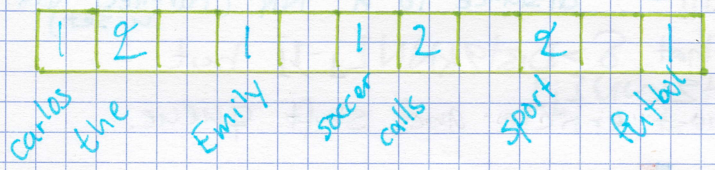
Document representation

Word count document representation

Bag of words model
- Ignore order of words
- Count # of instances of each word in vocabulary



"Carlos calls the sport futbol. Emily calls the sport soccer"

Issues with word counts — Rare words

Common words in doc: "the", "player", "field", "goal"

Dominate rare words like "futbol", "Messi"

TF-IDF document representation

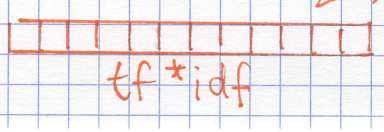
Emphasizes important words
- Appears frequently in document (common locally)

Term frequency = word counts

- Appears rarely in corpus (rare globally)

$$\text{Inverse doc freq.} = \log \frac{\#\text{docs}}{1 + \#\text{ docs using word}}$$

Trade off: local frequency vs. global rarity



Distance metrics: Euclidean and scaled Euclidean

Distance metric:
Defining notion of "closest"

In 1D, just Euclidean distance:

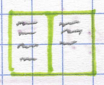
$$distance(x_i, x_q) = |x_i - x_q|$$

In multiple dimensions:
- can define many interesting distance functions
- most straightforwardly, might want to weight different dimensions differently

Weighting different features

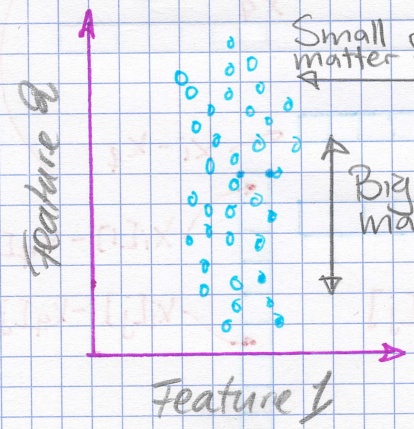
Reasons:
- Some features are more relevant than others

#bedrooms
#bathrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront



title
abstract
main body
conclusion

- Some features vary more than others



Specify weights as a function of feature spread

For feature j:

$$\frac{1}{\max_i(x_i[j]) - \min_i(x_i[j])}$$

Scaled Euclidean distance

Formally, this is achieved via

$$\text{distance}(x_i, x_q) = \sqrt{a_1 (x_i[1] - x_q[1])^2 + \ldots + a_d (x_i[d] - x_q[d])^2}$$

a_1 … a_d: weight on each feature (defining relative importance)

Setting weights as 0 or 1 is equivalent to feature selection { Effect of binary weights

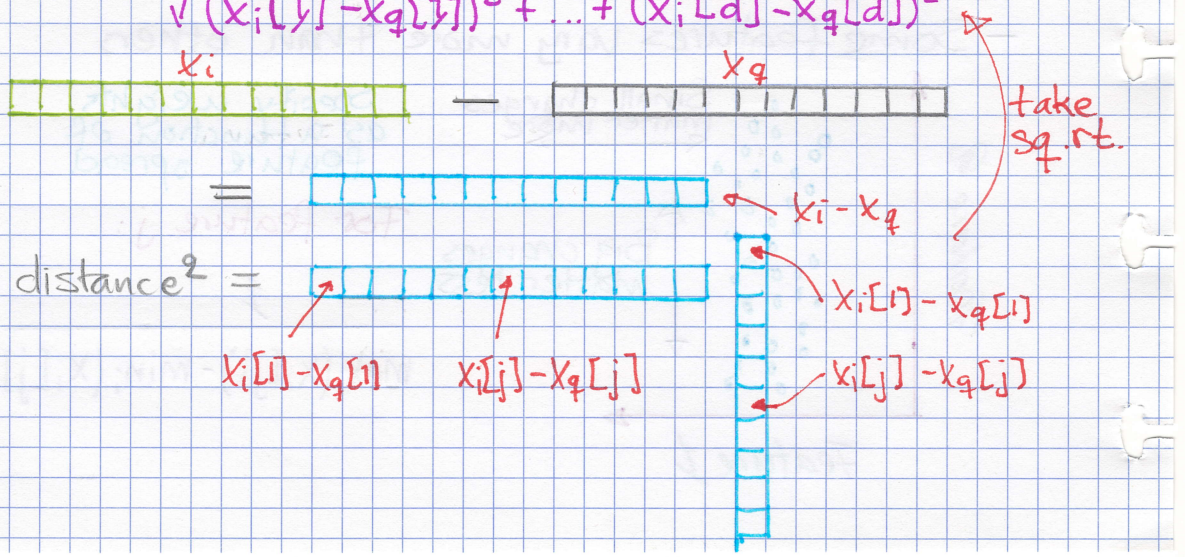
Feature engineering/selection is important, but hard

Writing (scaled) Euclidean distance using (weighted) inner products

(non-scaled) Euclidean distance

Defined in terms of inner product

$$\text{distance}(x_i, x_q) = \sqrt{(x_i - x_q)^T (x_i - x_q)} = \sqrt{(x_i[1] - x_q[1])^2 + \ldots + (x_i[d] - x_q[d])^2}$$

