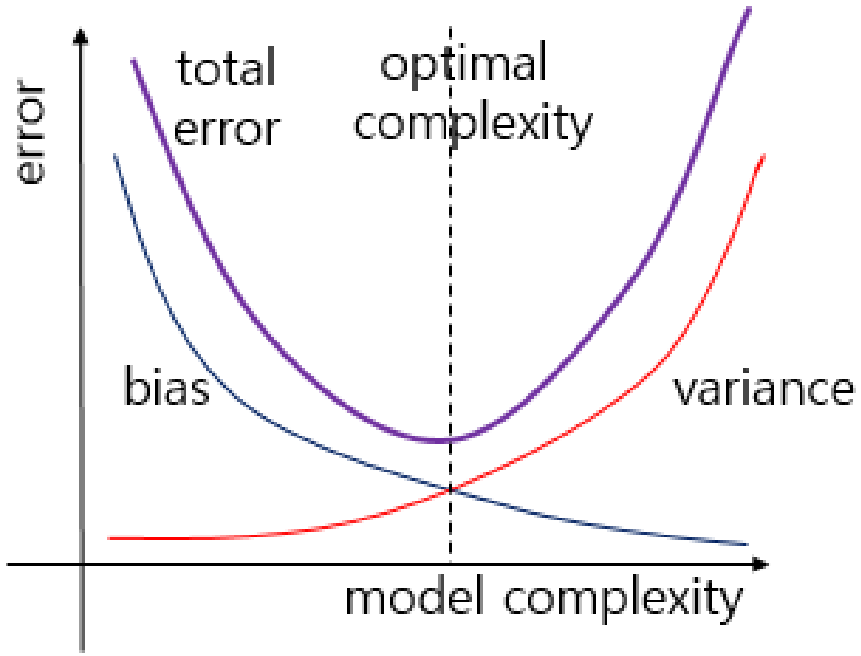
source: http://scott.fortmann-roe.com/docs/BiasVariance.html

Bias

- Bias is due to the simplifying assumption made by a model to make the target function easier to learn.
- Since a model with high bias does not fit well even for a given training set, it usually results in underfitting.

Variance

- Variance is the amount that the estimate of the target function will change if we use a different training data.
- In general, the more complex (flexible) the model is, the more variance it has.
- A model with high variance is usually an overfitted model.


