Machine Learning I: Fractal 2

Rajendra Nagar

Assistant Professor
Department of Electircal Engineering
Indian Institute of Technology Jodhpur

Supervised Learning

Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set.

Supervised Learning

Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set. Here, $\mathbf{x}_i$ is the i^{th} training input, e.g. an image, and y_i is the corresponding label, e.g. "cat".

Supervised Learning

Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set. Here, $\mathbf{x}_i$ is the i^{th} training input, e.g. an image, and y_i is the corresponding label, e.g. "cat". Let $\mathcal{X}$ be the set of all inputs and let $\mathcal{Y}$ be the set of all possible output labels

Supervised Learning

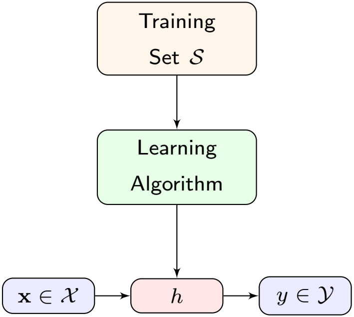
Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set. Here, $\mathbf{x}_i$ is the i^{th} training input, e.g. an image, and y_i is the corresponding label, e.g. "cat". Let $\mathcal{X}$ be the set of all inputs and let $\mathcal{Y}$ be the set of all possible output labels and let $h : \mathcal{X} \to \mathcal{Y}$ be a predictor.

Supervised Learning

Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set. Here, $\mathbf{x}_i$ is the i^{th} training input, e.g. an image, and y_i is the corresponding label, e.g. "cat". Let $\mathcal{X}$ be the set of all inputs and let $\mathcal{Y}$ be the set of all possible output labels and let $h : \mathcal{X} \to \mathcal{Y}$ be a predictor. Then, our goal is to find h such that $h(\mathbf{x}_i)$ is equal to the true label of the input $\mathbf{x}_i$.

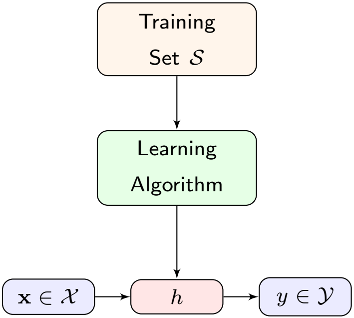
Supervised Learning

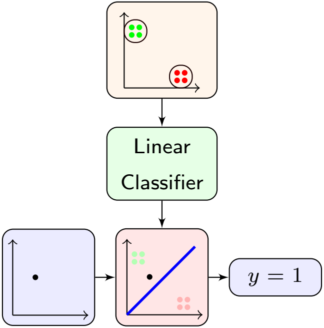
Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set. Here, $\mathbf{x}_i$ is the i^{th} training input, e.g. an image, and y_i is the corresponding label, e.g. "cat". Let $\mathcal{X}$ be the set of all inputs and let $\mathcal{Y}$ be the set of all possible output labels and let $h : \mathcal{X} \to \mathcal{Y}$ be a predictor. Then, our goal is to find h such that $h(\mathbf{x}_i)$ is equal to the true label of the input $\mathbf{x}_i$.



Supervised Learning

Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set. Here, $\mathbf{x}_i$ is the i^{th} training input, e.g. an image, and y_i is the corresponding label, e.g. "cat". Let $\mathcal{X}$ be the set of all inputs and let $\mathcal{Y}$ be the set of all possible output labels and let $h : \mathcal{X} \to \mathcal{Y}$ be a predictor. Then, our goal is to find h such that $h(\mathbf{x}_i)$ is equal to the true label of the input $\mathbf{x}_i$.



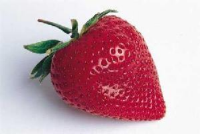


k-Nearest Neighbors Classifier

The idea is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbors in the training set.

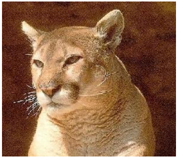
k-Nearest Neighbors Classifier

The idea is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbors in the training set.



"strawberry"

"elephant"

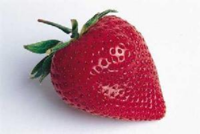


"cougar"

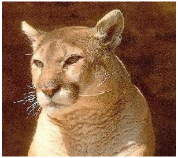
"butterfly"

k-Nearest Neighbors Classifier

The idea is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbors in the training set.


