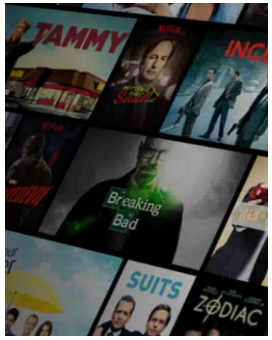
Features?

Images

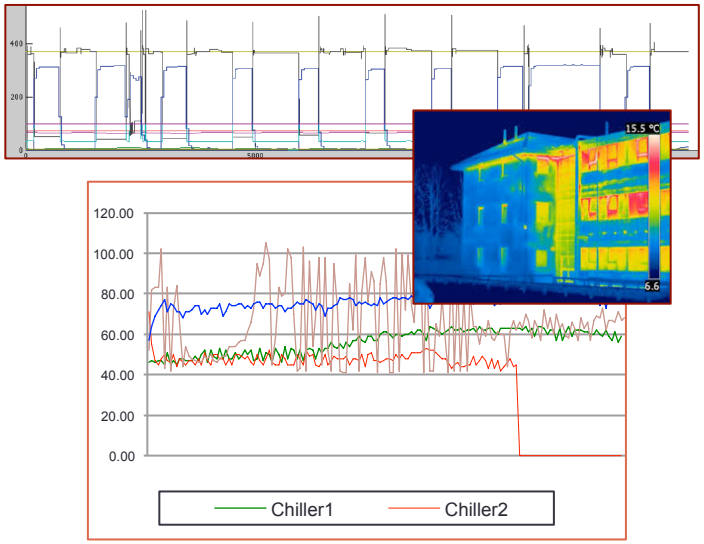
Books

Videos



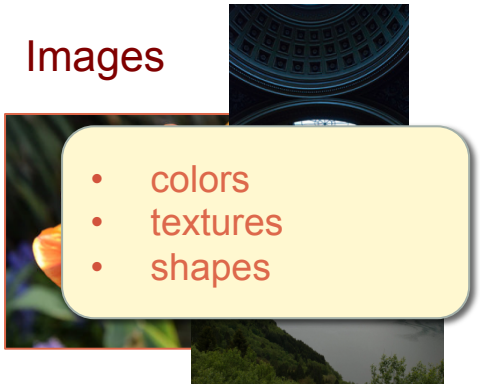
Social network

Sensor readings



Features?

Images



- colors
- textures
- shapes

Books

- words
- authors
- publishers

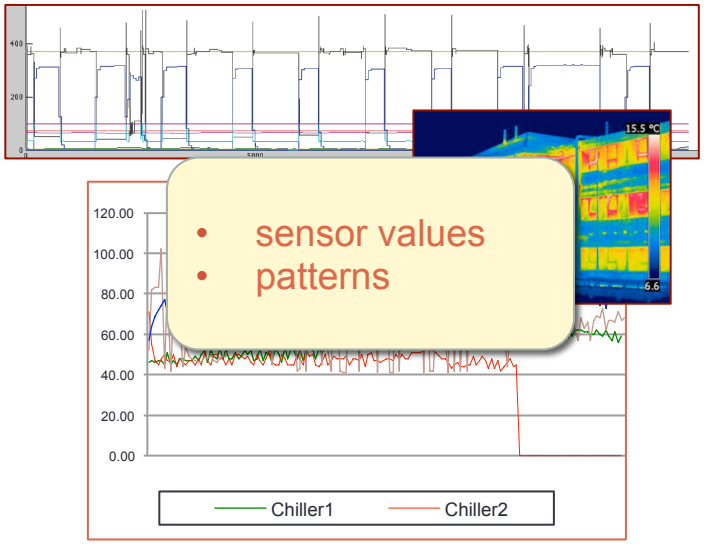
Videos

- actors
- ratings
- directors

Social network

- connections
- likes
- interactions

Sensor readings



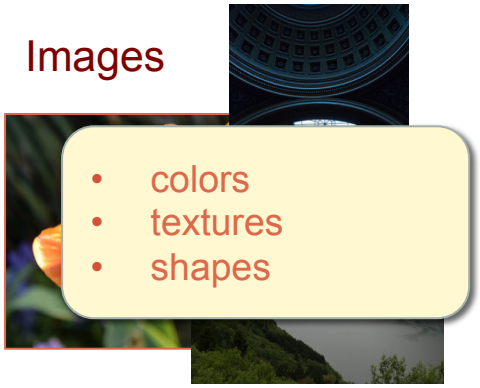
- sensor values
- patterns

Data representation?

- Sets
  - Does a feature instance exists in a given object?
- Vectors
  - How many times a feature instance exists in the given object?
  - What is the strength of a feature instance is in the given object?
- Strings
  - What is the order of feature instances in the given object?
- Trees
  - What is the hierarchy of the feature instance in the given object?
- Graphs
  - What are the pairwise relationships among feature instances in the given object?

Data representation?

Images



- colors
- textures
- shapes

Books

- words
- authors
- publishers

Videos

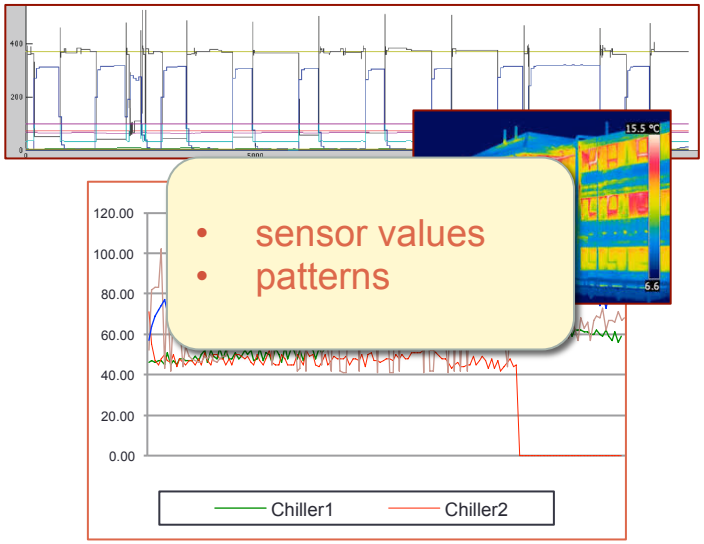
- actors
- ratings
- directors

count

Social network

- connections
- likes
- interactions

Sensor readings



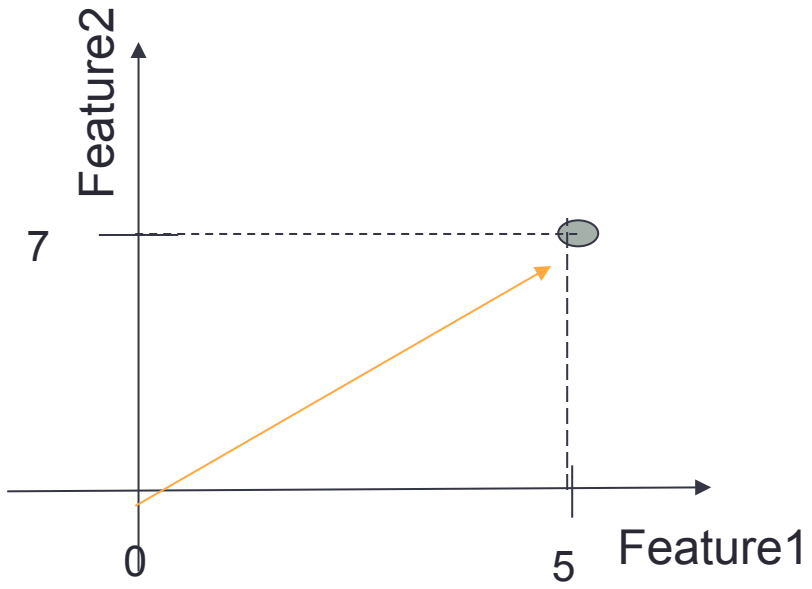
- sensor values
- patterns

How do we represent counts?

- Feature1 occurs 5 times

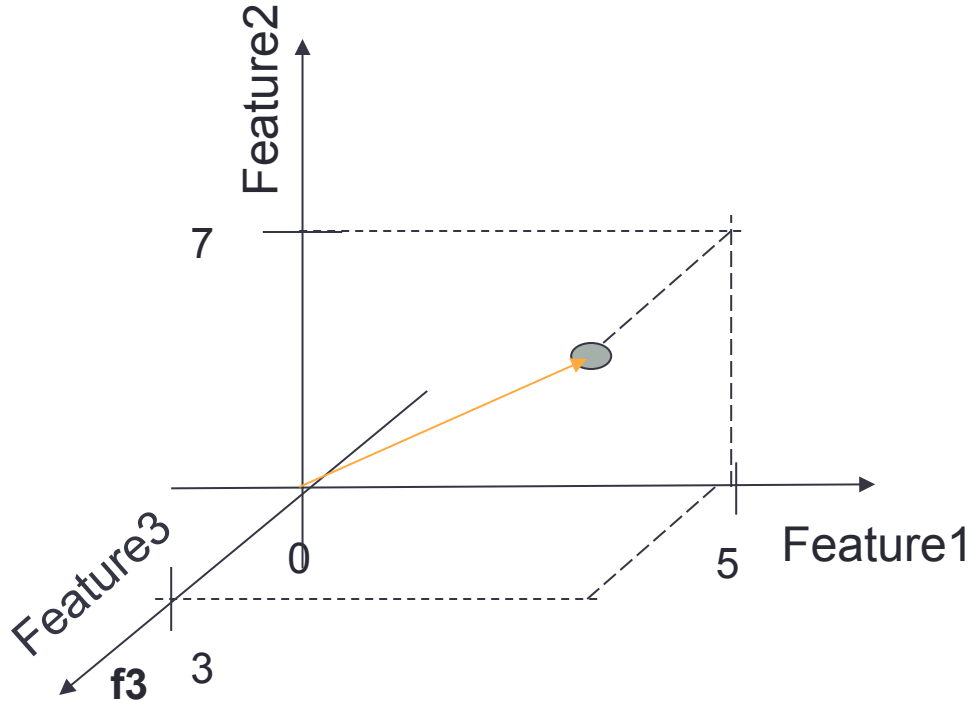
How do we represent counts?

- Feature1 occurs 5 times
- Feature2 occurs 7 times

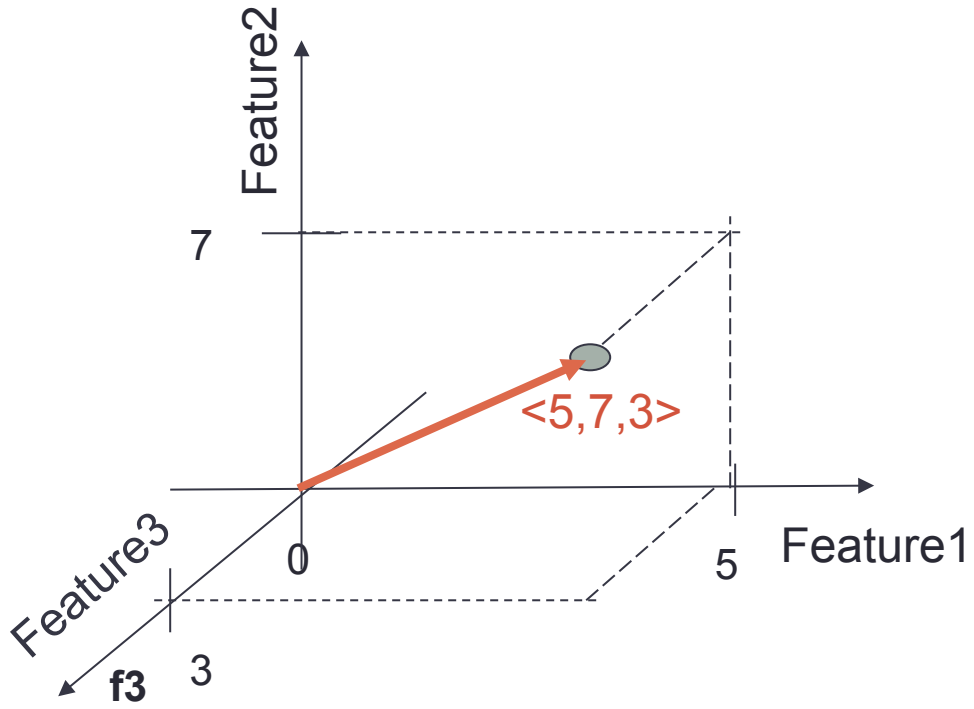


How do we represent counts?

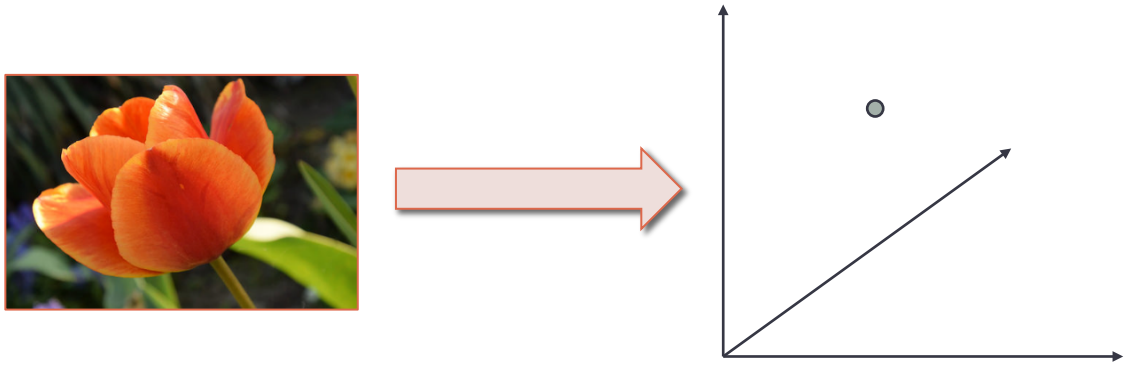
- Feature1 occurs 5 times
- Feature2 occurs 7 times
- Feature3 occurs 3 times



