Quick Primer on Machine Learning: Unsupervised Learning

Giri Iyengar

Cornell University

giyengar@gmail.com

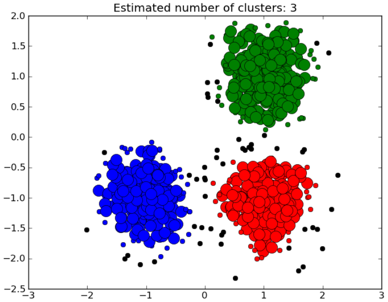
Jan 31, 2018

Overview

1. Clustering
2. Anomaly detection
3. Learning Latent Representations
4. Neural Network based approaches

Overview

Clustering

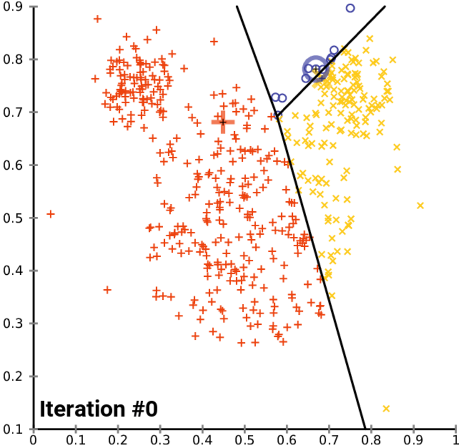

Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2
3

Clustering

- Given some data $(x_1, x_2, \ldots, x_N)$
- Represent them *compactly*
- Frequently done as part of exploratory analysis of your data
- Sometimes also done to make subsequent ML algorithms work better (e.g. make your classifier better)

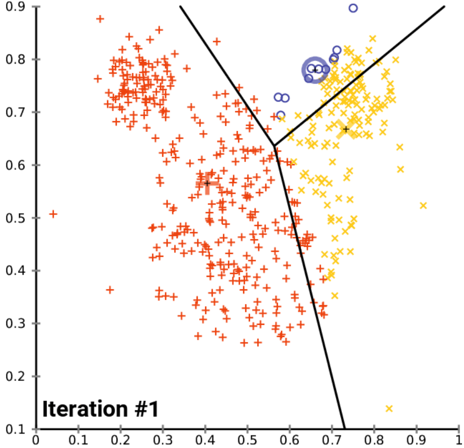
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



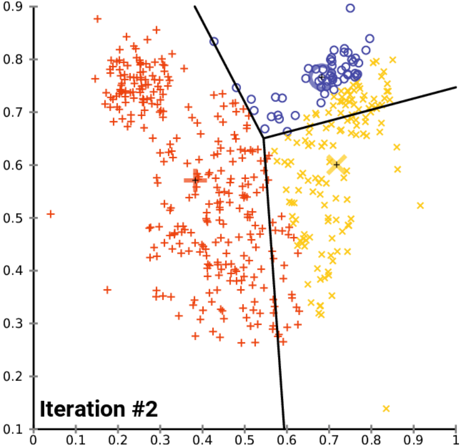
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



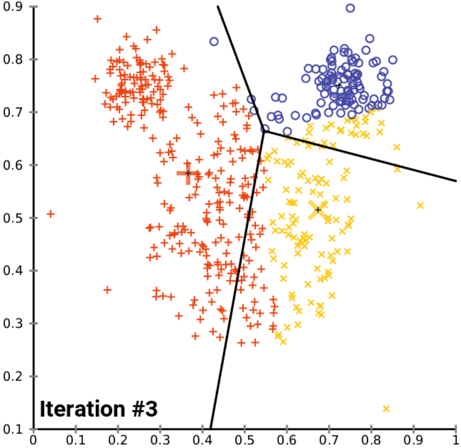
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



