Unsupervised Learning

1-бөлім: Бақылаусыз оқыту дегеніміз не?

Деректер жинағы мен алгоритмнен бастап, жоғары деңгейде бақылаусыз оқытудың не екенін түсінуден бастайық.

Бақылаусыз оқыту

Бізде белгілерсіз деректер жинағы бар. Біздің мақсатымыз - деректер құрылымы туралы қызықты нәрсе білу:

- Деректер жиынында жасырылған кластерлер.
- Шектеу: ерекше және/немесе қызықты деректер нүктелері.
- Шуда жасырылған пайдалы сигнал, мысалы. шулы телефон арқылы адамның сөйлеуі.

Бақылаусыз оқытудың компоненттері

Жоғары деңгейде бақыланбайтын машинаны оқыту мәселесі келесі құрылымға ие:

$$\text{Dataset} + \text{Algorithm} \rightarrow \text{Unsupervised Model}$$

Бақыланбайтын модель деректердегі қызықты құрылымды сипаттайды. Мысалы, ол қызықты жасырын кластерлерді анықтай алады.

Бақыланбайтын оқу деректер жинағы

Бақыланбайтын оқу деректер жинағының алғашқы мысалы ретінде біз Ирис гүлінің мысалын қолданамыз, бірақ біз белгілерді алып тастаймыз.

Біз осы деректер жинағын жүктеуден бастаймыз.

In [6]:
```
# import standard machine learning libraries
import numpy as np
import pandas as pd
from sklearn import datasets

# Load the Iris dataset
iris = datasets.load_iris()

print(iris.DESCR)
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
```

	Min	Max	Mean	SD	Class Correlation	
sepal length:	4.3	7.9	5.84	0.83	0.7826	
sepal width:	2.0	4.4	3.05	0.43	-0.4194	
petal length:	1.0	6.9	3.76	1.76	0.9490	(high!)
petal width:	0.1	2.5	1.20	0.76	0.9565	(high!)

:Missing Attribute Values: None
:Class Distribution: 33.3% for each of 3 classes.
:Creator: R.A. Fisher
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
:Date: July, 1988

The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken from Fisher's paper. Note that it's the same as in R, but not as in the UCI Machine Learning Repository, which has two wrong data points.

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

.. topic:: References

- Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).
- Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis. (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1. See page 218.
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed

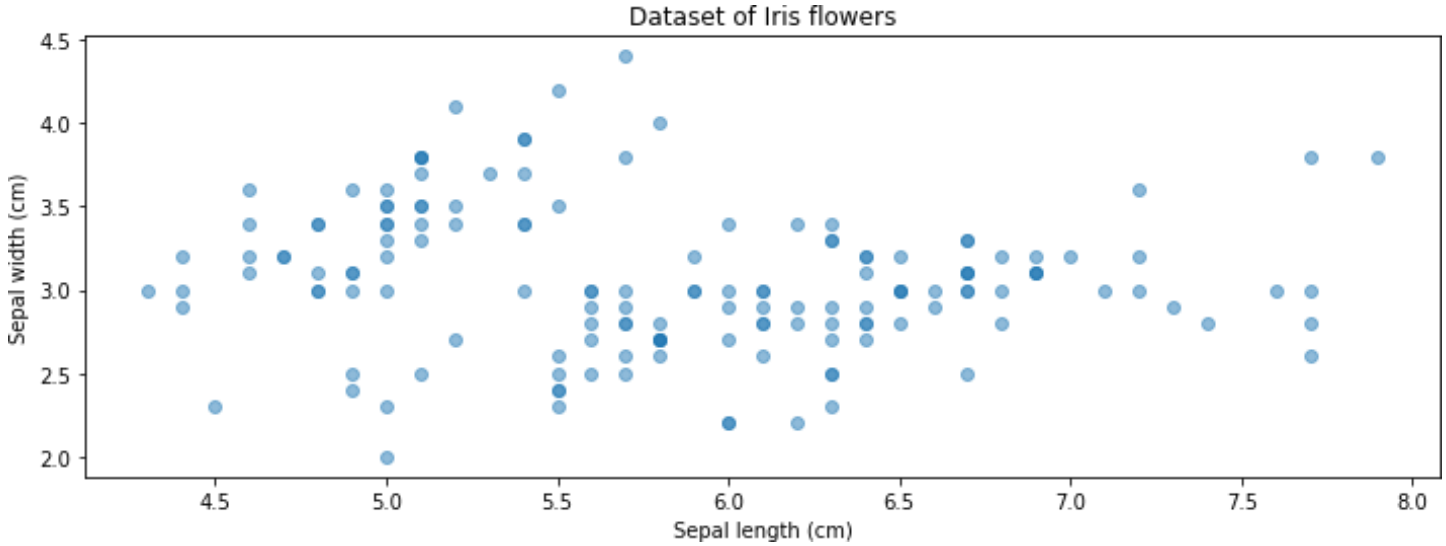
Environments". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, 67-71.
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64. Cheeseman et al"s AUTOCLASS II conceptual clustering system finds 3 classes in the data.
- Many, many more ...

