

Evaluering, Clustering

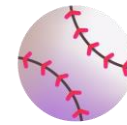
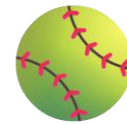
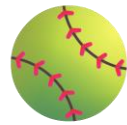
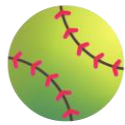
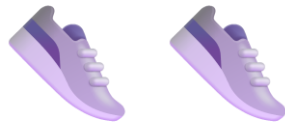
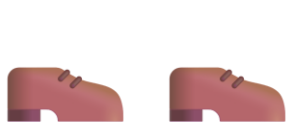
Leksjon 4

IN2110, vår 2024

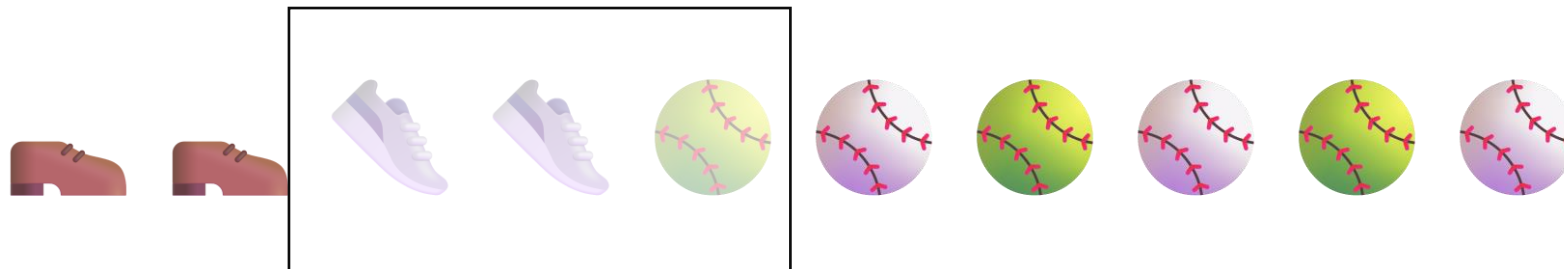
Egil Røningstad

1. Evaluering av klassifikasjon
2. Introduksjon til clustering

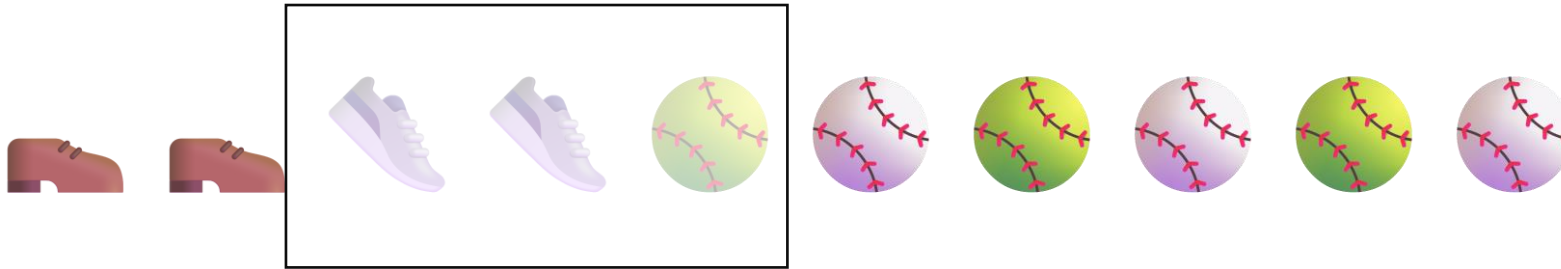
Finn skoene!



Finn skoene!



Hvor bra var dette?