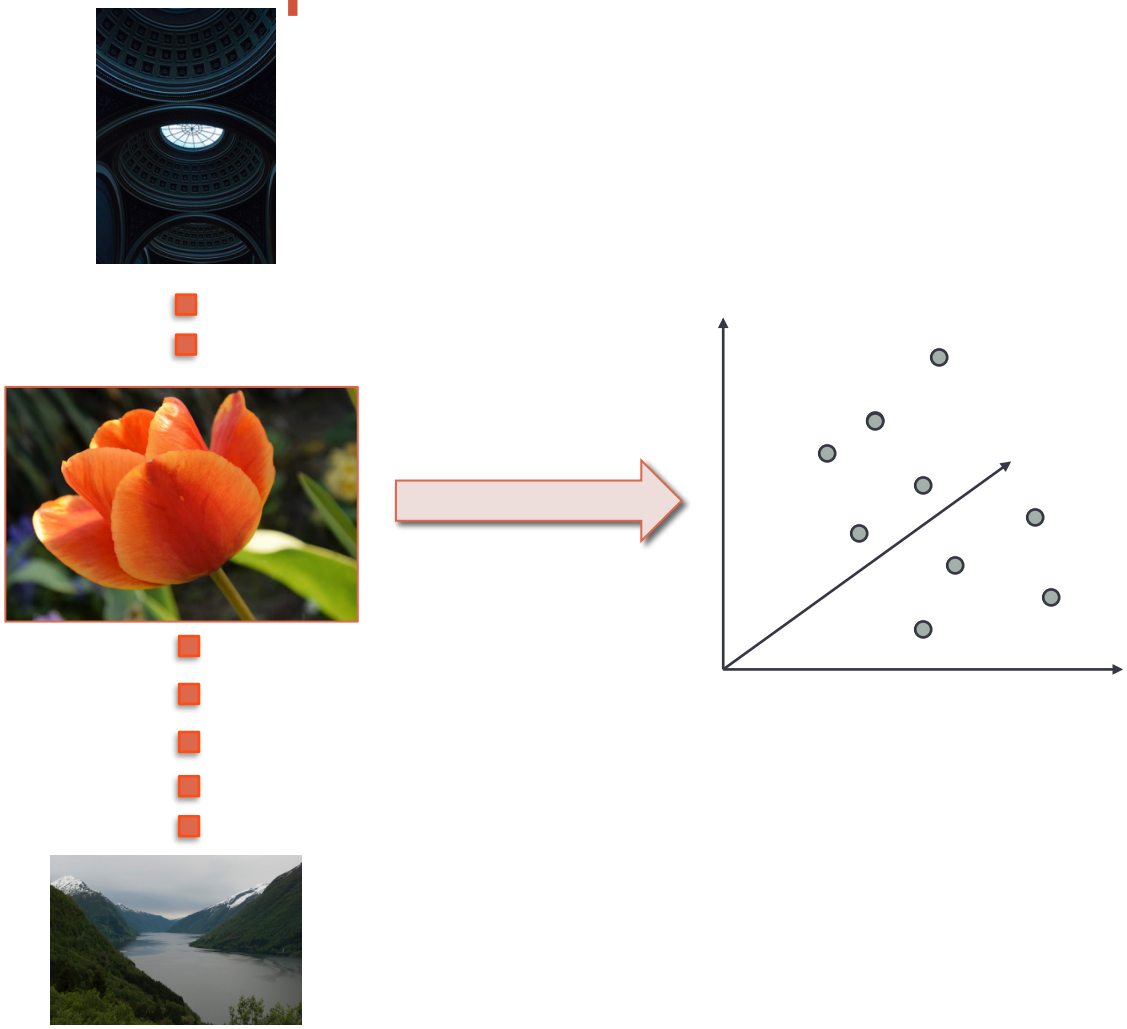
Vector representation



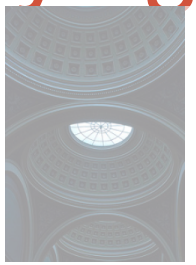
Vector representation

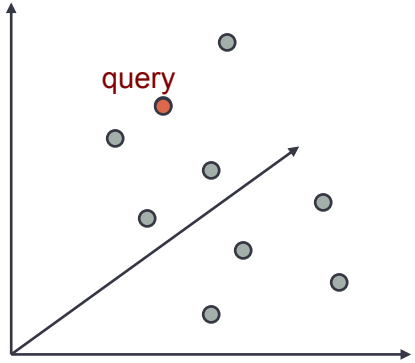


Vector representation

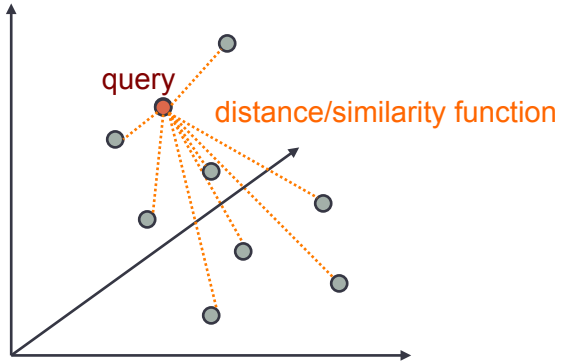


Querying

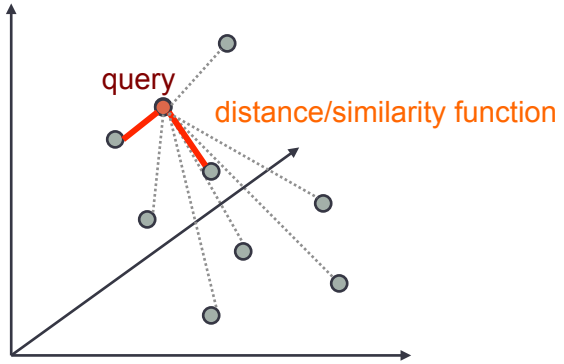




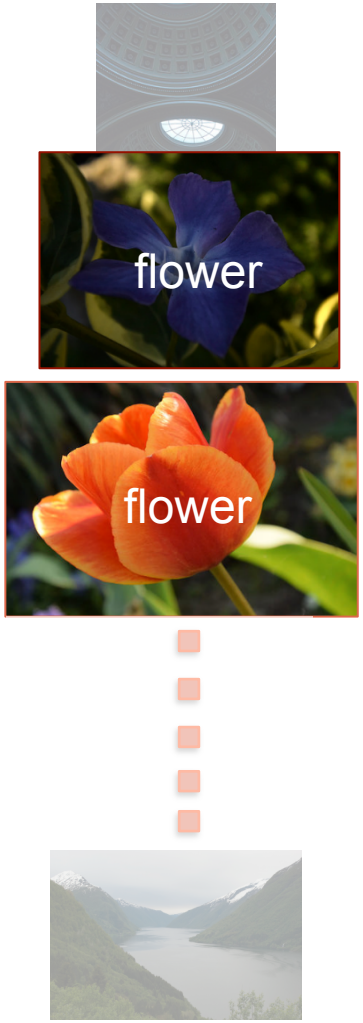
Distance/similarity measurement



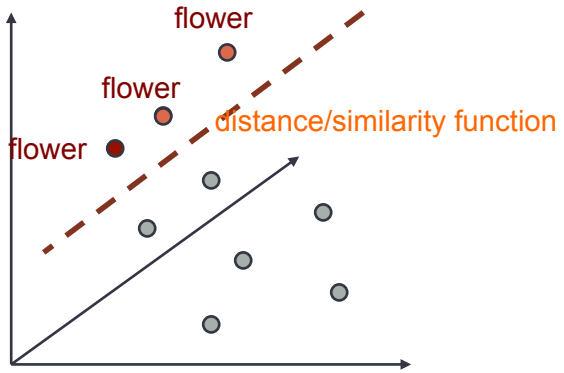
Top-k retrieval…



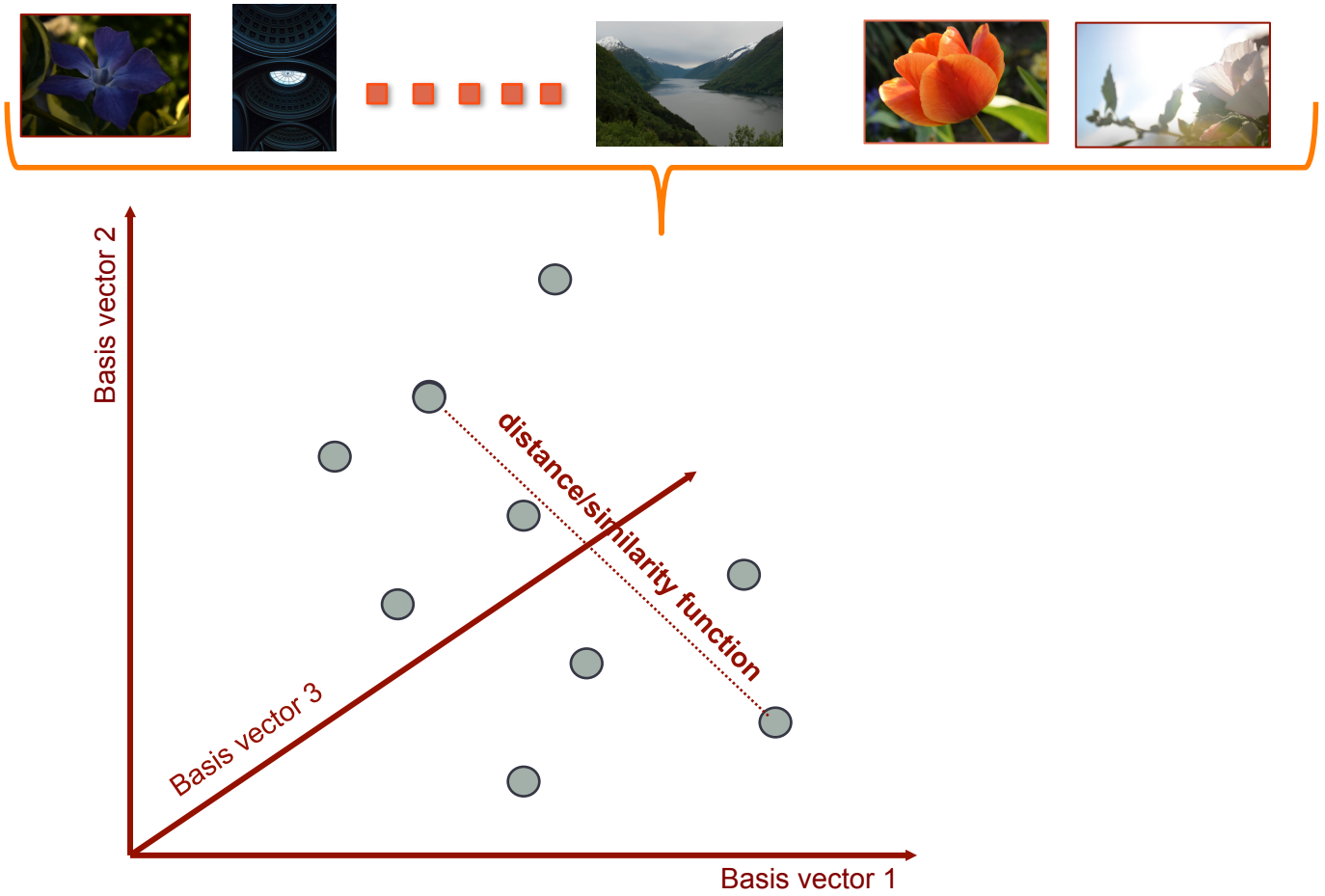
Classification



Classification



Vector spaces



Vector Space

Definition (Vector space): *The set* $\mathbb{S}$ *is a vector space iff for all* $\vec{v}_i$, $\vec{v}_j$, $\vec{v}_k \in \mathbb{S}$ *and for all* $c, d \in \mathbb{R}$, *the following axioms hold:*

- $\vec{v}_i + \vec{v}_j = \vec{v}_j + \vec{v}_i$
- $(\vec{v}_i + \vec{v}_j) + \vec{v}_k = \vec{v}_j + (\vec{v}_i + \vec{v}_k)$
- $\vec{v}_i + \vec{0} = \vec{v}_i$ *(for some* $\vec{0} \in \mathbb{S}$*)*
- $\vec{v}_i + (-\vec{v}_i) = \vec{0}$ *(for some* $-\vec{v}_i \in \mathbb{S}$*)*
- $(c + d)\vec{v}_i = (c\vec{v}_i) + (d\vec{v}_i)$
- $c(\vec{v}_i + \vec{v}_j) = c\vec{v}_i + c\vec{v}_j$
- $(cd)\vec{v}_i = c(d\vec{v}_i)$
- $1.\vec{v}_i = \vec{v}_i$

The elements of $\mathbb{S}$ *are called vectors.*

Basis of a Vector Space

Definition (Linear independence and basis): *Let* $V = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ *be a set of vectors in a vector space* $\mathbb{S}$. *The vectors in* V *are said to be* linearly independent *if*

$$\left(\sum_{i=1}^{n} c_i \vec{v}_i = \vec{0}\right) \longleftrightarrow \quad c_1 = c_2 = \cdots = c_n = 0.$$

The linearly independent set V *is said to be a basis for* $\mathbb{S}$ *if for every vector,* $\vec{u} \in \mathbb{S}$, *there exist constants* c_1 *through* c_n *such that*

$$\vec{u} = \sum_{i=1}^{n} c_i \vec{v}_i.$$

- The basis functions should be
  - non-redundant
  - complete

Inner product

Definition (Inner product and orthogonality): *The inner product on a vector space* $\mathbb{S}$ *is a function* $\mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}$ *such that*

- $\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{u}$,
- $(c_1\vec{u} + c_2\vec{v}) \cdot \vec{w} = c_1(\vec{u} \cdot \vec{w}) + c_2(\vec{v} \cdot \vec{w})$, *and*
- $\forall_{\vec{v} \neq \vec{0}} \quad \vec{v} \cdot \vec{v} > 0$.

The vectors $\vec{u}$ *and* $\vec{v}$ *are said to be* orthogonal *if*

$$\vec{u} \cdot \vec{v} = 0.$$

- A set of vectors whose projections onto each other are of 0 length can be used as basis vectors (if they are also complete)

A set of mutually orthogonal vectors are also linearly independent

Norm

Definition (Norms and orthonormal basis): *A* norm *(commonly denoted as* $\|\cdot\|$*) is a function that measures the length of vectors. A vector,* $\vec{v}$*, is said to be normalized if* $\|\vec{v}\| = 1$*. A basis,* $V = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$*, of the vector space* $\mathbb{S}$ *is said to be orthonormal if*

$$\forall_{\vec{v}_i, \vec{v}_j} \quad \vec{v}_i \cdot \vec{v}_j = \delta_{i,j},$$

such that if $i = j$*,* $\delta_{i,j} = 1$ *and* 0 *otherwise.*[2]

P-norms

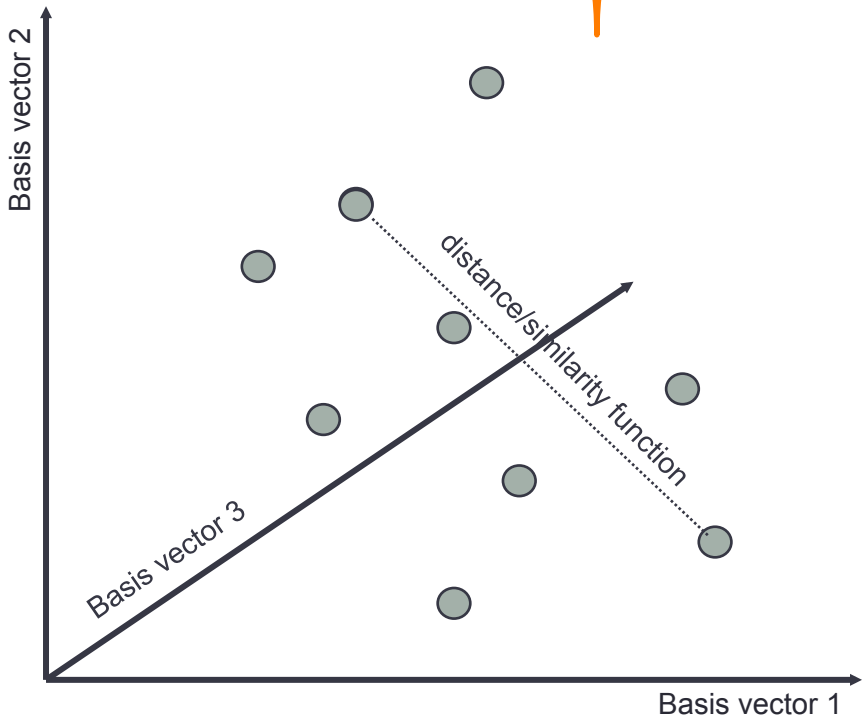
The most commonly used family of norms are the p-norms. Given a vector $\vec{v} = \langle w_1, \ldots, w_n \rangle$, the p-norm is defined as

$$\|\vec{v}\|_p = \left(\sum_{i=1}^{n} |w_i|^p \right)^{\frac{1}{p}} .$$

At the limit, as p goes to infinity, this gives the max-norm

$$\|\vec{v}\|_\infty = \max_{i=1\ldots n} \{|w_i|\} .$$

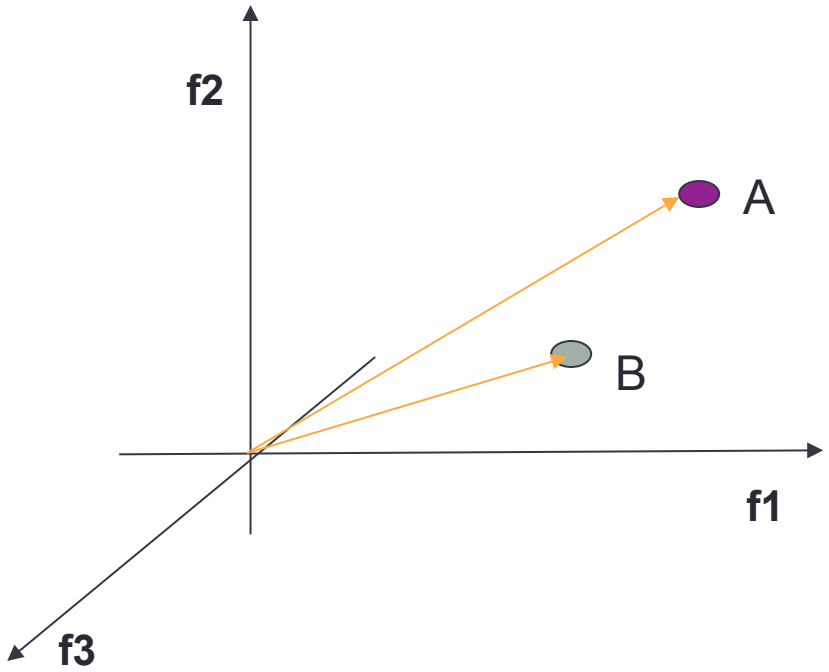
Vector spaces



- What are good features to use as basis vectors?
- How many features do we need as basis vectors?
- What is a good distance/similarity function?

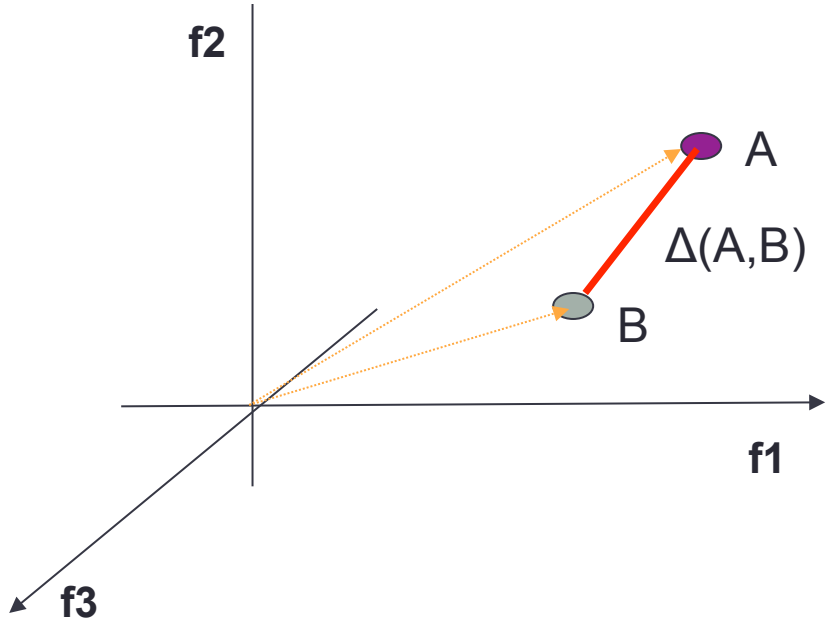
Distance between two images???

- Given A and B vectors, can we measure how different they are?

