Lecture 15: 3/8/17

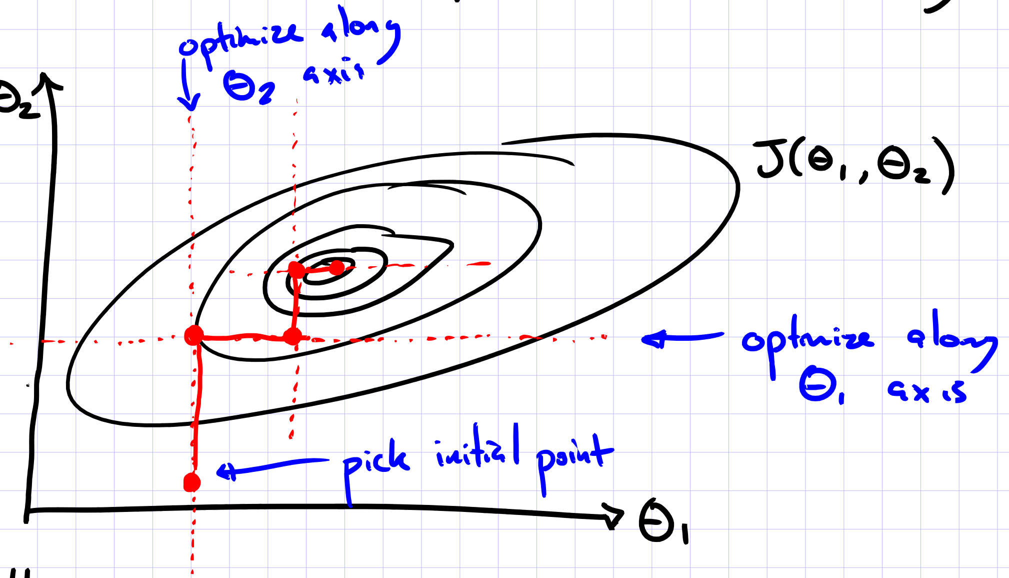
Optimization Background: Coordinate Descent

Goal: Minimize a function $J(\vec{\theta})$

e.g. $\hat{\theta} = \underset{\vec{\theta}}{\operatorname{argmin}}\ J(\vec{\theta})$

Idea: Pick one dimension, and minimize along that dimension.

Ex:



Algorithm:

① Choose initial point $\vec{\theta}$

② Repeat until stopping criterion is reached.

All the blue variables are "fixed" while we minimize over green vars.

$$\theta_1 = \underset{\theta_1}{\operatorname{argmin}}\ J(\theta_1, \theta_2, \ldots, \theta_M)$$

$$\theta_2 = \underset{\theta_2}{\operatorname{argmin}}\ J(\theta_1, \theta_2, \ldots, \theta_M)$$

$$\vdots$$

$$\theta_M = \underset{\theta_M}{\operatorname{argmin}}\ J(\theta_1, \theta_2, \ldots, \theta_M)$$

each step is an exact line search along some axis

③ Return $\vec{\theta}$

Block Coordinate Descent

Here: An example w/ two blocks $\vec{\alpha}, \vec{\beta}$, where $\vec{\theta} = \begin{bmatrix} \vec{\alpha} \\ \vec{\beta} \end{bmatrix}$

Goal: $\vec{\alpha}, \vec{\beta} = \underset{\vec{\alpha}, \vec{\beta}}{\operatorname{argmin}}\ J(\vec{\alpha}, \vec{\beta})$, $\vec{\alpha} \in \mathbb{R}^A$ $\vec{\beta} \in \mathbb{R}^B$

Idea: Minimize over an entire group of variables at a time.

Algorithm:

① Choose initial point $\vec{\alpha}, \vec{\beta}$

② Repeat until stopping criterion

$$\vec{\alpha} = \underset{\vec{\alpha}}{\operatorname{argmin}}\ J(\vec{\alpha}, \vec{\beta})$$

$$\vec{\beta} = \underset{\vec{\beta}}{\operatorname{argmin}}\ J(\vec{\alpha}, \vec{\beta})$$

blue is "fixed" aka. held constant, while minimizing over green.

Clustering

← our first example of unsupervised learning

GOAL: partition unlabeled instances into groups of "similar" points

Input: Unlabeled data: $\mathcal{D} = \{\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(N)}\}$, $\vec{x}^{(i)} \in \mathbb{R}^M$

★ We do not know the labels of the training examples.

