Introduction to Machine Learning

Thomas Lux

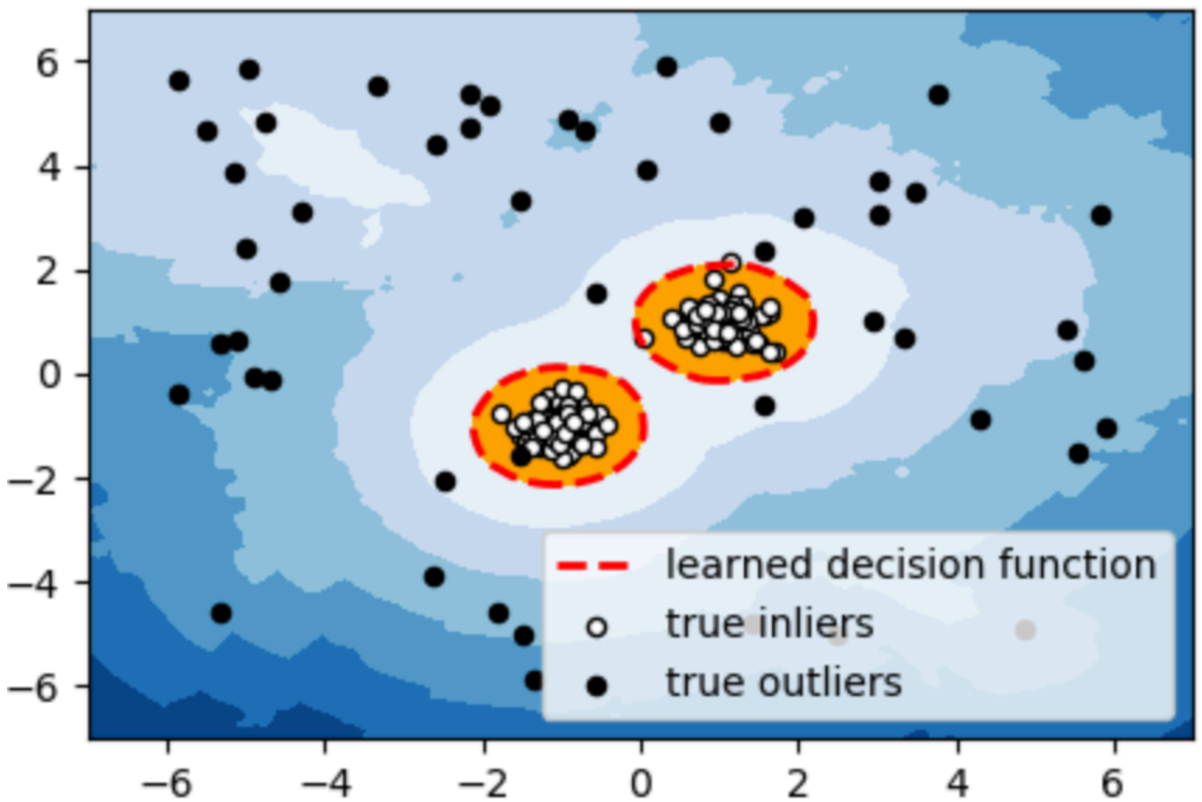
COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH