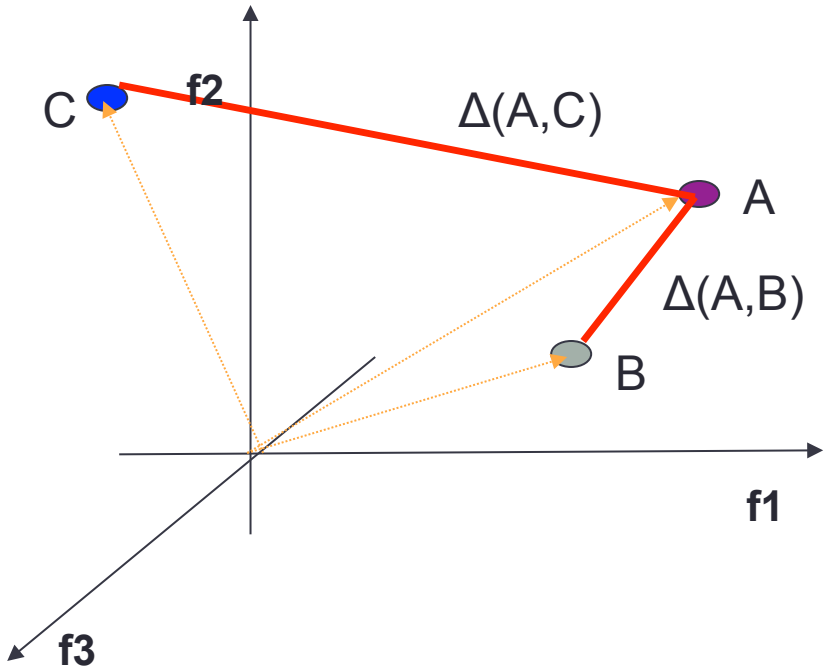


Euclidean distance

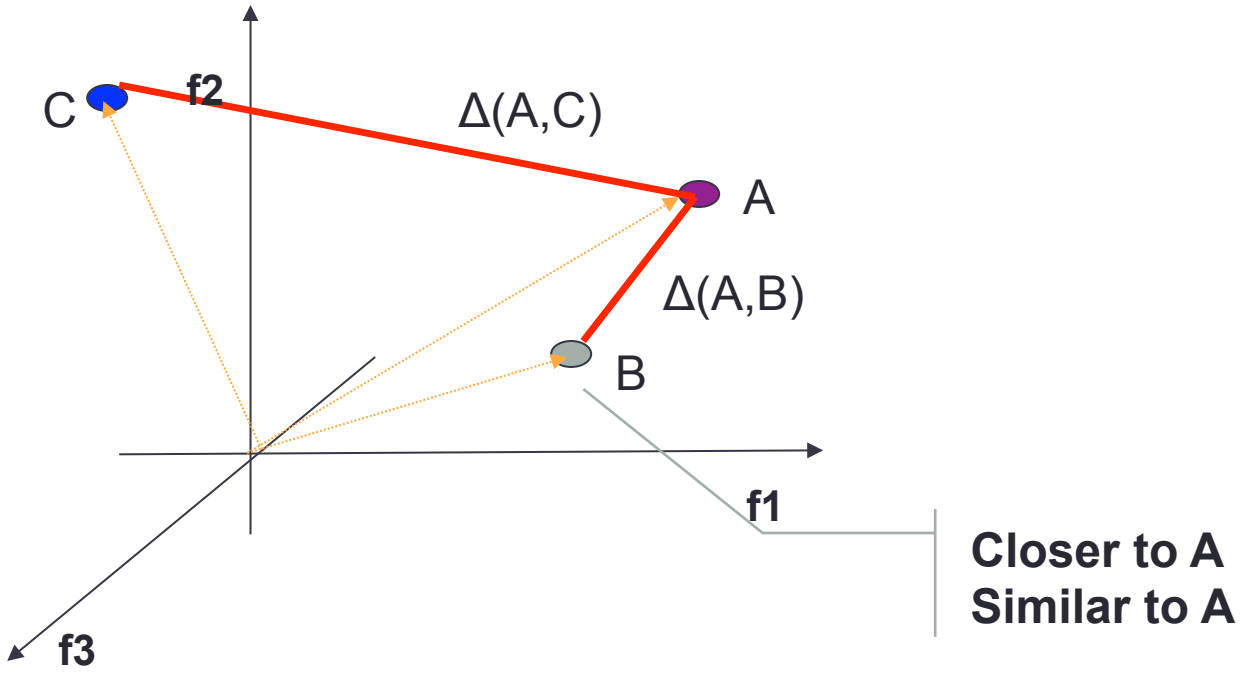
$$\Delta(A,B) = \sqrt{(a1-b1)^2 + (a2-b2)^2 + (a3-b3)^2}$$



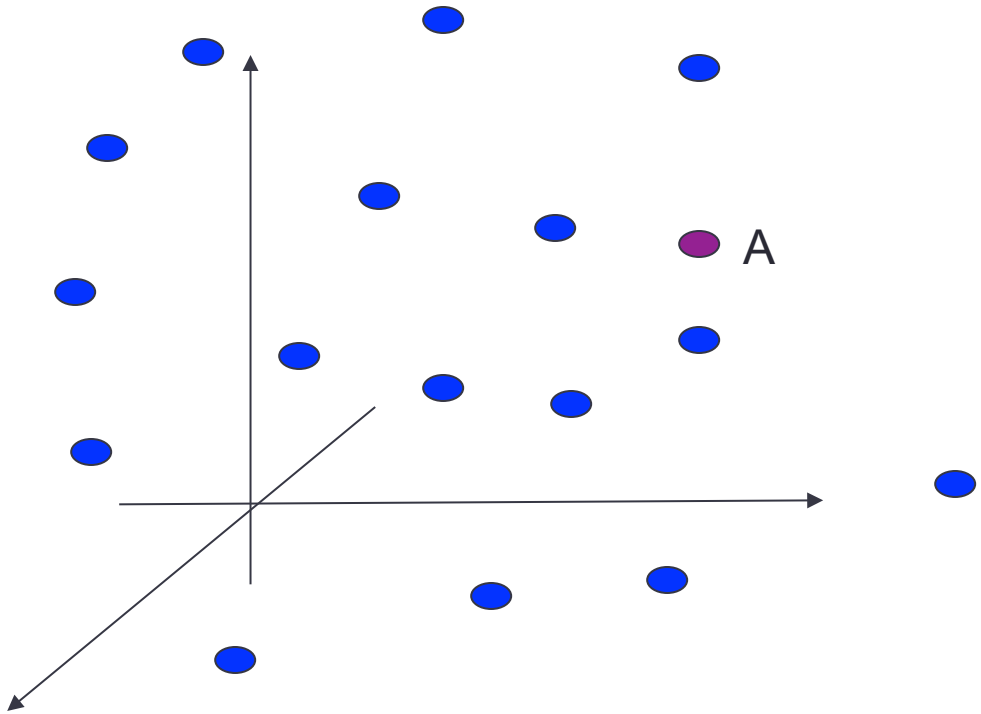
Which image is more similar to A?



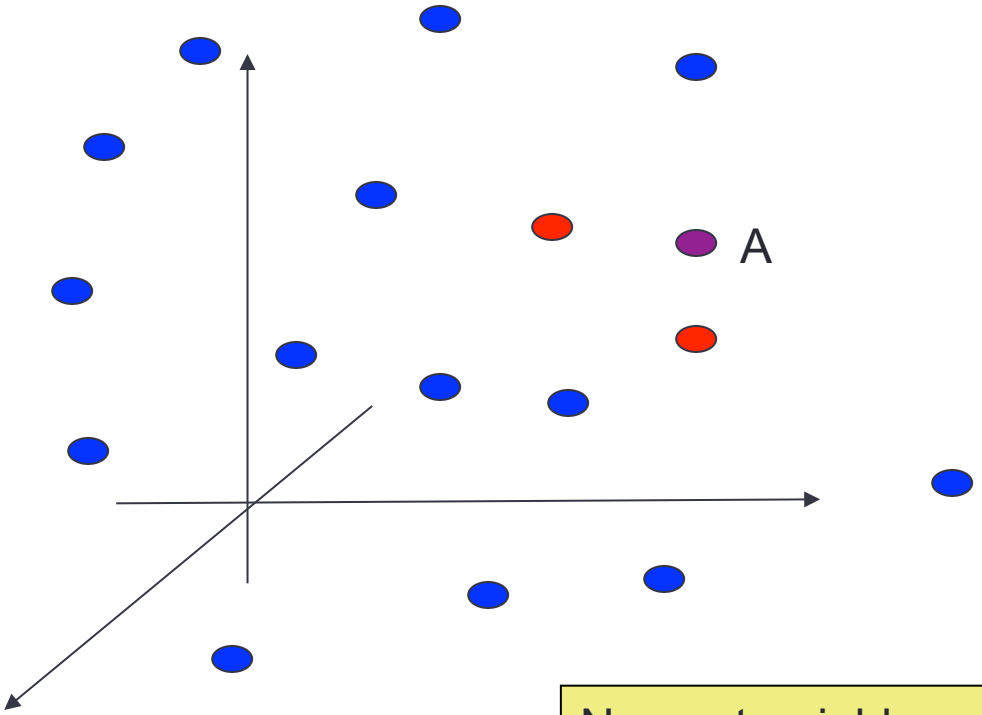
Which image is more similar to A?



“Find 2 most similar images to A”

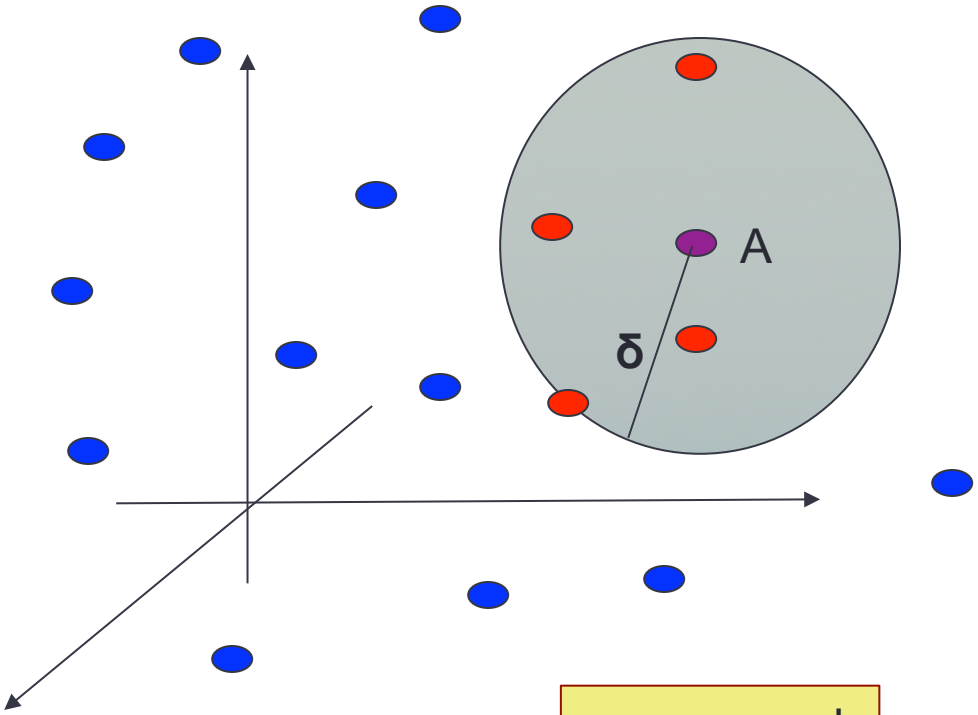


“Find 2 most similar images to A”



Nearest-neighbor search

“Find images at most δ different from A”



range search

What is a good measure then??

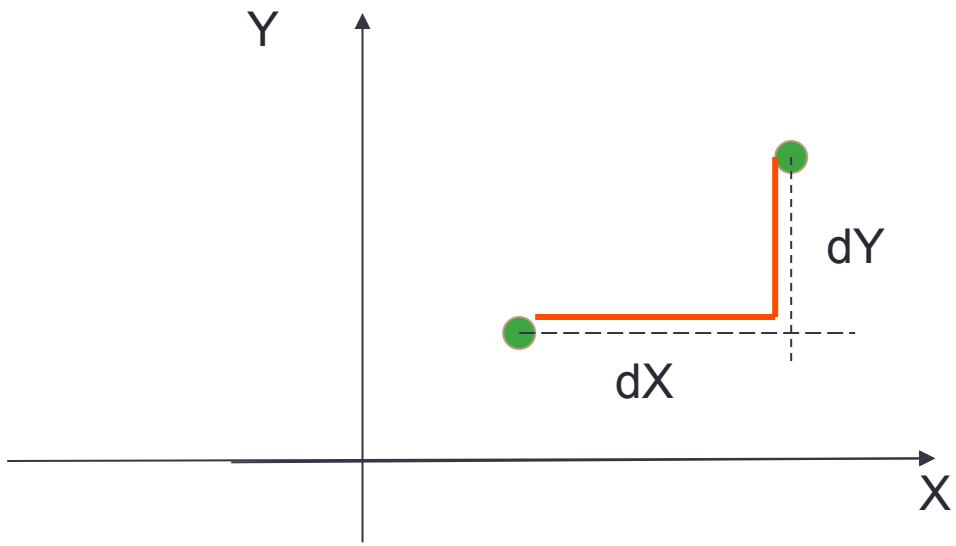
- Application dependent...
- ...but, distances in a metric space help indexing!

Metric model: axioms

- Any function *d* expressing a metric distance must satisfy the following axioms:
  - self-minimality: $d(s,s) = 0$
  - minimality $d(s_1,s_2) >= d(s_1,s_1)$
  - symmetry $d(s_1, s_2) = d(s_2, s_1)$
  - triangular inequality $d(s_1,s_2) + d(s_2,s_3) >= d(s_1,s_3)$

Metric distances (Minkowski metrics)

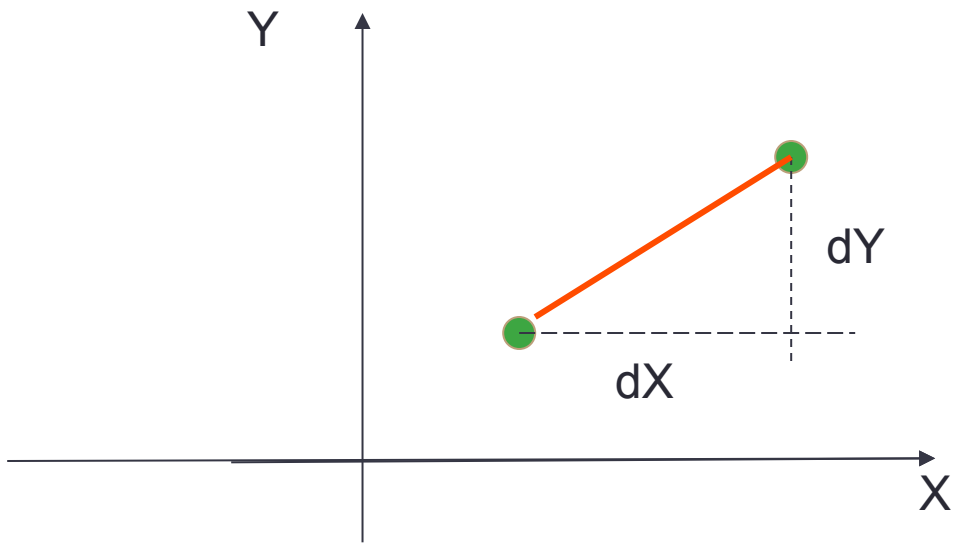
- L1-metric: d = (dX+dY)



Also called Manhattan Distance

Metric distances (Minkowski metrics)

- L2-metric: $d = (dX^2+dY^2)^{1/2}$



Also called Euclidean Distance

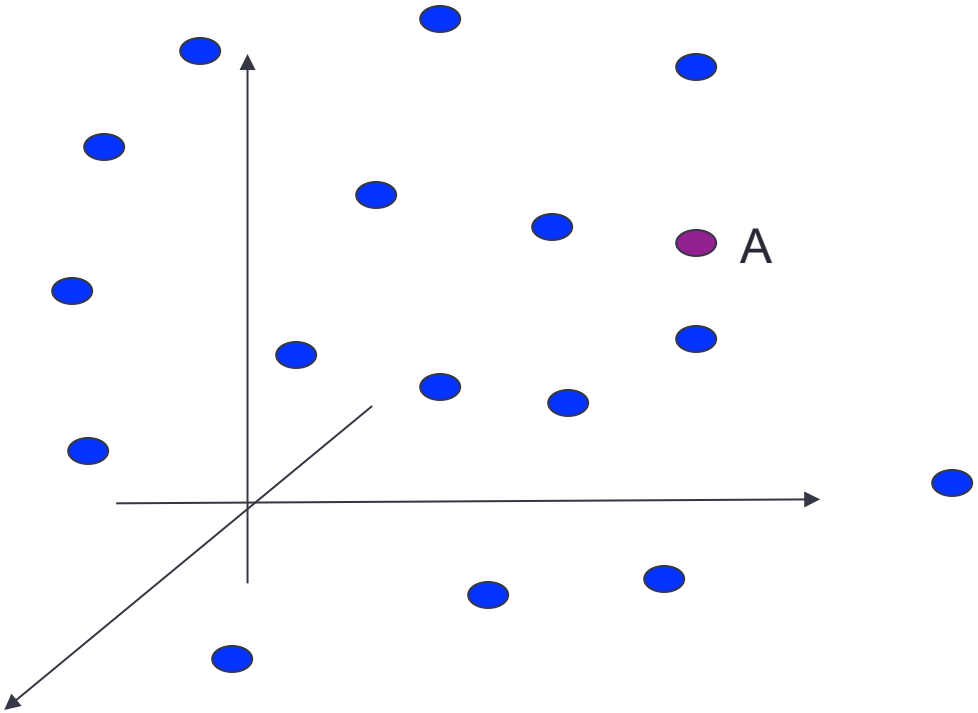
Metric distances (Minkowski metrics)