error
total
error
optimal
complexity
bias
variance
model complexity

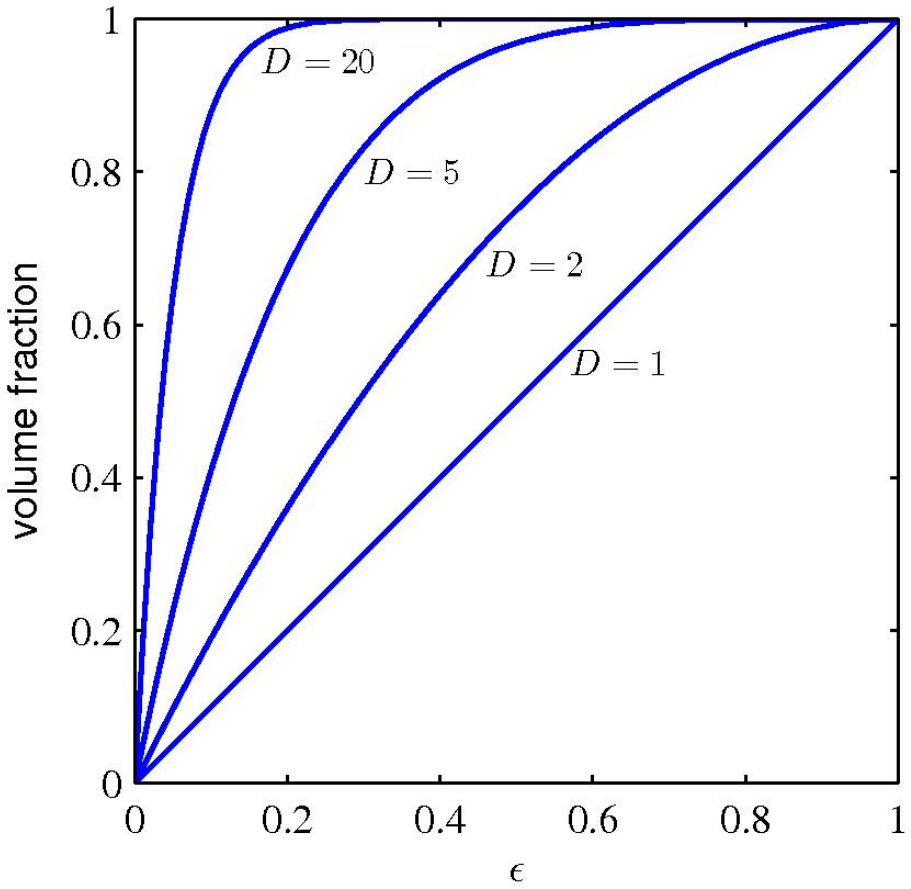
The Curse of Dimensionality

- In polynomal fitting: when the input vector $\mathbf{x}$ is in a D-dimensional space, the number of unknowns $\mathbf{w}$ becomes D^M.
- Not all the intuitions developed in spaces of low dimensionality will generalize to spaces of high dimensionality.

- Consider a sphere of radius $r = 1$ in a D-dimensional space.
- What is the fraction of the volume of the sphere that lies between radius $r = 1 - \epsilon$ and $r = 1$?
- Note that the volume of a sphere of radius r in a D-dimensional space is $V_D(r) = K_D r^D$. So the required fraction is given by

$$\frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = 1 - (1-\epsilon)^D$$

- In spaces of high dimensionality, most of the volume of a sphere is concentrated in a thin shell near the surface!
- We need an exponentially large quantity of training data in order to ensure that there exist no empty regions.



source: Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer 2006.

Three Categories in Machine Learning

- supervised learning: input/target vectors in the training set
- unsupervised learning: no target vectors in the training set
- reinforcement learning: actions and rewards