Machine Learning

Unsupervised *vs.* **Supervised**

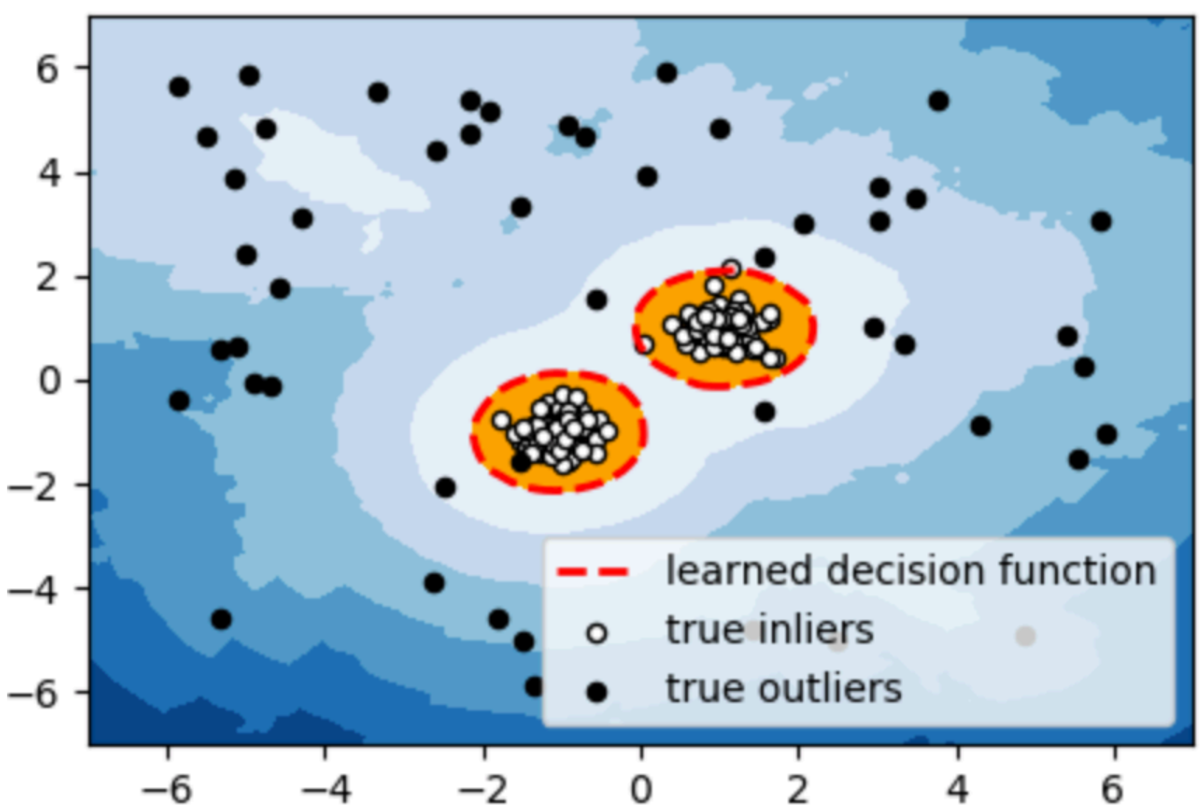
Machine Learning

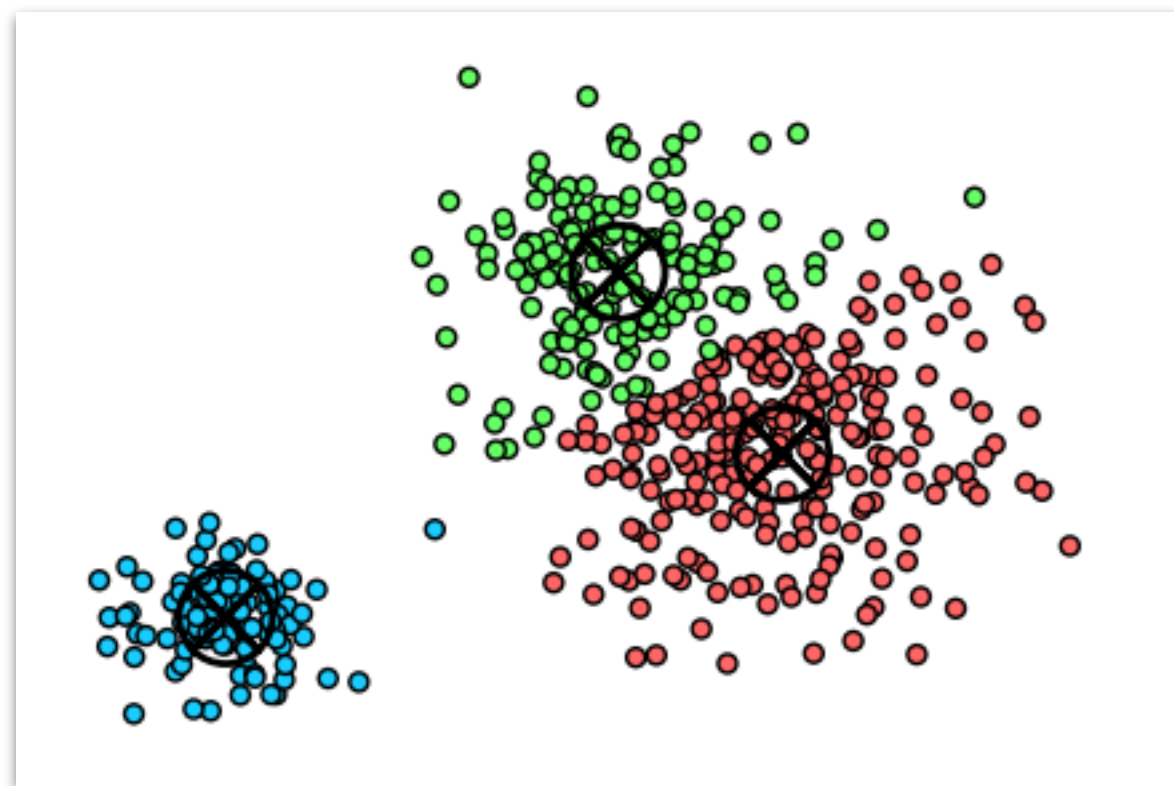
Unsupervised *vs.* Supervised



Machine Learning

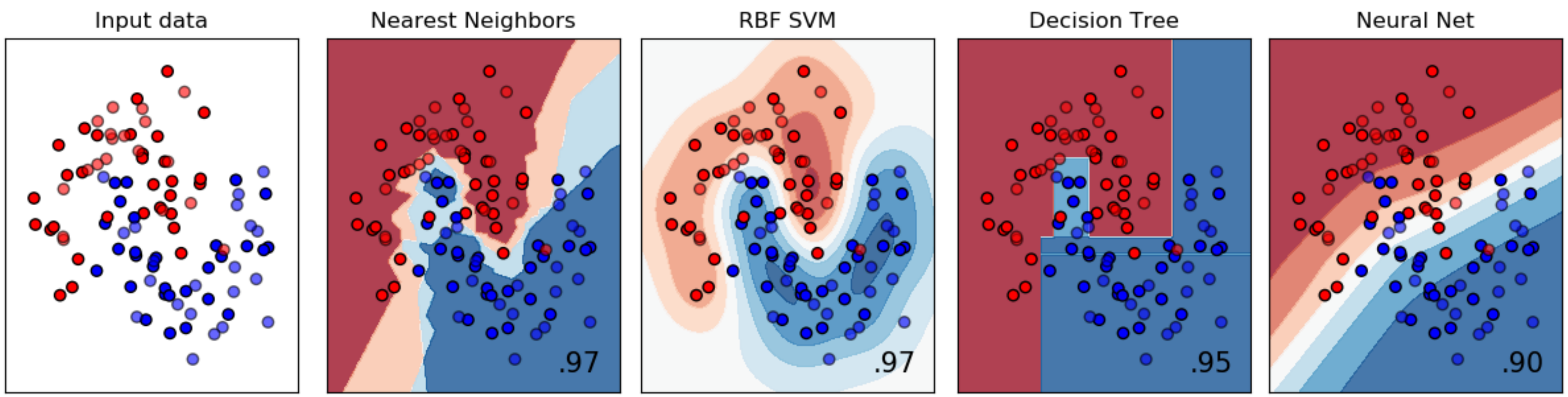
Unsupervised *vs.* Supervised





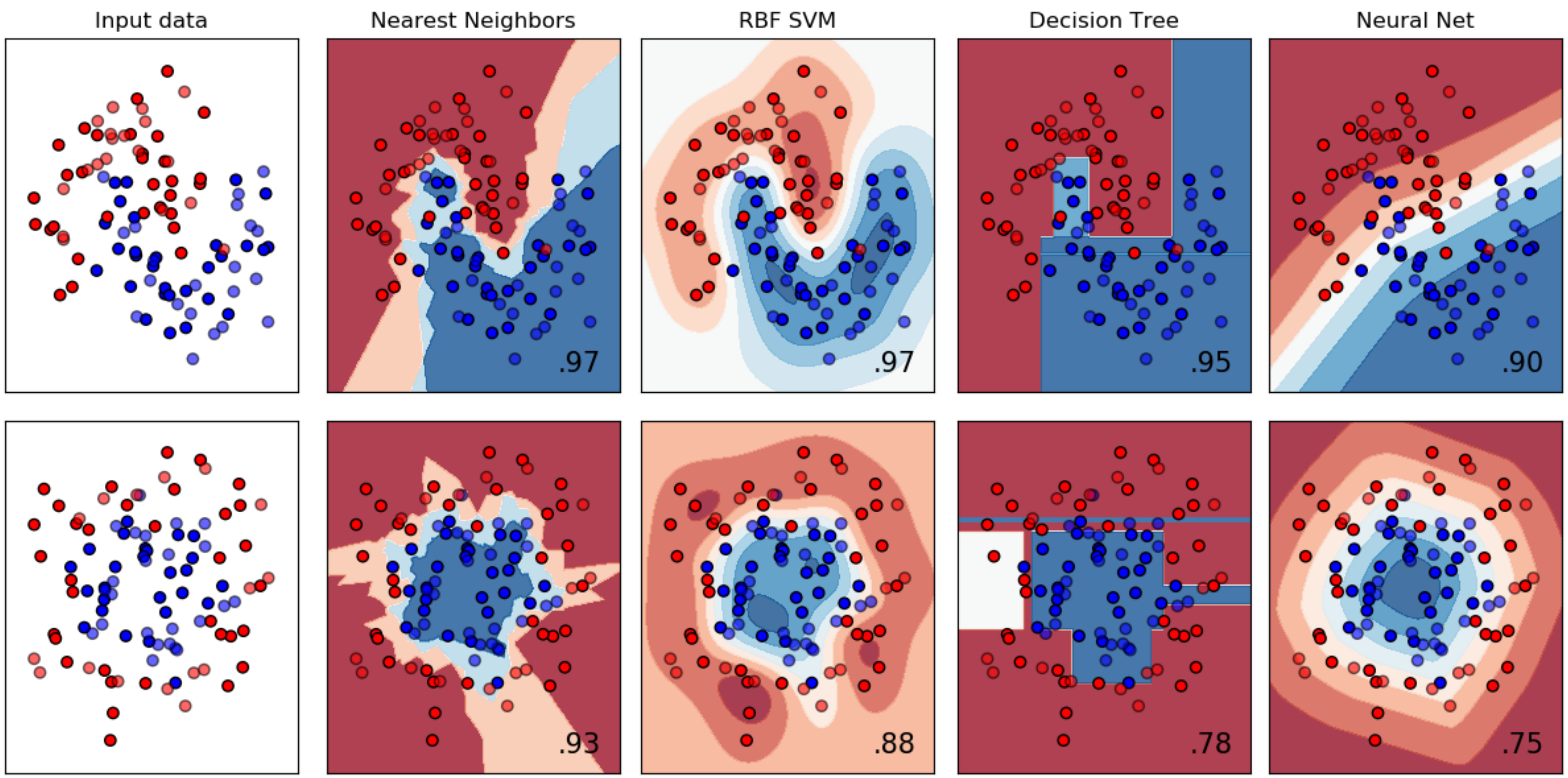
Machine Learning

Unsupervised *vs.* Supervised



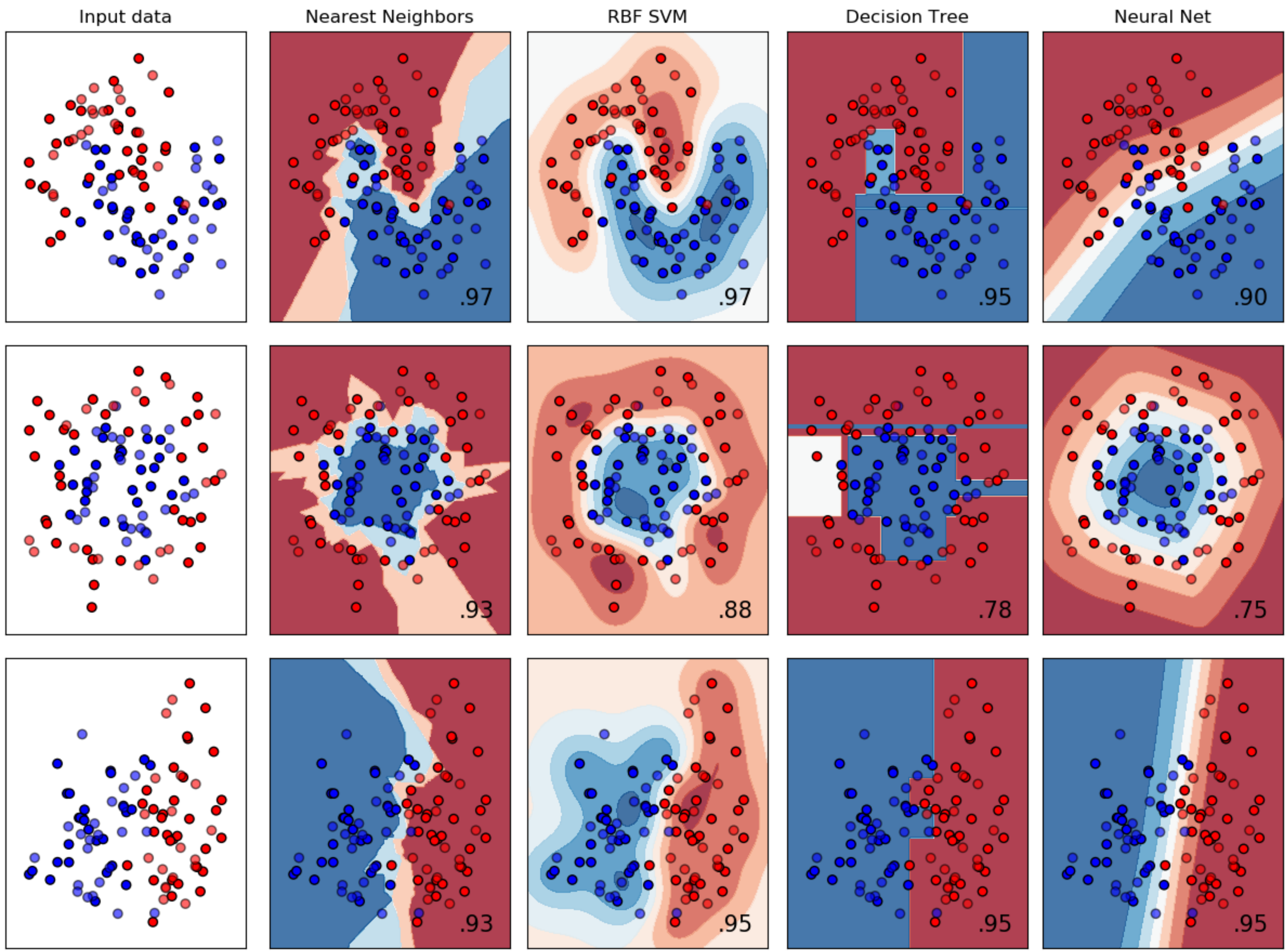
Machine Learning

Unsupervised *vs.* Supervised



Machine Learning

Unsupervised *vs.* Supervised



Unsupervised: Clustering

Unsupervised: Clustering

Apriori Algorithm

Find the most frequently occurring combinations of values in data.

Transactions

0: $\{a, d, e\}$
1: $\{b, c, d\}$
2: $\{a, c, e\}$
3: $\{a, c, d, e\}$
4: $\{a, e\}$
5: $\{a, c, d\}$
6: $\{b, c\}$
7: $\{a, c, d, e\}$
8: $\{b, c, e\}$
9: $\{a, d, e\}$

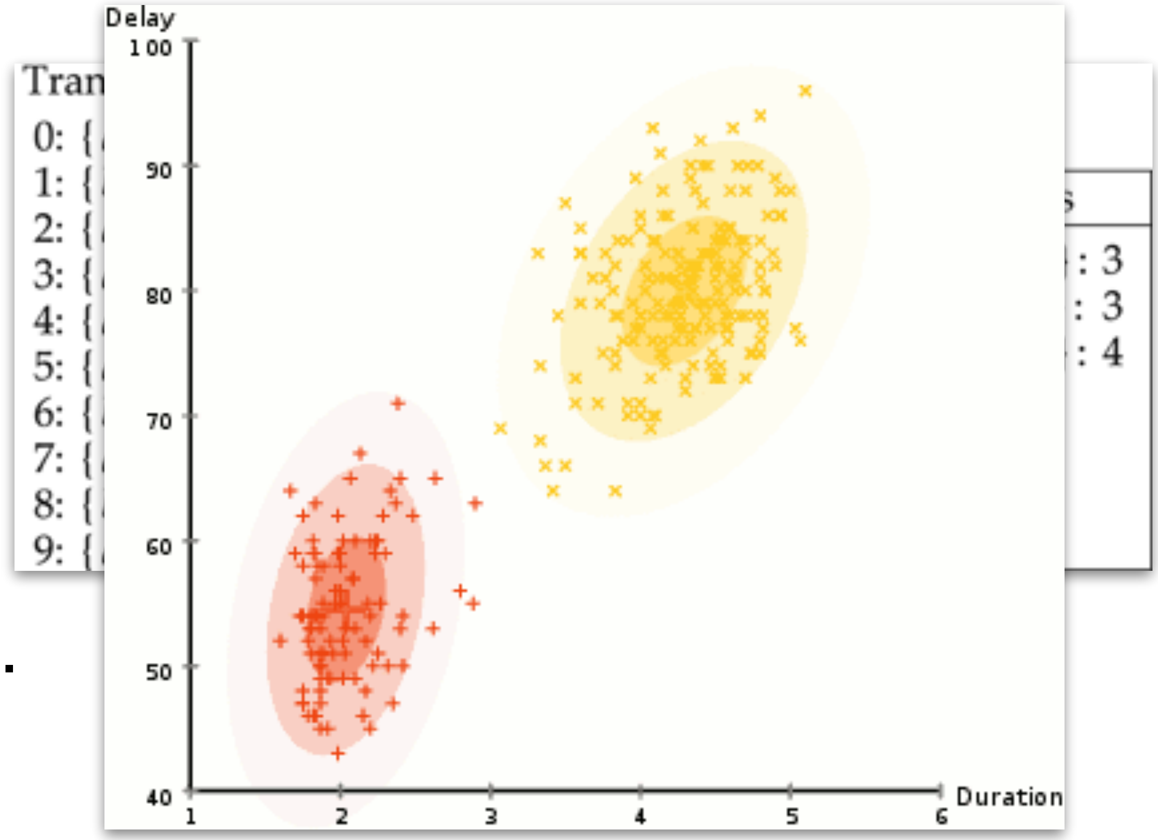
Frequent item sets (with support)
(minimum support: $s_{\min} = 3$)

0 items	1 item	2 items	3 items
$\emptyset$: 10	$\{a\}$: 7	$\{a, c\}$: 4	$\{a, c, d\}$: 3
	$\{b\}$: 3	$\{a, d\}$: 5	$\{a, c, e\}$: 3
	$\{c\}$: 7	$\{a, e\}$: 6	$\{a, d, e\}$: 4
	$\{d\}$: 6	$\{b, c\}$: 3	
	$\{e\}$: 7	$\{c, d\}$: 4	
		$\{c, e\}$: 4	
		$\{d, e\}$: 4	