"strawberry"


"elephant"


"cougar"


"butterfly"


"??"

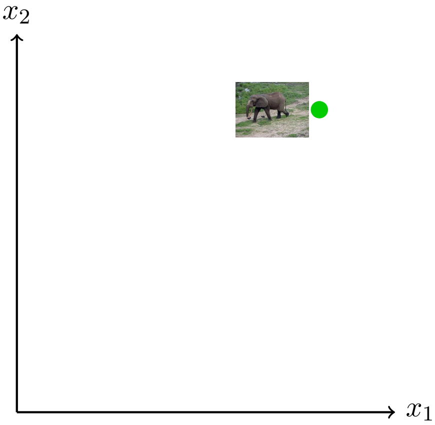
Images Source: L. Fei-Fei, R. Fergus and P. Perona. One-Shot learning of object categories. IEEE Trans. Pattern Recognition and Machine Intelligence. In press.

k-Nearest Neighbors Classifier

The features that are used to describe the domain points are relevant to their labelings in a way that makes close-by points likely to have the same label.

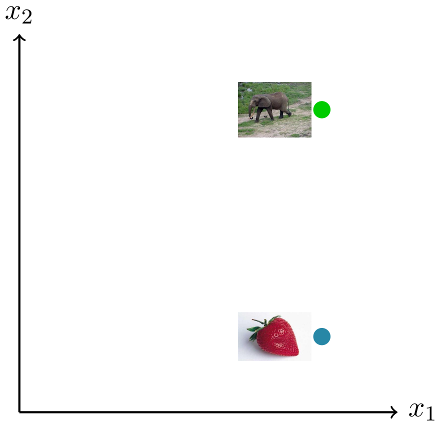
k-Nearest Neighbors Classifier

The features that are used to describe the domain points are relevant to their labelings in a way that makes close-by points likely to have the same label.



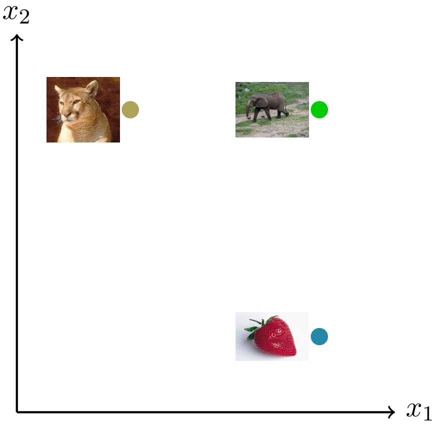
k-Nearest Neighbors Classifier

The features that are used to describe the domain points are relevant to their labelings in a way that makes close-by points likely to have the same label.



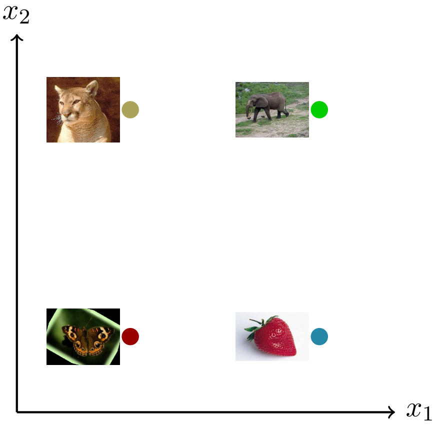
k-Nearest Neighbors Classifier

The features that are used to describe the domain points are relevant to their labelings in a way that makes close-by points likely to have the same label.



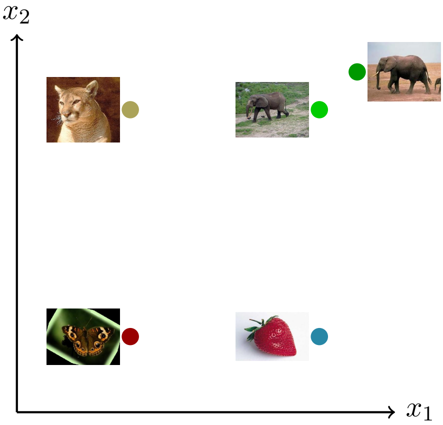
k-Nearest Neighbors Classifier

The features that are used to describe the domain points are relevant to their labelings in a way that makes close-by points likely to have the same label.



k-Nearest Neighbors Classifier

The features that are used to describe the domain points are relevant to their labelings in a way that makes close-by points likely to have the same label.