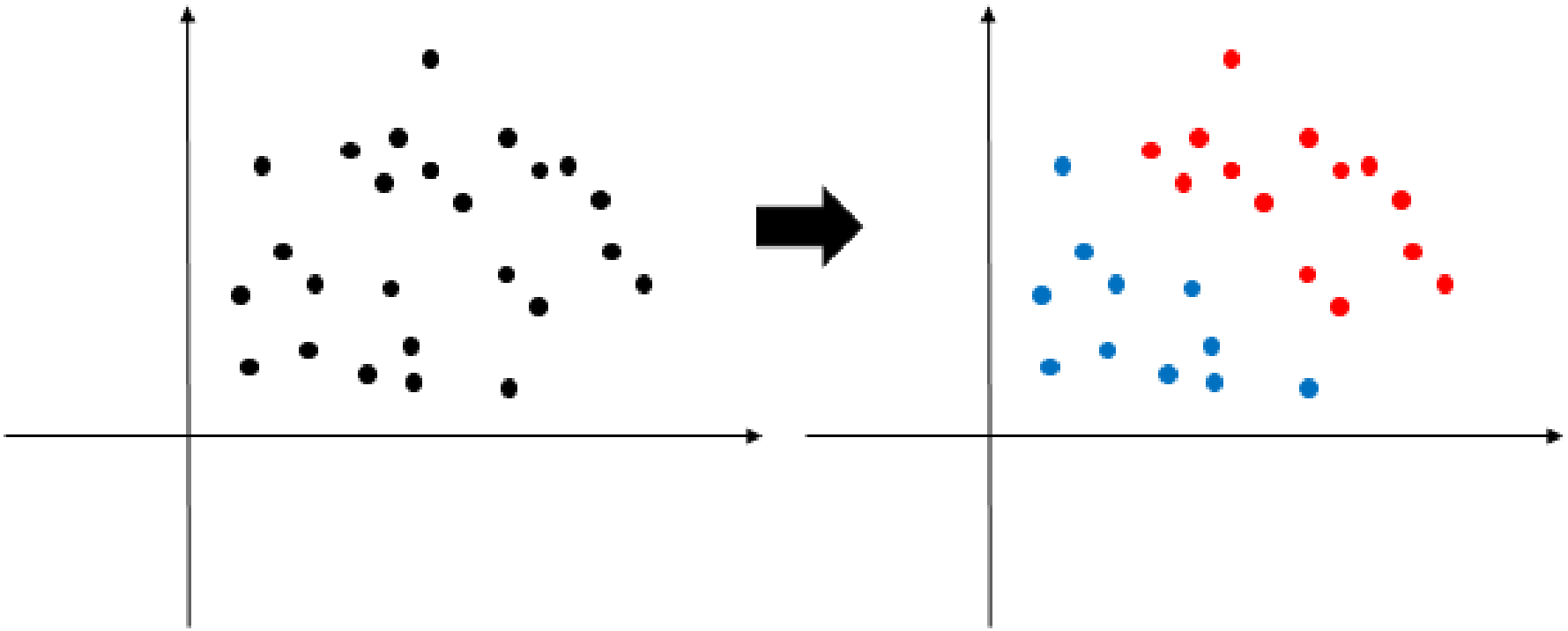
Supervised Learning

- input vectors $\mathbf{x}$: information on pixels
- target value $y(\mathbf{x})$: cat or dog

- examples: regression, classification

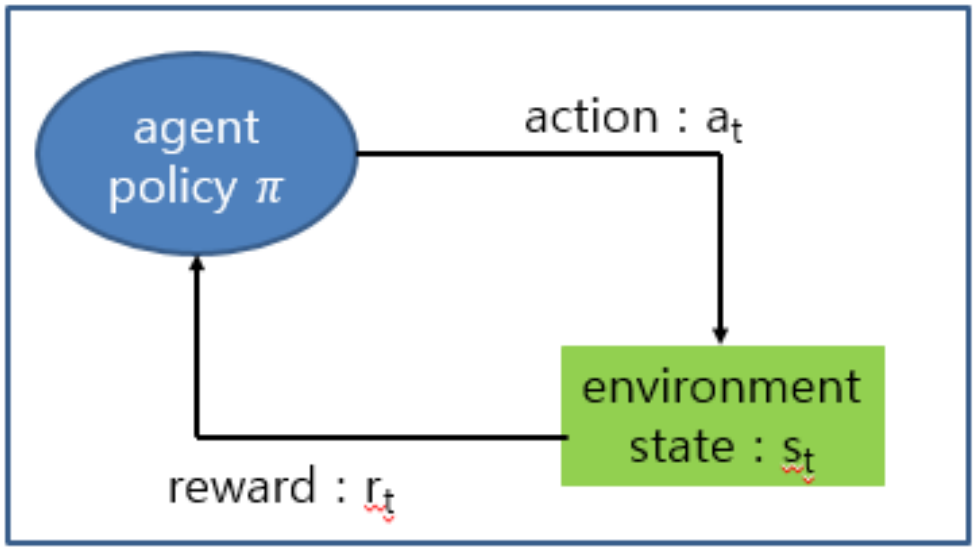
Unsupervised Learning

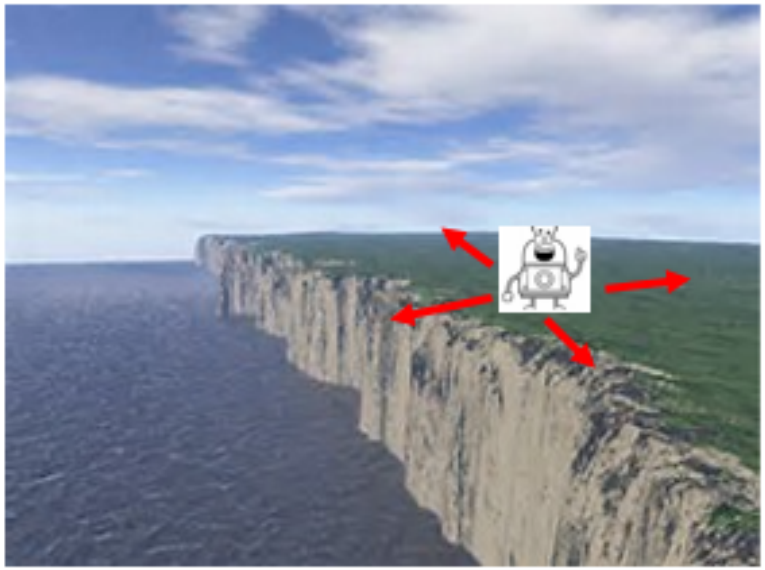
- Unsupervised learning is a self-organized learning from data that has not been labeled, classified, or categorized.