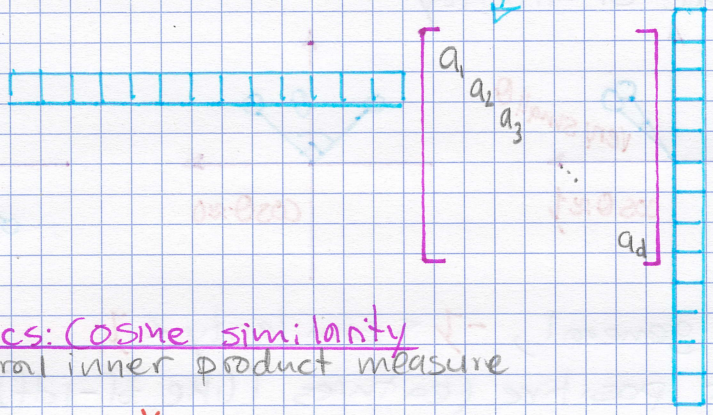


Scaled Euclidean distance

Defined in terms of inner product

$$\text{distance}(x_i, x_q) = \sqrt{(x_i - x_q)^T A (x_i - x_q)}$$

$$\sqrt{a_1 (x_i[1] - x_q[1])^2 + \ldots + a_d (x_i[d] - x_q[d])^2}$$

$\text{distance}^2 =$ 

Distance metrics: Cosine similarity

Another natural inner product measure

x_q

1	0	0	0	5	3	0	0	1	0	0	0	0

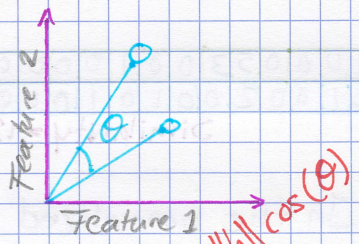
x_i

3	0	0	0	2	0	0	1	0	1	0	0	0

Similarity $= x_i^T x_q$

$$= \sum_{j=1}^{d} x_i[j]\, x_q[j]$$

$= 13$



1	0	0	0	5	3	0	0	1	0	0	0	0

0	0	1	0	0	0	9	0	0	6	0	4	0

Similarity $= 0$

Cosine similarity — normalize

$$\text{Similarity} = \sum_{j=1}^{d} x_i[j]\, x_q[j]$$

$$\sqrt{\sum_{j=1}^{d} (x_i[j])^2} \sqrt{\sum_{j=1}^{d} (x_q[j])^2}$$

$$= \frac{x_i^T x_q}{\|x_i\| \|x_q\|} = \cos\theta$$

$a^T b = \|a\| \|b\| \cos(\theta)$

first normalize

$\left(\frac{x_i}{\|x_i\|}\right)^T \left(\frac{x_q}{\|x_q\|}\right)$

- Not a proper distance metric
- Efficient to compute for sparse vecs

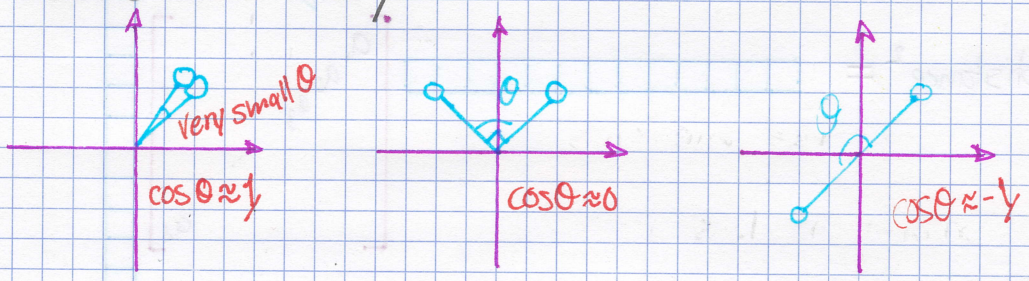
Normalize

1	0	0	0	5	3	0	0	1	0	0	0	0

$$\sqrt{1^2+5^2+3^2+7^2}$$

1/6	0	0	0	5/6	3/6	0	0	1/6	0	0	0	0

Cosine similarity



In general, $-1 \leq$ similarity ≤ 1

For positive features (like tf-idf) } our focus

$0 \leq$ similarity ≤ 1

Define distance = 1 - similarity

To normalize or not and other distance considerations

To normalize or not?

1	0	0	0	5	3	0	0	1	0	0	0	0
3	1	0	0	2	0	0	1	0	1	0	0	0

Similarity = 13

2	0	0	0	10	6	0	0	2	0	0	0	0
6	2	0	0	4	0	0	2	0	2	0	0	0

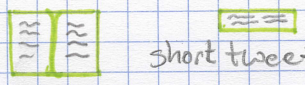
Similarity = 52

In the normalized case

1/6	0	0	0	5/6	3/6	0	0	1/6	0	0	0	0
3/4	1/4	0	0	2/4	0	0	1/4	0	1/4	0	0	0

Similarity = 13/24

But not always desired...



Normalizing can make dissimilar objects appear more similar

Common Compromise:
Just cap maximum word counts

Other distance metrics
- Mahalanobis
- rank-based
- correlation-based
- Manhattan
- Jaccard
- Hamming
- ...

Combining distance metrics
Example of document features:

1. Text of document
   - Distance metric: Cosine similarity
2. # of reads of doc
   - Distance metric: Euclidean distance

Add together with user-specified weights