The Notion of Distance

- Let $\mathcal{X}$ be the set of instances.

The Notion of Distance

- Let $\mathcal{X}$ be the set of instances.
- We also assume that we have a notion of finding distance between the elements.

The Notion of Distance

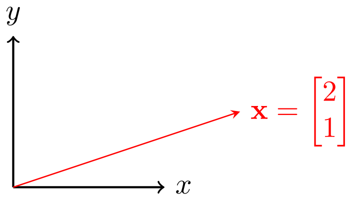
- Let $\mathcal{X}$ be the set of instances.
- We also assume that we have a notion of finding distance between the elements.
- That is, $\rho : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a function that returns the distance between any two elements of $\mathcal{X}$.

The Notion of Distance

- Let $\mathcal{X}$ be the set of instances.
- We also assume that we have a notion of finding distance between the elements.
- That is, $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a function that returns the distance between any two elements of $\mathcal{X}$.
- For example, if $\mathcal{X} = \mathbb{R}^d$ then ρ can be the Euclidean distance, $\rho(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2 = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2}$.

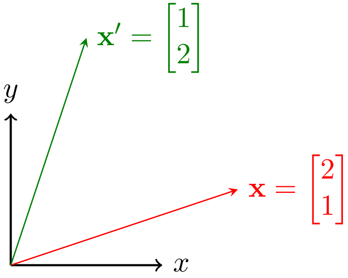
The Notion of Distance

- Let $\mathcal{X}$ be the set of instances.
- We also assume that we have a notion of finding distance between the elements.
- That is, $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a function that returns the distance between any two elements of $\mathcal{X}$.
- For example, if $\mathcal{X} = \mathbb{R}^d$ then ρ can be the Euclidean distance, $\rho(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2 = \sqrt{\sum_{i=1}^{d} (x_i - x_i')^2}$.



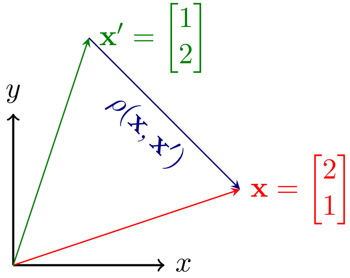
The Notion of Distance

- Let $\mathcal{X}$ be the set of instances.
- We also assume that we have a notion of finding distance between the elements.
- That is, $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a function that returns the distance between any two elements of $\mathcal{X}$.
- For example, if $\mathcal{X} = \mathbb{R}^d$ then ρ can be the Euclidean distance, $\rho(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2 = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2}$.



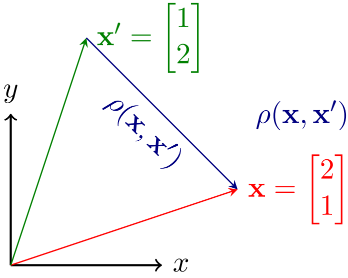
The Notion of Distance

- Let $\mathcal{X}$ be the set of instances.
- We also assume that we have a notion of finding distance between the elements.
- That is, $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a function that returns the distance between any two elements of $\mathcal{X}$.
- For example, if $\mathcal{X} = \mathbb{R}^d$ then ρ can be the Euclidean distance, $\rho(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2 = \sqrt{\sum_{i=1}^{d} (x_i - x_i')^2}$.



The Notion of Distance

- Let $\mathcal{X}$ be the set of instances.
- We also assume that we have a notion of finding distance between the elements.
- That is, $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a function that returns the distance between any two elements of $\mathcal{X}$.
- For example, if $\mathcal{X} = \mathbb{R}^d$ then ρ can be the Euclidean distance, $\rho(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2 = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2}$.



$$\rho(\mathbf{x}, \mathbf{x}') = \sqrt{(1-2)^2 + (2-1)^2} = \sqrt{2}$$

k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.

k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.

