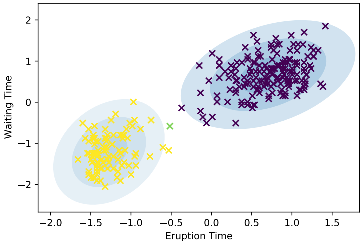


CSE 151A

Intro to Machine Learning

Lecture 15 – Part 01

Supervised and Unsupervised Learning

Supervised Learning

- We tell the machine the "right answer".
  - There is a **ground truth**.
- Data set: $\{(\vec{x}^{(i)}, y_i)\}$.
- **Goal**: learn relationship between features $\vec{x}^{(i)}$ and labels y_i.
- **Examples**: classification, regression.

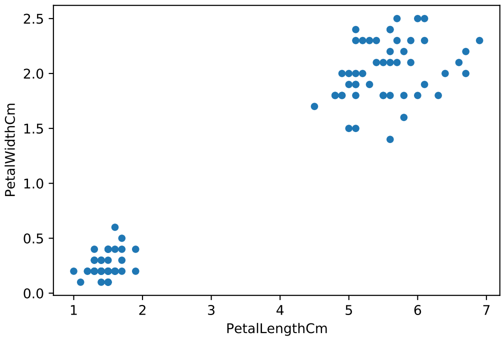
Unsupervised Learning

- We don’t tell the machine the “right answer”.
  - In fact, there might not be one!
- Data set: $\vec{x}^{(i)}$ (usually no test set)
- **Goal**: learn the **structure** of the data itself.
  - To discover something, for compression, to use as a feature later.
- **Example**: **clustering**

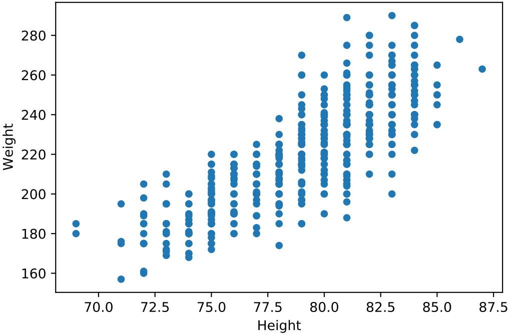
Example

- We gather measurements $\vec{x}^{(i)}$ of a bunch of flowers.
- **Question**: how many species are there?
- **Goal**: **cluster** the similar flowers into groups.

Example



Example

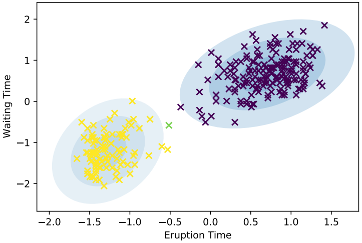

Weight
280
260
240
220
200
180
160
70.0
72.5
75.0
77.5
80.0
82.5
85.0
87.5
Height

Clustering and Dimensionality

- Groups emerge with more features.
- But too many features, and groups disappear.
  - **Curse of dimensionality**.
- **Also**: We can't see in $d > 3$.

Ground Truth

- If we don’t have labels, we can’t measure accuracy.
- Sometimes, labels don’t exist.
- Example: cluster customers into types by previous purchases.



CSE 151A

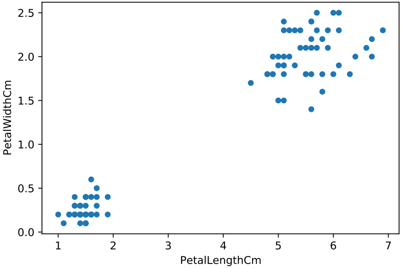
Intro to Machine Learning

Lecture 15 – Part 02

K-Means Clustering

Learning

- **Goal**: turn clustering into optimization problem.
- **Idea**: clustering is like **compression**



K-Means Objective

- **Given**: data, $\{\vec{x}^{(i)}\} \in \mathbb{R}^d$ and a parameter k.
- **Find**: k cluster centers $\vec{\mu}^{(1)}, \ldots, \vec{\mu}^{(k)}$ so that the average squared distance from a data point to nearest cluster center is small.
- The **k-means objective function**:

$$\mathrm{Cost}(\vec{\mu}^{(1)}, \ldots, \vec{\mu}^{(k)}) = \frac{1}{n} \sum_{i=1}^{n} \min_{j \in \{1,\ldots,k\}} \|\vec{x}^{(i)} - \vec{\mu}^{(j)}\|^2$$

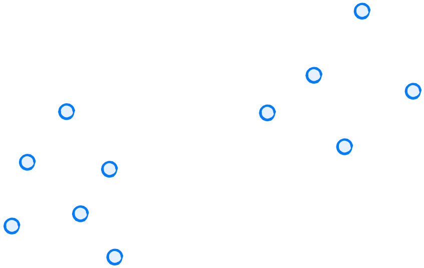
Optimization

- **Goal**: find $\vec{\mu}^{(1)}, \ldots, \vec{\mu}^{(k)}$ minimizing k-means objective function.
- **Problem**: this is NP-Hard.
- We use a heuristic instead of solving exactly.