Reinforcement Learning

- Reinforcement learning (RL) is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.
- The environment is usually modeled by Markov Decision Process (MDP).

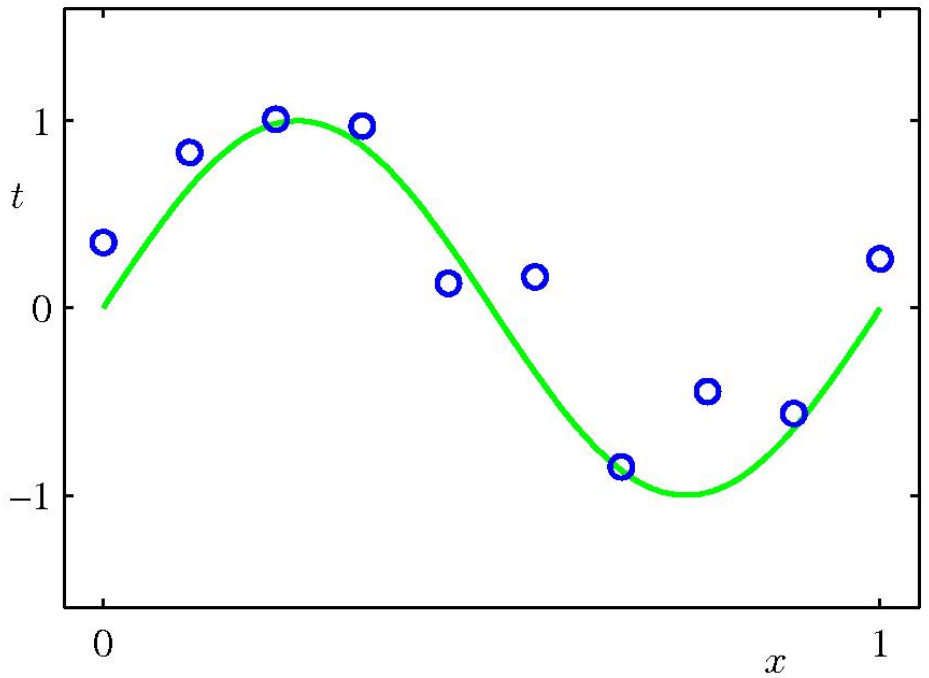




Machine Learning Example: Polynomial Curve Fitting

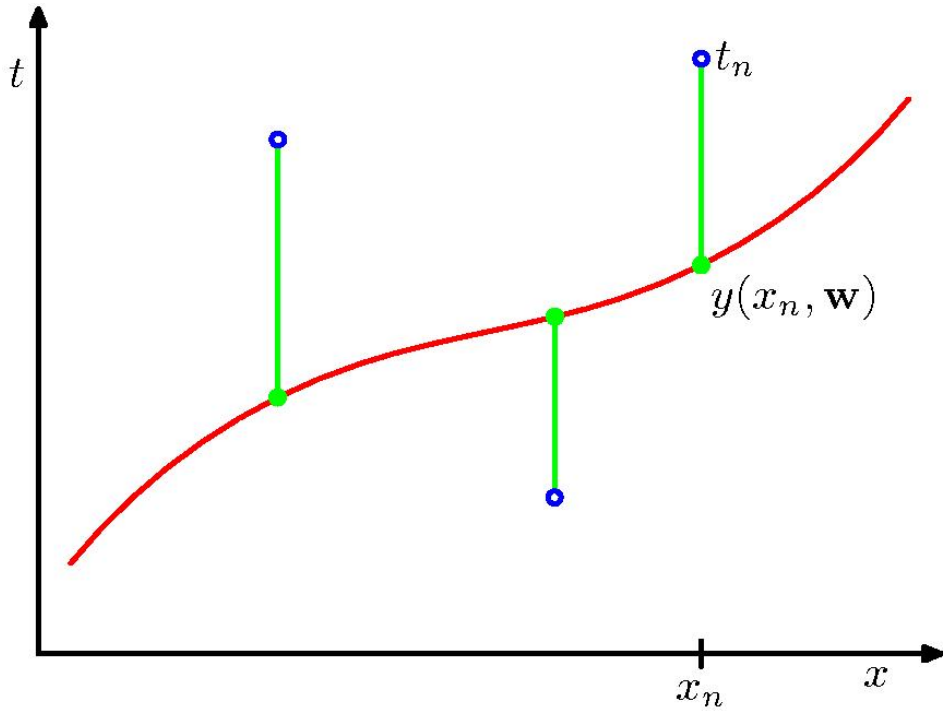
Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer 2006.

- Training set: $\{(x_1, t_1), (x_2, t_2), \cdots, (x_n, t_n)\}$
- Goal: predict the target $\hat{t}$ for a new input value $\hat{x}$



- For a given training set, consider a polynomial with coefficients $\mathbf{w} = (w_0, w_1, \cdots, w_M)$ $y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j x^j$.
- The objective is to minimize the sum of squared errors

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (y(x_i, \mathbf{w}) - t_i)^2$$



More Questions?

- Model Selection: How to choose M?
- Regularization: Adding a penalty term, e.g.,

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y(x_i, \mathbf{w}) - t_i)^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$

Probability Theory and Decision Theory

- Probability theory allows to express the uncertaity of the target value.
- Decision theory allows to make optimal predictions.

Questions in Machine Learning

- Is this A or B?: Classification algorithms
- Is this weird?: Anomaly detection algorithms
- How much or How many?: Regression algorithms
- How is this organized?: Clustering algorithms, Dimensionality reduction
- What should I do next?: Reinforcement learning algorithms

source: Brandon Rohrer's breakdown of the "5 questions data science answers"