k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$

k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$

$\left(\begin{bmatrix}0\\0\end{bmatrix}, \text{'car'}\right)$

k-Nearest Neighbors Classifier

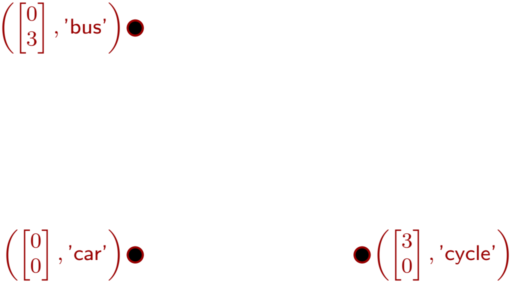
- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$

k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

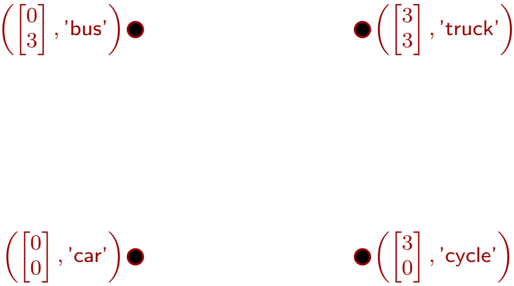
$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$



k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

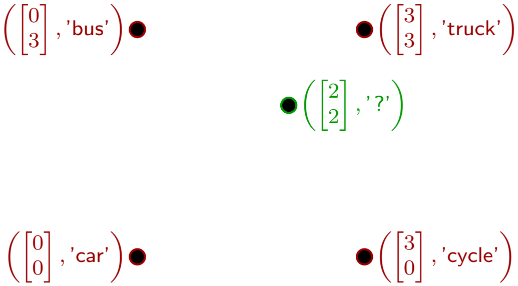
$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$



k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

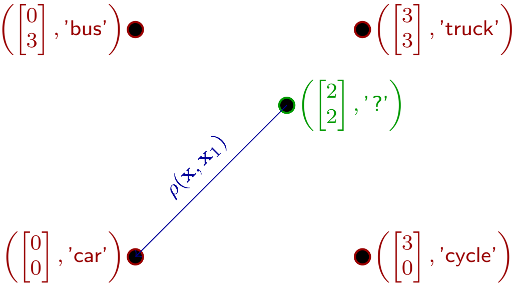
$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$



k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

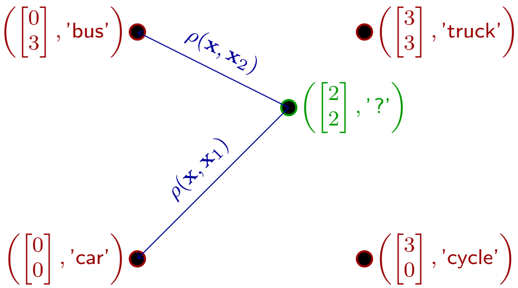
$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$



k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

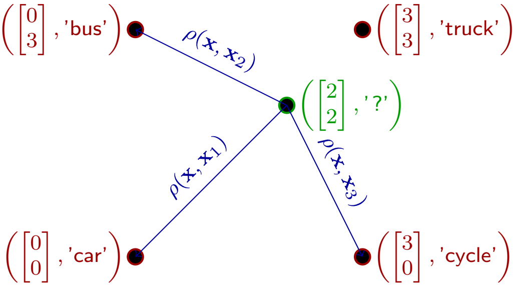
$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$



k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

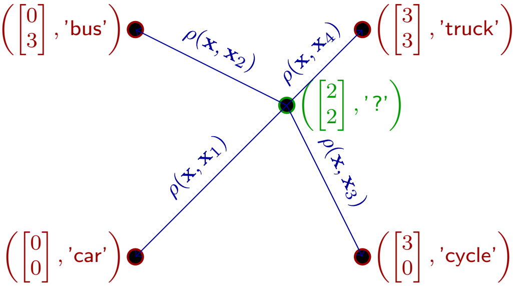
$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$



k-Nearest Neighbors Classifier

- Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set.
- For each $\mathbf{x} \in \mathcal{X}$, let $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ be reordering of $\{1, 2, \ldots, m\}$ according to their distance to $\mathbf{x}$, $\rho(\mathbf{x}, \mathbf{x}')$.
- That is, for all $i < m$,

$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})})$$



Algorithm 1 k-NN Binary Classification

1: **Input:** a training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$

2: **Output:** For every test point $\mathbf{x} \in \mathcal{X}$
return the majority label among $\{y_{\pi_i(\mathbf{x})} : i \leq k\}$.