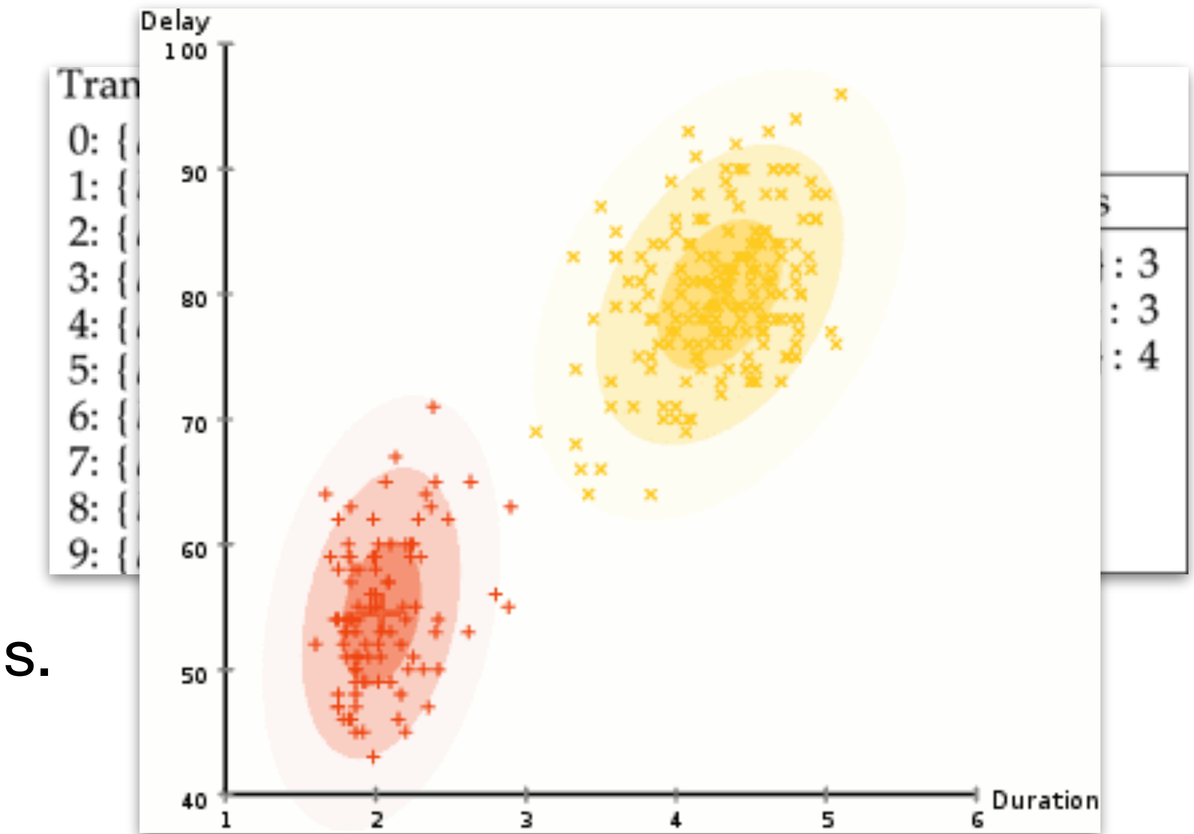
Unsupervised: Clustering

Apriori Algorithm
Find the most frequently occurring combinations of values in data.

Expectation Maximization
Identify the most likely statistical distribution that matches observations.



Unsupervised: Clustering



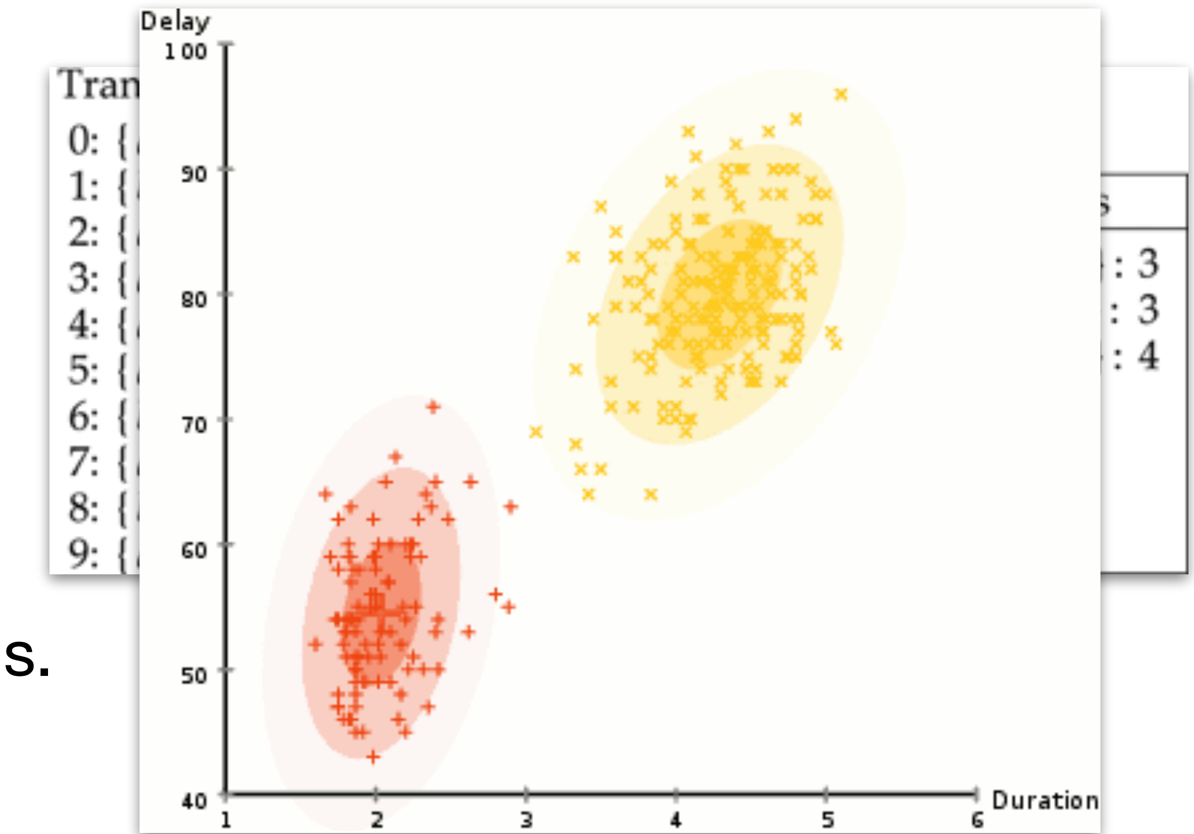
Apriori Algorithm
Find the most frequently occurring combinations of values in data.

Expectation Maximization
Identify the most likely statistical distribution that matches observations.

K-Means
Identify stable cluster centers that are also the mean of cluster members.

Unsupervised: Clustering

Apriori Algorithm
Find the most frequently occurring combinations of values in data.



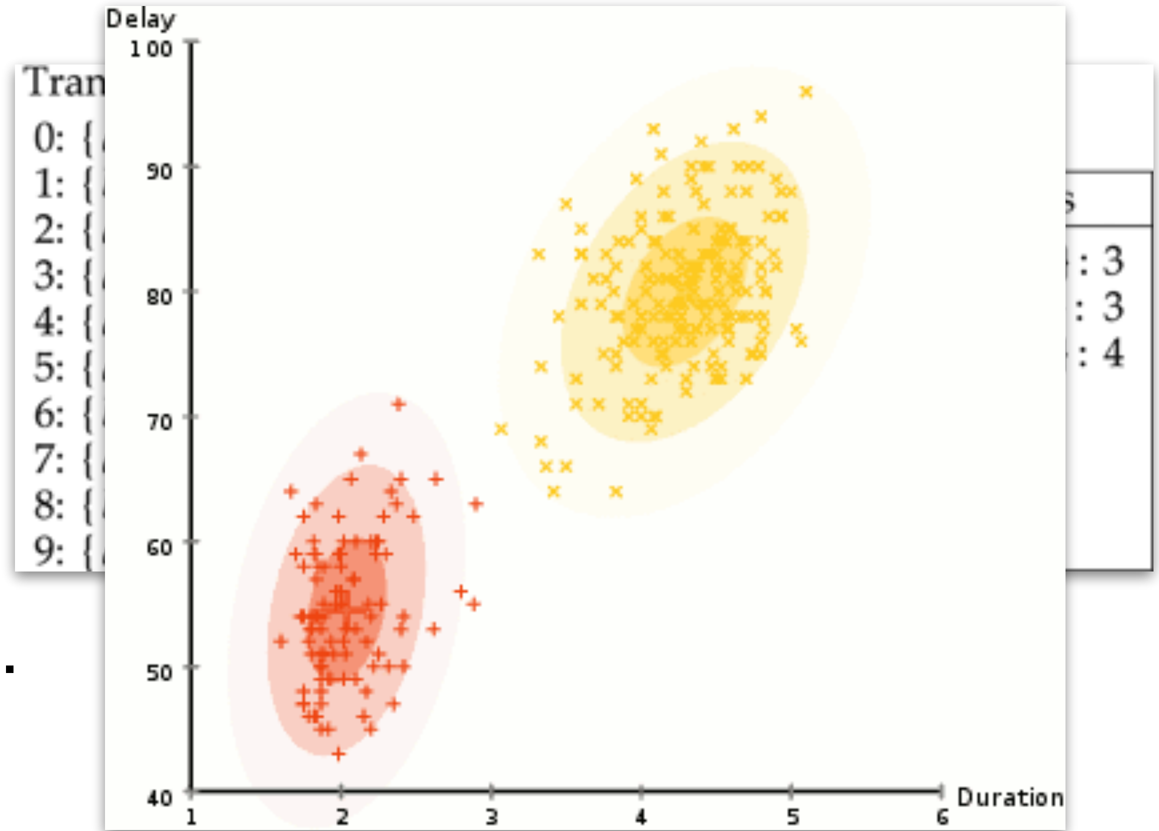
Expectation Maximization
Identify the most likely statistical distribution that matches observations.

K-Means
Identify stable cluster centers that are also the mean of cluster members.

1) $C_i = \left\{ x^{(j)} \;\middle|\; c^{(i)} = \underset{c}{\operatorname{argmin}} \; \|x^{(j)} - c\|_2 \right\}$ Associate points with cluster centers, $c^{(i)}$.

where c are cluster centers, x are data, C_i is the i-*th* cluster, $c^{(i)}$ is the center of the i-*th* cluster.

Unsupervised: Clustering



Apriori Algorithm

Find the most frequently occurring combinations of values in data.

Expectation Maximization

Identify the most likely statistical distribution that matches observations.

K-Means

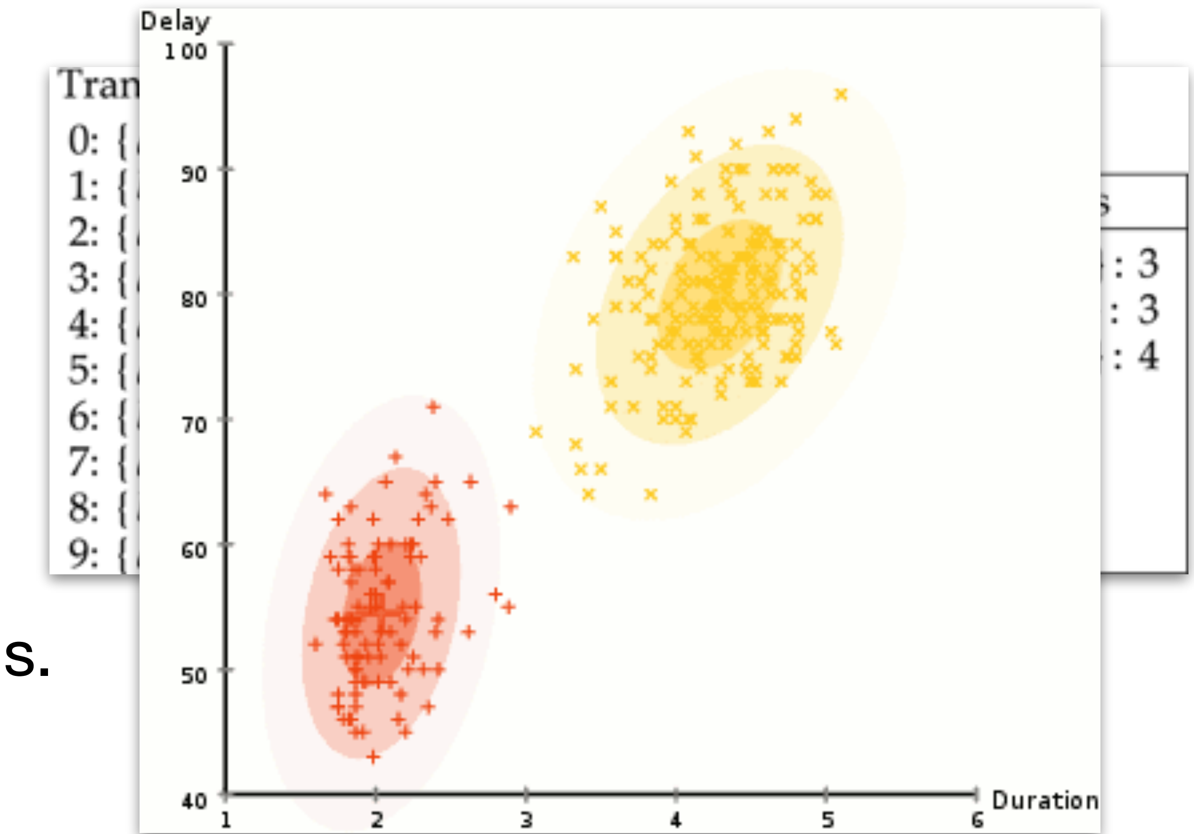
Identify stable cluster centers that are also the mean of cluster members.