Output:

View #1:

Labeled Instances

$\{(\vec{x}^{(1)}, z^{(1)}), (\vec{x}^{(2)}, z^{(2)}), \ldots, (\vec{x}^{(N)}, z^{(N)})\}$

where $z^{(i)} \in \{1, \ldots, K\}$

these cluster assignments are predictions

# of clusters

View #2:

Clusterings $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K$ ← capital C

$\mathcal{C}_j = \{\vec{x}^{(i)} : z^{(i)} = j\}$

← the points in the j^{th} partition.

Important Questions:

- How many clusters are there?
- How do we define "similarity" between points?

Objective-Based Clustering

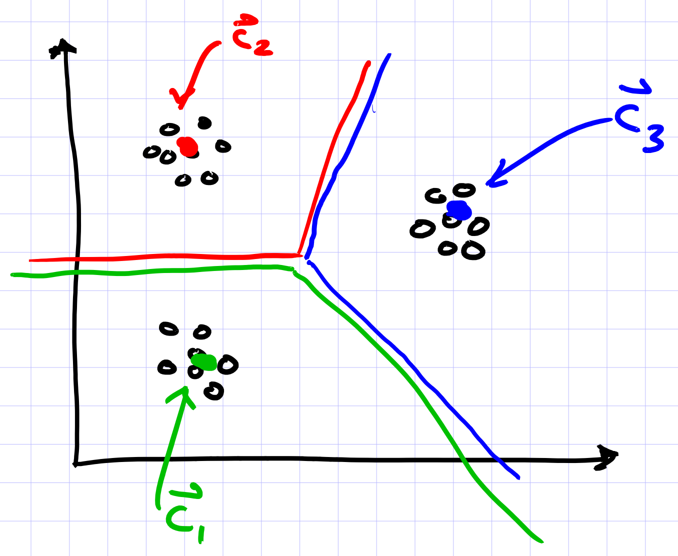
Ex: K-Means Objective.

Input: $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

lower case c.

Cluster Centers: $\vec{c} \triangleq \{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_K\}$

Decision Rule: Assign $\vec{x}^{(i)}$ to its nearest cluster center $\vec{c}_j$



Voronoi Diagram

Objective:

$$\underbrace{\vec{c} = \underset{\vec{c}}{\operatorname{argmin}} \sum_{i=1}^{N} \overbrace{\min_{j \in \{1,\ldots,K\}} \|\vec{x}^{(i)} - \vec{c}_j\|^2}}$$

Each point should as close as possible to its cluster center.

← c_j is the cluster center

All points should be close to their cluster centers.

Equivalent Objective:

Let $\vec{z}$ be all the cluster assignments $\vec{z} = [z^{(1)}, \ldots, z^{(N)}]$
$z^{(i)} \in \{1, \ldots, K\}$

$$\vec{c} = \underset{\vec{c}}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{z^{(i)}} \|x^{(i)} - c_{z^{(i)}}\|^2$$

push min out of sum.

$$\equiv \quad \vec{c}, \vec{z} = \underset{\vec{c}, \vec{z}}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{N} \|x^{(i)} - c_{z^{(i)}}\|^2}$$

→ call this $J_{KMeans}(\vec{c}, \vec{z})$

$$= \underset{\vec{c}, \vec{z}}{\operatorname{argmin}} J_{KMeans}(\vec{c}, \vec{z})$$

Question: How should we optimize $J_{KMeans}(\vec{c}, \vec{z})$?

Computational Complexity of K-Means Obj. Minimization Problem

① Objective is nonconvex

② NP-Hard, even for $k=2$ ← # clusters

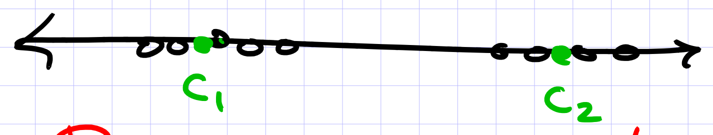
even for $M=2$ ← # features

Easy Case #1 : $K=1$

$$\vec{c}_1 = \underset{\vec{c}_1}{\operatorname{argmin}} \sum_{i=1}^{N} \|\vec{x}^{(i)} - \vec{c}_1\|^2$$

$$= \frac{1}{N} \sum_{i=1}^{N} \vec{x}^{(i)}$$ ← mean

Easy Case #2 : $M=1$



Dynamic programming in time $O(N^2K)$

K-Means in Practice

- Solve minimization problem heuristically w/ Block Coord. Descent.