Algorithm 2 k-NN Regression

1: **Input:** a training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$

2: **Output:** For every test point $\mathbf{x} \in \mathcal{X}$
return the average target of the k nearest neighbors

$$h(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{k} y_{\pi_i(\mathbf{x})}.$$

The Curse of Dimensionality

- The 1-NN rule might fail if the number of examples is smaller than $\frac{(c+1)^d}{2}$.

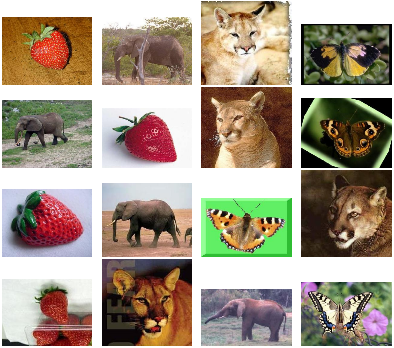
The Curse of Dimensionality

- The 1-NN rule might fail if the number of examples is smaller than $\frac{(c+1)^d}{2}$.
- The exponential dependence of the dataset size on the dimension of the data points is known as the curse of dimensionality.

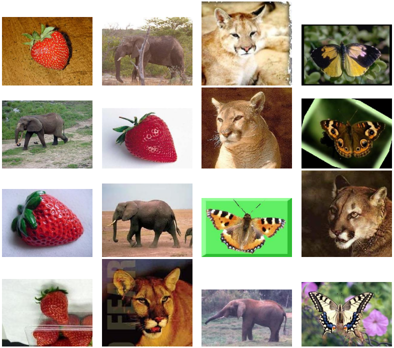
The Curse of Dimensionality

- The 1-NN rule might fail if the number of examples is smaller than $\frac{(c+1)^d}{2}$.
- The exponential dependence of the dataset size on the dimension of the data points is known as the curse of dimensionality.
- As a result, NN is usually performed in practice after dimensionality reduction preprocessing step.