Біз бұл деректер жинағын 2D форматында бейнелей аламыз. Біз енді жапсырма ақпаратын пайдаланбайтынымызды ескеріңіз.

In [10]:
```
from matplotlib import pyplot as plt
plt.rcParams['figure.figsize'] = [12, 4]

# Visualize the Iris flower dataset
plt.scatter(iris.data[:,0], iris.data[:,1], alpha=0.5)
plt.ylabel("Sepal width (cm)")
plt.xlabel("Sepal length (cm)")
plt.title("Dataset of Iris flowers")
```

Out[10]:
```
Text(0.5, 1.0, 'Dataset of Iris flowers')
```



Бақыланбайтын оқыту алгоритмі

Біз бұл деректер жиынтығын танымал бақылаусыз оқыту алгоритмі K-орталарына кіріс ретінде пайдалана аламыз.

- Алгоритм деректерден жасырын К кластерлерді табуға тырысады.
- Әрбір кластер өзінің центроидімен (оның орташа мәні) сипатталады.
- Кластерлер деректердегі қызықты құрылымды ашады.

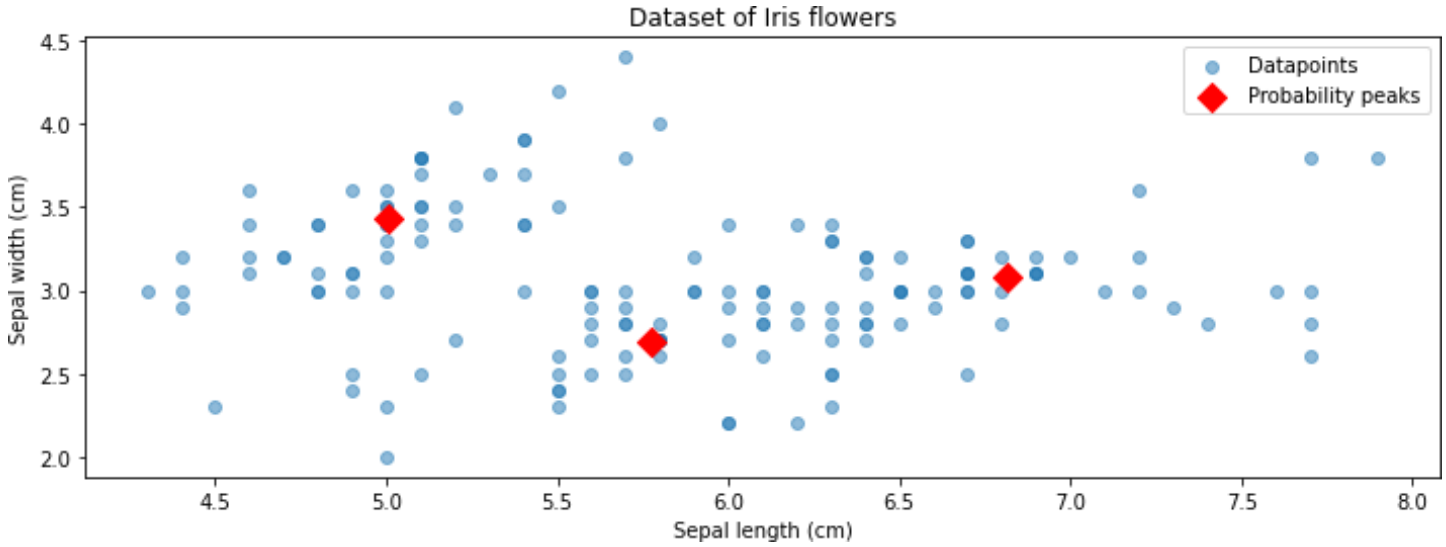
In [7]:
```
# fit K-Means with K=3
from sklearn import cluster
model = cluster.KMeans(n_clusters=3)
model.fit(iris.data[:,[0,1]])
```

Out[7]:
```
KMeans(n_clusters=3)
```

Осы деректер жинағындағы K-мағынасын іске қосу үш кластерді анықтайды.