- L3-metric; $d = (dX^3+dY^3)^{1/3}$
- .....
- .....
- L(infinity): d = max{X,Y}

…metric model

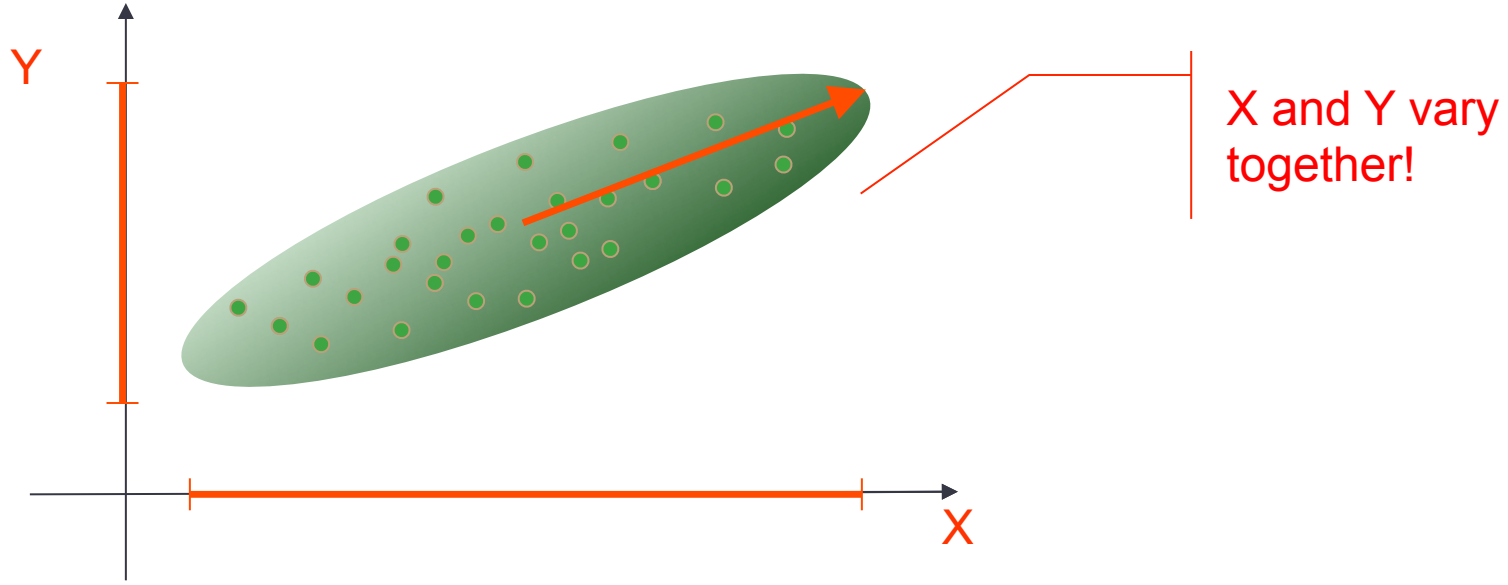
- Well suited for certain kinds of similarity evaluation, such as color based comparisons
- Consistent with widely used approaches from computer vision and pattern recognition communities
  - results suggest that the L1 metric may better capture human notions of image similarity.
- Makes it relatively easy to index data, modeled as vectors of properties, in terms of classical multi-dimensional indexing techniques.

Are there other similarity/distance measures?



Mahalonobis Distance

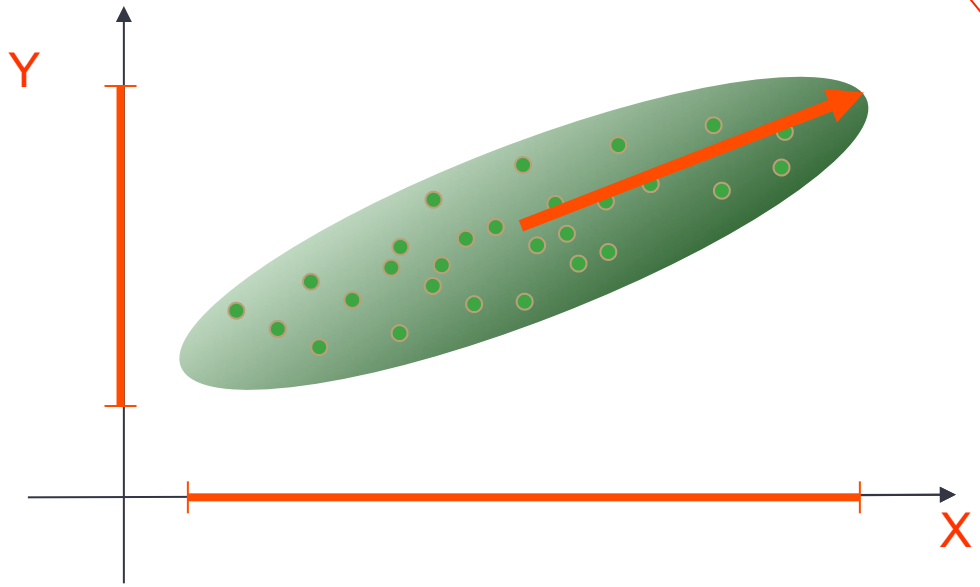
$$\Delta_{Euc}\left(\vec{a},\vec{b}\right)=\sqrt{\left(\vec{a}-\vec{b}\right)^{T}\left(\vec{a}-\vec{b}\right)}$$



$$S[x,y] = Cov(x,y) = E\left((x-\mu_x)(y-\mu_y)\right)$$

Mahalonobis Distance

$$\Delta_{Mah}\left(\vec{a},\vec{b}\right) = \sqrt{\left(\vec{a}-\vec{b}\right)^T S^{-1}\left(\vec{a}-\vec{b}\right)}$$



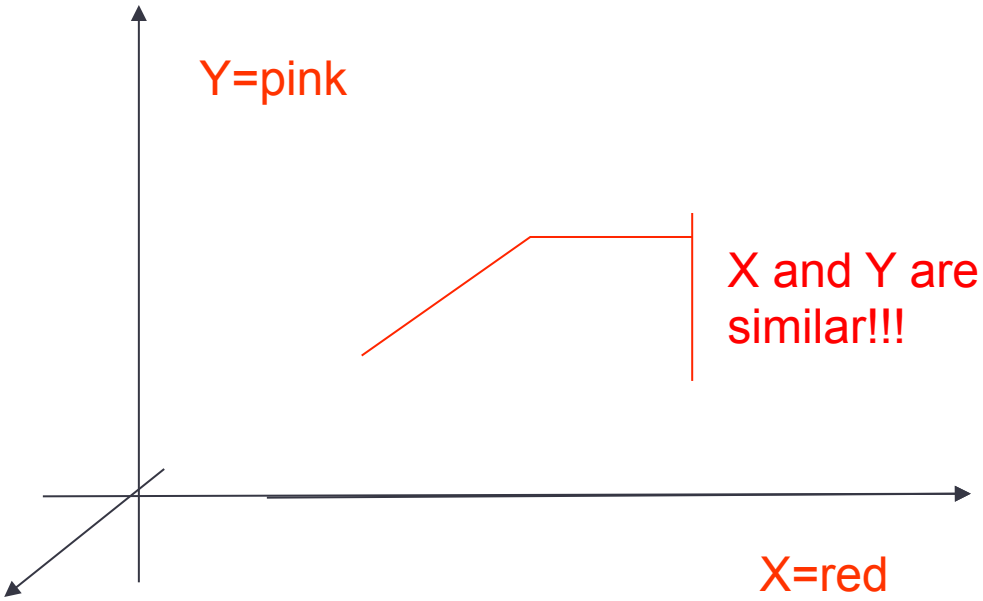
Covariance matrix

Covariance matrix

$$S[i,j] = Cov(i,j) = E\left((i-\mu_i)(j-\mu_j)\right)$$

Quadratic Distance

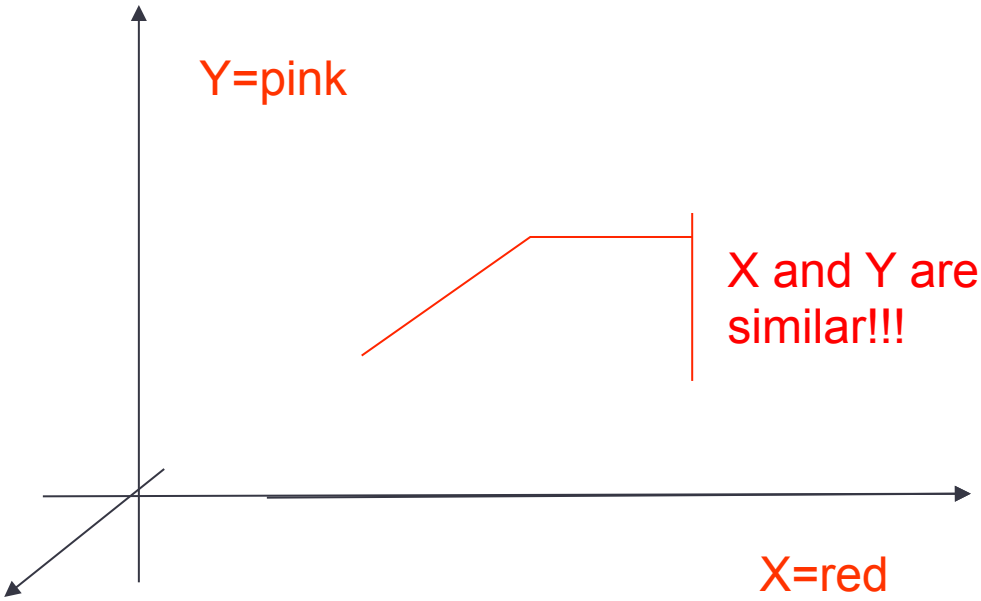
$$\Delta_{Euc}\left(\vec{a},\vec{b}\right)=\sqrt{\left(\vec{a}-\vec{b}\right)^{T}\left(\vec{a}-\vec{b}\right)}$$



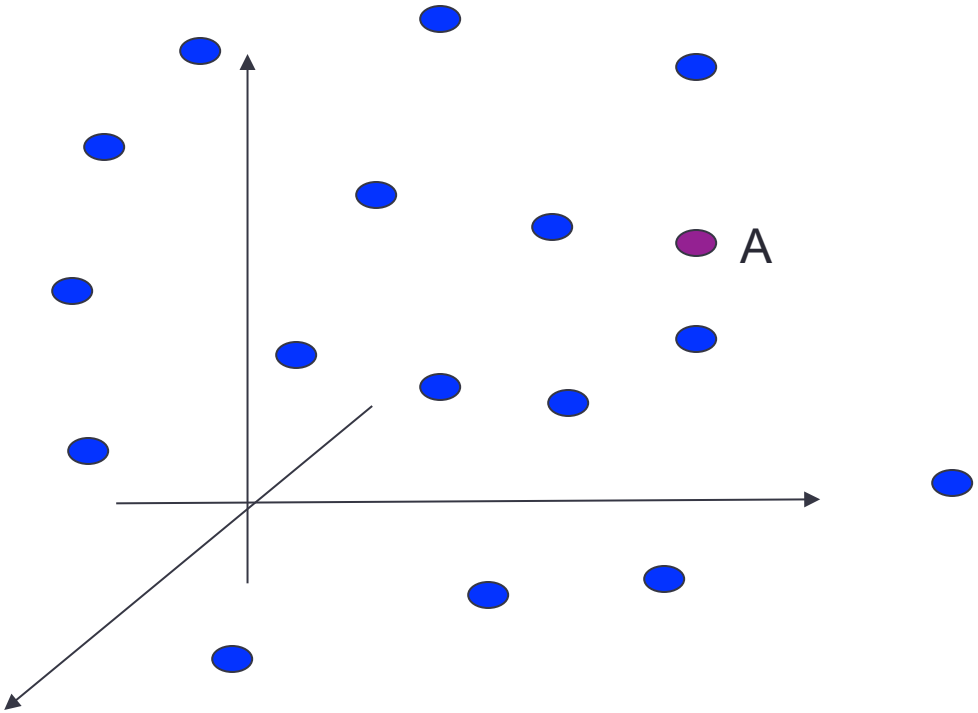
Quadratic Distance

$$\Delta_{quad}\left(\vec{a},\vec{b}\right)=\sqrt{\left(\vec{a}-\vec{b}\right)^{T}A^{-1}\left(\vec{a}-\vec{b}\right)}$$

Inverse of the feature similarity matrix



Are there other similarity measures?



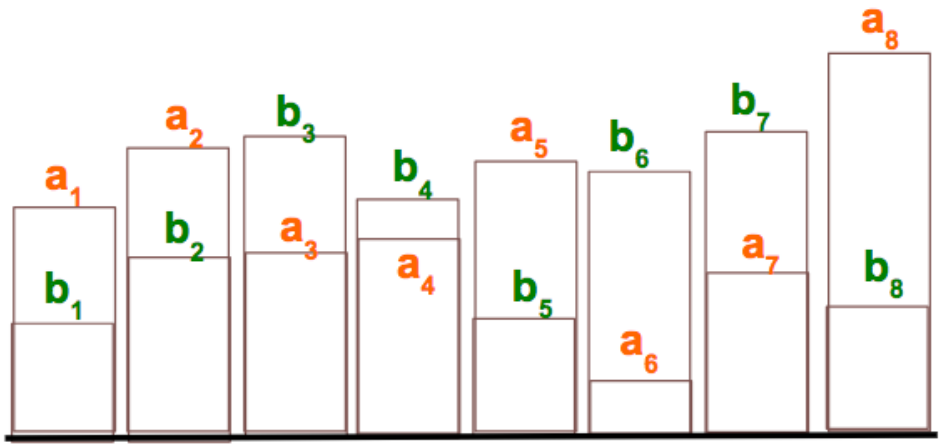
Intersection similarity

$$\text{sim}_{\text{int}}\left(\vec{a},\vec{b}\right)=\frac{\sum_{i=1..n}\min(a_i.b_i)}{\sum_{i=1..n}\max(a_i.b_i)}$$

Intersection Similarity

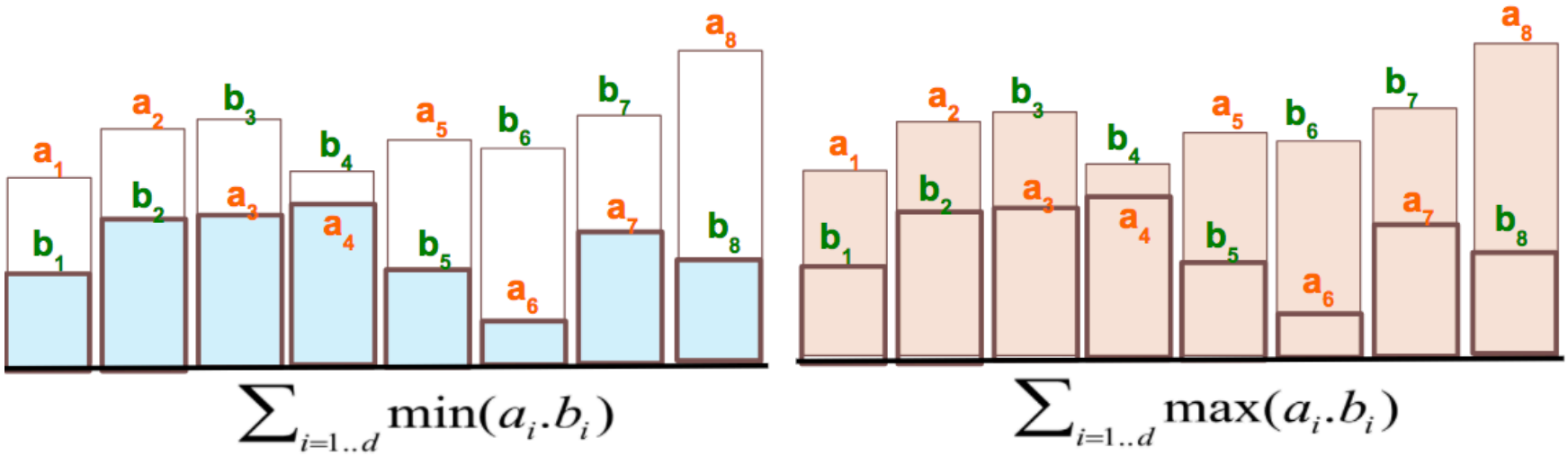
Consider two vectors

$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$

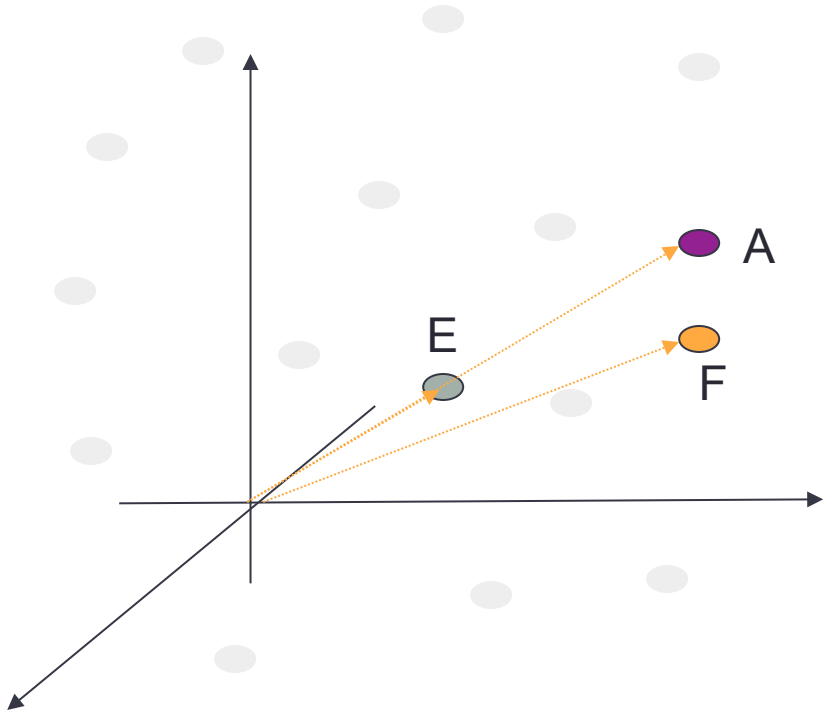


Intersection Similarity

$$\mathrm{sim}_{\mathrm{int}}\left(\vec{a},\vec{b}\right)=\frac{\sum_{i=1..d}\min(a_i.b_i)}{\sum_{i=1..d}\max(a_i.b_i)}$$

