IN2110: Språkteknologiske metoder

Ordvektorer

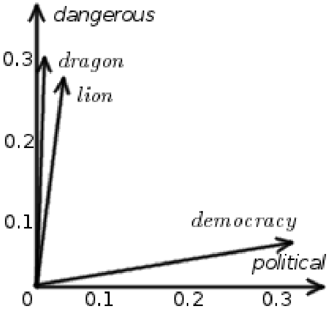
Eivind Alexander Bergem

Språkteknologigruppen (LTG)

19. februar, 2019

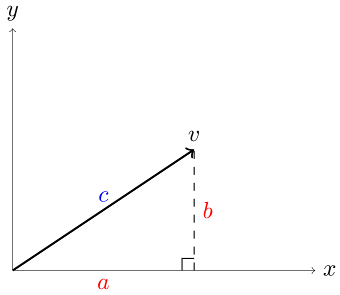
Quantifying similarity

- We can use the same metrics for document and word vectors:
  - Euclidean distance
  - Cosine similarity



Vector norms

We calculate the norm just like we calculate the length of the hypotenuse using the Pythagorean theorem!



$$c = \sqrt{a^2 + b^2}$$

$$||v|| = \sqrt{\sum_{i=1}^{n} {v_i}^2}$$

Unit vectors

A vector $\hat{v}$ is a unit vector when

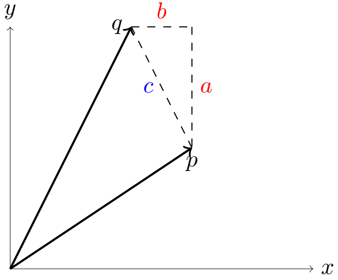
$$||\hat{v}|| = \sqrt{\sum_{i=1}^{n} \hat{v}_i^2} = 1$$

To get unit vector $\hat{v}$ from vector v, divide values by vector norm

$$\hat{v}_i = \frac{v_i}{||v||}$$

Measuring distance

We can use Euclidean distance to measure distance



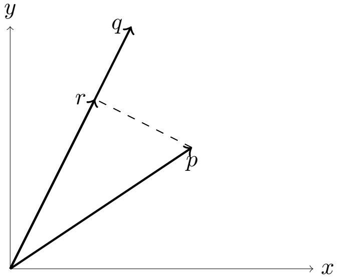
$$c = \sqrt{a^2 + b^2}$$

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Measuring similarity

We can use a dot-product to measure similarity

$$\text{dot-product}(p, q) = p \cdot q = \sum_{i=1}^{n} p_i q_i$$



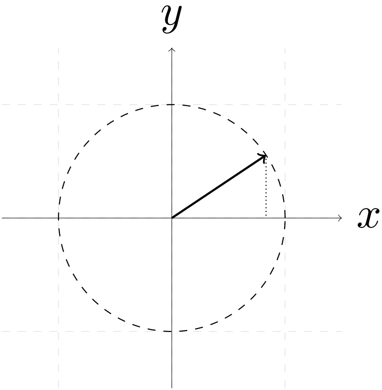
$$p \cdot q = ||r|| \times ||q||$$

and when q is a unit vector

$$p \cdot q = ||r||$$

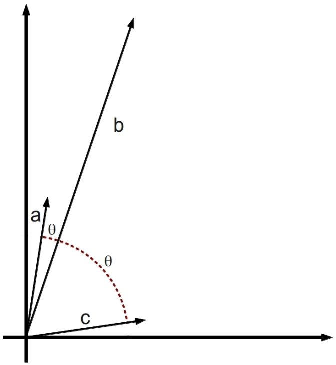
Unit circle and cosines

- When working with unit vectors, dot-product like projecting onto the x-axis.
- Value of 1 when vectors point in the same direction.
- Value of -1 when they point in opposite directions.



Measuring angles

- The dot-product is sensitive to vector norms.
- Measure angle between vectors to ignore vector norms.



Cosine similarity

We use cosine similarity to measure the angle between vectors p and q

$$\text{similarity}(p, q) = \frac{p \cdot q}{||p|| \times ||q||}$$

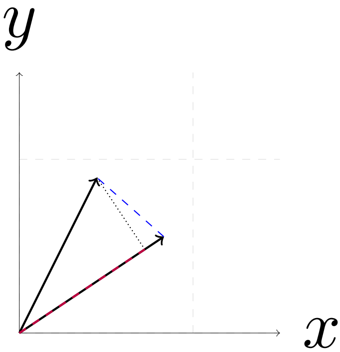
This can be seen as a normalized dot-product that is invariant to vector length. When p and q are unit vectors we can drop the denominator

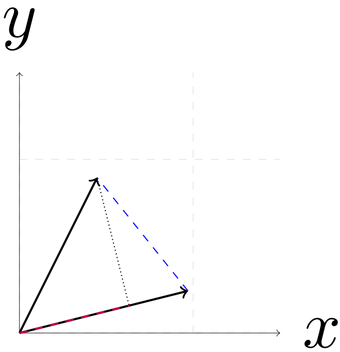
$$\text{similarity}(p, q) = p \cdot q = \sum_{i=1}^{n} p_i q_i$$

making cosine similarity equivalent to taking the dot-product.

Relative rank

When using unit vectors, cosine similarity and eucliean distance have the *same relative rank order*





Computational complexity

- Euclidean distance
  - Square root is expensive.
  - For sparse vectors, need to consider the union of the non-zero values in the two vectors.
- Cosine similarity
  - Vector norms are expensive.
  - Dot product is cheap, only need to consider intersections of the non-zero values in the two vectors.
- When using unit vectors, euclidean distance and cosine similarity are rank equivalent.
- TL;DR: Normalize to unit vectors and use dot-product in place of full cosine similarity.

Weighing terms in word-word matrix

- Problem with count based methods:
  - Frequent context terms are not that informative.
  - Functional words: “the”, “and”, “of”, etc.
- For documents we can use tf-idf to give higher values for more informative terms.

PMI – Intuitions

- Words that frequently occur together are more informative.
- Very frequent words are less informative.
- Words that occur together more frequently than would be expected are very informative.

PMI – Definition

$$\text{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}$$

Pointwise mutual information (PMI) easures how more often w and c occurs together than what would be expected by chance. Positive value means more frequent and negative means less frequent.

PMI – Problems

- Negative values are unreliable. Notion of *unrelatedness* is problematic.
- Solution: Use only positive values.

$$\text{PPMI}(w, c) = \max\left(\log_2 \frac{P(w, c)}{P(w)P(c)}, 0\right)$$

- Rare words get high values.
- Solution: Use modified function to calculate $P(c)$.

$$P_\alpha(c) = \frac{\text{count}(c)^\alpha}{\sum_c \text{count}(c)^\alpha}$$

- Using $\alpha < 1$ increases $P(c)$ and lowers PMI for rare events.

PMI – Summary

- PMI increases value for informative words.
- Use PPMI to ignore negative values.
- Use $P(c)^\alpha$ to reduce PMI of infrequent words.