1) $C_i = \left\{ x^{(j)} \;\middle|\; c^{(i)} = \underset{c}{\operatorname{argmin}} \; \|x^{(j)} - c\|_2 \right\}$ Associate points with cluster centers, $c^{(i)}$.

2) $c^{(i)} = \sum_{(x \in C^i)} x \, / \, \left| C^i \right|$ Update cluster centers to be mean of points.

where c are cluster centers, x are data, C_i is the i-*th* cluster, $c^{(i)}$ is the center of the i-*th* cluster.

Unsupervised: Clustering



Apriori Algorithm

Find the most frequently occurring combinations of values in data.

Expectation Maximization

Identify the most likely statistical distribution that matches observations.

K-Means

Identify stable cluster centers that are also the mean of cluster members.

1) $C_i = \left\{ x^{(j)} \;\middle|\; c^{(i)} = \underset{c}{\operatorname{argmin}} \; \|x^{(j)} - c\|_2 \right\}$ Associate points with cluster centers, $c^{(i)}$.

2) $c^{(i)} = \sum_{(x \in C^i)} x \, / \, \left| C^i \right|$ Update cluster centers to be mean of points.

where c are cluster centers, x are data, C_i is the i-*th* cluster, $c^{(i)}$ is the center of the i-*th* cluster.

Demonstration

K-Means Clustering

Error Measure: Silhouette Score

Error Measure: Silhouette Score

$$s\left(x^{(i)}\right) = \frac{b\left(x^{(i)}\right) - a\left(x^{(i)}\right)}{\max\left(b\left(x^{(i)}\right),\ a\left(x^{(i)}\right)\right)},$$

where $a(x)$ is the average distance between x and members of its cluster, $b(x)$ is the smallest average distance between x and all members of another cluster.

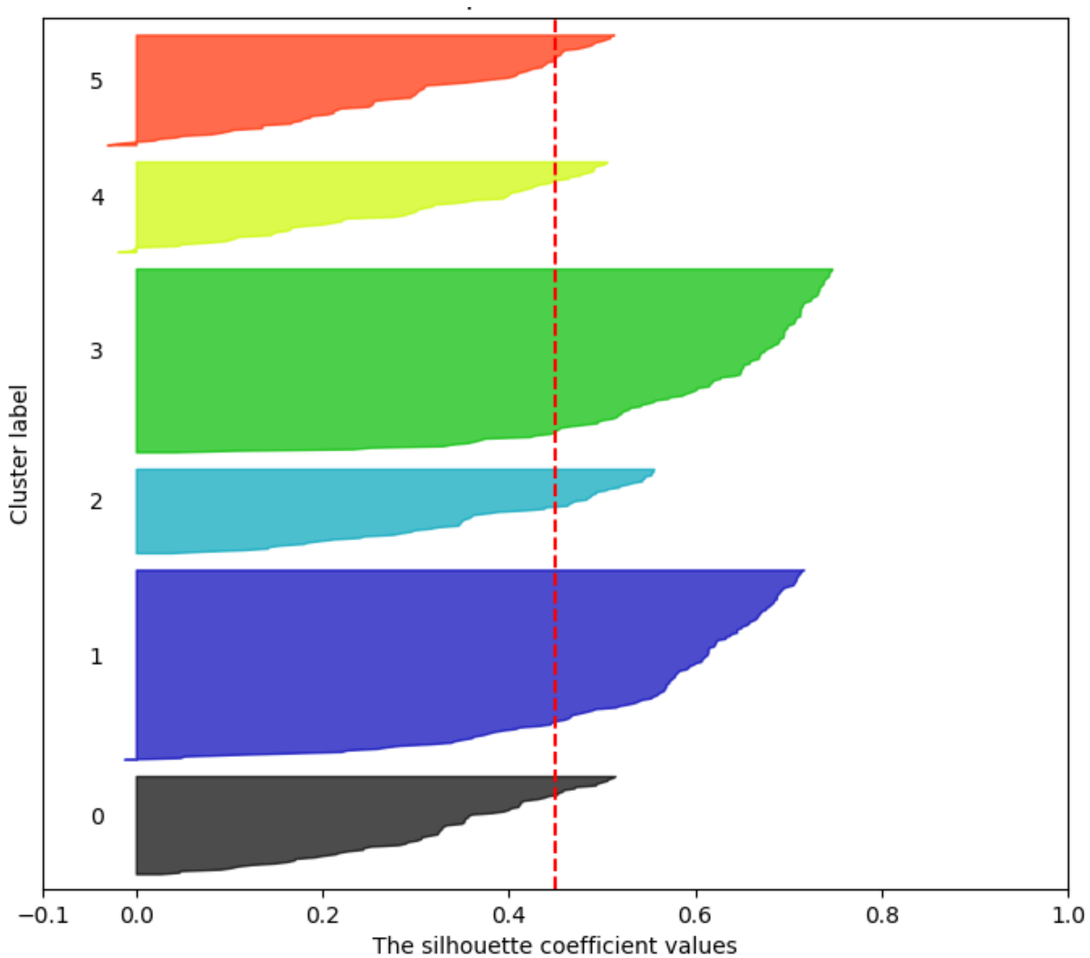
$s(x^{(i)}) \rightarrow 1$ – point is perfectly clustered.
$s(x^{(i)}) \sim 0$ – point is neutral, between clusters.
$s(x^{(i)}) < 0$ – point is poorly clustered.

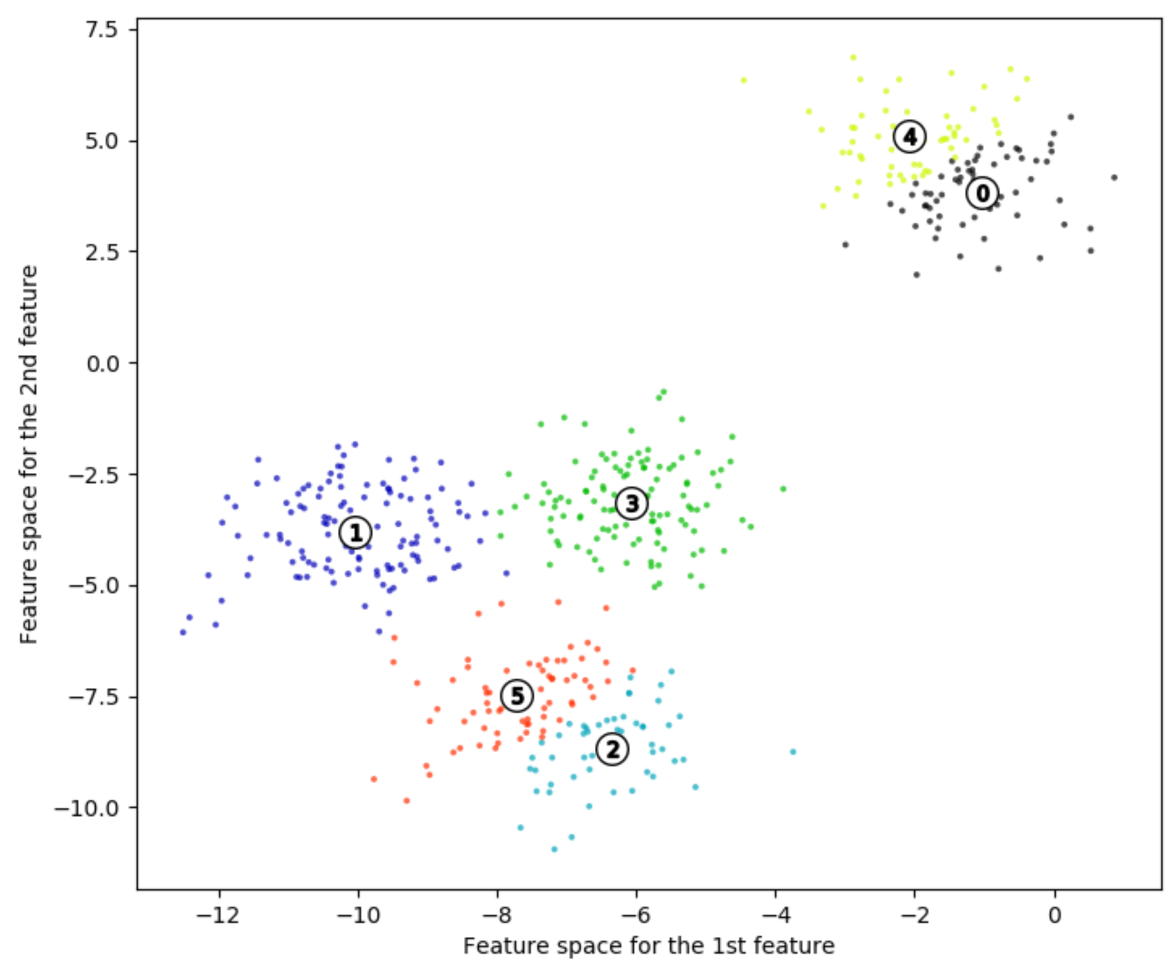
Error Measure: Silhouette Score

$$s\left(x^{(i)}\right) = \frac{b\left(x^{(i)}\right) - a\left(x^{(i)}\right)}{\max\left(b\left(x^{(i)}\right),\, a\left(x^{(i)}\right)\right)},$$

where $a(x)$ is the average distance between x and members of its cluster, $b(x)$ is the smallest average distance between x and all members of another cluster.

$s(x^{(i)}) \rightarrow 1$ – point is perfectly clustered.
$s(x^{(i)}) \sim 0$ – point is neutral, between clusters.
$s(x^{(i)}) < 0$ – point is poorly clustered.