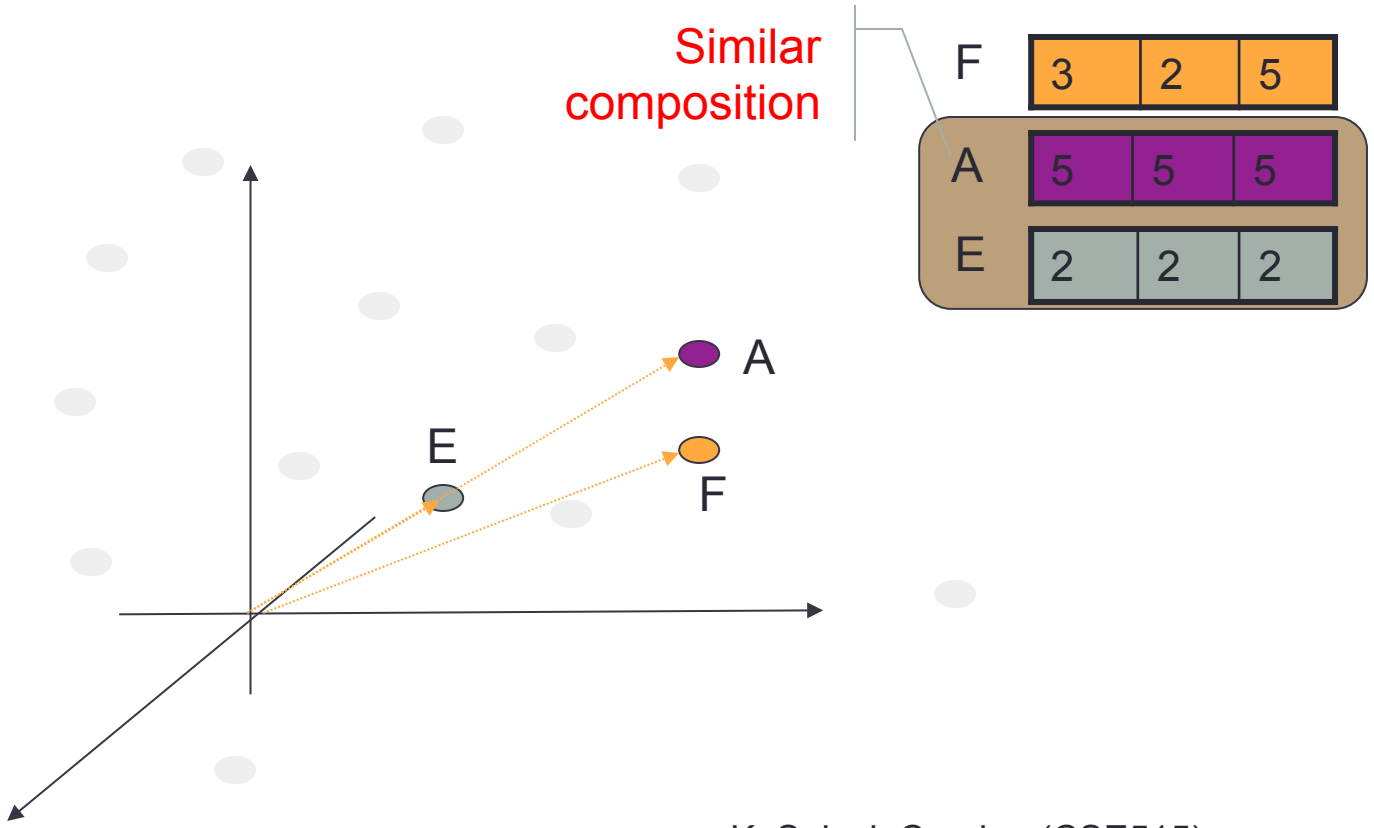


Let's try angles...

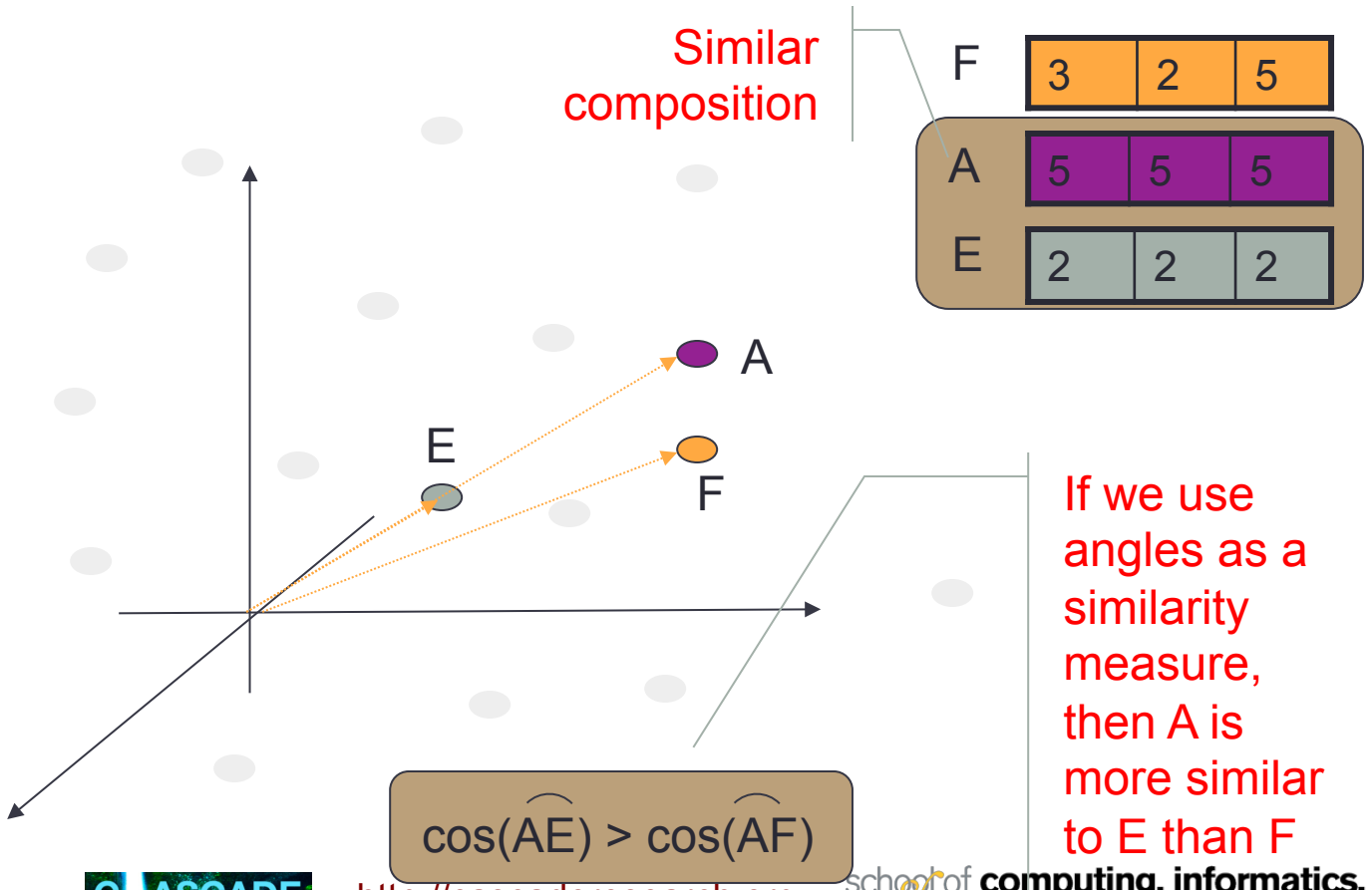


F	3	2	5
A	5	5	5
E	2	2	2

Let's try angles...



Let's try angles...



Angle-based measures

- Given

  - Dot product $\vec{a} = \langle a_1, a_2, \ldots, a_n \rangle \qquad \vec{b} = \langle b_1, b_2, \ldots, b_n \rangle$

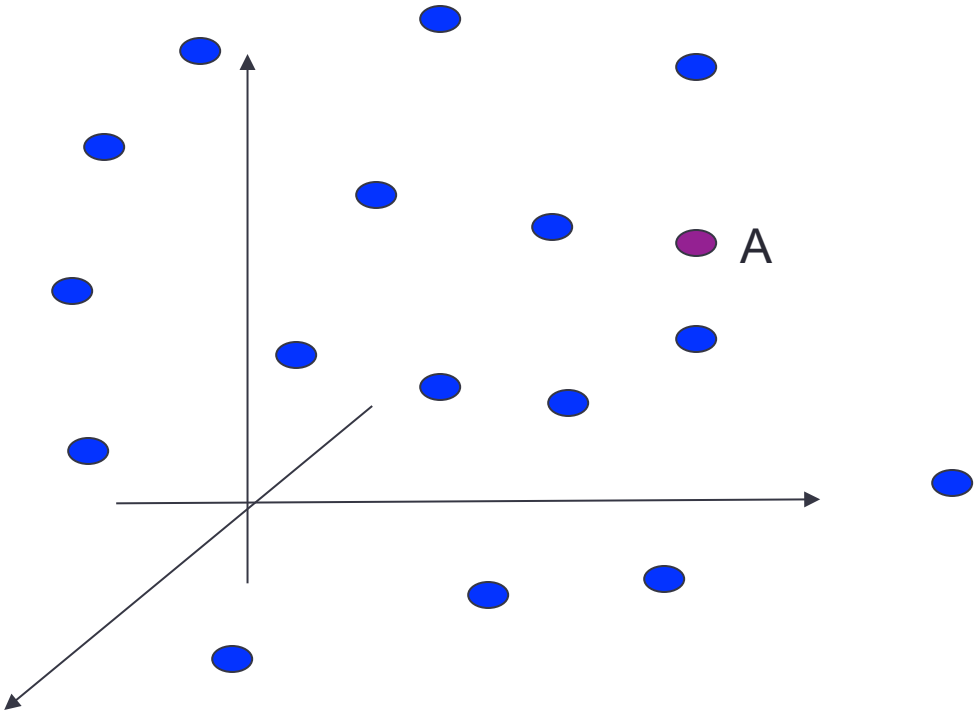
$$sim_{dot}(\vec{a}, \vec{b}) = \vec{a}.\vec{b} = \sum_{i=1}^{n} a_i b_i$$

  - Cosine similarity

$$sim_{\cos}(\vec{a}, \vec{b}) = \cos(\vec{a}, \vec{b}) = \frac{\vec{a}.\vec{b}}{|\vec{a}||\vec{b}|}$$

Cosine and dot product are the same if the vectors are unit length

Are there other similarity measures?



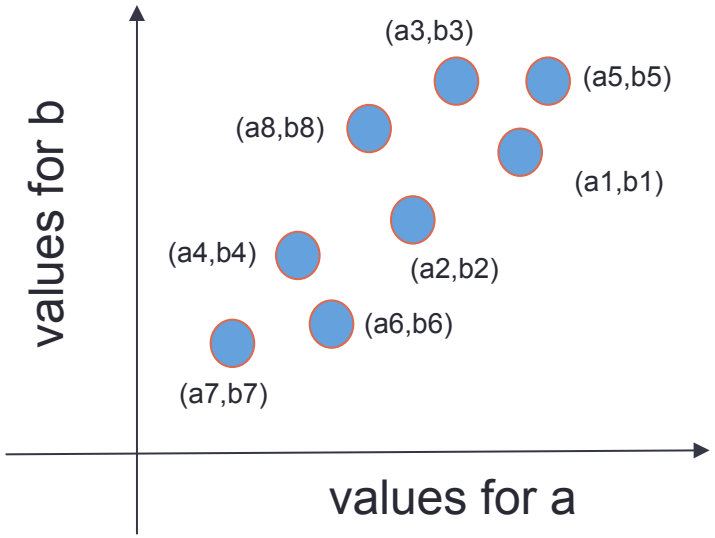
Other Commonly used Similarity / Distance Measures

- Pearson's correlation (similarity measure)
  - linear correlation (the strength of linear association) among the corresponding components of two vectors
- KL-Divergence (distance measure)
  - how one vector (interpreted as a probability distribution) diverges from the other
- Earth-movers distance (distance measure)
  - how one vector (interpreted as a probability distribution) diverges from the other
- Signal-to-noise ratio (similarity measure)
  - how one signal (represented as a vector) approximates another signal

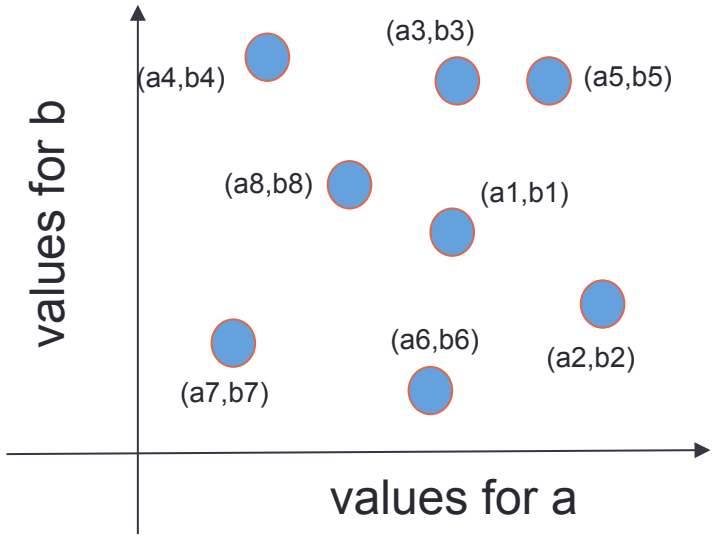
Pearson Correlation

- Consider two object vectors

$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$



Two positively correlated objects



Two un-correlated objects

KL-distance
(Kullback-Leibler divergence)

$$\Delta_{KL}\left(\vec{a},\vec{b}\right)=\sum_{i=1..n} a_i \log \frac{a_i}{b_i}$$

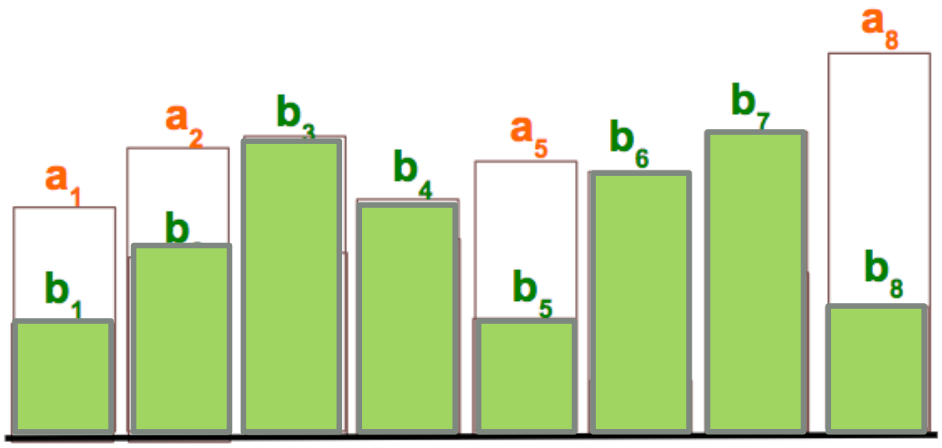
Relative entropy between two probability distributions

$$\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i = 1$$

Earth Mover's Distance (EMD)
(Wasserstein metric)

- Consider two probability distributions

$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$

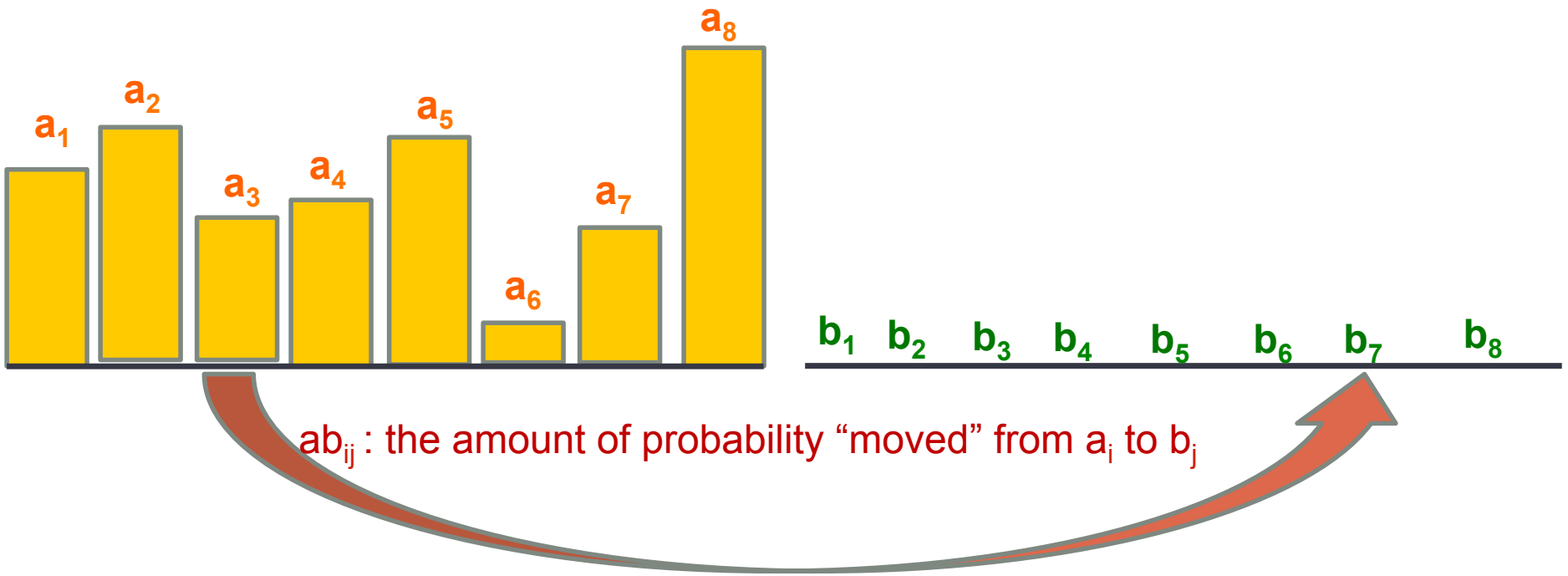


How can we convert vector a to vector b by redistributing the weights of vector a?