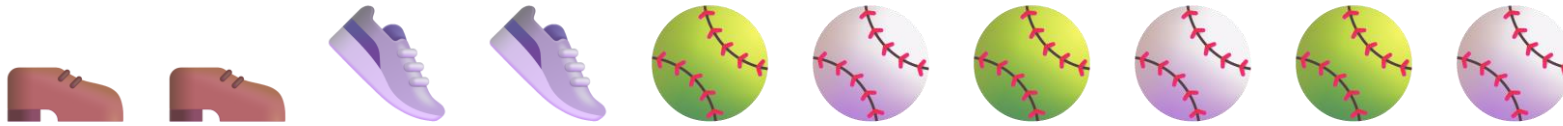
- Av totalt 10 gjenstander, ble $5+2 = 7$ gjenstander rett klassifisert
- Av totalt 4 sko, ble halvparten hentet
- Av 3 hentede gjenstander, var 2 korrekte

Accuracy: $7 / 10 = 70\%$

Recall: $2 / 4 = 50\%$

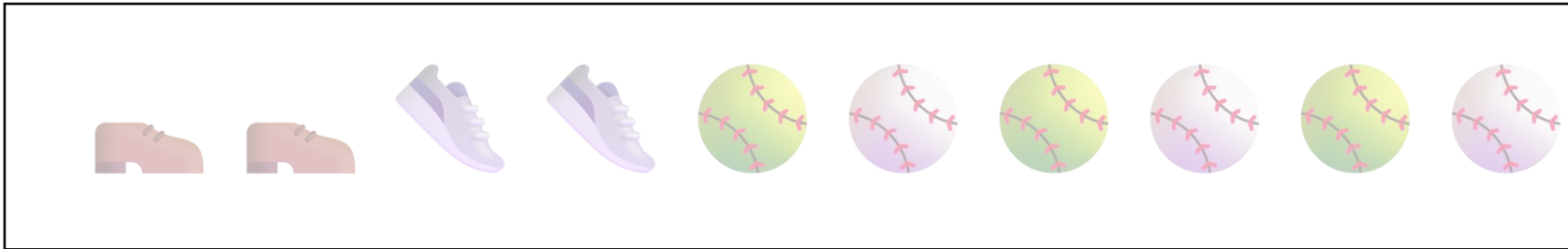
Precision: $2 / 3 = 67\%$

Finn skoene!

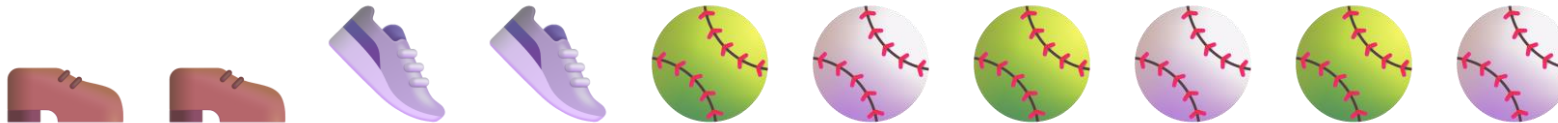


Jeg aner ikke hva han mener

Hvor bra var dette?



- Av totalt 10 gjenstander, ble 4 gjenstander rett klassifisert
- Av totalt 4 sko, ble alle hentet
- Av 10 hentede gjenstander, var 4 korrekte



Accuracy: $4 / 10 = 40\%$

Recall: $4 / 4 = 100\%$

Precision: $4 / 10 = 40\%$

Hva om vi hadde hundre tennisballer og fire sko?

- Hvis hunden kommer tilbake med ingenting, ville den ha klassifisert 100 av 104 gjenstander rett.
- 96% rett ved å bare si at alt tilhører kategorien det er mest av.
- God accuracy men lite nytteverdi.

Boka: imagine that we looked at a million tweets, and let's say that only 100 of them are discussing their love (or hatred) for our pie while the other 999,900 are tweets about something completely unrelated. Imagine a simple classifier that stupidly classified every tweet as "not about pie". This classifier would have 999,900 true negatives and only 100 false negatives for an accuracy of $999,900/1,000,000$ or 99.99%! What an amazing accuracy level!