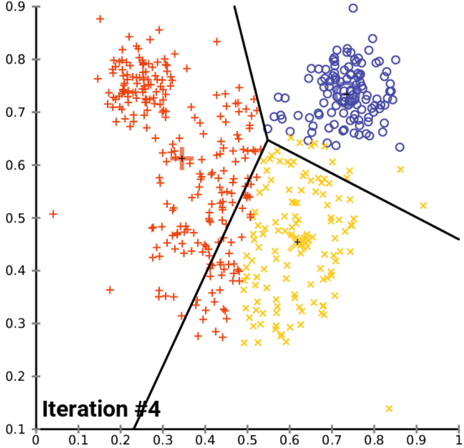
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



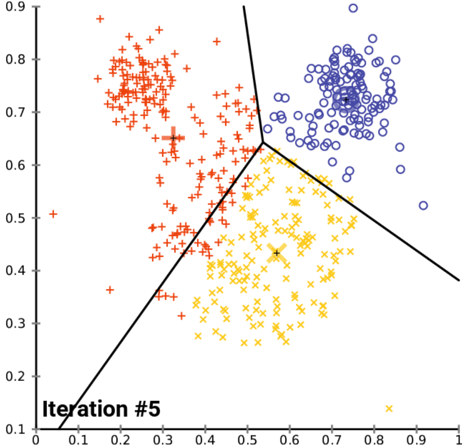
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



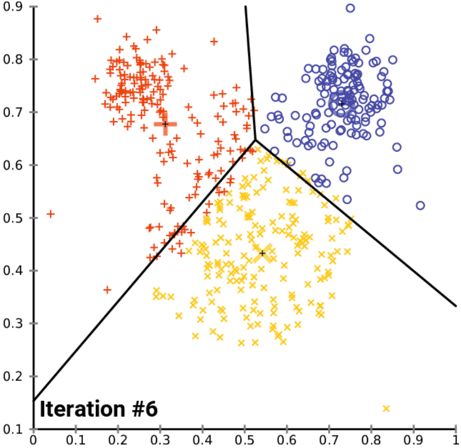
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



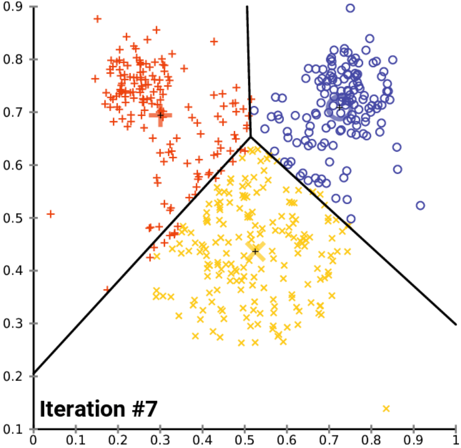
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



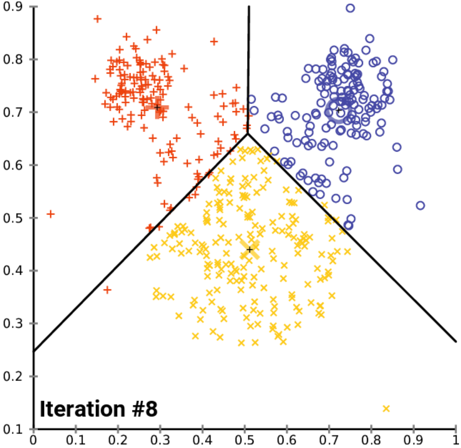
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



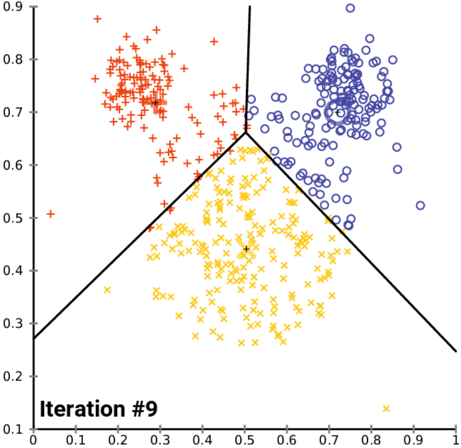
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