Lloyd's Algorithm for K-Means

- Initialize centers, $\vec{\mu}^{(1)}, \ldots, \vec{\mu}^{(k)}$ somehow.
- Repeat until convergence:
    - Assign each point $\vec{x}^{(i)}$ to closest center
    - Update each $\vec{\mu}^{(i)}$ to be mean of points assigned to it

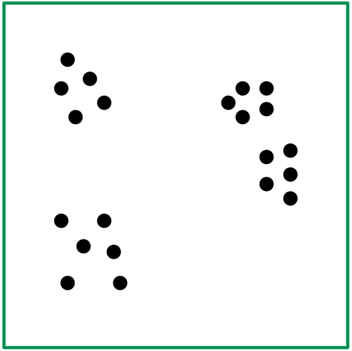
Example

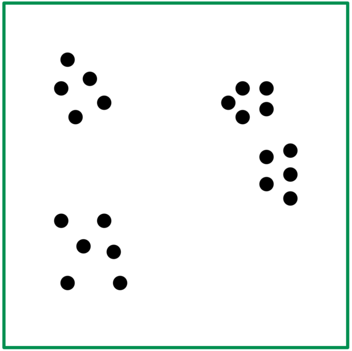


Theory

- Each iteration reduces cost.
- This guarantees convergence to a **local** min.
- Initialization is very important.

Example





Initialization Strategies

- Basic Approach: Pick k data points at random.
- Better Approach: **k-means++**:
  - Pick first center at random from data.
  - Let $C = \{\vec{\mu}^{(1)}\}$ (centers chosen so far)
  - Repeat $k - 1$ more times:
    - Pick random data point $\vec{x}$ according to distribution

$$\mathbb{P}(\vec{x}) \propto \min_{\vec{\mu} \in C} \|\vec{x} - \mu\|^2$$

    - Add $\vec{x}$ to C

Picking k

- How do we know how many clusters the data contains?

Plot of K-Means Objective

Applications of K-Means

- Discovery
- Vector Quantization
  - Find a finite set of representatives of a large (possibly infinite) set.

Example #1

- Cluster animal descriptions.
- 50 animals: grizzly bear, dalmatian, rabbit, pig, ...
- 85 attributes: long neck, tail, walks, swims, ...
- 50 data points in $\mathbb{R}^{85}$. Run k-means with $k = 10$

Results

1. zebra
2. spider monkey, gorilla, chimpanzee
3. tiger, leopard, wolf, bobcat, lion
4. hippopotamus, elephant, rhinoceros
5. killer whale, blue whale, humpback whale, seal, walrus, dolphin
6. giant panda
7. skunk, mole, hamster, squirrel, rabbit, bat, rat, weasel, mouse, raccoon
8. antelope, horse, moose, ox, sheep, giraffe, buffalo, deer, pig, cow
9. beaver, otter
10. grizzly bear, dalmatian, persian cat, german shepherd, siamese cat, fox, chihuahua, polar bear, collie

1. zebra
2. spider monkey, gorilla, chimpanzee
3. tiger, leopard, fox, wolf, bobcat, lion
4. hippopotamus, elephant, rhinoceros, buffalo, pig
5. killer whale, blue whale, humpback whale, seal, otter, walrus, dolphin
6. dalmatian, persian cat, german shepherd, siamese cat, chihuahua, giant panda, collie
7. beaver, skunk, mole, squirrel, bat, rat, weasel, mouse, raccoon
8. antelope, horse, moose, ox, sheep, giraffe, deer, cow
9. hamster, rabbit
10. grizzly bear, polar bear

Example #2

- How do we represent images of different sizes as fixed length feature vectors for use in classification tasks?

Visual Bags-of-Words

- **Idea**: build a “dictionary” of image patches.
- Extract all $\ell \times \ell$ image patches from all training images.
- Cluster them with k-means.
  - Each cluster center is now a dictionary “word”
- Represent an image as a histogram over $\{1, 2, \ldots, k\}$ by associating each patch with nearest center.

Online Learning

- What if the dataset is huge?
  - It doesn't even fit in memory.
- What if we're continuously getting new data?
  - Don't want to retrain with every new point.
- We can update the model **online**.

Sequential k-Means

- Set the centers $\vec{\mu}^{(1)}, \ldots, \vec{\mu}^{(k)}$ to be first k points

- Set counts to be $n_1 = n_2 = \ldots = n_k = 1$.

- Repeat:
  - Get next data point, $\vec{x}$
  - Let $\vec{\mu}^{(j)}$ be closest center
  - Update $\vec{\mu}^{(j)}$ and n_j:

$$\vec{\mu}^{(j)} = \frac{n_j \vec{\mu}^{(j)} + \vec{x}}{n_j + 1} \qquad n_j = n_j + 1$$

K-Means

- Perhaps the most popular clustering algorithm.
- **Fast, easy to understand.**
- **Assumes spherical clusters.**

Example