Linear Data Analysis
Unsupervised Learning: K-Means Clustering

Cain Susko

Queen's University
School of Computing

March 8, 2022

a A Conceptual Hierarchy of Machine Learning

the main concepts of thislecutre are: Hpyer planes, learning through clustering, the k-means algorithm.
we will use a simple Hierarchy of machine learning. there are 2 heirarchies:

- Unsupervised - find natural clusters
  - Binary Clustering
  - Multi-Class CLustering
- Supervised - cluster using predetermined labels
  - Binary Labelling
  - Multi-Class Labelling

b Hyperplane of Seperation

in 2D, we can seperate data with a line, in 3D we can seperate data with a plane. in N dimensions we can seperate data using a hyperplane.
A hyper plane is a vector space + a reference point

$2D$	$line$	$\vec{l}(z) = \vec{r} + z\vec{d}$
$3D$	$plane$	$\vec{p}(z) = \vec{r} + z_1\vec{d_1} + z_2\vec{d_2}$

Suppose $\mathbb{R}^m$. The Hyperplane $\mathbb{H}$ is defined as point $\vec{v}$ is in $\mathbb{H}$:
for $\vec{w} \neq \sigma$ and $b \in \mathbb{R}$:

$$\vec{w} \cdot \vec{v} + b = 0$$

$$\vec{w} \cdot \vec{v} = -b$$

where $\vec{w}$ is the **normal vector** and b is the **biased scalar**. We may also sometimes need the unit normal vector, which is:

$$\vec{n} \cdot \vec{v} = -a$$

HyperPlane Partitioning

given the point $\vec{u} \in \mathbb{R}^m$ and the hyperplane $\mathbb{H} : \vec{h}, \vec{r}$
to find out if $\vec{u}$ is on one side or the other of $\mathbb{H}$, we must project the reference point $\vec{r}$ and the point $\vec{u}$ into the vectorspace of the hyperplane: $\vec{h}$.

$$\begin{aligned}(\vec{u} - \vec{r}) \cdot \vec{h} &= \vec{h} \cdot (\vec{u} - \vec{r}) \\ &= \vec{h} \cdot \vec{u} - \vec{h} \cdot \vec{r} \\ &= \vec{h} \cdot \vec{u} + a\end{aligned}$$

and thus, we will define $\vec{u}$ to be on the positive side of $\mathbb{H}$ because $\vec{h} \cdot \vec{u} + a \geq 0$ so do we need a unit normal? (see the final equation in b). Recall a inequality holds if both sides are multiplied by the same factor.
thus, if $\vec{u}$ is in the positive side of $\mathbb{H}$ then:

$$\vec{h} \cdot \vec{u} + a \geq 0$$

additionally, let $\vec{w} = ||\vec{w}||\vec{h}$. therefore:

$$\begin{aligned}||\vec{w}||(\vec{h} \cdot \vec{u} + a) &\geq ||\vec{w}||0 \\ \equiv ||\vec{w}||(\vec{h} \cdot \vec{u}) + b &\geq 0 \\ \equiv \vec{w} \cdot \vec{u} + b &\geq 0\end{aligned}$$

so what does this tell us? from evaluating the equation $\vec{w} \cdot \vec{u} + b$, if the result is positive, the point $\vec{u}$ is in the positive side of $\mathbb{H}$ and if the result is negative, $\vec{u}$ is on the negative side.
Also note that, $\vec{h} \cdot (\vec{u} - \vec{r})$ is the orthogonal distance from $\vec{u}$ to the hyperplane $\mathbb{H}$

Example suppose were given a unit normal: $\vec{h} = \begin{bmatrix}1\\0\end{bmatrix}$ with a reference point of $\vec{r} = \begin{bmatrix}2\\0\end{bmatrix}$ and a biased scalar of $a = -2$. we know that $\vec{h} \cdot \vec{u}$ equals the first entry of $\vec{u}$:

$$\vec{h} \cdot \vec{u} = u_1$$

this thus implies that:

$$\vec{h} \cdot \vec{u} + a \geq 0 \equiv u_1 - 2 \geq 0 \equiv u_1 \geq 2$$

thus we can say that any point who's first entry is greater than 2 is on the positive side of $\mathbb{H}$ and everything with first entry less than 2 is on the negative side of $\mathbb{H}$

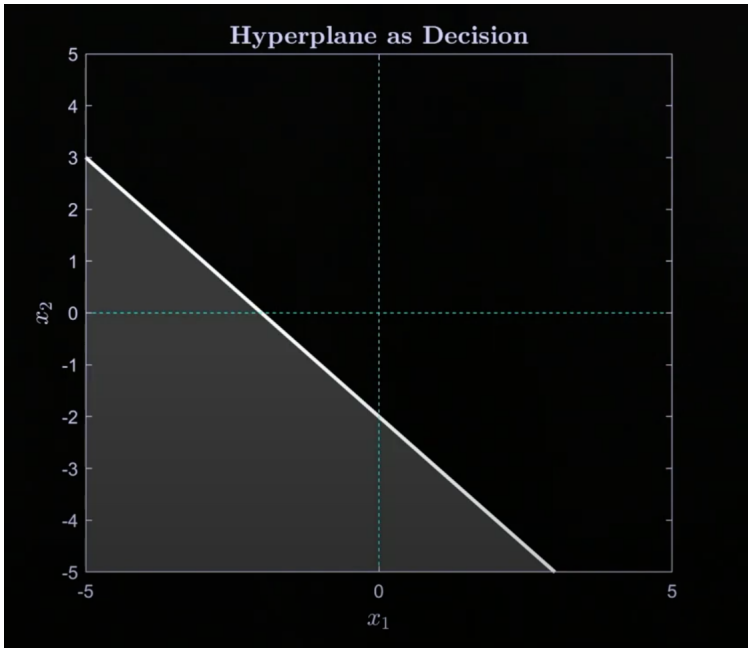
Example given:

$$\vec{w} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, b = 2$$

now, say the point is the zero vector, then:

$$\vec{w} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2 \geq 0$$

thus, the origin (0,0) is on the positive side of this hyperplane. we can view



$\mathbb{H}$ as a line or a 'decision' that the model will make.

