MATH 462 LECTURE NOTES WEEK 1

ADAM M. OBERMAN

1. Week 1: k means clustering

This note covers lectures 1 and 2. References

- Clustering [SSBD14, Chapter 22]
- Vector Calculus [DFO20, Chapter 5]

1.1. **Introduction and problem setup.** In k-means clustering, we want to partition the data into k sets, where each partition contains similar data. In our case we consider vector data and use distance as measure of similarity.

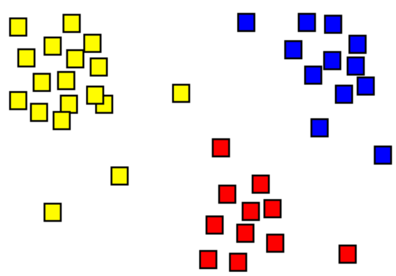

Figure 1. Example of a $k = 3$ cluster

Givens.

- a dataset, S^m, consisting of m vectors in d-dimensions, $\mathbb{R}^d$.

$$S^m = \{x_1 \dots, x_m\}$$

- k, the number of partitions required.

Goal: We want to partition the data into k disjoint sets,

$$S^m = C_1 \cup C_2 \cup \cdots \cup C_k$$

in such a way that 'similar' points belong to the same partition. Each partition C_j is represented by a vector, w_j, which is called a 'mean'.

Date: September 26, 2022.

Model: Similarity is a semantic[1] relation. It is replaced by the a mathematical relation of distance. The distance function we use is the usual Euclidean distance, $d(x, x') = \|x - x'\|$, where

$$\|x - y\|^2 = (x_1 - y_1)^2 + \cdots + (x_d - y_d)^2$$

Formally our model substitutes semantic similarity for *geometric* similarity via

$$d(x, x') \text{ small means } x \text{ and } x' \text{ are similar}$$

Method: The k-means algorithm.

Randomly choose initial means $W = (w_1, \ldots, w_k)$.

- Assign each point x in dataset S^m to the cluster C_i corresponding to the closest mean w_i.
- Update the means by setting w_i to be the mean of the vectors in the cluster C_i

Repeat until convergence (meaning the w don't change).

Example 1.1. Do a one dimensional example.

1.2. **Discussion.** Clustering is visually simple and the algorithm is also simple to implement and understand.

In what follows, we will *deliberately make things complicated*. Why? We are using this digestible example of k-means clustering to introduce some concepts which will appear later in a more complicated context.

Analysis:

- We will analyze the problem, using simple examples to show what can happen.
- We will give a variational interpretation of the algorithm, and prove that each step of the algorithm improves the cluster, until the algorithm terminates at a fixed point.

2. Analysis via examples

[Pictures]

3. Analysis via loss

3.1. **Hypothesis class of partition functions.** Given k vectors $w_1, \ldots w_k$, written as the single array of vectors $W = (w_1, \ldots, w_k)$ define the hypothesis class of functions

$$\mathcal{H} = \{h_W : \mathbb{R}^d \to \mathbb{R}^d \mid W = (w_1, \ldots, w_k) \in \mathbb{R}^{d \times k}\}$$

where each function is given by

$$h_W(x) = w^*(x) = \operatorname*{arg\,min}_{w \in \{w_1, \ldots, w_k\}} \|x - w_i\|^2 \tag{1}$$

So $h_W(x)$ returns *the closest* w_i *to* x.[2]

Note: $h_W(x)$ is *piecewise constant*. The pieces are determined by the sets

$$V_j = \{x \in \mathbb{R}^d \mid h(x) = w_j\}$$

which are the Voronoi cells corresponding to the points `https://en.wikipedia.org/wiki/Voronoi_diagram`. See Figure 2.

[1]semantic: relating to meaning

[2]We leave the function undefined at the points where there is more than one minimizer

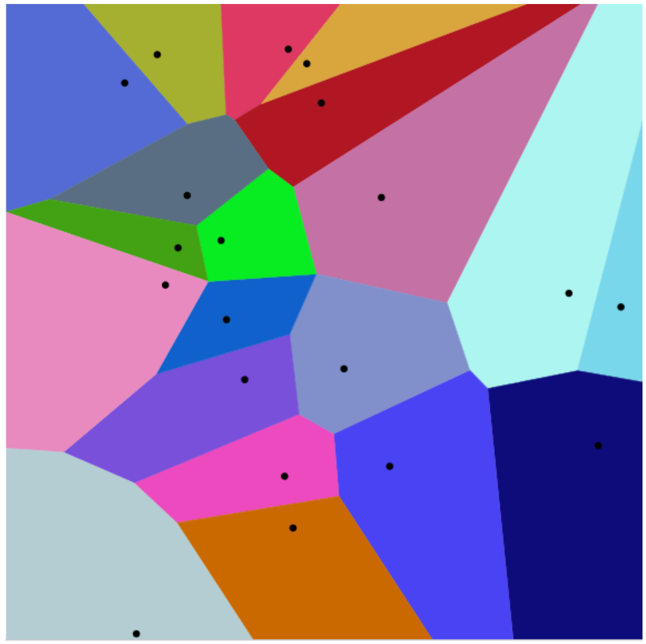

FIGURE 2. Voronoi diagram illustrating the function

Define the partition C_j by

$$C_j = \{x \in S^m \mid h(x) = w_j\} \tag{2}$$

3.2. **Loss functional.**

Definition 3.1 (Empirical Loss functional)**.** Given a dataset S^m and a function $h : \mathbb{R}^d \to \mathbb{R}^d$, define the empirical loss functional to be the average squared distance from a point to its image under the transformation $h(x)$,