Regression

- Predicting continuous values
- Drug response, Stock prices, Housing prices.
- Examples: Linear/nonlinear Regression, etc.

Classification

- Identifying the category the data belongs to.
- Spam detection, Image recognition, etc.
- Examples: SVM, nearest neighbors, random forest, Logistic Regression

Two Types of Classifiers

Consider a pair $(\mathbf{X}, \mathbf{Y})$ with an input $\mathbf{X}$ and its label $\mathbf{Y}$.

- Generative models vs. Discriminative models
- Generative models
  - Assume some function forms for $P(\mathbf{Y})$ and $P(\mathbf{X}|\mathbf{Y})$.
  - Estimate the parameters of $P(\mathbf{Y})$ and $P(\mathbf{X}|\mathbf{Y})$ from the training data.
  - Compute $P(\mathbf{Y}|\mathbf{X})$ by Bayes Theorem.
- Discriminative models
  - Assume some function form for $P(\mathbf{Y}|\mathbf{X})$.
  - Estimate the parameters of $P(\mathbf{Y}|\mathbf{X})$ from the training data.

Clustering

- Grouping of similar data
- Grouping customers and experiment outcomes, etc.
- Examples: k-Means

Dimensionality Reduction

- Reducing the dimension or the number of random variables to consider.
- Visualization, Increased efficiency, etc.
- Examples: PCA, Singular Value Decomposition

References:

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer 2006.
2. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning, Springer, 2nd ed., 2008.
3. Richard Duda, Peter Hart and David Stork, Pattern Classification, 2nd ed. John Wiley & Sons, 2001.
4. Tom Mitchell, Machine Learning. McGraw-Hill, 1997.
5. etc.
6. Online Courses: Andrew Ng: http://ml-class.org/