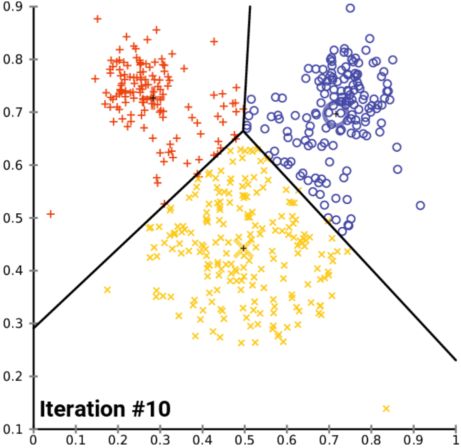
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



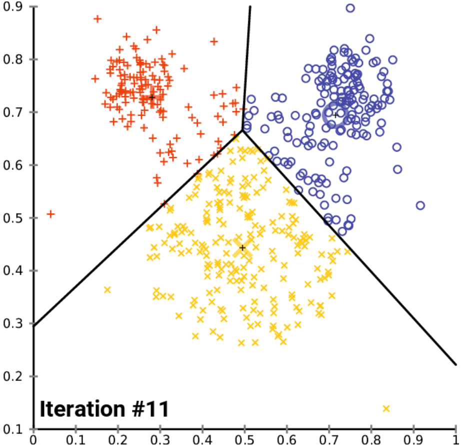
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



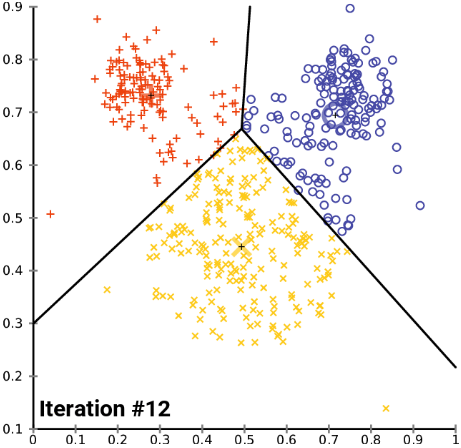
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



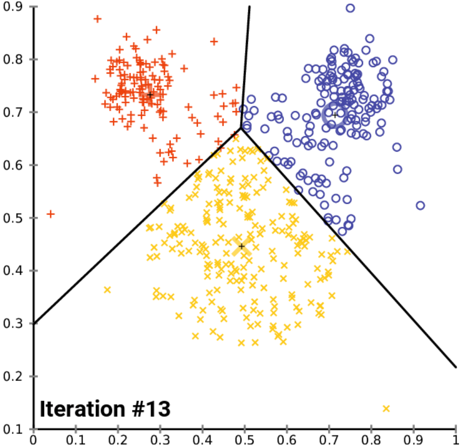
kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen

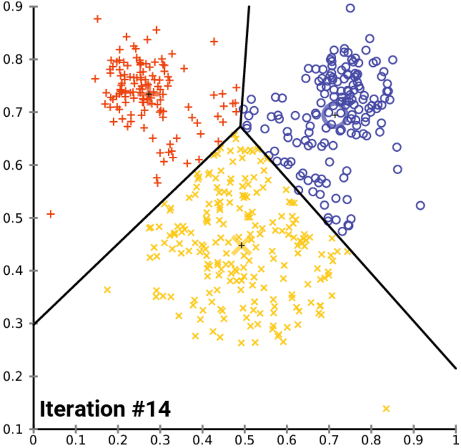


kMeans

- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



- Classic algorithm. Workhorse and useful in many, many contexts
- Very simple. Start with a random number of centroids
- Assign points to each centroid based on proximity
- Recalculate the new centroids from all points assigned to each
- Repeat till no change of assignment of points happen