In [16]:
```
# display the clusters in 2D
plt.scatter(iris.data[:,0], iris.data[:,1], alpha=0.5)
plt.scatter(model.cluster_centers_[:,0], model.cluster_centers_[:,1], marker='D'
, c='r', s=100)
plt.ylabel("Sepal width (cm)")
plt.xlabel("Sepal length (cm)")
plt.title("Dataset of Iris flowers")
plt.legend(['Datapoints', 'Probability peaks'])
```

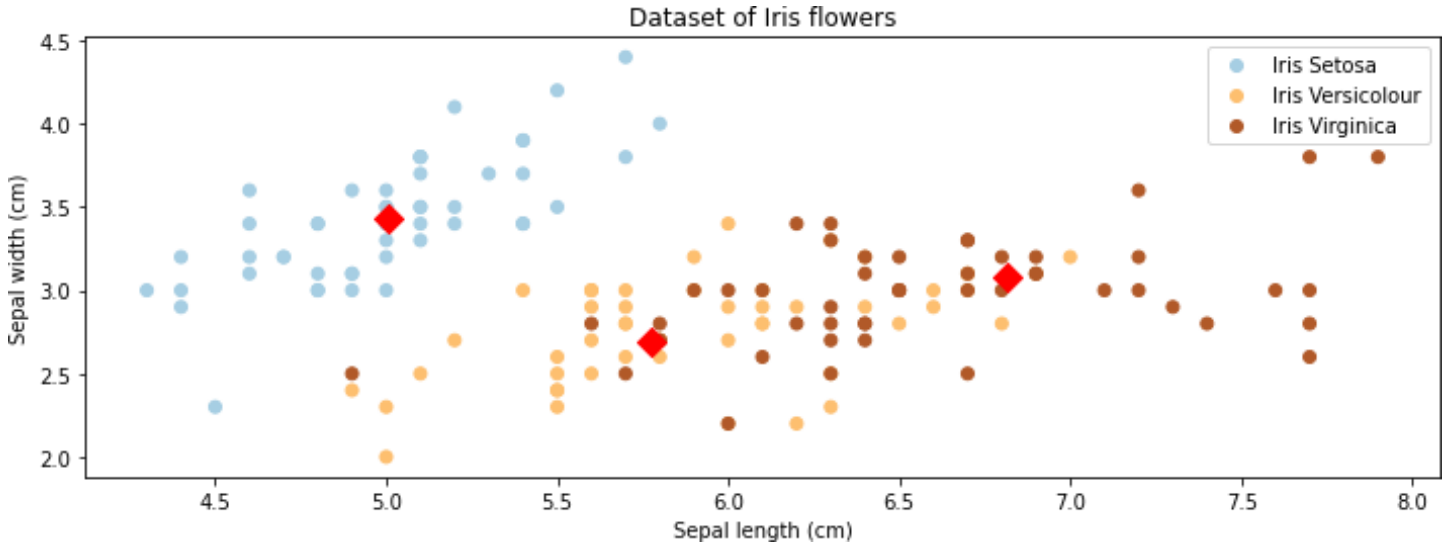
Out[16]:
```
<matplotlib.legend.Legend at 0x1231df4a8>
```



Бұл кластерлер деректер жинағында табылған, біз белгілерден алатын гүлдердің үш түріне сәйкес келеді.