$$\widehat{L}(h) = L(h, S^m) = \frac{1}{m} \sum_{i=1}^{m} \|h(x_i) - x_i\|^2 \tag{3}$$

Remark 3.2. The term *functional* is used for a function L that inputs another function and returns a number. The correct notation is $L(h, S^m)$ to indicate the dependence on the dataset. The term *empirical* and the notation $\widehat{L}$ is a shorthand which hides the dependence of the loss on the dataset. The loss (3) is an example of a typical loss functional, which has the form

$$\widehat{L}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i), x_i)$$

in the case of the loss $\ell(x_1, x_2) = \|x_1 - x_2\|^2$, The k-means loss functional by

$$\widehat{L}(h_W) = \frac{1}{m}\sum_{i=1}^{m} \|h_W(x_i) - x_i\|^2$$

Lemma 3.3. *Given a function h_W of the form* (1)*, we can write*

$$\widehat{L}(h_W) = \frac{1}{m}\sum_{j=1}^{k}\sum_{x\in C_j} \|x - w_j\|^2$$

Proof. Rewrite the loss as

$$\begin{aligned}
\widehat{L}(h_W) &= \frac{1}{m}\sum_{j=1}^{k}\sum_{x\in C_j} \|x - h_W(x)\|^2 && \text{since } C_1, \dots C_k \text{ is partition of } S^m \\
&= \frac{1}{m}\sum_{j=1}^{k}\sum_{x\in C_j} \|x - w_j\|^2 && \text{by definition (2)}
\end{aligned}$$

□

3.3. **Algorithm.** Here we rewrite the simple k-means algorithm described above in terms of the hypothesis.

Given an initial (e.g. random) choice of W^0, for any t, given W^t, define

$$w_j^{t+1} = \arg\min_{w\in\mathbb{R}^d} \sum_{x\in C_j} \|x - w\|^2, \qquad j = 1, \dots, k \tag{4}$$

thus *in each cluster, the w_j^t is updates to one which improves the sum of the distances over the cluster*

Remark 3.4. In other parts of the course, we will consider algorithms which update the loss using a gradient with respect to the weights. However, in this case, gradient based algorithm are not appropriate because h_W is piecewise constant, so not really differentiable in W.

Lemma 3.5. *Suppose we update h_W according to* (4)*. Then we have*

$$\widehat{L}(h_W^{t+1}) \le \widehat{L}(h_W^t)$$

with a strict inequality, unless $W^{t+1} = W^t$

4. (NOT COVERED) Interpretation: Generative and Discriminate

4.1. **Generative model.** A generative model is a way of generating new data points. For example, in statistics, Gaussian model, learn the parameters (mean and variance), and can then generate new data, provided we can sample from a Gaussian (which we can).

We can interpret the k-means function h_W as a generative model for the data, as follows.

Definition 4.1. Given the means μ_i, define σ_i^2 to be the variances of each cluster, and p_i to be the fraction of data points in the cluster. Generate a new data point as follows:

- Choose an index j from $1, \dots, k$, with probability p_j.
- Generate a point x from the d-dimensional Gaussian with mean μ_j and variance σ_j.

We can ask the question, when will the generative model determined by the parameters 'match' the samples. For example, if the samples were generated by k Gaussians, variances σ_j^2 and means μ_j. Can we recover the parameters of the Gaussians.

Remark 4.2. In general this is a hard problem to solve, but if the Gaussians are widely separated with small (say constant) σ_j, then we expect the method to work

Example 4.3.
- Find a simple example where we recover the generating distribution. (Hint: can do a one dimensional example, with $k = 2$, and with points uniform on two intervals).
- Generalize this to a $d = 2$ example (with circles instead of intervals).
- Change the $d = 1$ example so it fails to recover the distribution (Hint: make the samples unbalanced, so more often from one).
- Change the $d = 2$ example so it fails to recover the distribution, using a different method from the previous example. (Hint: make the samples come from squares instead of circles).

4.2. **Discriminative model.** A discriminative model is one where we make a decision (e.g. classification). We can interpret the k-means function h_W as a discriminator as follow:

Definition 4.4. Given h_W. Given two points x, x'. Then x, x' are similar according to the function h_W if $h_W(x) = h_W(x')$

Example 4.5. Give examples where h_W succeeds or fails as a discriminator.

5. Exercises

The exercises are in a separate document

References

[DFO20] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.