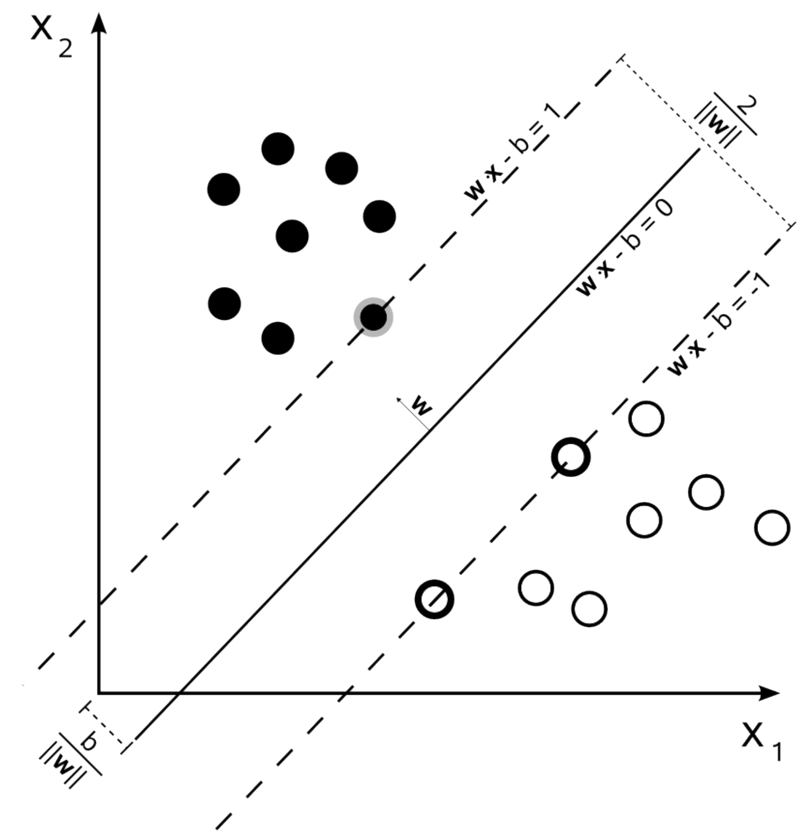




Supervised: Classification

Supervised: Classification

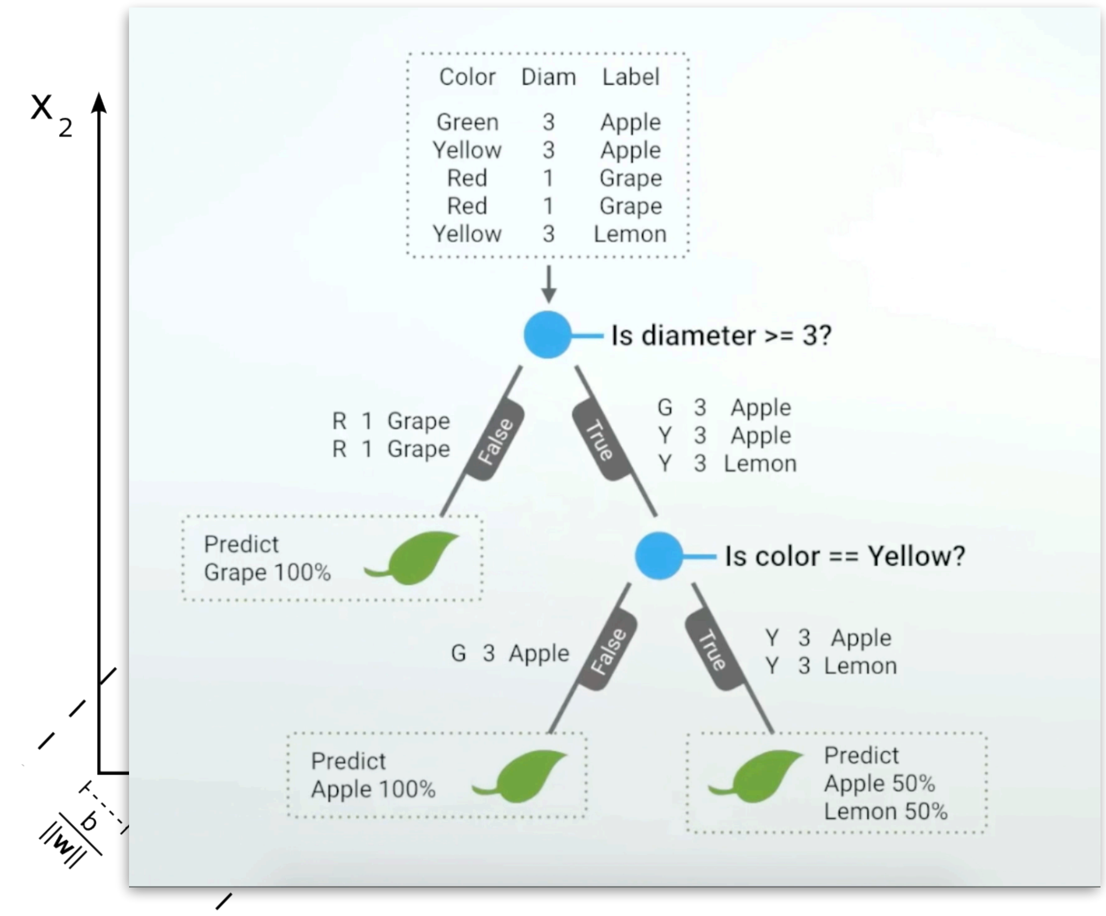
Support Vector Machine
Find the largest-margin
boundary between classes.



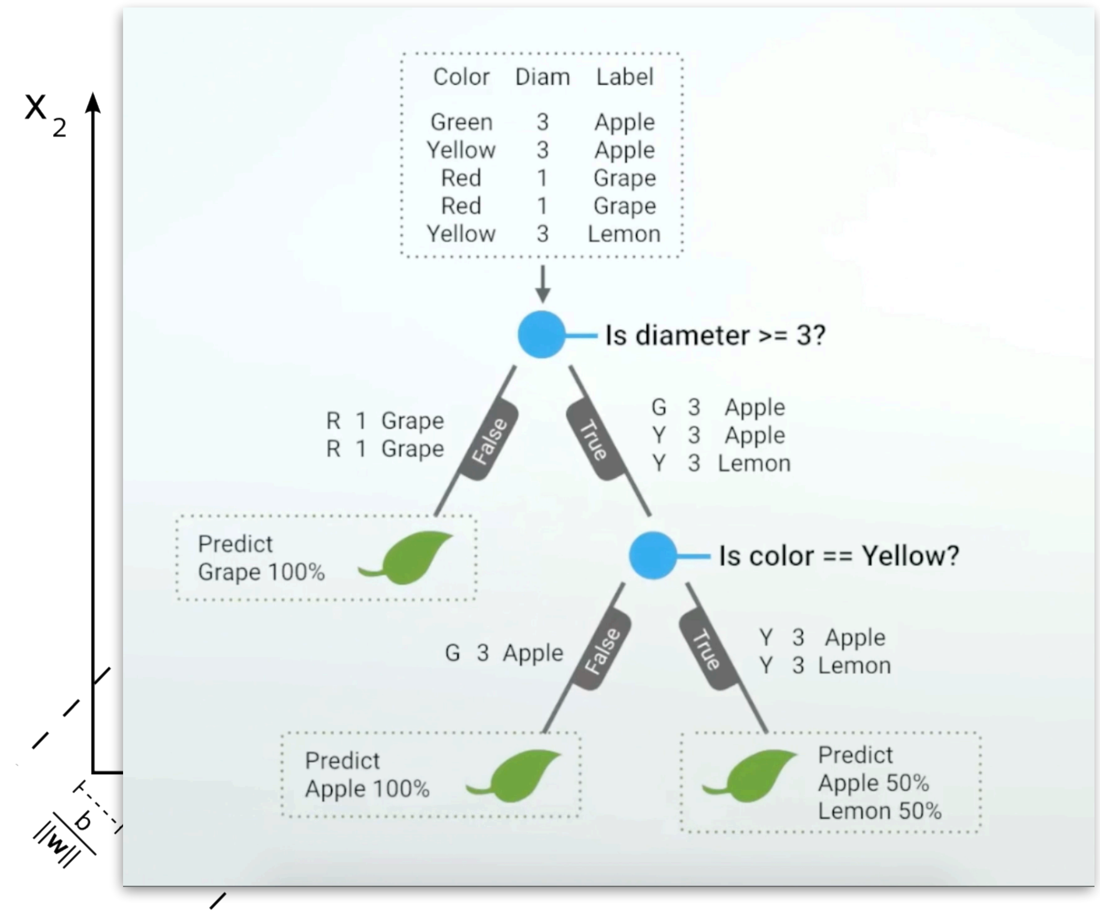
Supervised: Classification

Support Vector Machine
Find the largest-margin boundary between classes.

Decision Tree
Find the most class-divisive feature value and split data, repeat.



Supervised: Classification



Support Vector Machine

Find the largest-margin boundary between classes.

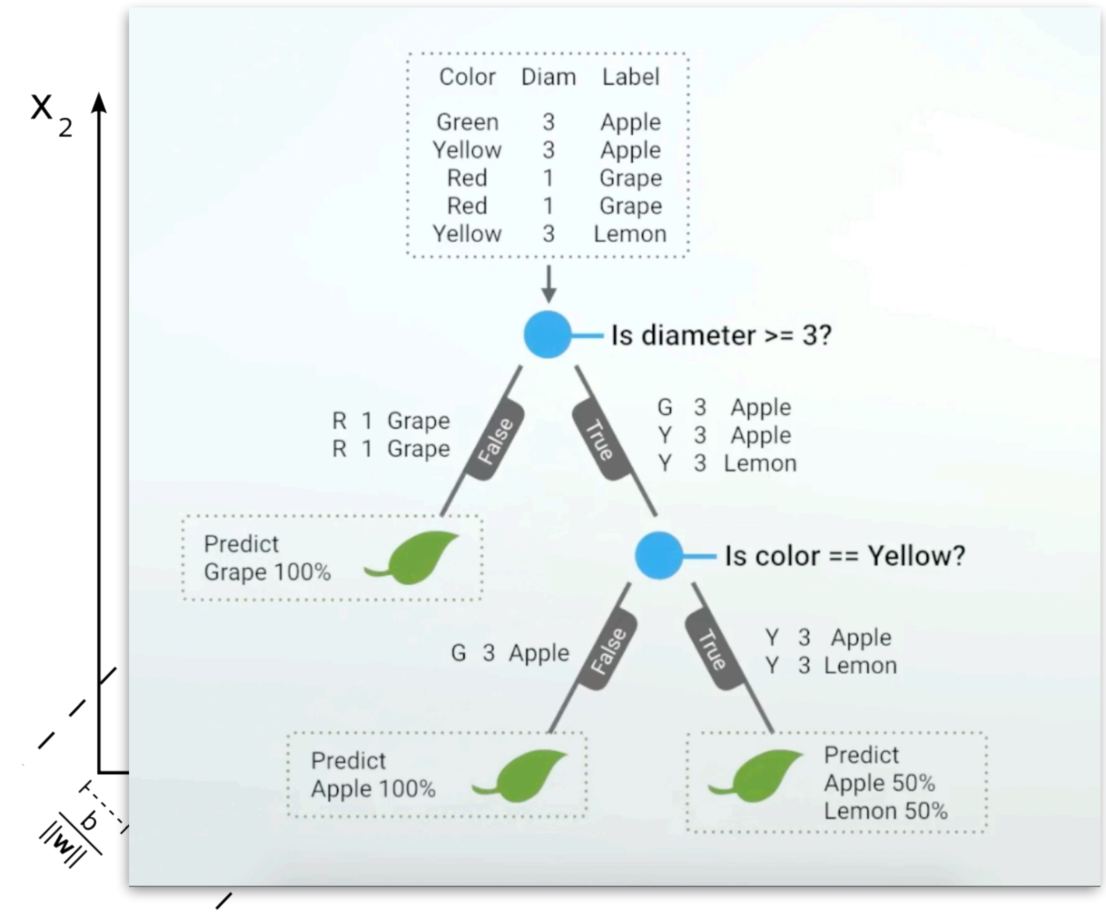
Decision Tree

Find the most class-divisive feature value and split data, repeat.

Neural Network

Find the composition of boundaries that best-separates classes.

Supervised: Classification



Support Vector Machine
Find the largest-margin boundary between classes.

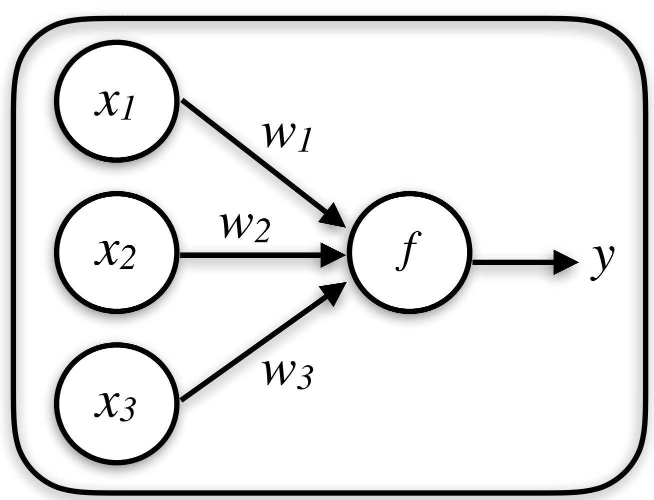
Decision Tree
Find the most class-divisive feature value and split data, repeat.

Neural Network
Find the composition of boundaries that best-separates classes.