In [17]:
```
p1 = plt.scatter(iris.data[:,0], iris.data[:,1], alpha=1, c=iris.target, cmap='Paired')
plt.scatter(model.cluster_centers_[:,0], model.cluster_centers_[:,1], marker='D', c='r', s=100)
plt.ylabel("Sepal width (cm)")
plt.xlabel("Sepal length (cm)")
plt.title("Dataset of Iris flowers")
plt.legend(handles=p1.legend_elements()[0], labels=['Iris Setosa', 'Iris Versicolour', 'Iris Virginica'])
```

Out[17]:
```
<matplotlib.legend.Legend at 0x123247fd0>
```



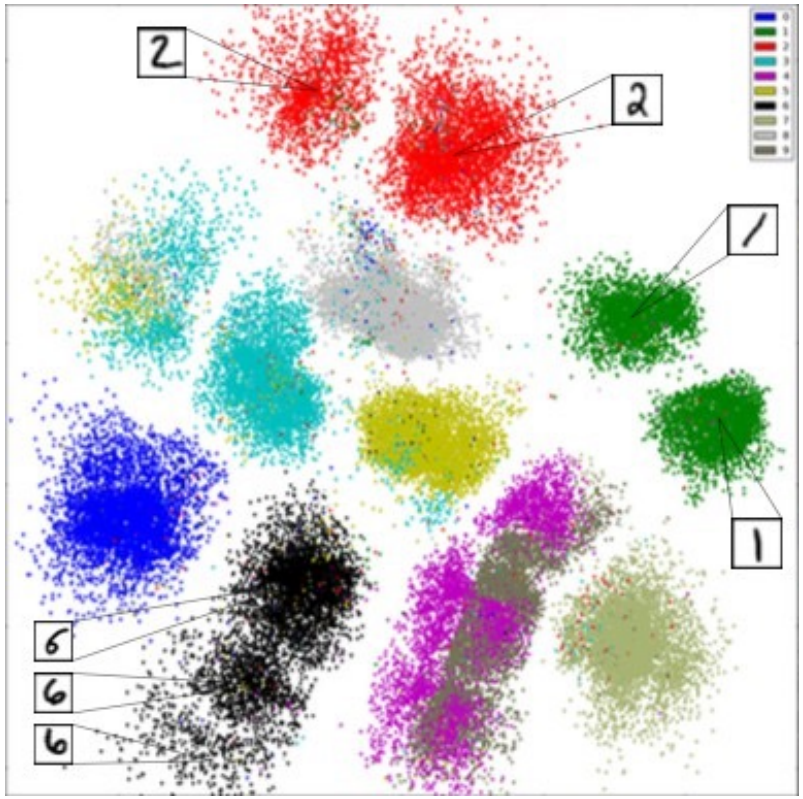
Бақылаусыз оқытуды қолдану

Бақылаусыз оқытудың көптеген қолданбалары бар:

- Көрнекілігі: деректердегі пайдалы жасырын құрылымды анықтау және қолжетімді ету.
- Аномалияны анықтау: жақын арада бұзылуы мүмкін зауыт құрамдастарын анықтау.
- Сигналдарды өшіру: шулы жазбадан адам сөзін шығару.

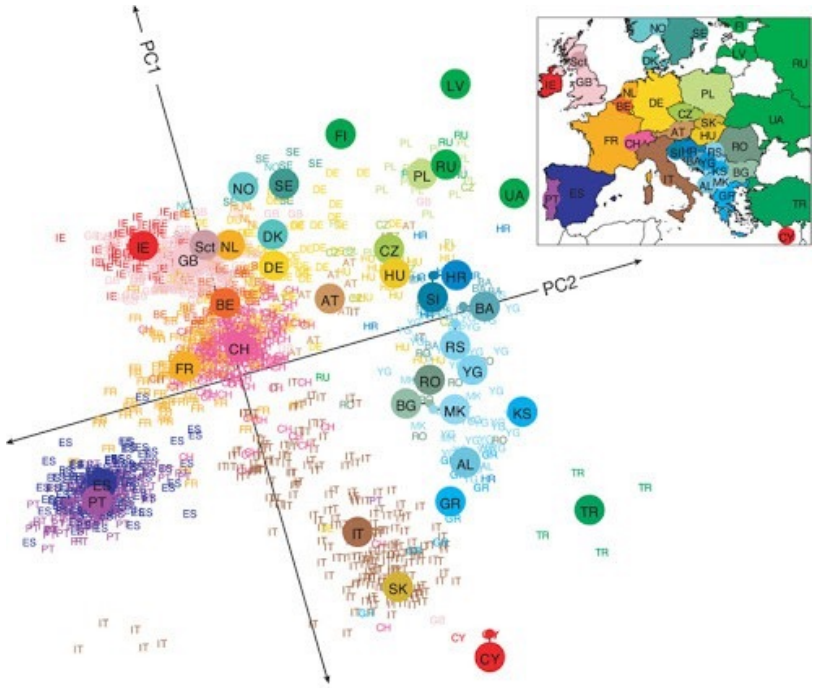
Қолдану: Цифрлардағы құрылымды ашу

Бақыланбайтын оқыту цифрлардағы құрылымды ешқандай белгілерсіз таба алады.