Surely we should be happy with this classifier? But of course this fabulous 'no pie' classifier would be completely useless, since it wouldn't find a single one of the customer comments we are looking for.

The 2-by-2 confusion matrix

gold standard labels

gold positive gold negative

*system
output
labels*

system
positive

system
negative

true positive

false positive

false negative

true negative

$$\text{precision} = \frac{tp}{tp+fp}$$

$$\text{recall} = \frac{tp}{tp+fn}$$

$$\text{accuracy} = \frac{tp+tn}{tp+fp+tn+fn}$$

Evaluation: Precision

- % of items the system detected (i.e., items the system labeled as positive) that are in fact positive (according to the human gold labels)

$$\textbf{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Evaluation: Recall

- % of items actually present in the input that were correctly identified by the system.

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Why Precision and recall

- Our dumb pie-classifier
 - Just label nothing as "about pie"

Accuracy=99.99%

but

Recall = 0

- (it doesn't get any of the 100 Pie tweets)

Precision and recall, unlike accuracy, emphasize true positives:

- finding the things that we are supposed to be looking for.

A combined measure: F

- F measure: a single number that combines P and R:

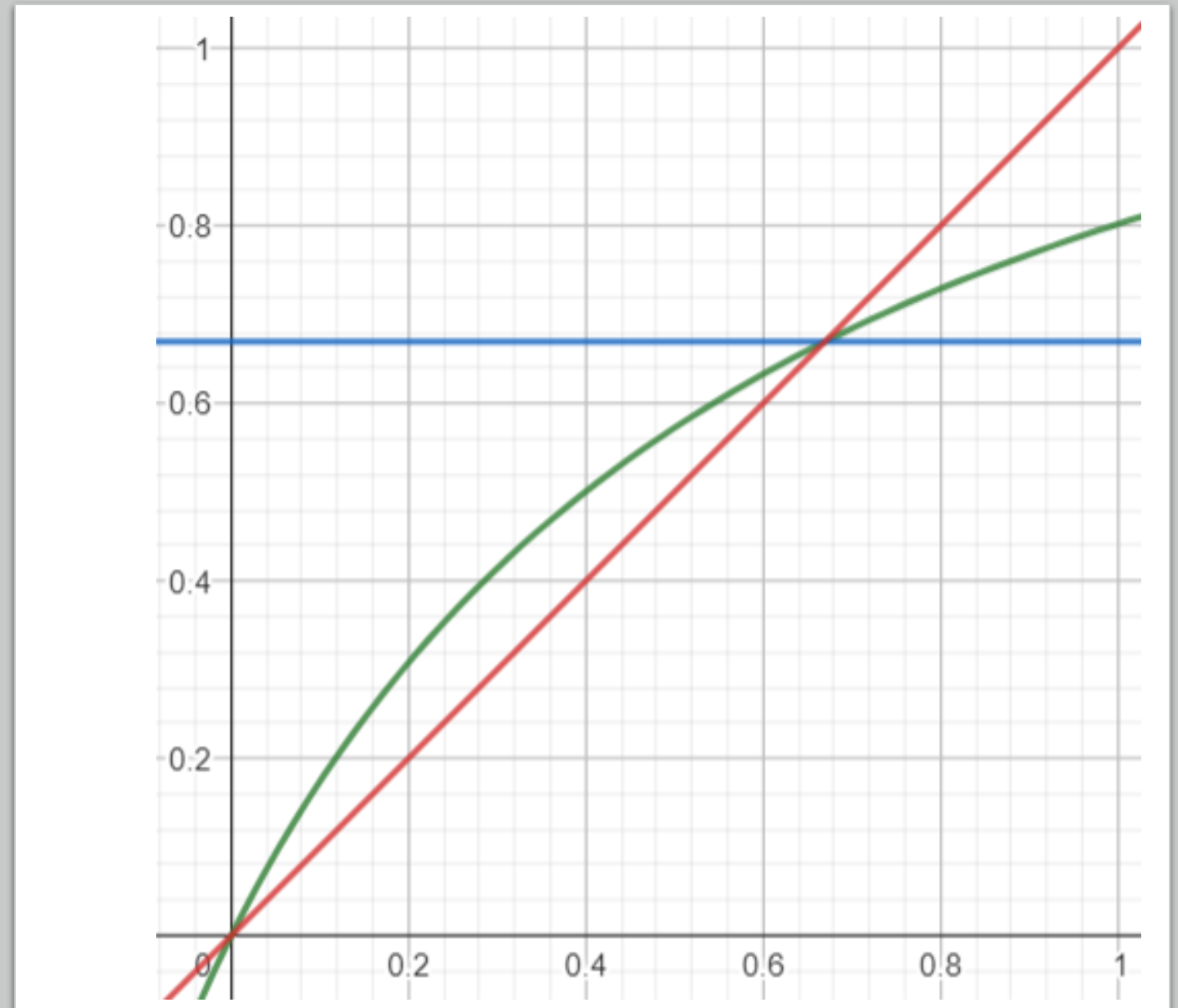
$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- We almost always use balanced F_1 (i.e., $\beta = 1$)