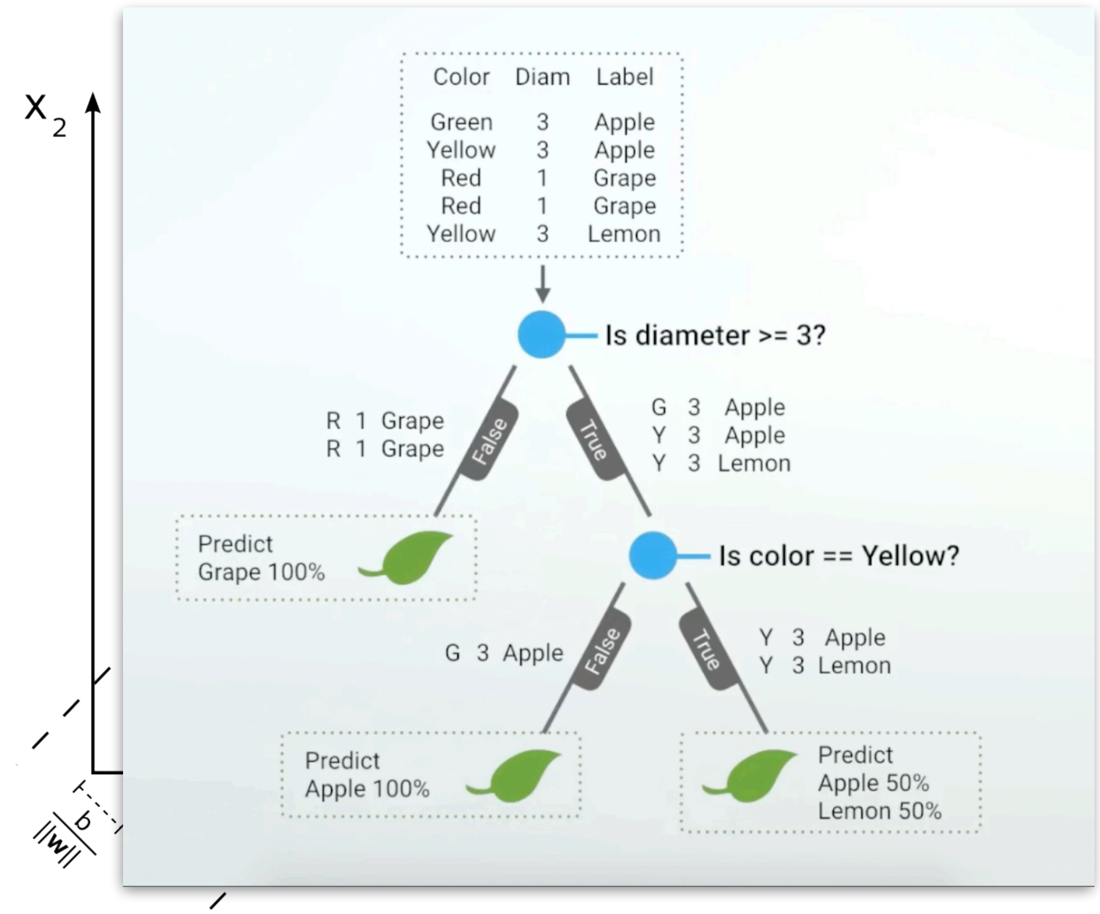
Use error gradient to solve

$$\min_{w} \left\| f(w \cdot x) - y_{\text{true}} \right\|$$

where f is an *activation function*.



Supervised: Classification



Support Vector Machine
Find the largest-margin boundary between classes.

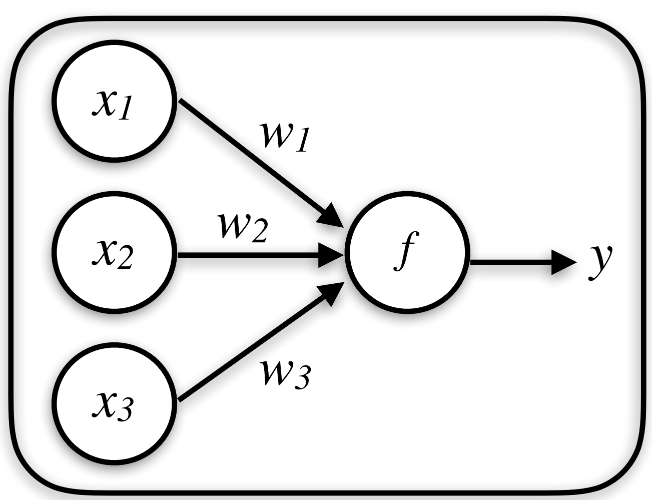
Decision Tree
Find the most class-divisive feature value and split data, repeat.

Neural Network
Find the composition of boundaries that best-separates classes.

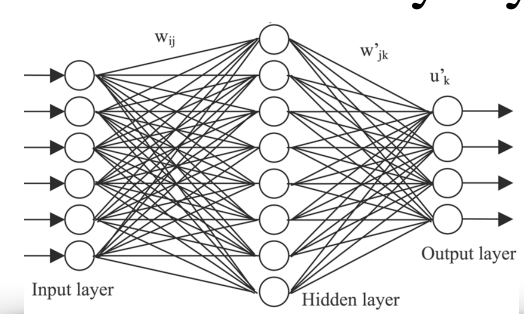
Use error gradient to solve

$$\min_{w} \left\| f(w \cdot x) - y_{\text{true}} \right\|$$

where f is an *activation function*.



Extend this framework to have many nodes and many layers.



Demonstration

Neural Network Classification

Error Measure: Confusion Matrix

Error Measure: Confusion Matrix

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

Source: https://en.wikipedia.org/wiki/Confusion_matrix

Error Measure: Confusion Matrix

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

Source: https://en.wikipedia.org/wiki/Confusion_matrix

From the CM you can compute:

Accuracy, Sensitivity, Specificity,

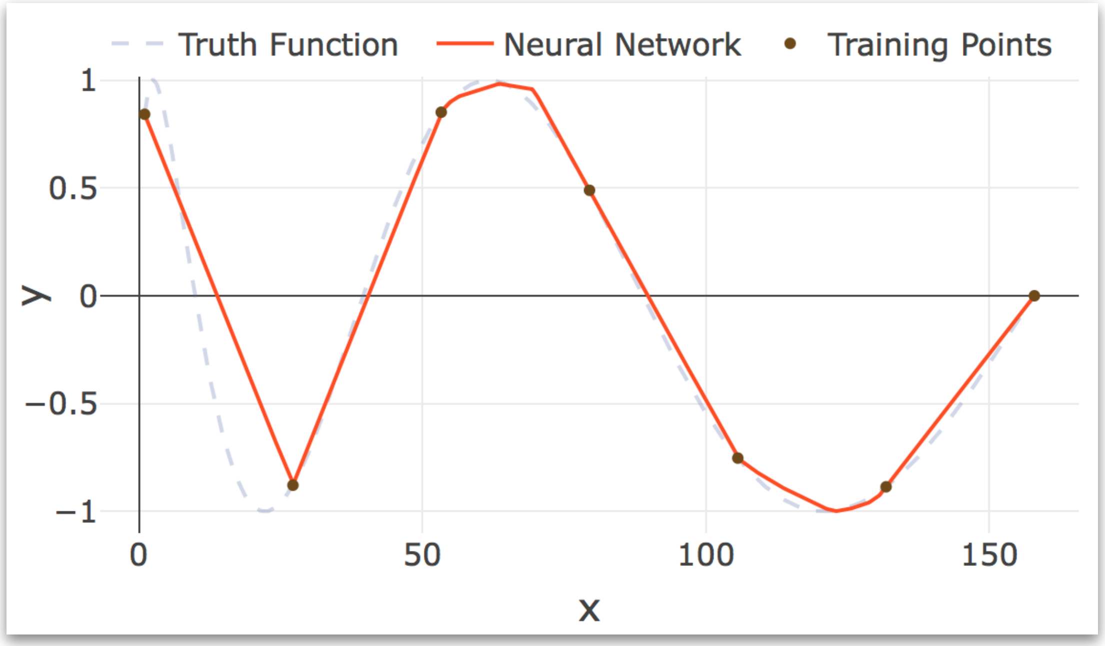
Matthews Correlation Coefficient

Supervised: Regression

Supervised: Regression

Neural Network Regressor

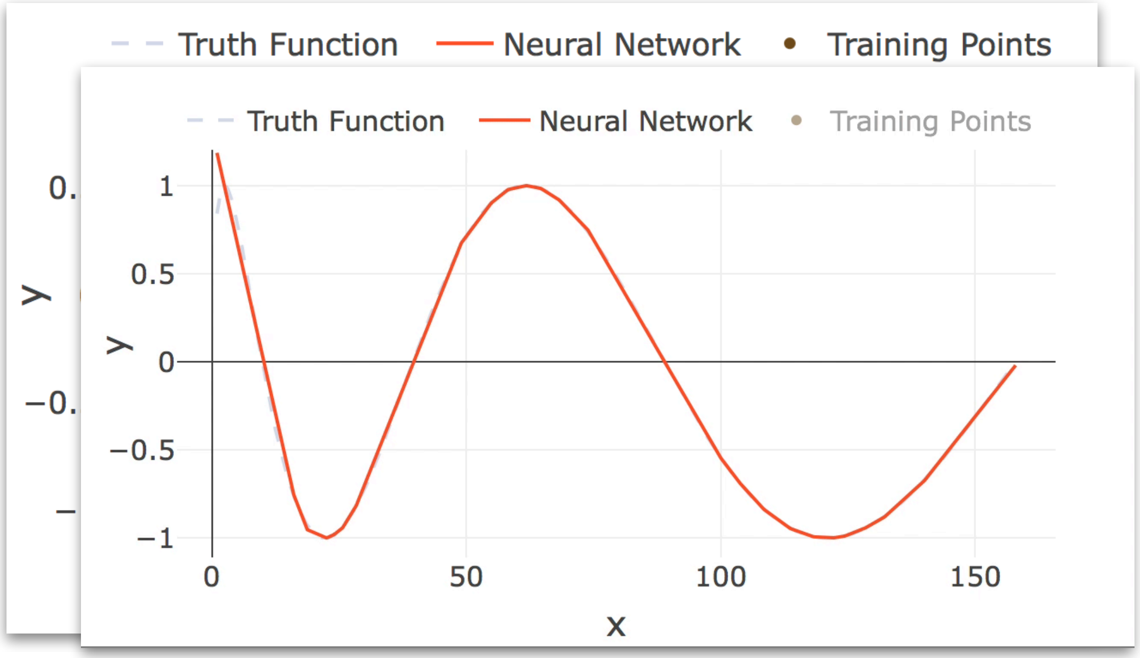
Compose (find parameters for) a set of functions to best fit the provided data.



Supervised: Regression

Neural Network Regressor

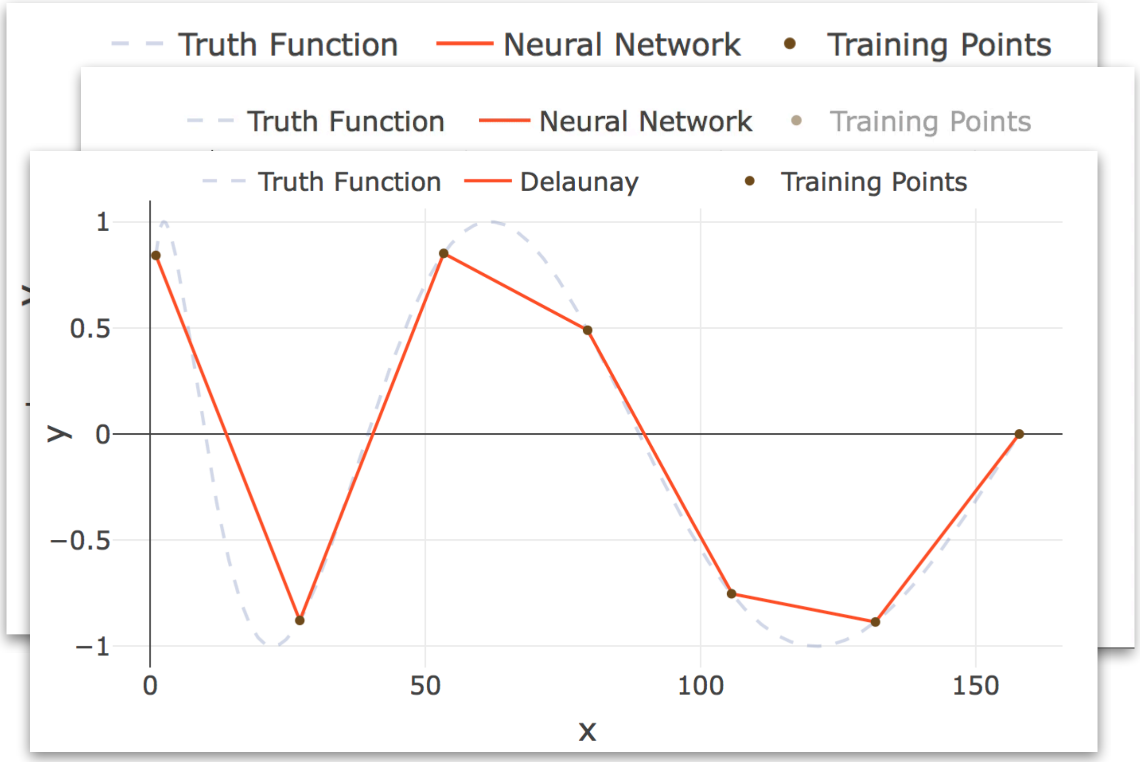
Compose (find parameters for) a set of functions to best fit the provided data.



Supervised: Regression

Neural Network Regressor
Compose (find parameters for) a set of functions to best fit the provided data.

Delaunay Triangulation
Construct a simplicial mesh (piecewise linear local approximations) from data.