Gaussian Mixture Models

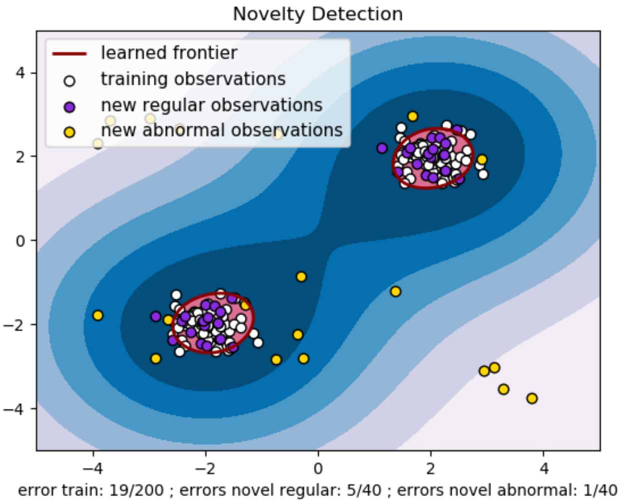
- Given a set of data points $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N})$
- Model them using a set of Gaussians centered around different points in the space
- Model the distribution as $p(\mathbf{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\mu_\mathbf{i}, \mathbf{\Sigma_i^2})$
- Evaluate the fit by estimating the data likelihood $p(x|\theta)$, for a given parameter setting θ
- **What are the parameters that determine the fit?**

Overview

Anomaly Detection

- Given a set of data points $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N})$
- Figure out if a new data point is *anomalous*
- Attack, Denial of Service, Virus, Failure, ...

Anomaly Detection



Anomaly Detection

- kNN
- Local outlier rejection
- One-class SVM

Overview

Learning Latent Representations

- Very popular in Natural Language Processing
- Latent Semantic Analysis
- Dimensionality Reduction (e.g. PCA)
- Independent Component Analysis
- Word-embeddings (e.g. Word2Vec)
- Autoencoders

Latent Semantic Analysis

- Consider the **Term-Document** matrix
- Approximate this with a low-rank approximation
- This helps eliminate **noise** and **merge** similar words

- d1: Romeo and Juliet.
- d2: Juliet: O happy dagger!
- d3: Romeo died by dagger.
- d4: ?Live free or die?, that?s the New-Hampshire?s motto.
- d5: Did you know, New-Hampshire is in New-England.

Latent Semantic Analysis

Documents >>	d1	d2	d3	d4	d5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
die	0	0	0	1	0
new-hampshire	0	0	0	1	1

Table: Document-Term matrix

Principal Components Analysis

- Given a k-dimensional dataset with possibly correlated dimensions
- Project the data into a set of orthogonal dimensions that are much smaller than k
- Useful for exploratory data analysis
- Dimensionality reduction prior to modeling using another ML algorithm

Independent Component Analysis

- Attempts to decompose a multi-variate signal into independent, additive, non-gaussian components
- Example: Separating individual sources from a mixed audio signal (cocktail party effect)
- As long as the independence assumptions are valid, produces good results
- ICA Demo

Overview


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
dziękuję
bedankt
спасибо
Баярлалаа
рахмат
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
grazie
arigatô
dakujem
мерси
감사합니다
ありがとう
ευχαριστώ
tapadh leat
dank je
hvala
sagolun
takk
taiku
tack
mersi
barka
welalin
spas
kia ora
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
murakoze
tenki
djiere dieuf
tau
дякую
mamnun
chnorakaloutioun
gratias ago
gracies
sulpáy
faafetai lava
kiitos
dankie
dhanyavad
nanni
nandri
bayarlalaa
gracie
mauruuru
köszönöm
enkosi
sobodi
dêkuji
mèsi
didi madloba
kam sah hamnida
najis tuke
rahmat
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
merce
trugarez
xiexie
dankon
akun
aciu