$$F_1 = \frac{2PR}{P + R}$$

F1 på Geogebra

- Blå: Precision = 0,7
- Rød: Recall fra 0 til 1
- Grønn: F1



Classification and Naive Bayes

Evaluation with more than two classes

Confusion Matrix for 3-class classification

		<i>gold labels</i>			
		urgent	normal	spam	
<i>system output</i>	urgent	8	10	1	precision_u = $\frac{8}{8+10+1}$
	normal	5	60	50	precision_n = $\frac{60}{5+60+50}$
	spam	3	30	200	precision_s = $\frac{200}{3+30+200}$
		recall_u = $\frac{8}{8+5+3}$	recall_n = $\frac{60}{10+60+30}$	recall_s = $\frac{200}{1+50+200}$	

**FORSKJELL PÅ OM TRUE
URGENT BLE
KLASSIFISERT SOM
NORMAL ELLER SPAM?**

How to combine P/R from 3 classes to get one metric

- Macroaveraging:
 - compute the performance for each class, and then average over classes
- Microaveraging:
 - collect decisions for all classes into one confusion matrix
 - compute precision and recall from that table.

Macroaveraging and Microaveraging

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

$$\text{precision} = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

$$\text{precision} = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

$$\text{precision} = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
system yes	268	99
system no	99	635

$$\text{microaverage precision} = \frac{268}{268+99} = .73$$

$$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = .60$$

2. Unsupervised learning - clustering

- *Can you sort the Lego bricks?*
 - ▣ (No instruction on how)
- You may choose sorting on
 - ▣ Color, or
 - ▣ Size, or
 - ▣ Shape, or
 - ▣ A combination
- I cannot know beforehand what you choose, but
- The result might be helpful



Clustering

- Eksempelanvendelse:
Avgrense hvilke dokumenter
man skal beregne
dokumentlikhet for

Search result clustering for better navigation



The screenshot shows the Vivisimo search engine interface. At the top, the Vivisimo logo is on the left, followed by a search bar containing 'jaguar' and a dropdown menu set to 'the Web'. To the right of the search bar is a blue 'Search' button and links for 'Advanced Search' and 'Help'.

Below the search bar, there are two main sections. On the left, under the heading 'Clustered Results', is a sidebar with a tree view of clusters. The clusters are: 'jaguar (208)', 'Cars (74)', 'Club (34)', 'Cat (23)', 'Animal (13)', 'Restoration (10)', 'Mac OS X (8)', 'Jaguar Model (8)', 'Request (5)', 'Mark Webber (6)', and 'Maya (5)'. A 'More' link is at the bottom of the sidebar. Below the sidebar is a 'Find in clusters:' section with a text input field labeled 'Enter Keywords' and a red 'Go' button.