Earth Mover's Distance (EMD)
(Wasserstein metric)

- Consider two probability distributions

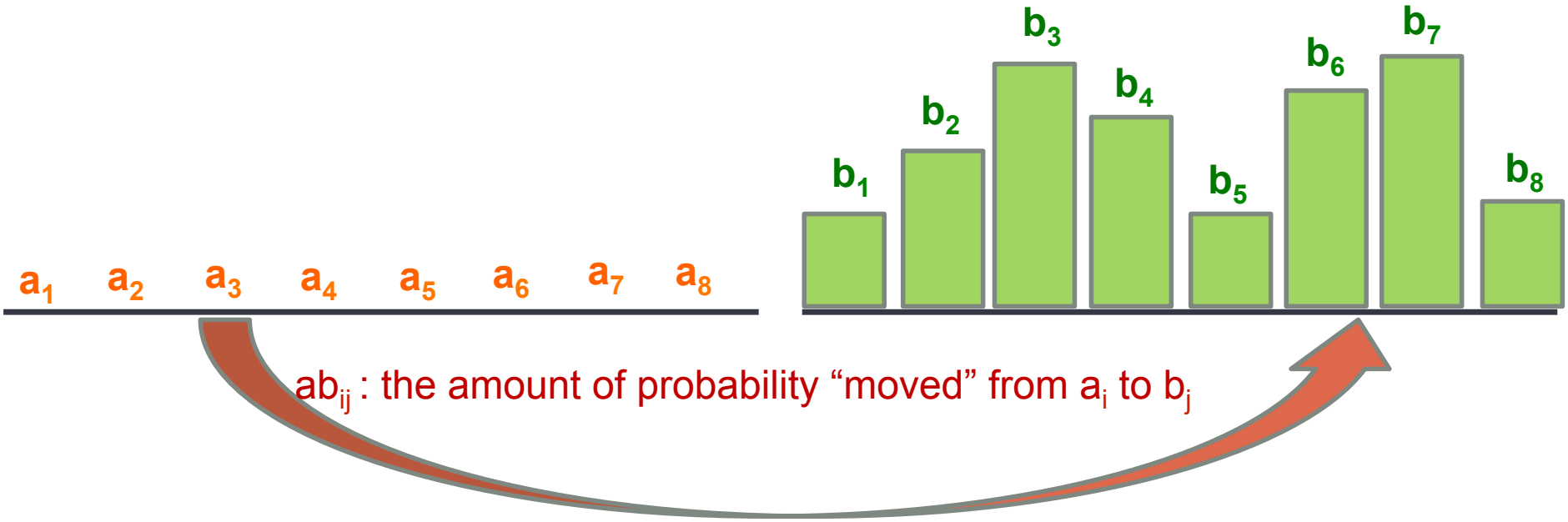
$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$



Earth Mover's Distance (EMD)
(Wasserstein metric)

- Consider two probability distributions

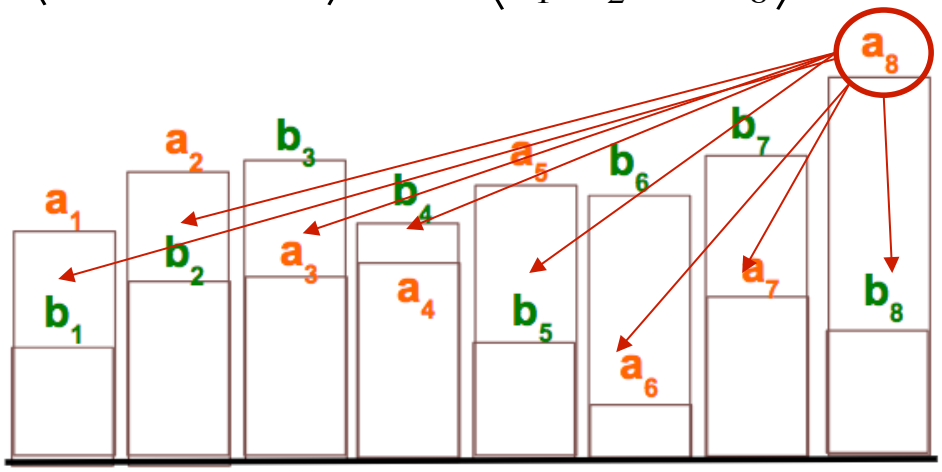
$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$



Earth Mover's Distance (EMD)
(Wasserstein metric)

- Consider two probability distributions

$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$



ab_{ij} : the amount of "probability" transferred from a_i to b_j

Earth Mover's Distance (EMD)
(Wasserstein metric)

- Consider two probability distributions

$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$

- Minimize $\cos t_{ij} = \sum_{i=1..n} \sum_{j=1..n} c_{ij} \times ab_{ij}$

- Such that

$$ab_{ij} \geq 0 \qquad \sum_{j=1..n} ab_{ij} = a_i \qquad \sum_{i=1..n} ab_{ij} = b_j \qquad \sum_{i=1..n} \sum_{j=1..n} ab_{ij} = 1$$

Earth Mover's Distance (EMD)
(Wasserstein metric)

- Consider two probability distributions

$$\vec{a} = \langle a_1, a_2, \ldots, a_8 \rangle \quad \vec{b} = \langle b_1, b_2, \ldots, b_8 \rangle$$

$$\Delta_{EMD}\left(\vec{a}, \vec{b}\right) = \frac{\sum_{i=1..n} \sum_{j=1..n} \cos t_{ij}}{\sum_{i=1..n} \sum_{j=1..n} ab_{ij}} = \sum_{i=1..n} \sum_{j=1..n} \cos t_{ij}$$

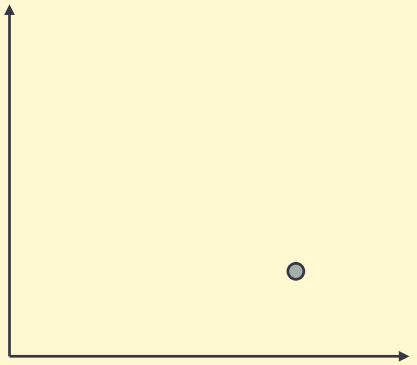
Signal-to-noise ratio (SNR)

$$sim_{snr}\left(\vec{a},\vec{b}\right) = 20\log_{10}\frac{\sqrt{\sum_{i=1..n} a_i^2}}{\sqrt{\sum_{i=1..n}\left(a_i - b_i\right)^2}}$$

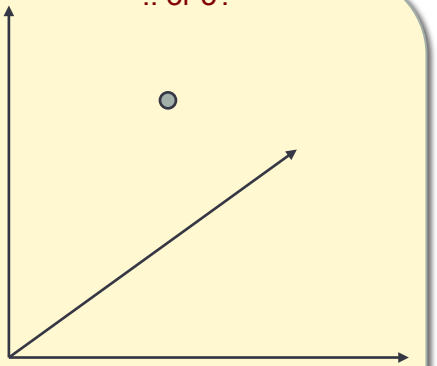
How many features do we need?

..just 1?

.. maybe 2?



.. or 3?



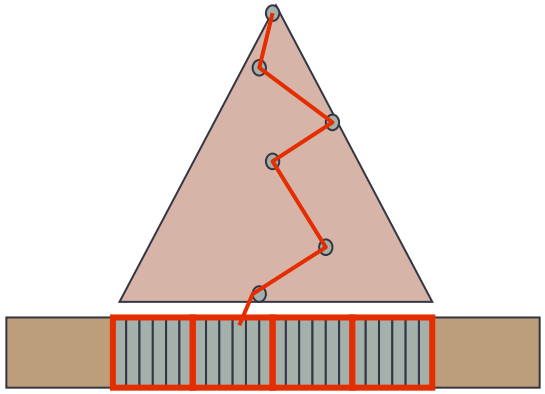
- ..or, many many more (100s, 1000s)???

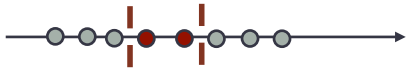
We want as few features as possible

- Dimensionality curse: The more dimensions we have, the less efficient and effective search and analysis become

- Efficiency: Search data structures are not very efficient at high dimensions

- Effectiveness: The more dimensions we have, the more data we need to prevent overfitting



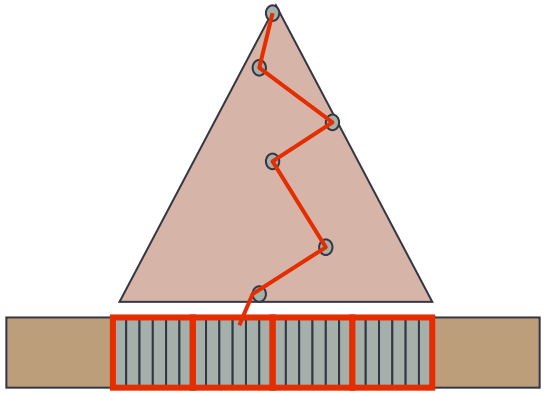


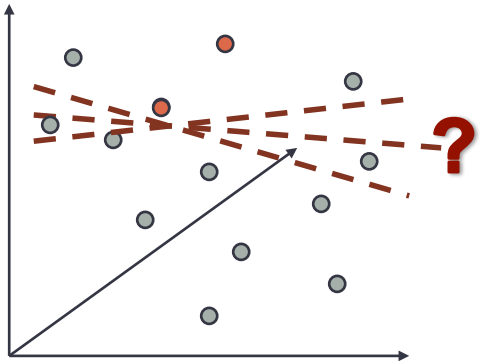
We want as few features as possible

- Dimensionality curse: The more dimensions we have, the less efficient and effective search and analysis become

- Efficiency: Search data structures are not very efficient at high dimensions

- Effectiveness: The more dimensions we have, the more data we need to prevent overfitting



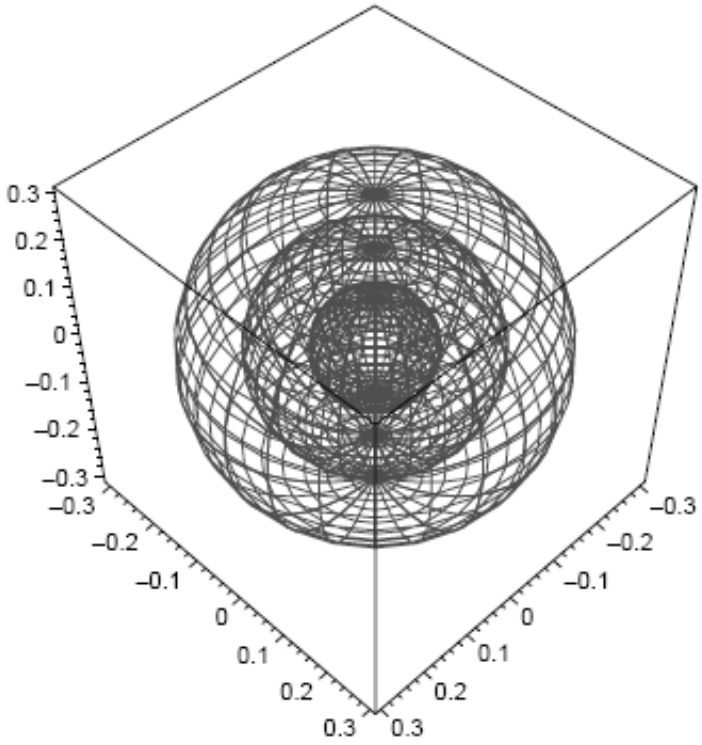


Feature selection

- more features mean more storage space, more feature extraction time, and higher cost of index management.
- some of the index structures require exponential storage space in terms of the features that are used for indexing
- more features means pairwise object similarity/distance computations are more expensive.
- searches in multi-dimensional vector spaces suffer from an inherent problem, called "dimensionality curse"

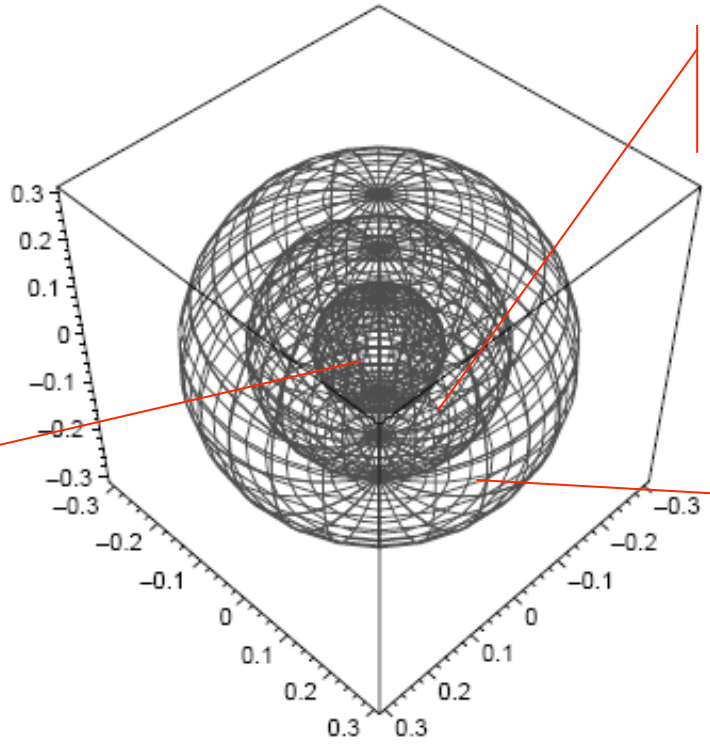
Dimensionality Curse

- Consider a query point and three alternative ranges:



Dimensionality Curse

- Consider a query point and three alternative ranges:



Volume of the second slice: $(4/3)\pi((2r)^3 - r^3) = (2^3\text{-}1^3)\ I$

Volume of the innermost sphere: $I=(4/3)\pi r^3$

Volume of the third slice: $(4/3)\pi((3r)^3 - (2r)^3) = (3^3 - 2^3)\ I$

Dimensionality Curse

- In n-dimensional space, if the number of points in the inner most sphere is *I*, then
  - number of points in the second slice is $O(2^{n-1} I)$
  - number of points in the second slice is $O(3^{n-1} I)$
  - number of points in the second slice is $O(4^{n-1} I)$

- This means that most of the points lie in the outermost slice!!!!

Power Law and Dimensionality Curse

- Power law: most real world data is such that given *d*-dimensional space, the number of pairs of elements within a given distance, *r*, follows the formula

$$pairs(r) = c \times r^d$$

Dimensionality curse

- If

Distance between closest and furthest points converge

$$\lim_{n\to\infty} \frac{\text{var}(\Delta_n(\vec{q},\vec{o}))}{\text{expected}(\Delta_n(\vec{q},\vec{o}))} = 0$$

..then, nearest neighbor search does not make any sense.

What do we need to consider when selecting features?

- Application semantics
- Perception power
- Object description power
- Query description power
- Query workload
- Discrimination power

Perception-based feature selection

- E.g., lossy compression
  - Image may contain details that human eye can not recognize
  - Use a color model that reflects user perception
    - Compress those colors that are not easy to perceive
  - Color table:
    1. reduce the number of colors to 256 (1 byte per pixel)
    2. Cluster similar colors into a single bucket and assign a single color to the bucket
    3. the set of buckets is called color table

Good feature..

- A good feature is significant and enables us to differentiate objects from others as much as possible

- A good feature corresponds to users' perception as much as possible
  - Relevance feedback!!!!