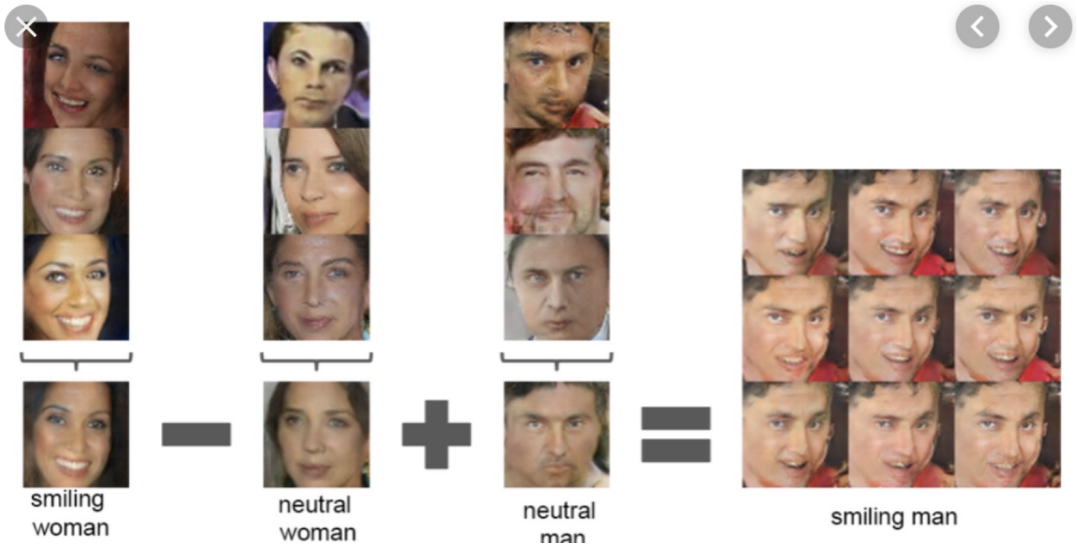
Қолданылуы: ДНҚ талдауы

ДНҚ-ға қолданылатын өлшемді азайту Еуропа елдерінің географиясын ашады:



Қолданылуы: Адам беттері

Терең оқытуға негізделген заманауи бақыланбайтын алгоритмдер адам бетінің деректер жиынындағы құрылымды ашады.

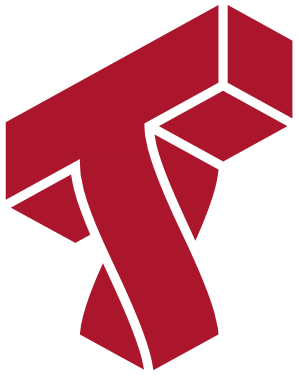


Бұл курста бақылаусыз оқыту

Біз бақылаусыз оқыту мәселелерінің бірнеше түрін қарастырамыз.

- Тығыздықты бағалау
- Аномалия мен жаңалықты анықтау
- Кластерлеу
- Өлшемді азайту

Әрі қарай, біз кейбір белгілерді орнатудан бастаймыз.



2-бөлім: Бақылаусыз оқыту тілі

Әрі қарай, бақылаусыз оқыту мәселесін формальды түрде қалай анықтауға болатынын қарастырайық.

Бақыланбайтын оқыту мәселесінің компоненттері

At a high level, an unsupervised machine learning problem has the following structure:

$$\underbrace{\text{Dataset}}_{\text{Attributes}} + \underbrace{\text{Learning Algorithm}}_{\text{Model Class} + \text{Objective} + \text{Optimizer}} \rightarrow \text{Unsupervised Model}$$

Бақыланбайтын модель деректердегі қызықты құрылымды сипаттайды. Мысалы, ол қызықты жасырын кластерлерді анықтай алады.