On the right, under the heading 'Top 208 results of at least 20,373,974 retrieved for the query jaguar (Details)', is a list of search results. The first four results are:

1. [Jag-lovers - THE source for all Jaguar information](#) [new window] [frame] [cache] [preview] [clusters]
... Internet! Serving Enthusiasts since 1993 The Jag-lovers Web Currently with 40661 members The Premier **Jaguar** Cars web resource for all enthusiasts Lists and Forums Jag-lovers originally evolved around its ...
[www.jag-lovers.org](#) - Open Directory 2, Wisenut 8, Ask Jeeves 8, MSN 9, Looksmart 12, MSN Search 18
2. [Jaguar Cars](#) [new window] [frame] [cache] [preview] [clusters]
[...] redirected to [www.jaguar.com](#)
[www.jaguarcars.com](#) - Looksmart 1, MSN 2, Lycos 3, Wisenut 6, MSN Search 9, MSN 29
3. [http://www.jaguar.com/](#) [new window] [frame] [preview] [clusters]
[www.jaguar.com](#) - MSN 1, Ask Jeeves 1, MSN Search 3, Lycos 9
4. [Apple - Mac OS X](#) [new window] [frame] [preview] [clusters]
Learn about the new OS X Server, designed for the Internet, digital media and workgroup management Download a technical factsheet.
[www.apple.com/macosx](#) - Wisenut 1, MSN 3, Looksmart 26

Clustering methods

□ Hierarchical

- ▣ Creates a tree structure of hierarchically nested clusters.

□ Flat

- ▣ Tries to directly decompose the data into a set of clusters.
- ▣ What we will focus on.
- ▣ Given a set of objects $O = \{o_1, \dots, o_n\}$,
construct a set of clusters $C = \{c_1, \dots, c_k\}$,
where each object o_i is assigned to a cluster c_j .
- ▣ We will consider one algorithm: k-means clustering

K-means clustering

1. Decide on the number of clusters: k
2. Choose a set of arbitrary centroids: $\mu_1, \mu_2, \dots, \mu_k$
3. For each item, x , in the training data,
 - ▣ find the nearest centroid μ_i , and assign x to class C_i
4. For each resulting class C_i , calculate and find the new centroid μ_i .
5. Classify each item according to the new centroids
6. Repeat from 4

Clustering- demos

- <https://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means/>
- <http://shabal.in/visuals/kmeans/6.html>

Why does this work? How does this work?

39

- The goal is a mapping
$$\gamma: O \rightarrow C = \{C_1, C_2, \dots, C_k\}$$
- We need a tool, F ,
 - ▣ to measure the performance of γ
- The goal is to find a γ that **optimizes** F , in symbols
$$\hat{\gamma} = \operatorname{argmax}_{\gamma} F(\gamma)$$
- F is called an objective function
- Several possible objectives:
 - ▣ High similarity (=small distance) within the clusters (intra-cluster)
 - ▣ Low similarity (high distance) between the clusters (inter-clusters)

Within cluster sum of squares (intra-cluster)