What does “significant” mean

- Information theoric sense:
  - An event is more significant if it carries more information

What does “significant” mean

- Information theoric sense:
  - An event is more significant if it carries more information
  - An event which has high occurrence rate caries less information
    - Solar eclipse is more interesting then sunset

High frequency ----- less information
Low frequency ----- high information

Entropy

- Total (or expected) information content
  - Uncertainty!!

$$H(X) \equiv \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)}$$

Information content of the event

Entropy (example)

- Total information content (uncertainty)

$$H(X) \equiv \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)}$$

$p(A)$	$p(B)$	$-log_2 p(A)$	$-log_2 p(B)$	$\frac{p(A)}{log_2 p(A)}$	$\frac{p(B)}{log_2 p(B)}$	$H(E)$
0.05	0.95	1.3	0.022	0.216	0.07	0.29
0.5	0.5	1	1	0.5	0.5	1
0.95	0.05	0.022	1.3	0.07	0.216	0.29

Entropy (example)

- Total information content (uncertainty)

$$H(X) \equiv \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)}$$

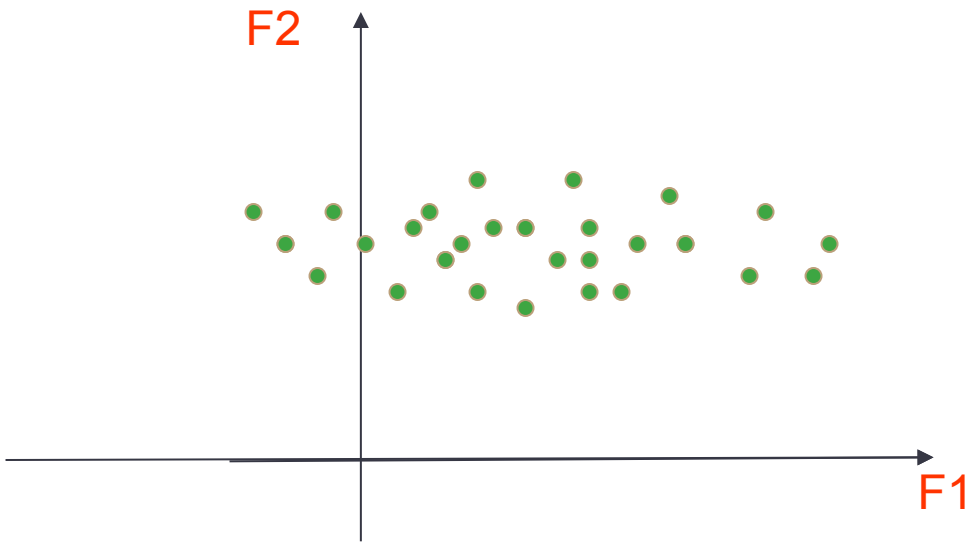
P(a) = 0.5, P(b) = 0.5 → **H = 1** — more uncertain, more information

P(a) = 0.95, P(b) = 0.05 → **H = 0.29** — less uncertain, less information

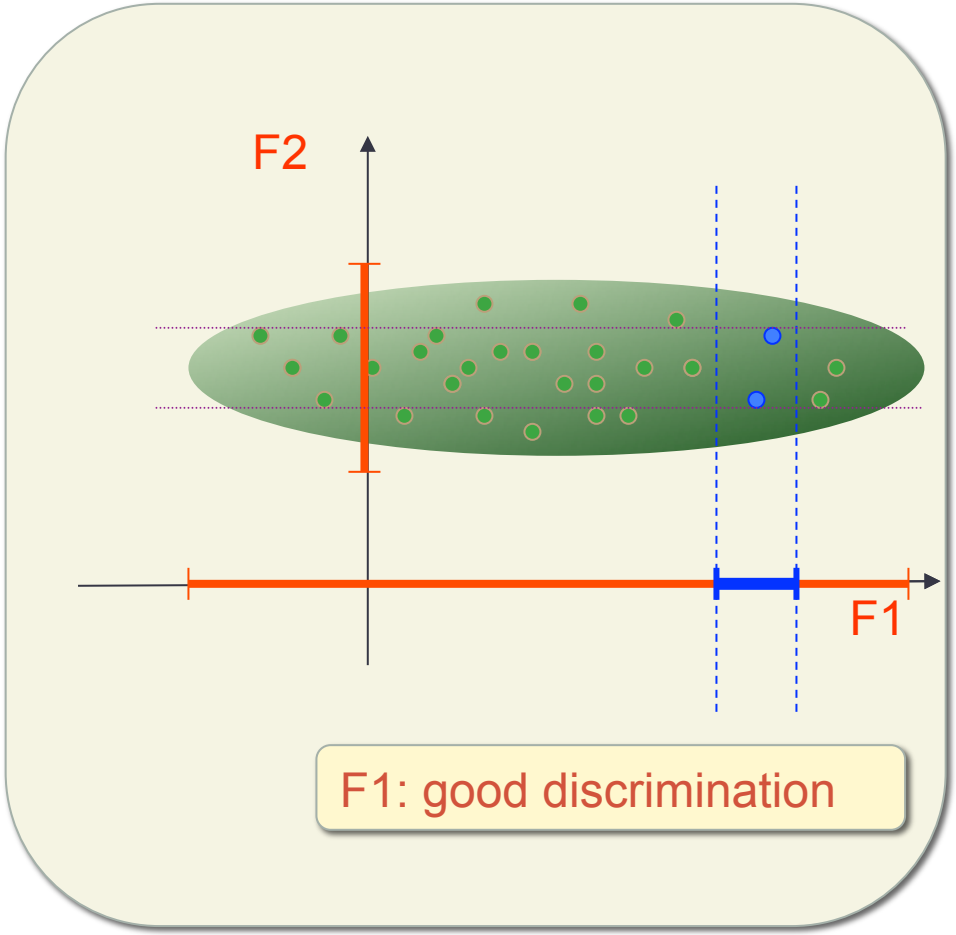
Conditional Entropy

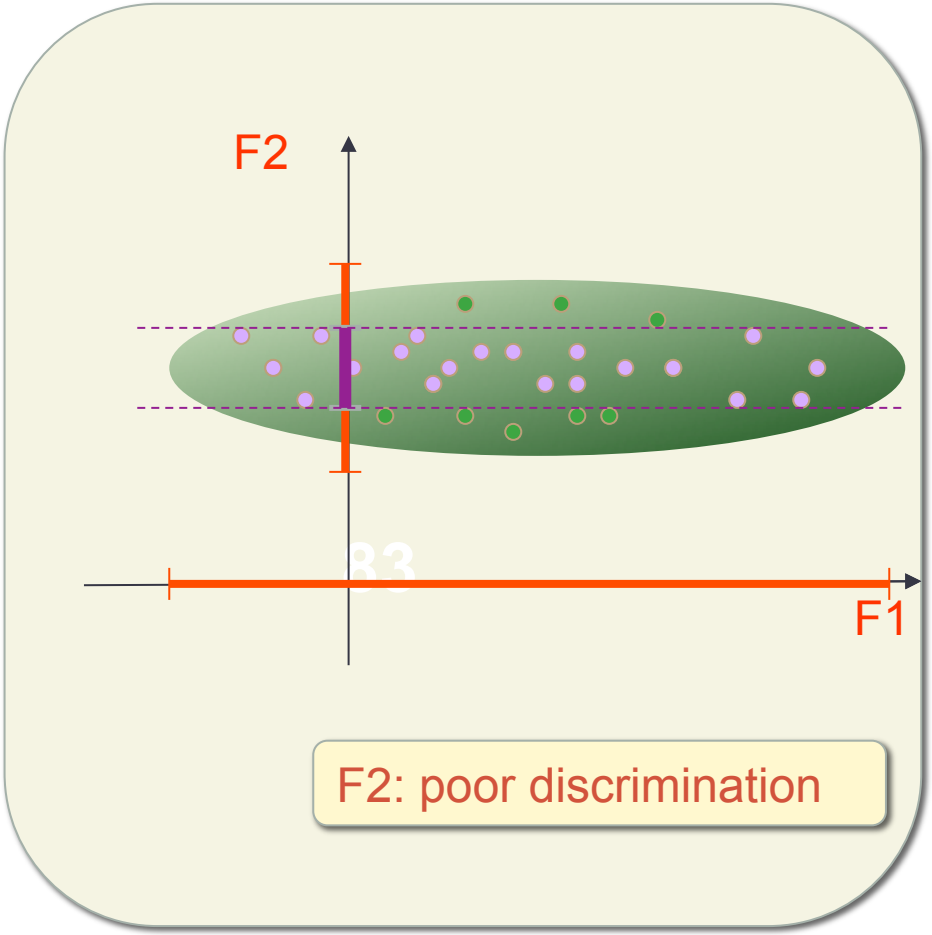
- X takes values from $\{ x_1, \ldots, x_n \}$
  - $\Sigma_i\, p(X=x_i) = 1$
- Y takes values from $\{ y_1, \ldots, y_m \}$
  - $\Sigma_i\, p(Y=y_i) = 1$
- Conditional entropy of X given $Y=y_j$ is:
  - $H(X \mid Y=y_j) = \Sigma_i\, p(X=x_i \mid Y=y_j) * \lg 1/\, p(X=x_i \mid Y=y_j)$
- Conditional entropy of X given Y is:
  - $H(X \mid Y) = \Sigma_j\, p(Y=y_j)\, \Sigma_i\, p(X=x_i \mid Y=y_j) * \lg 1/p(X=x_i \mid Y=y_j)$

Feature selection…



Which feature is better?





Text (as a collection of keywords)…..

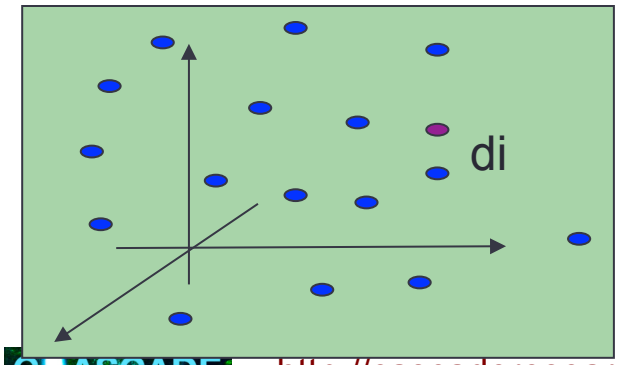
- Each document is represented as a multi-set of keywords
  - Content words (terms)
  - Non-content words (stop words)

Vector representation

- Given a set of keywords, each document is represented as a vector:

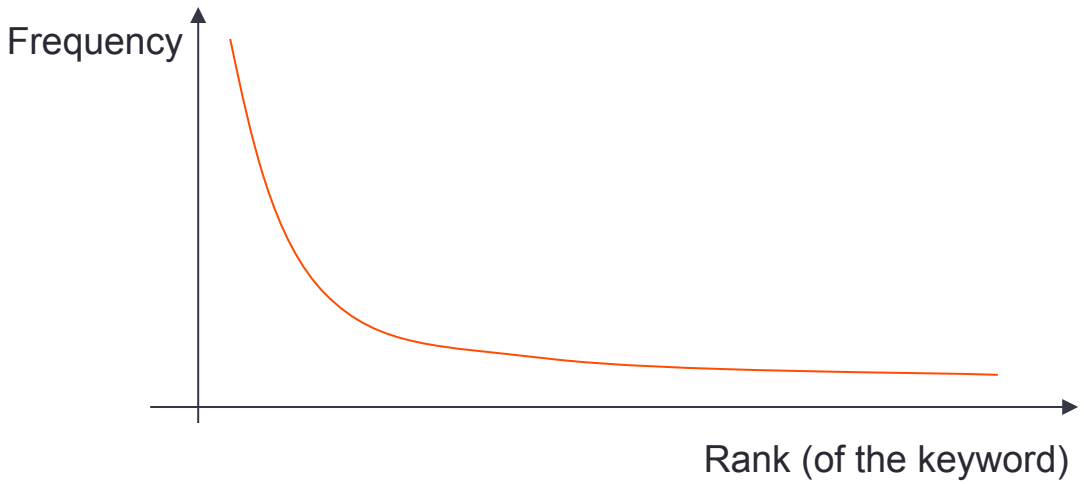
  di = <wi1,wi2,wi3,……,win>

  where
  - wij = 0, if the keyword does not occur in di
  - wij > 0, if the keyword occurs in di



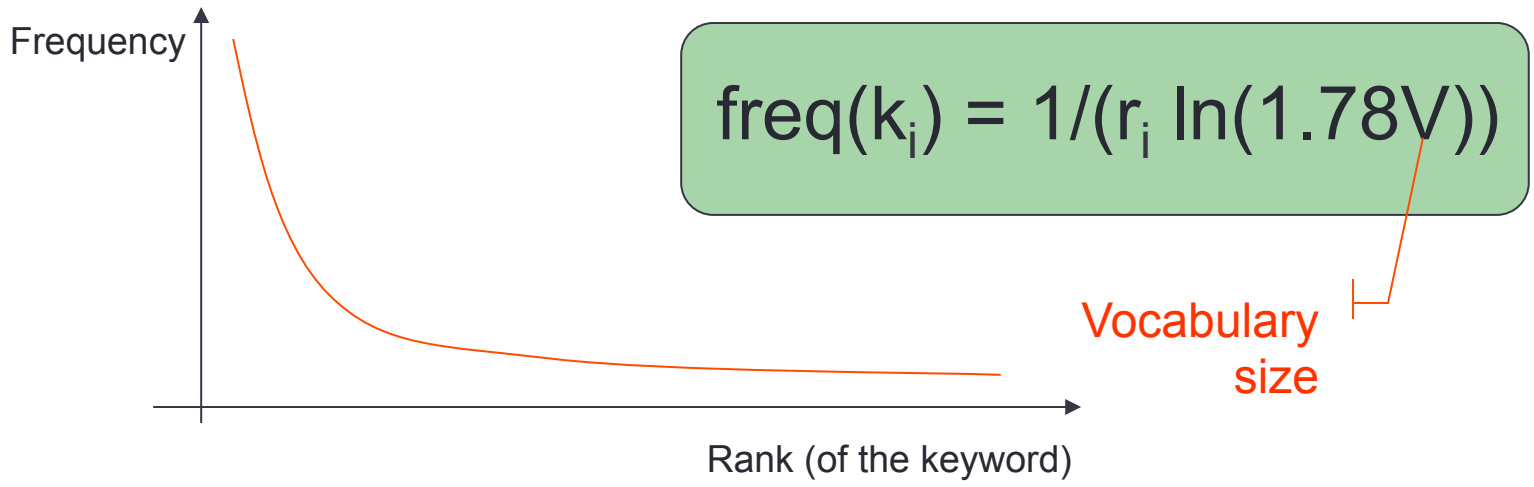
Zipfian Distribution

- The frequency of the k^{th} most frequent word in a collection is $(1/k)^{\Theta}$ times the most frequent word.



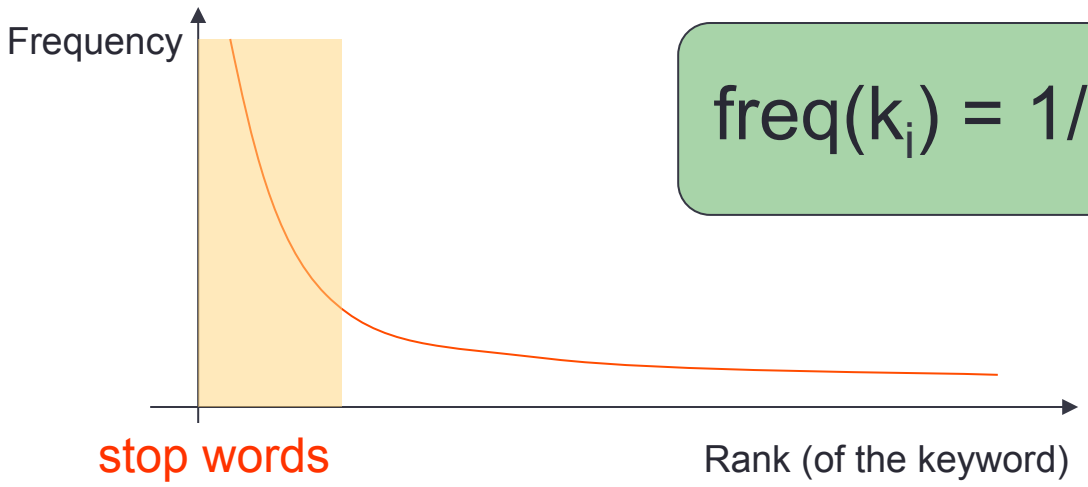
Zipfian Distribution

- The frequency of the k^{th} most frequent word in a collection is $(1/k)^{\Theta}$ times the most frequent word.



Zipfian Distribution

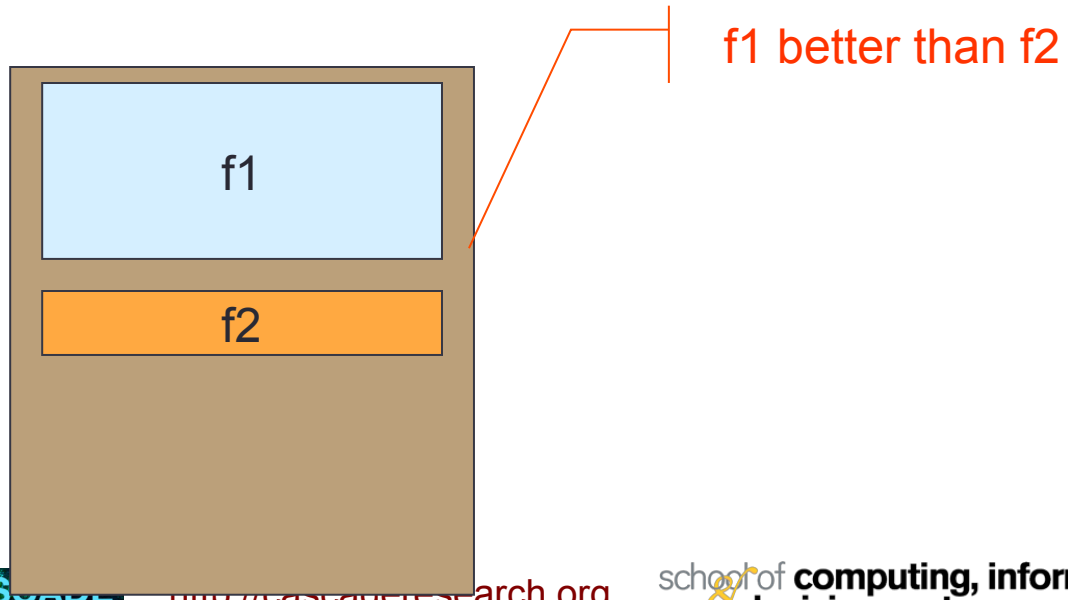
- The frequency of the k^{th} most frequent word in a collection is $(1/k)^{\Theta}$ times the most frequent word.



$$freq(k_i) = 1/(r_i \ln(1.78V))$$

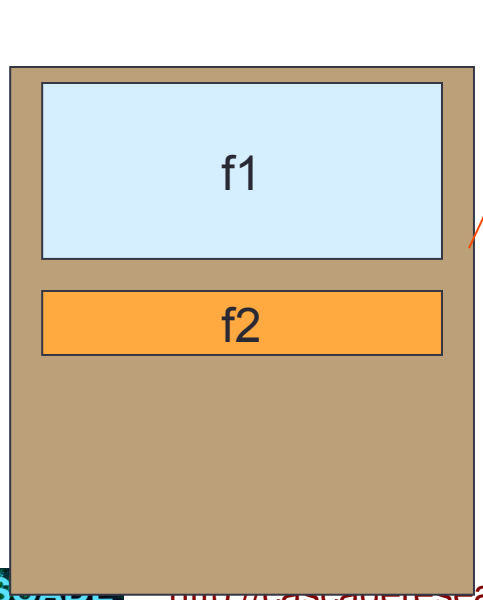
What are the weights????