c Basics of Vector Clustering

we shall expolre Vector Clustering. Assume a data vector is: $\vec{x} \in \mathbb{R}^m$
Additionally, a cluster is:

- A set s_i with $\vec{x}_j$ entries or
- A centriod $\vec{g}_i$ with neightbors.

Note: we will primarily use the **centriod** definition of a cluster

What is a Cluster thus, the question is; how do we define a cluster? A simple approach we can use is to treat vectors uniformly or
we can treat vectors hierarchically, meaning we will allow subsets.

l_2 **Centriod of a Set** the concept is to associate the centriod (e.g. mean) with a cluster S_i which is written as $\vec{g}_i$. We can do this in 2 ways:

i Define mean in terms of cluster

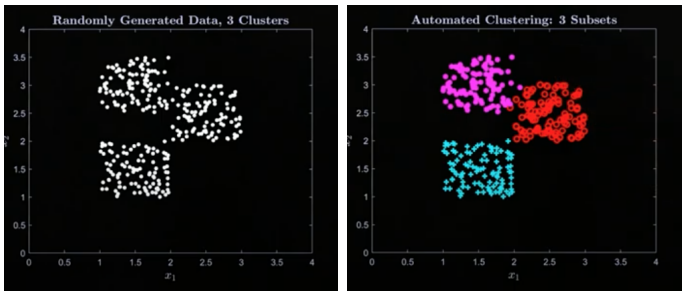
$$\vec{g}_i =^{def} \frac{\sum_{\vec{u} \in S_i} \vec{u}}{m_i}$$

ii Define cluster in terms of mean

$$S_i =^{def} \{\vec{u} \mid \forall_l ||\vec{u} - \vec{g}_i||^2 \geq ||\vec{u} - \vec{g}_l||^2\}$$

note that m is the mean of the vector. and $||x - y||^2$ is the distance between the points x, y

Observe the below example of clustering:



The number of clusters matters, as without the magenta, part of the magenta cluster would be clustered in with the red instead.

Funrthermore, clustering can change depending on new data, as the clusters' bounds will change with new points.

d K-Means Clustering

in clustering a problem that often comes up is: how many sets do we use? Usually, the user gives us this number as k such that:

$$S = S_1, S_2, ..., S_k$$

the difficulty is for $\vec{x} \in \mathbb{R}^m$ the complexity is $O(m^{nk+1})$
Instead, we will use Lloyd's k-means algorithm, which has a complexity of

$$O(mnkl)$$

where m is for the number of entries in each vector. n for each vector. k for each cluster, and l for each iteration.
the code for this in MatLab would be:

```
kmeans(Xmat, k)
```

Recall that a cluster is either a set S_i with $\vec{x}_i$ entries or a centriod $\vec{g}_i$ with neighbors.

Consider altering these defintions. For example: initialize contriods $\vec{g}_i$. The pseudo-code for this would be:

```
while not converged:
        for each vecx_j:
                assign index i of nearest vecg_i
                for each i:
                        compute vecg_i from points w/ index i
```

Note: the vector norm matters (ie. L_2 norm vs. Frobenius norm). for the Euclidian norm L_2 use the mean of $\vec{x}_i$. for the Manhattan norm L_1 use the median of $\vec{x}_i$

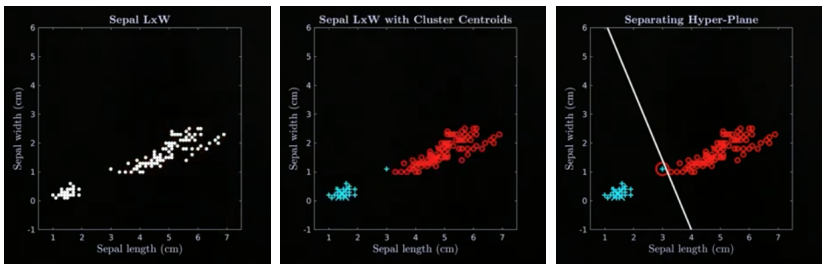
Clustering the Iris Data

this section will explore the best known dataset in machine learning: Fisher's Iris Data Set.

in MatLab, we can use `load fisheriris`. to get the data from there, we will use `Xmat = meas(:, 3:4);` to extract columns 3 and 4 from the data set.

What we want to do is find the index array and the row oriented centiods (means) for $k = 2$: `[kidx, kmrow] = kmeans(Xmat, 2)`

finally, we can plot the results



But We can see that there is a cyan data vector that is beyond the bounds of the majority of the cyan cluster.

But this can be explained by a seperating hyper-plane that partitions that 2 clusters. For every 2 centriods, there is a hyper-plane that divides them.