Clustering

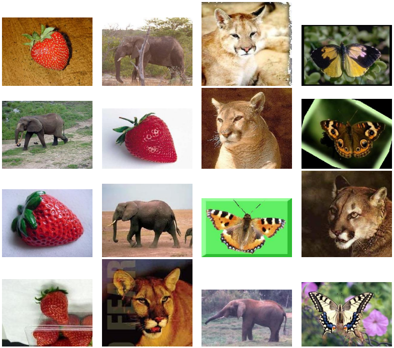


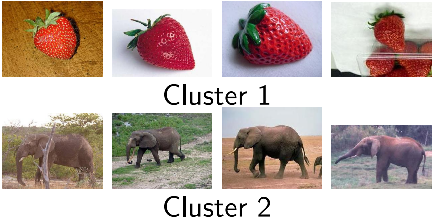
Clustering



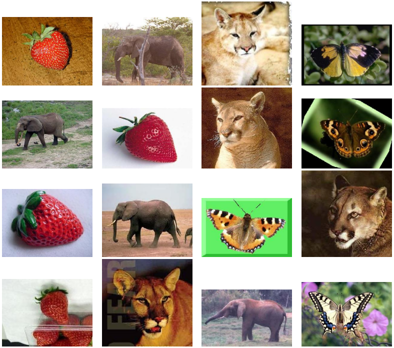
Cluster 1

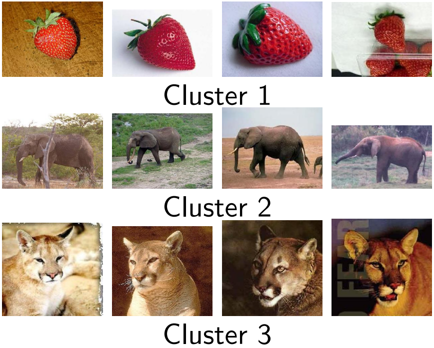
Clustering



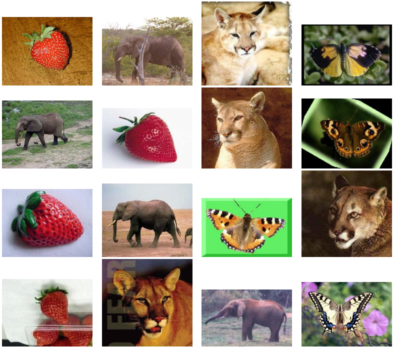


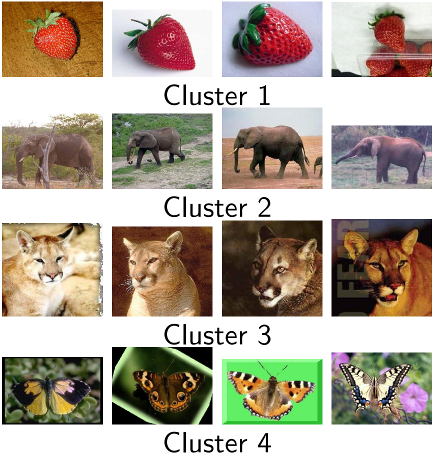
Clustering





Clustering





Clustering

- Identifying meaningful groups among the data points.

Clustering

- Identifying meaningful groups among the data points.
- For example, computational biologists cluster genes on the basis of similarities in their expression.

Clustering

- Identifying meaningful groups among the data points.
- For example, computational biologists cluster genes on the basis of similarities in their expression.
- Retailers cluster customers, on the basis of their customer profiles, for the purpose of targeted marketing.

Clustering

- Identifying meaningful groups among the data points.
- For example, computational biologists cluster genes on the basis of similarities in their expression.
- Retailers cluster customers, on the basis of their customer profiles, for the purpose of targeted marketing.
- Astronomers cluster stars on the basis of their spatial proximity.

Clustering

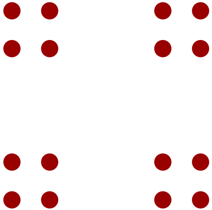
- Identifying meaningful groups among the data points.
- For example, computational biologists cluster genes on the basis of similarities in their expression.
- Retailers cluster customers, on the basis of their customer profiles, for the purpose of targeted marketing.
- Astronomers cluster stars on the basis of their spatial proximity.
- Clustering is the task of grouping a set of objects such that similar objects end up in the same group and dissimilar objects are separated into different groups.

Clustering

A set of elements, $\mathcal{X}$, and a distance function over it. That is, a function $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ that gives distance between two elements.

Clustering

A set of elements, $\mathcal{X}$, and a distance function over it. That is, a function $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ that gives distance between two elements.



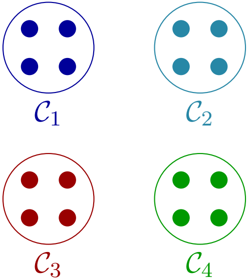
$\mathcal{X}$

Clustering

A partition of the input domain $\mathcal{X}$ into subsets. That is, $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ where $\cup_{i=1}^{k} \mathcal{C}_i = \mathcal{X}$, and $\mathcal{C}_i \cap \mathcal{C}_j = \phi, \forall i \neq j$.

Clustering

A partition of the input domain $\mathcal{X}$ into subsets. That is, $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ where $\cup_{i=1}^{k} \mathcal{C}_i = \mathcal{X}$, and $\mathcal{C}_i \cap \mathcal{C}_j = \phi, \forall i \neq j$.



$$\mathcal{X} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4$$

Clustering Algorithms

- Connectivity Based: Hierarchical Clustering

Clustering Algorithms

- Connectivity Based: Hierarchical Clustering
- Centroid models: k-Means

Clustering Algorithms

- Connectivity Based: Hierarchical Clustering
- Centroid models: k-Means
- Graph-based models: Spectral Clustering

Clustering Algorithms

- Connectivity Based: Hierarchical Clustering
- Centroid models: k-Means
- Graph-based models: Spectral Clustering
- Distribution models: Expectation Maximization

Clustering Algorithms

- Connectivity Based: Hierarchical Clustering
- Centroid models: k-Means
- Graph-based models: Spectral Clustering
- Distribution models: Expectation Maximization
- Density models: DBSCAN

Clustering Algorithms

- Connectivity Based: Hierarchical Clustering
- Centroid models: k-Means
- Graph-based models: Spectral Clustering
- Distribution models: Expectation Maximization
- Density models: DBSCAN
- Neural models: Self-organizing map

Clustering Algorithms

- Connectivity Based: Hierarchical Clustering
- Centroid models: k-Means
- Graph-based models: Spectral Clustering
- Distribution models: Expectation Maximization
- Density models: DBSCAN
- Neural models: Self-organizing map