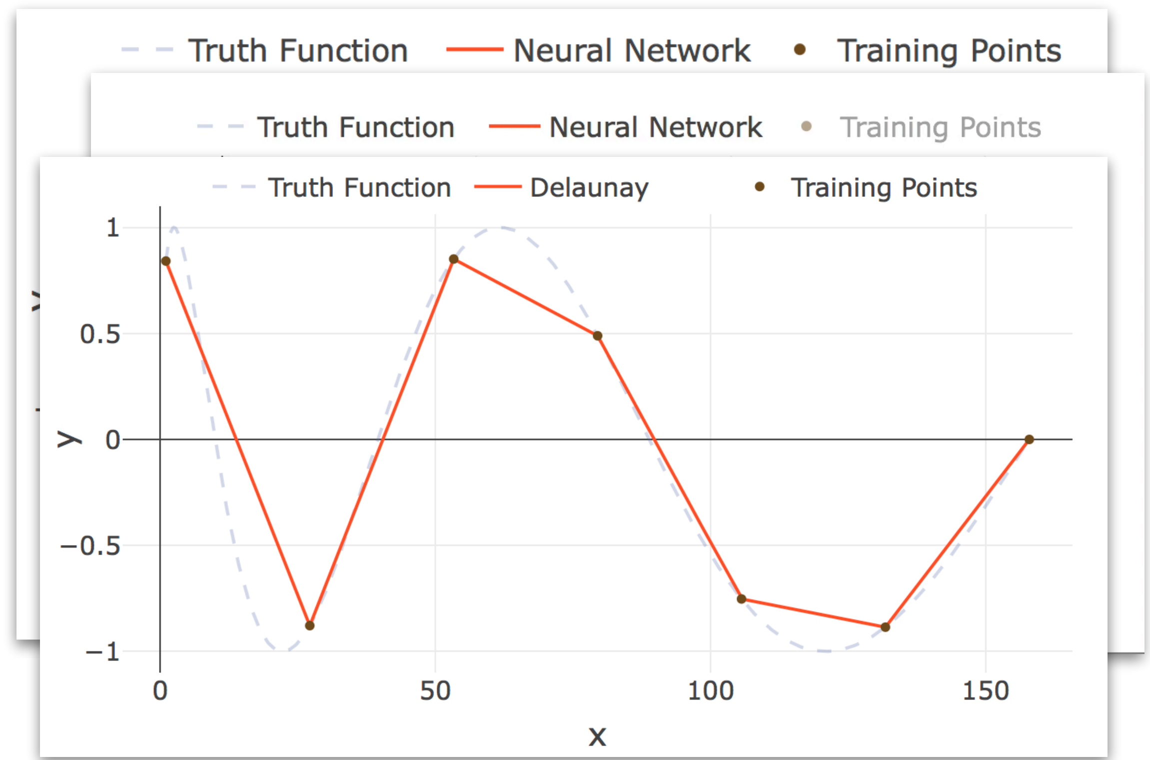
Supervised: Regression

Neural Network Regressor

Compose (find parameters for) a set of functions to best fit the provided data.

Delaunay Triangulation

Construct a simplicial mesh (piecewise linear local approximations) from data.



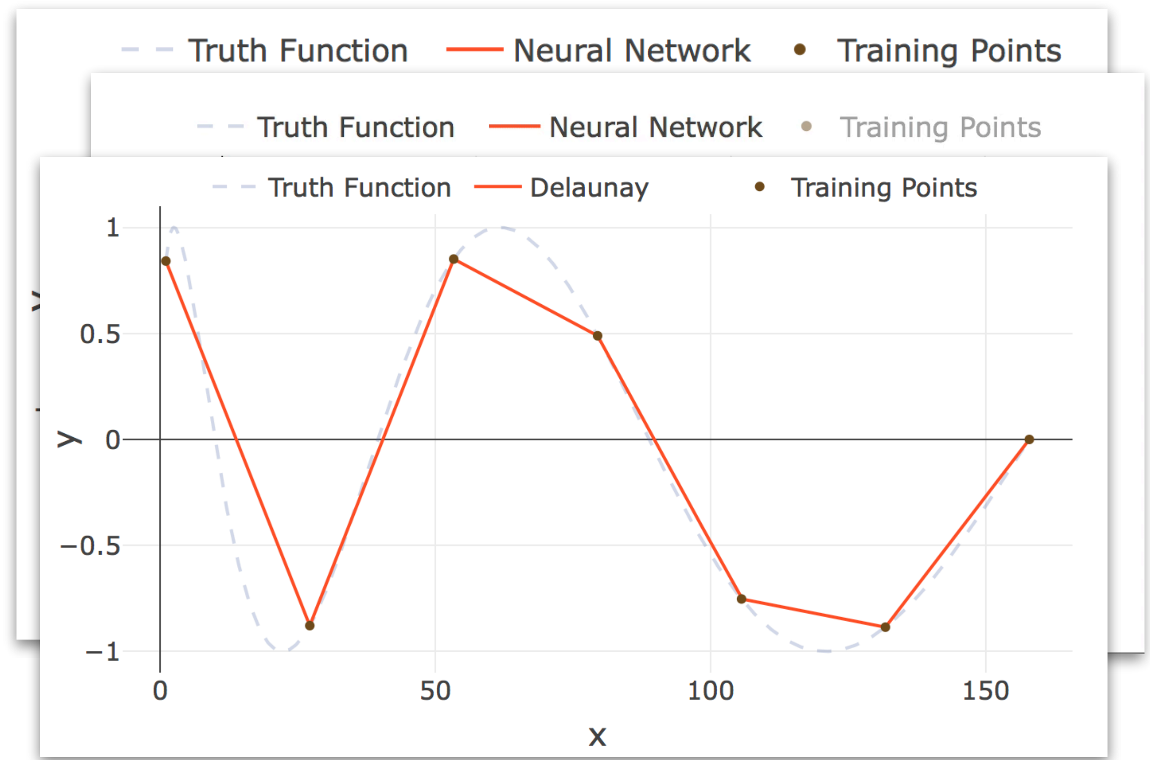
Linear Regression

Solve for those coefficients on a linear function that minimize the Euclidean distance between all function responses and provided data.

Supervised: Regression

Neural Network Regressor
Compose (find parameters for) a set of functions to best fit the provided data.

Delaunay Triangulation
Construct a simplicial mesh (piecewise linear local approximations) from data.

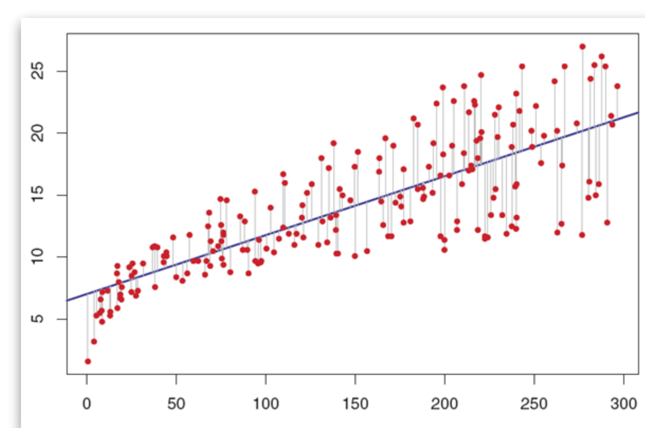


Linear Regression
Solve for those coefficients on a linear function that minimize the Euclidean distance between all function responses and provided data.

Use linear algebra to solve

$$\min_{w} \left\| Xw - y_{\text{truth}} \right\|_2$$

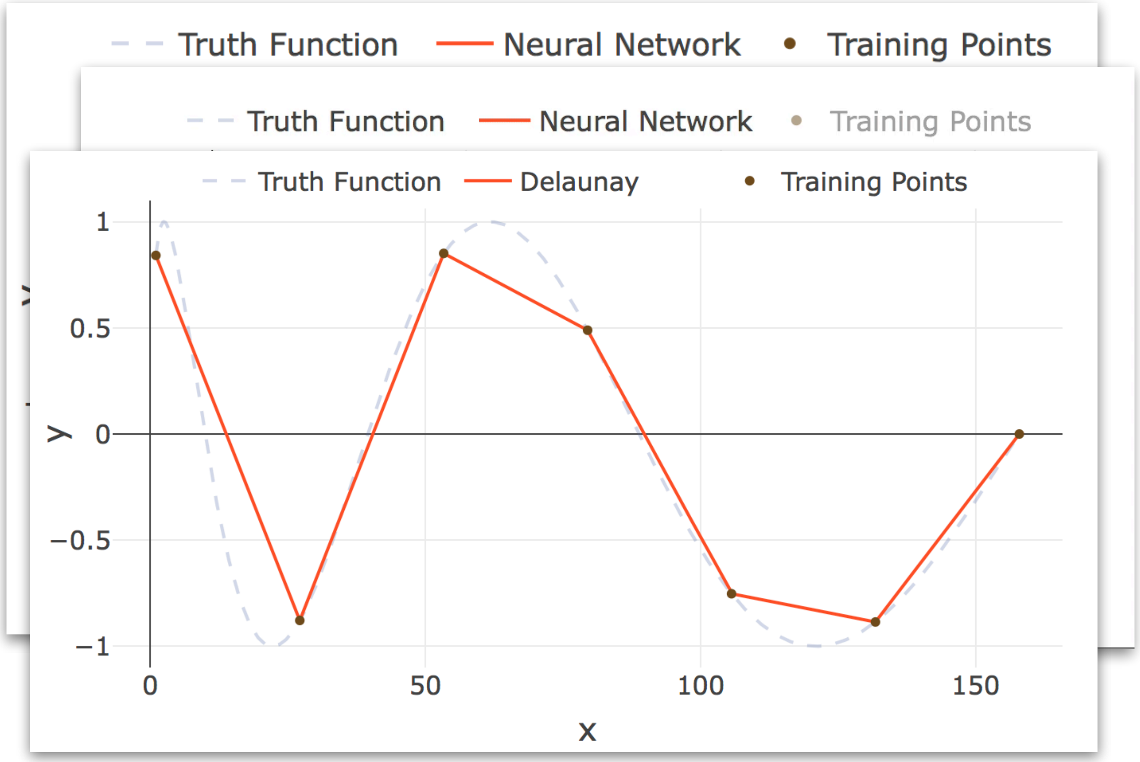
where X is a matrix of row-vector points and w, y are vectors.



Supervised: Regression

Neural Network Regressor
Compose (find parameters for) a set of functions to best fit the provided data.

Delaunay Triangulation
Construct a simplicial mesh (piecewise linear local approximations) from data.



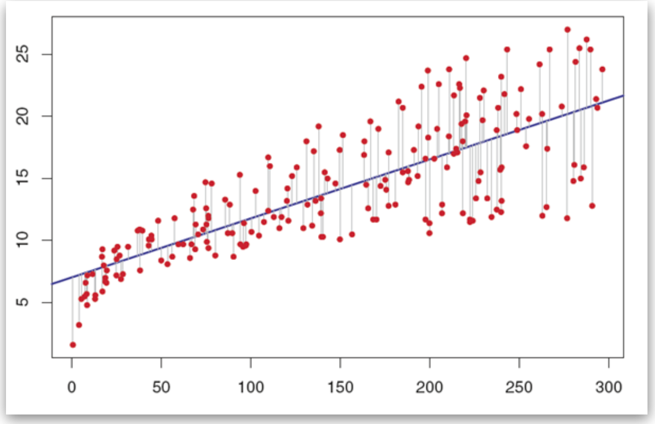
Linear Regression
Solve for those coefficients on a linear function that minimize the Euclidean distance between all function responses and provided data.