Бақыланбайтын деректер жинағы: Белгілеу

Біз n өлшемін бақылаусыз оқытуға арналған деректер жиынтығы ретінде анықтаймыз

$$Q = \{x^{(i)} \mid i = 1, 2, \ldots, n\}$$

Әрбір x(i) e Rd кірісті, d атрибуттарының немесе мүмкіндіктерінің векторын білдіреді.

Деректерді тарату

Біз деректер жиыны ықтималдық үлестірімінен таңдалған деп есептейміз & оны біз деректерді тарату деп атаймыз. Біз мұны былай белгілейміз

$$x \sim \&.$$

Деректер жинағы $Q = \{x(i) \mid i = 1, 2, \ldots, n\}$ & ішінен тәуелсіз және бірдей таралған (IID) үлгілерден тұрады.

Деректерді тарату: IID үлгісі

Тренинг мысалдарының тәуелсіз және бірдей таратылуындағы негізгі болжам (IID).

- Әрбір жаттығу мысалы бірдей таратудан алынған.
- Бұл бөлу алдыңғы оқу мысалдарына байланысты емес.

Мысалы: Тиынды аудару. Әрбір айналдырудың бастары мен құйрықтары бірдей ықтималдығы бар және алдыңғы айналдыруларға байланысты емес.

Қарсы мысал: жыл сайынғы санақ деректері. Әр жылдағы халық саны өткен жылғыға жақын болады.

Бақылаусыз оқыту алгоритмінің компоненттері

Бақылаусыз оқыту алгоритмін үш құрамдас бөліктен тұратындай қарастыруға болады:

- Модельдер класы: біз қарастыратын мүмкін болатын бақыланбайтын модельдер жиынтығы.
  Модельдің қаншалықты жақсы екенін анықтайтын мақсатты функция.
- Мақсат функциясына сәйкес модельдер класында ең жақсы болжамды модельді табатын оптимизатор

Үлгі: Белгілеу

Модельді функция деп айтамыз

$$f: X \rightarrow S$$

бұл x е X кірістерін s е S құрылымының қандай да бір түсінігімен салыстырады. Құрылымның көптеген анықтамалары болуы мүмкін (кластерлер, кіші өлшемді көріністер және т.б.) және біз көптеген мысалдарды көреміз.

Көбінесе модельдерде Θ жиынында тұратын Θ е Θ параметрлері болады. Содан кейін үлгіні былай жазамыз

$$f_\Theta : X \rightarrow S$$

Θ арқылы параметрленгенін белгілеу үшін.

Үлгі класы: Белгілеу

Формальды түрде үлгі класы жиын болып табылады

$$n \subseteq \{f \mid f : X \to S\}$$

құрылымдық элементтерге енгізу мүмкіндіктерін салыстыратын ықтимал үлгілердің.

ƒΘ модельдері кейбір Θ жиынында тұратын Θ е Θ параметрлерімен пареметрленген кезде. Осылайша біз де жаза аламыз

$$n = \{f_\theta \mid f : X \to S;\ \theta \text{e}\ \Theta\}.$$

Мақсаты: Белгілеу

Бұл интуицияны қабылдау үшін біз мақсаттық функцияны анықтаймыз (оны жоғалту функциясы деп те аталады)

$$J(f) : n \rightarrow [0, \infty),$$

ол Q = {x(i) | деректеріне ƒ қаншалықты "сәйкес келетінін" сипаттайды i = 1, 2, . . . , n}.

ƒ Θ е Θ арқылы параметрленгенде мақсат функцияға айналады

$J(\Theta) : \Theta \rightarrow [0, \infty)$.

Оңтайландырушы: Белгілеу

Оптимизатор J мақсатының ең кіші мәні бар f е n моделін табады.

$$\min_{f \text{ e } n} J(f)$$

Интуитивті түрде бұл оқу деректер жинағындағы деректерге ең жақсы «сәйкес келетін» функция.

f Θ е Θ арқылы параметрленгенде, оңтайландырғыш J(Θ) функциясын барлығынан кішірейтеді.

Θ е Θ.

Мысал: K-мағынасы

Мысал ретінде біз бұрын көрген K-Means алгоритмін қолданайық.

Еске салайық:

- Алгоритм деректерден жасырын K кластерлерді табуға тырысады.
- Әрбір кластер өзінің центроидімен (оның орташа мәні) сипатталады.
- Кластерлер деректердегі қызықты құрылымды ашады.