K-Means Algorithm :

① Given $\vec{x}^{(1)}, \dots, \vec{x}^{(N)}$

② Initialize cluster centers $\vec{c} = \{\vec{c}_1, \dots, \vec{c}_K\}$

Initialize cluster assignments $\vec{z} = \{z^{(1)}, \dots, z^{(N)}\}$

③ Repeat until objective stops changing.

a) $$\vec{c} = \underset{\vec{c}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\vec{x}^{(i)} - \vec{c}_{z^{(i)}}\|^2$$ (Min over centers, w/ assignments fixed)

b) $$\vec{z} = \underset{\vec{z}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\vec{x}^{(i)} - \vec{c}_{z^{(i)}}\|^2$$ (Min over assignments, w/ centers fixed)

$$\boxed{\begin{array}{l} \vec{a} = \underset{\vec{a}}{\operatorname{argmin}}\, f(a_1) + g(a_2) \\ \equiv \\ a_1 = \underset{a_1}{\operatorname{argmin}}\, f(a_1) \\ a_2 = \underset{a_2}{\operatorname{argmin}}\, g(a_2) \end{array}}$$

3a) decomposes:

$$\sum_{i=1}^{N} \|\vec{x}^{(i)} - \vec{c}_{z^{(i)}}\|^2 = \sum_{j=1}^{K} \sum_{i: z^{(i)}=j} \|\vec{x}^{(i)} - \vec{c}_j\|^2$$

$$\vec{c}_1 = \underset{\vec{c}_1}{\operatorname{argmin}} \sum_{i: z^{(i)}=1} \|\vec{x}^{(i)} - \vec{c}_1\|^2$$

$$\vec{c}_2 = \underset{\vec{c}_2}{\operatorname{argmin}} \dots$$

$\vdots$

$$\vec{c}_K = \underset{\vec{c}_K}{\operatorname{argmin}} \sum_{i: z^{(i)}=K} \|x^{(i)} - \vec{c}_K\|^2$$

Each is just Easy Case #1 :

$$\vec{c}_j = \underset{\vec{c}_j}{\operatorname{argmin}} \sum_{i: z^{(i)}=j} \|\vec{x}^{(i)} - \vec{c}_j\|^2$$

$=$ mean of points in cluster j

$$= \frac{1}{N_j} \sum_{i: z^{(i)}=j} \vec{x}^{(i)}$$

3b) decomposes

$$z^{(1)} = \underset{j}{\operatorname{argmin}} \|x^{(1)} - c_j\|^2$$

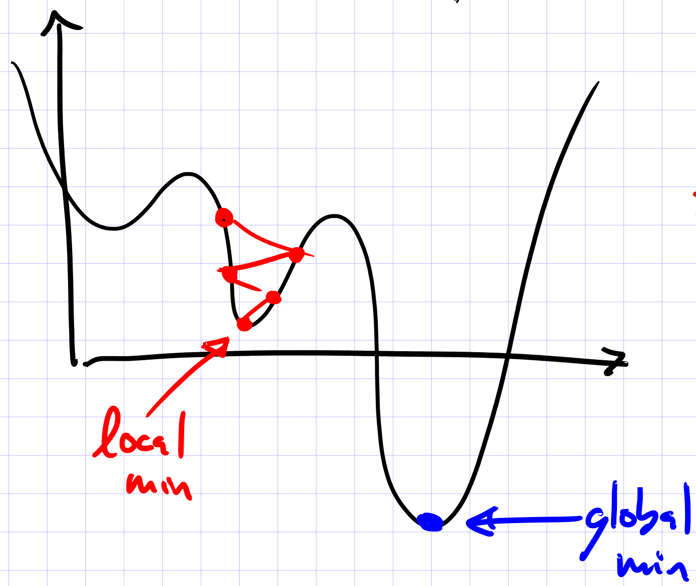
$\vdots$

$$z^{(N)} = \underset{j}{\operatorname{argmin}} \|x^{(N)} - c_j\|^2$$

just find the closest cluster center for each point $\vec{x}^{(i)}$.

Initialization

This is a local opt. alg. for a nonconvex objective.



K-Means just finds a local min.

$\Rightarrow$ How we initialize $\vec{c}, \vec{z}$ is crutial.

Three Options:

① Random: select points uniformly at random (w/o replacement) from dataset D

② Furthest Traversal:
select points in D s.t. c_j is as far as possible from $c_1, \ldots, c_{j-1}$

③ K-Means++:
interpolate between ① and ②
★ good theoretical guarantees.