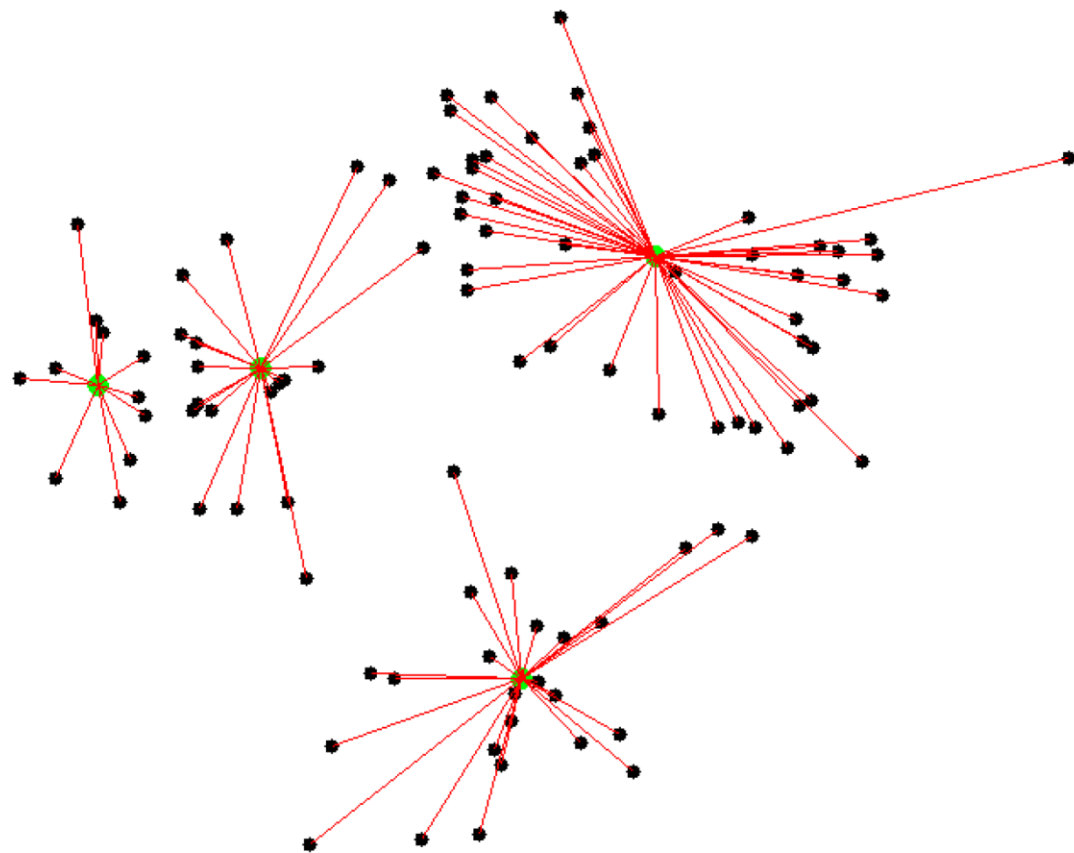
- For each cluster consider the sum of square distances:

$$SS_i = \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

- Sum over all classes

$$WCSS = \sum_{i=1}^k SS_i$$

- To optimize F , is to find the γ that yields the smallest $WCSS$



Applied to k -means

41

- For each iteration:

$$WCSS_{i+1} \leq WCSS_i$$

- Because:

- ▣ Given a class, C_i , the recalculated centroid is the unique point in space that minimizes SS_i
- ▣ If an item is moved from one class to another, its centroid-distance decreases