K-Means моделі

Біз K-Means арқылы қайтарылған модельді функция ретінде қарастыра аламыз

$$f_\theta : X \to S$$

әрбір x кірісін кластерге тағайындайтын s e S = {1, 2, … , K}.

Модельдің Ө параметрлері K центроидтары c1 , c2 , … cK e X болып табылады. Егер c+ x-ке ең жақын центроид болса, x класы + болады.

K-мағынасы

ƒΘ Q деректер жиынының жақсы кластерленуі екенін қалай анықтаймыз?

Біз нүктелер мен олардың ең жақын центроидтары арасындағы қашықтық минимум болатындай етіп c+ центроидтарын іздейміз:

$$J(\Theta) = \sum_{i=1}^{n} ||x^{(i)} - \text{centroid}(f_{\Theta}(x^{(i)}))||,$$

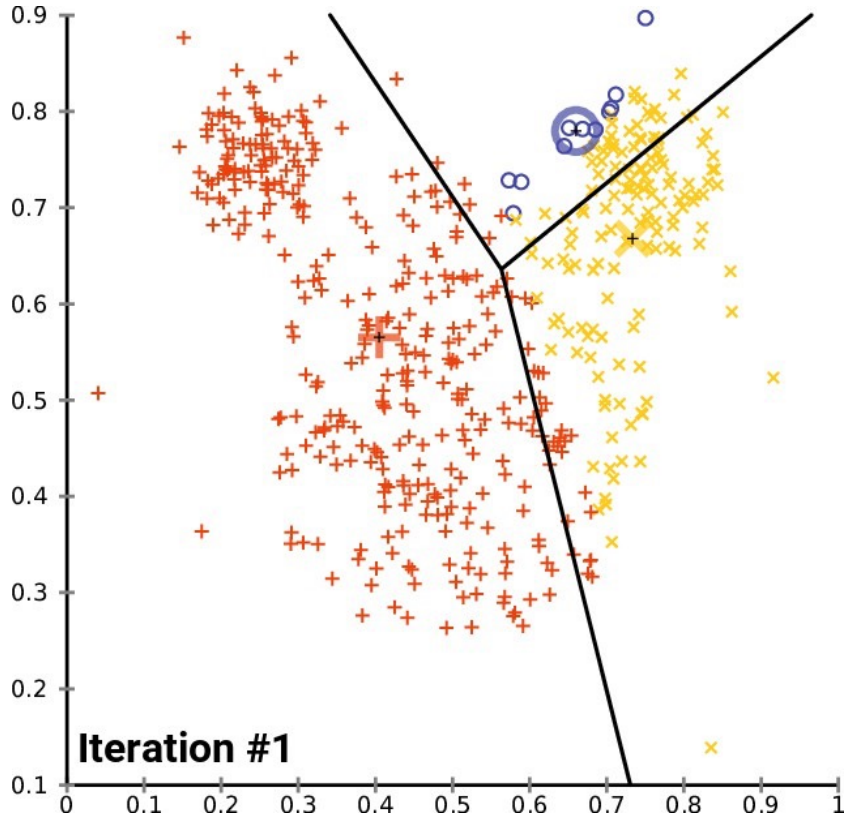
мұндағы центроид(+) = c+ кластер үшін центроидты білдіреді +.

K-Means Optimizer

Біз мұны бастапқы кездейсоқ кластерді $f(x)$ тағайындаудан бастап екі тоқтау процесінде оңтайландыруға болады.
Конвергенция болғанша қайталаңыз:

- Әрбір c+ мәнін оның {x(i) | кластерінің орталығы етіп орнатыңыз $f(x(i)) = +$}.
- x(i) ең жақын центроидтың кластерінде болатындай $f(x)$ кластерін жаңартыңыз.

Бұл ең жақсы көрнекі түрде суреттелген (Википедиядан (https://commons.wikimedia.org/wiki/File:K-means_convergence.gif)):



Алгоритм: K-Means

- **Түрі:** бақылаусыз оқыту (кластер)
- **Модельдік отбасы:** + центроидтар
- **Мақсатты функция:** Ең жақын центроидқа дейінгі қашықтықтардың
- қосындысы
  Оптимизатор: Итеративті оңтайландыру процедурасы.