Use linear algebra to solve

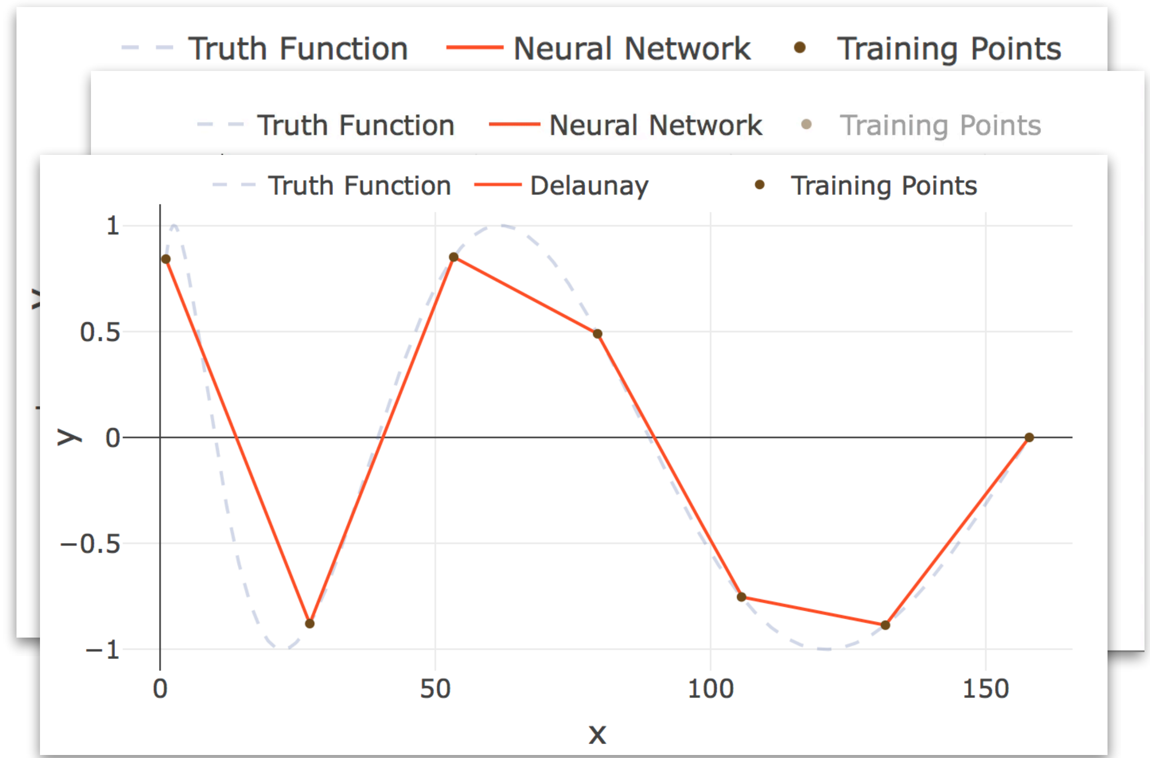
$$\min_{w} \left\| Xw - y_{\text{truth}} \right\|_2$$

where X is a matrix of row-vector points and w, y are vectors.

$$Xw = y$$



Supervised: Regression



Neural Network Regressor

Compose (find parameters for) a set of functions to best fit the provided data.

Delaunay Triangulation

Construct a simplicial mesh (piecewise linear local approximations) from data.

Linear Regression

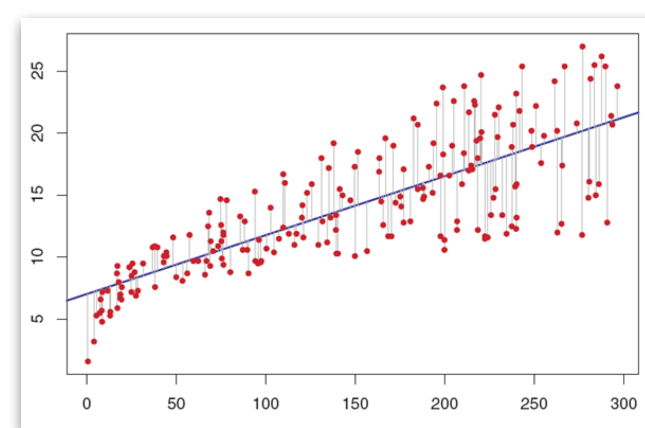
Solve for those coefficients on a linear function that minimize the Euclidean distance between all function responses and provided data.

Use linear algebra to solve

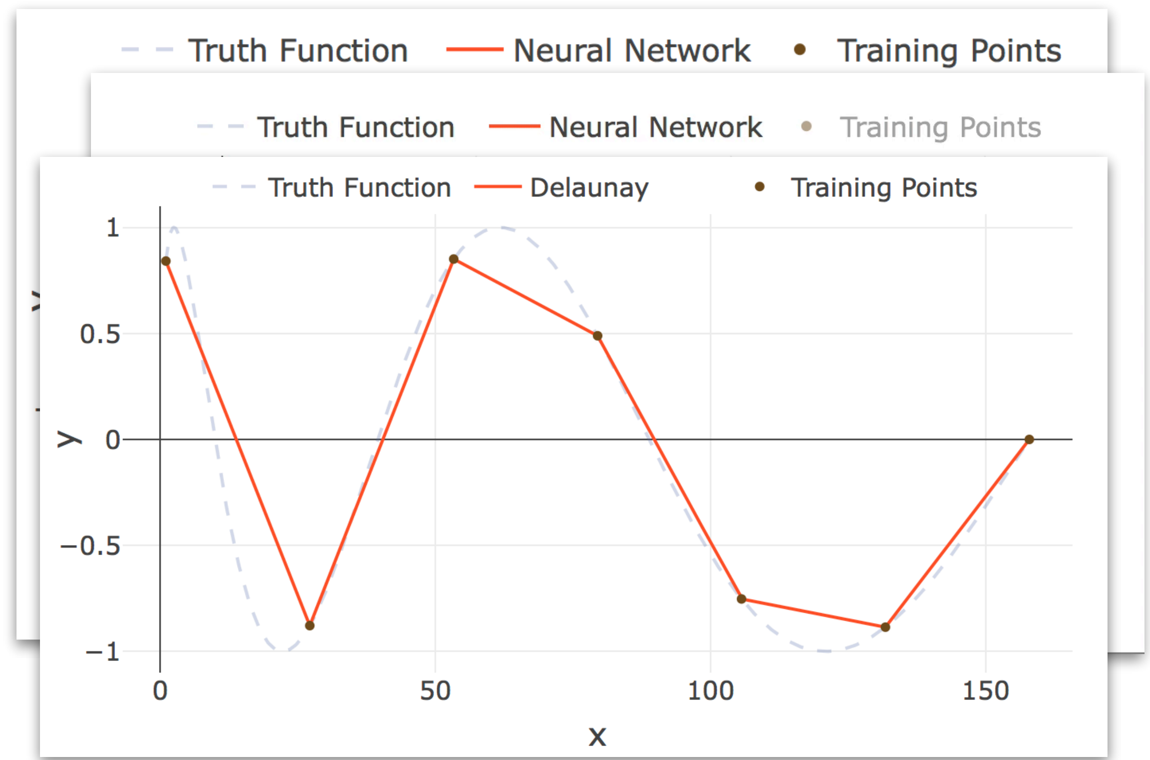
$$\min_{w} \left\| Xw - y_{\text{truth}} \right\|_2$$

where X is a matrix of row-vector points and w, y are vectors.

$$Xw = y \qquad (X^T X)w = X^T y$$



Supervised: Regression



Neural Network Regressor

Compose (find parameters for) a set of functions to best fit the provided data.

Delaunay Triangulation

Construct a simplicial mesh (piecewise linear local approximations) from data.

Linear Regression

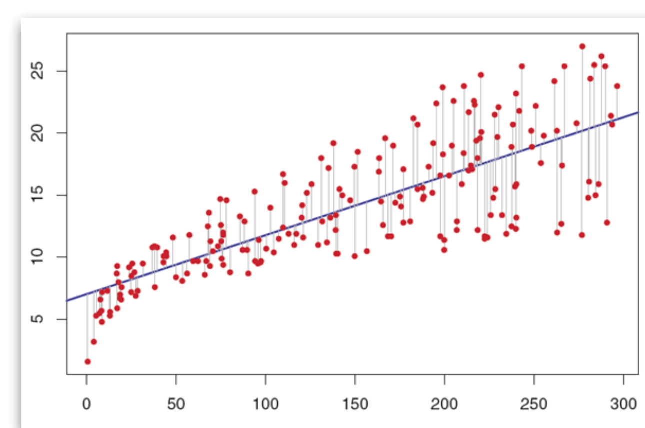
Solve for those coefficients on a linear function that minimize the Euclidean distance between all function responses and provided data.

Use linear algebra to solve

$$\min_{w} \left\| Xw - y_{\text{truth}} \right\|_2$$

where X is a matrix of row-vector points and w, y are vectors.

$$Xw = y \qquad (X^T X)w = X^T y \qquad w = (X^T X)^{-1} X^T y$$



Demonstration

Delaunay Triangulation

Error Measure: *p*-norm

$$\| x \|_p = \left(\sum_{i=1}^{d} | x_i |^p \right)^{1/p}$$

Error Measure: *p*-norm

$$\| x \|_p = \left(\sum_{i=1}^{d} | x_i |^p \right)^{1/p}$$

Common norms

"City-block distance" is the 1-norm

the sum of absolute values

$$\sum_{i=1}^{d} | x_i |$$

Error Measure: *p*-norm

$$\| x \|_p = \left(\sum_{i=1}^{d} | x_i |^p \right)^{1/p}$$

Common norms

“City-block distance” is the 1-norm
the sum of absolute values

$$\sum_{i=1}^{d} | x_i |$$

“Euclidean distance” is the 2-norm
the square-root of the sum of squares

$$\left(\sum_{i=1}^{d} x_i^2 \right)^{1/2}$$

Error Measure: *p*-norm

$$\|x\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}$$

Common norms

"City-block distance" is the 1-norm
the sum of absolute values

$$\sum_{i=1}^{d} |x_i|$$

"Euclidean distance" is the 2-norm
the square-root of the sum of squares

$$\left(\sum_{i=1}^{d} x_i^2 \right)^{1/2}$$

"Max norm" is the ∞-norm
the largest component

$$\left(\sum_{i=1}^{d} x_i^\infty \right)^{1/\infty} = \lim_{n \to \infty} \left(\sum_{i=1}^{d} x_i^n \right)^{1/n}$$

Error Measure: *p*-norm

$$\| x \|_p = \left(\sum_{i=1}^{d} | x_i |^p \right)^{1/p}$$

Common norms

“City-block distance” is the 1-norm
the sum of absolute values

$$\sum_{i=1}^{d} | x_i |$$

“Euclidean distance” is the 2-norm
the square-root of the sum of squares

$$\left(\sum_{i=1}^{d} x_i^2 \right)^{1/2}$$

“Max norm” is the ∞-norm
the largest component

$$\left(\sum_{i=1}^{d} x_i^\infty \right)^{1/\infty} = \lim_{n \to \infty} \left(\sum_{i=1}^{d} x_i^n \right)^{1/n}$$

“0-norm” is common, but it is **not actually a norm!!**
the sum of nonzero components

Error Measure: *p*-norm

$$\|x\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}$$

Properties of a norm

Given vectors **u**, **v** and constant a.

1) $\|u + v\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$

2) $\|a\mathbf{v}\| = |a| \; \|\mathbf{v}\|$

3) $\|\mathbf{v}\| = 0 \iff \mathbf{v} = \mathbf{0}$

Common norms

"City-block distance" is the 1-norm
the sum of absolute values

$$\sum_{i=1}^{d} |x_i|$$

"Euclidean distance" is the 2-norm
the square-root of the sum of squares

$$\left(\sum_{i=1}^{d} x_i^2 \right)^{1/2}$$

"Max norm" is the ∞-norm
the largest component

$$\left(\sum_{i=1}^{d} x_i^\infty \right)^{1/\infty} = \lim_{n \to \infty} \left(\sum_{i=1}^{d} x_i^n \right)^{1/n}$$

"0-norm" is common, but it is **not actually a norm!!**
the sum of nonzero components

Index of Terms (embedded links)

Extra Slides

Contingency Tables

Given two sequences of numbers **A** and **B**

We may consider what pairs occur occur between the sequences.

	A is 1	A is 2	…	A is n
B is 1	$\lvert A_1 \cap B_1 \rvert$	$\lvert A_2 \cap B_1 \rvert$	…	$\lvert A_n \cap B_1 \rvert$
B is 2	$\lvert A_1 \cap B_2 \rvert$	$\ddots$		$\vdots$
$\vdots$	$\vdots$			
B is n	$\lvert A_1 \cap B_n \rvert$	$\cdots$		$\lvert A_n \cap B_n \rvert$

Unsupervised: Clustering

Apriori Tree

Goal: Find the most frequently occurring combinations of values in data.
Pro: Fast to compute, works for categorical (or coarsely discretized) data.
Con: Not good for continuous (or high resolution discretized) data.

Expectation Maximization

Goal: Identify the most likely statistical distribution that matches observations.
Pro: Works for unbalanced clusters, achieves local convergence.
Con: No-guarantee solutions, slow convergence, choosing parameterization.

K-Means

Goal: Identify stable cluster centers that are also the mean of cluster.
Pro: Fast convergence, quick to compute.
Con: Potentially unstable solutions, even-sized clusters, choosing “k”.