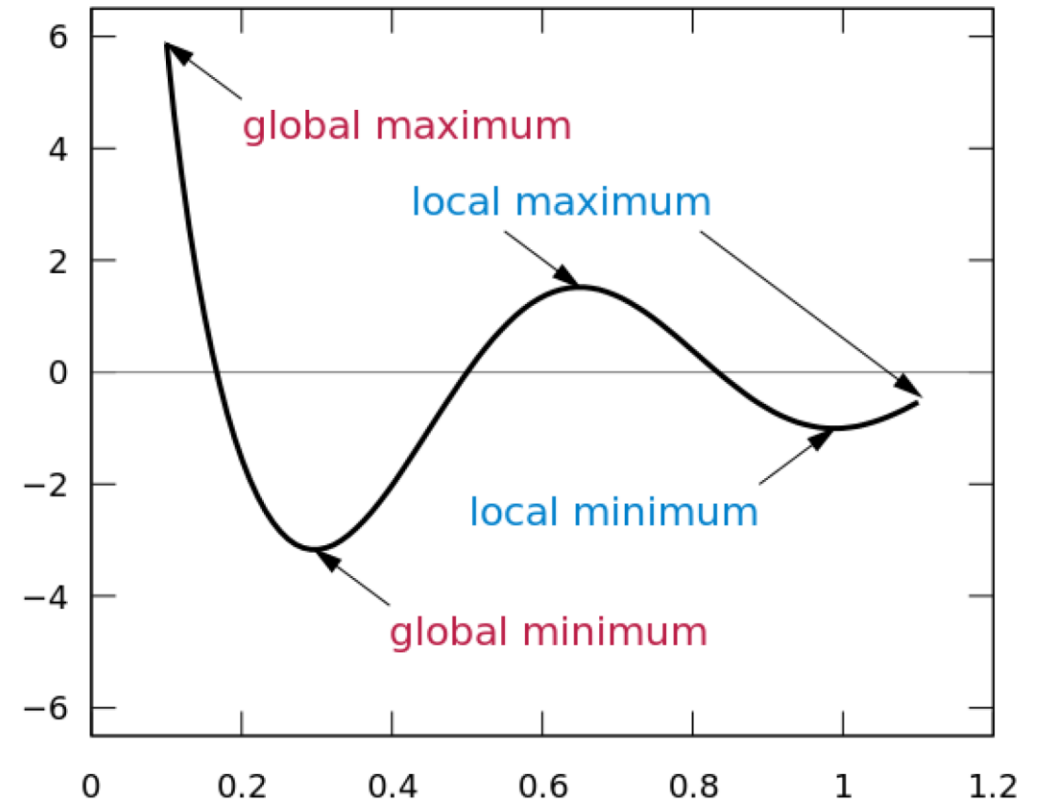
- Possible stopping criteria:

- ▣ Fixed number of iterations
- ▣ Clusters or centroids are unchanged between iterations.
- ▣ Threshold on the decrease of the objective function (absolute or relative to previous iteration)

Properties of k -means

42

- The time complexity is linear, $O(kn)$
- Guaranteed to converge, but not to find the global optimal solution:
 - ▣ Depends on choice of initial centroids



Comments

43

'Seeding'

- ▶ We **initialize** the algorithm by choosing **random seeds** that we use to compute the first set of centroids, e.g:
 - ▶ pick k random objects from the collection;
 - ▶ pick k random points in the space;
 - ▶ pick k sets of m random points and compute centroids for each set; etc.
 - ▶ The seeds can have a large impact on the resulting clustering.
 - ▶ **Outliers** are troublemakers.
- No prescribed way to choose k .
 - ▣ In particular, more k -s will always give better WCSS without being intuitively better.

Intrinsic evaluation of clustering

44

With labeled gold-data

- Run k -means on the gold set (without the labels).
- Compare the clusters to the classes:
 - ▣ Purity: a good cluster will have all members from the same class

Without using gold data

- We can use some intra-cluster or inter-cluster measure,
 - ▣ E.g., WCSS to compare which initial choice of centroids is better in k -means

Extrinsic evaluation

45

- See which clustering (or lack of clustering) yields the best results in a larger task
- For example: two versions of a recommender system, and measure some of:
 - ▣ User satisfaction
 - ▣ How many recommended articles they read, or click on
 - ▣ Improvement in sales

Flat Clustering: The good and the bad



Pros

- ▶ Conceptually **simple**, and easy to implement.
- ▶ **Efficient**. Typically linear in the number of objects.

Cons

- ▶ <https://github.com/Viliami/kmeans-cluster.git>
The dependence on random seeds as in k -means makes the clustering **non-deterministic**.
- ▶ The number of clusters k must be pre-specified. Often no principled means of *a priori* specifying k .
- ▶ Not as informative as the more structured clusterings produced by hierarchical methods.
- ▶ In general; often difficult to **evaluate** clustering.

OK kode å se på for regnestykkene

- <https://github.com/Viliami/kmeans-cluster.git>

```
▼ def euclid_distance(point1, point2, dimensions=2):  
    sum = 0  
    for i in range(dimensions):  
        sum += (point1[i]-point2[i])**2  
    return math.sqrt(sum)
```