- They need to capture how
  - good the term (feature) is in describing the content of the object



What are the weights????

- They need to capture how
  - good the term (feature) is in describing the content of the object

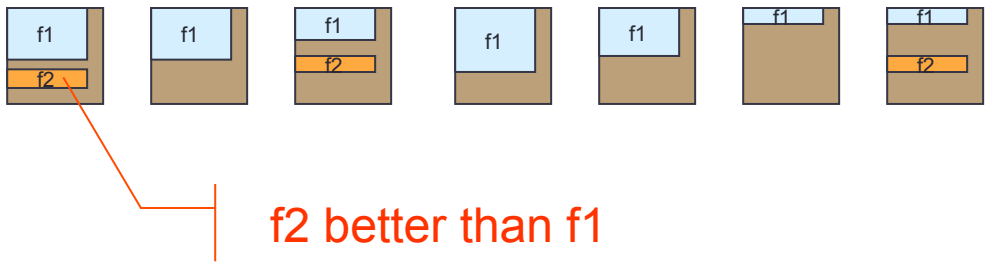


f1 better than f2

$$tf = \frac{n}{K}$$

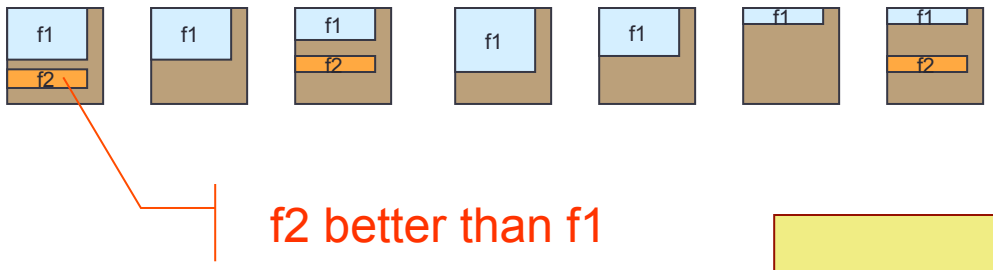
What are the weights????

- They need to capture how
  - differentiating the term (feature) is..



What are the weights????

- They need to capture how
  - differentiating the term (feature) is..



f2 better than f1

$$idf = \log(\frac{N}{m})$$

What are the weights????

- They need to capture how
  - good the term (feature) is in describing the content of the object
  - differentiating the term (feature) is..

$$tfidf = \frac{n}{K}\log(\frac{N}{m})$$

What are the weights????

- They need to capture how
  - good the term (feature) is in describing the content of the object
  - differentiating the term (feature) is..

$$norm\_tfidf = \frac{n}{K}\frac{\log(\frac{N}{m})}{\max idf}$$

Idf of the keyword

Experiment results suggest that

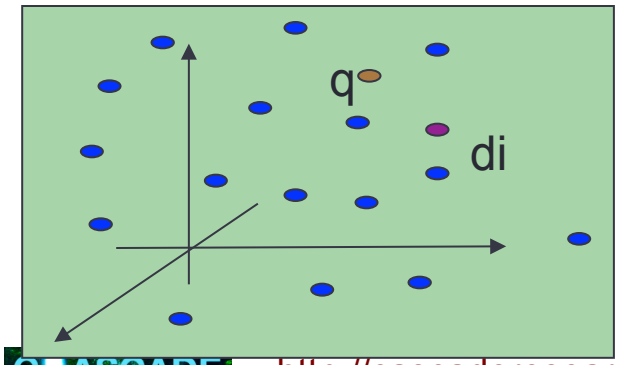
- Poor terms have high document frequency
- Good terms have low document frequency
  - Problem: may not be queried often enough to be useful
- Best terms have medium document frequency

How about query terms??

- Given a set of keywords, each query is also represented as a vector:

  q = <wq1,wq2,wq3,……,wqn>

  where
  - wqj = 0, if the keyword does not occur in dq
  - wqj > 0, if the keyword occurs in dq



How about query terms??

- They need to capture how
  - good the term (feature) is in describing the query
  - differentiating the term (feature) is..
  - Salton&Buckley suggests..

Maximum term frequency in the query

Frequency of the term in the query

$$tfidf = \left(0.5 + 0.5 \frac{\frac{n}{K}}{\max freq} \right) \log\left(\frac{N}{m}\right)$$

Relevance based feature significance

- *D*: all documents
- *R*: the set of objects known to be relevant to the user
- *I*: the set of objects known to be irrelevant to the user

$$log\left(\frac{p(f_k|R)(1-p(f_k|I))}{p(f_k|I)(1-p(f_k|R))}\right) \times \left(p(f_k|rel) - p(f_k|\overline{rel})\right)$$

assuming I = D – R:

$$log\left(\frac{r_k(|D-R| - (d_k - r_k))}{(d_k - r_k)(|R| - r_k)}\right) \times \left|\frac{r_k}{|R|} - \frac{d_k - r_k}{|D-R|}\right|$$

rk is the number of objects in **R** with **f**k
dk is the number of objects in **D** with **f**k

Entropy

- Total (or expected) information content
  - Uncertainty!!

$$H(X) \equiv \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)}$$

Information content of the event

Entropy (example)

- Total information content (uncertainty)

$$H(X) \equiv \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)}$$

$p(A)$	$p(B)$	$-log_2 p(A)$	$-log_2 p(B)$	$\frac{p(A)}{log_2 p(A)}$	$\frac{p(B)}{log_2 p(B)}$	$H(E)$
0.05	0.95	1.3	0.022	0.216	0.07	0.29
0.5	0.5	1	1	0.5	0.5	1
0.95	0.05	0.022	1.3	0.07	0.216	0.29

Entropy (example)

- Total information content (uncertainty)

$$H(X) \equiv \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)}$$

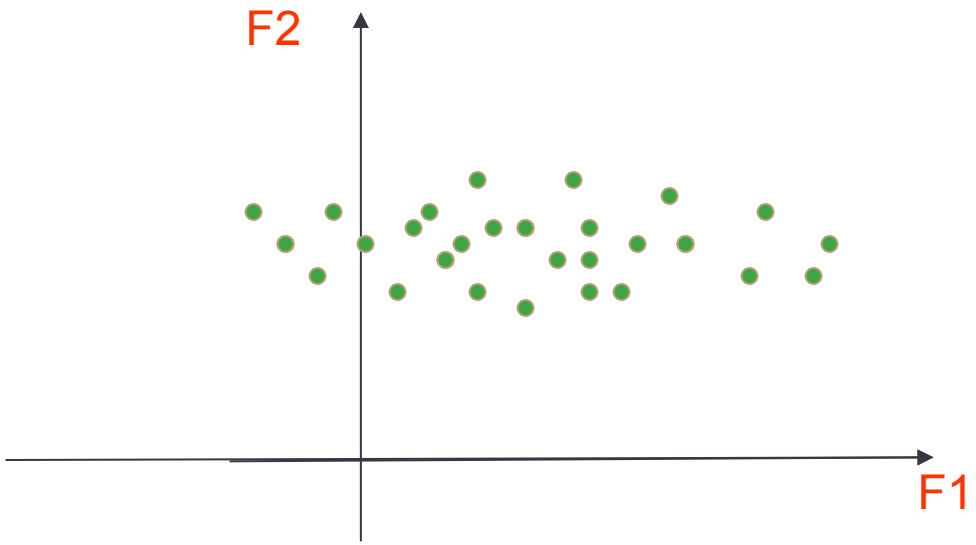
P(a) = 0.5, P(b) = 0.5 → **H = 1** — more uncertain, more information

P(a) = 0.95, P(b) = 0.05 → **H = 0.29** — less uncertain, less information

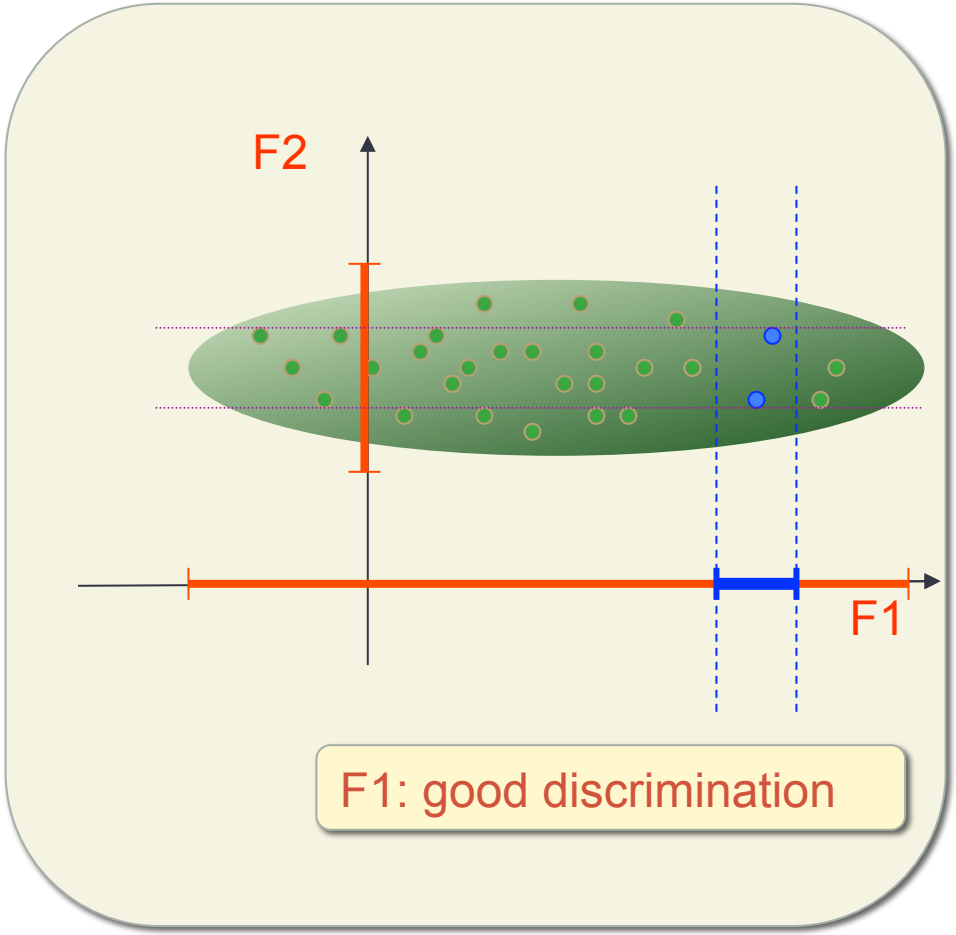
Conditional Entropy

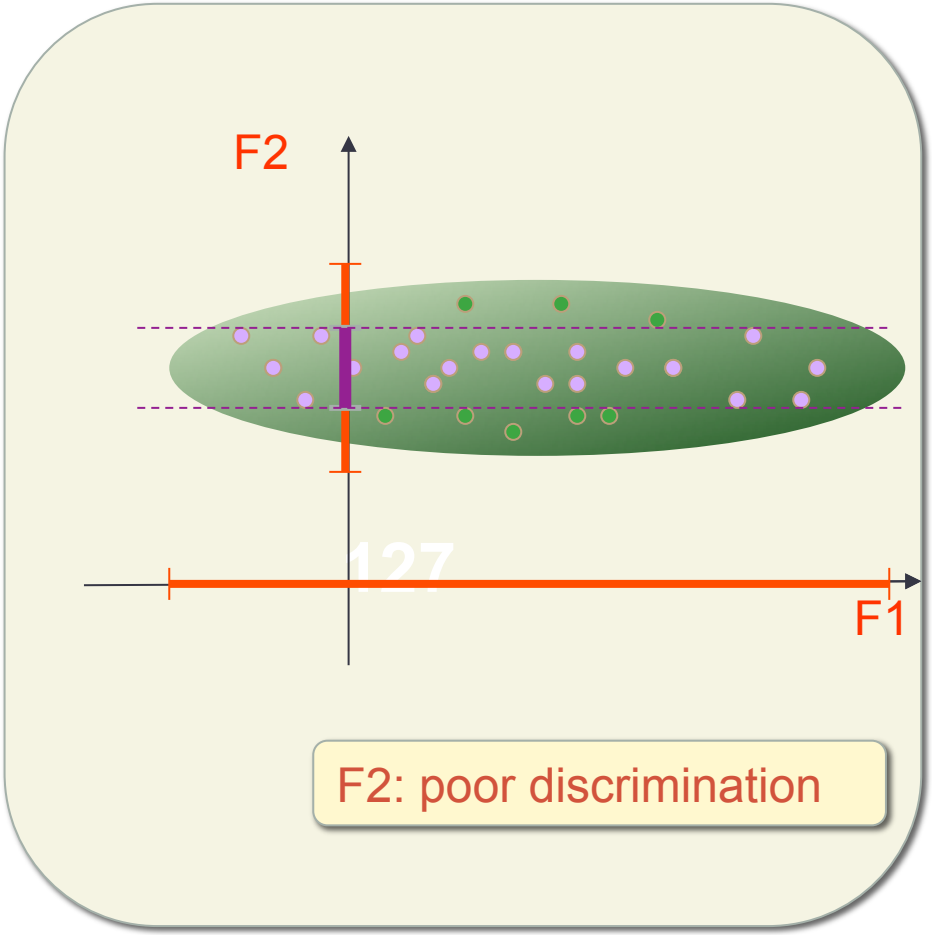
- X takes values from $\{ x_1, \ldots, x_n \}$
  - $\Sigma_i\, p(X=x_i) = 1$
- Y takes values from $\{ y_1, \ldots, y_m \}$
  - $\Sigma_i\, p(Y=y_i) = 1$
- Conditional entropy of X given $Y=y_j$ is:
  - $H(X \mid Y=y_j) = \Sigma_i\, p(X=x_i \mid Y=y_j) * \lg 1/\, p(X=x_i \mid Y=y_j)$
- Conditional entropy of X given Y is:
  - $H(X \mid Y) = \Sigma_j\, p(Y=y_j)\, \Sigma_i\, p(X=x_i \mid Y=y_j) * \lg 1/p(X=x_i \mid Y=y_j)$

Feature selection…

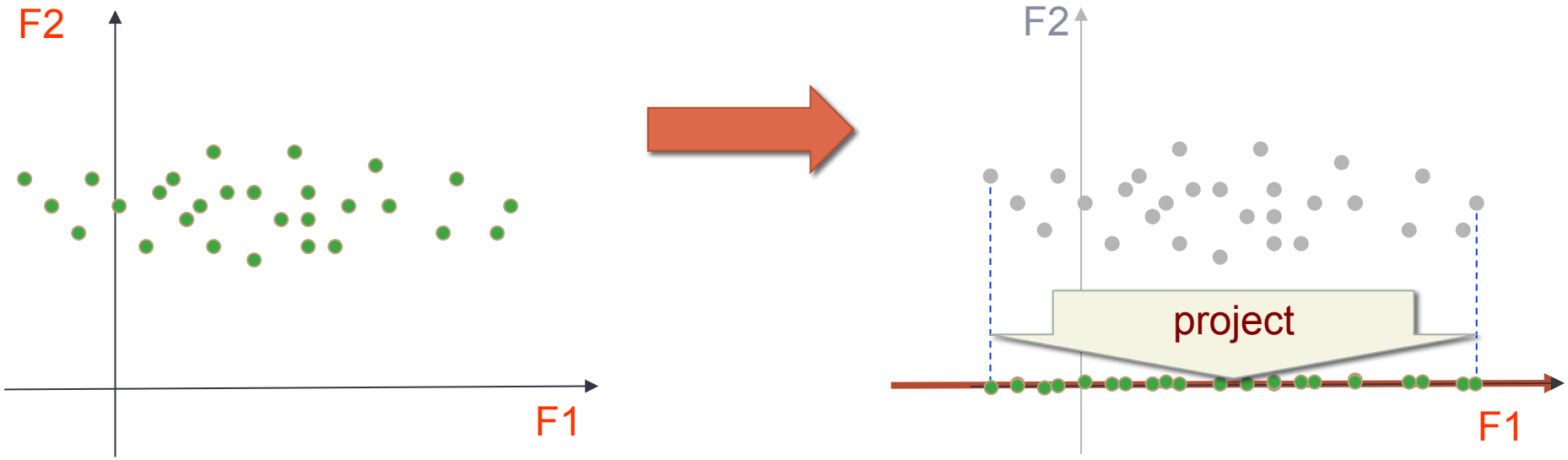


Which feature is better?

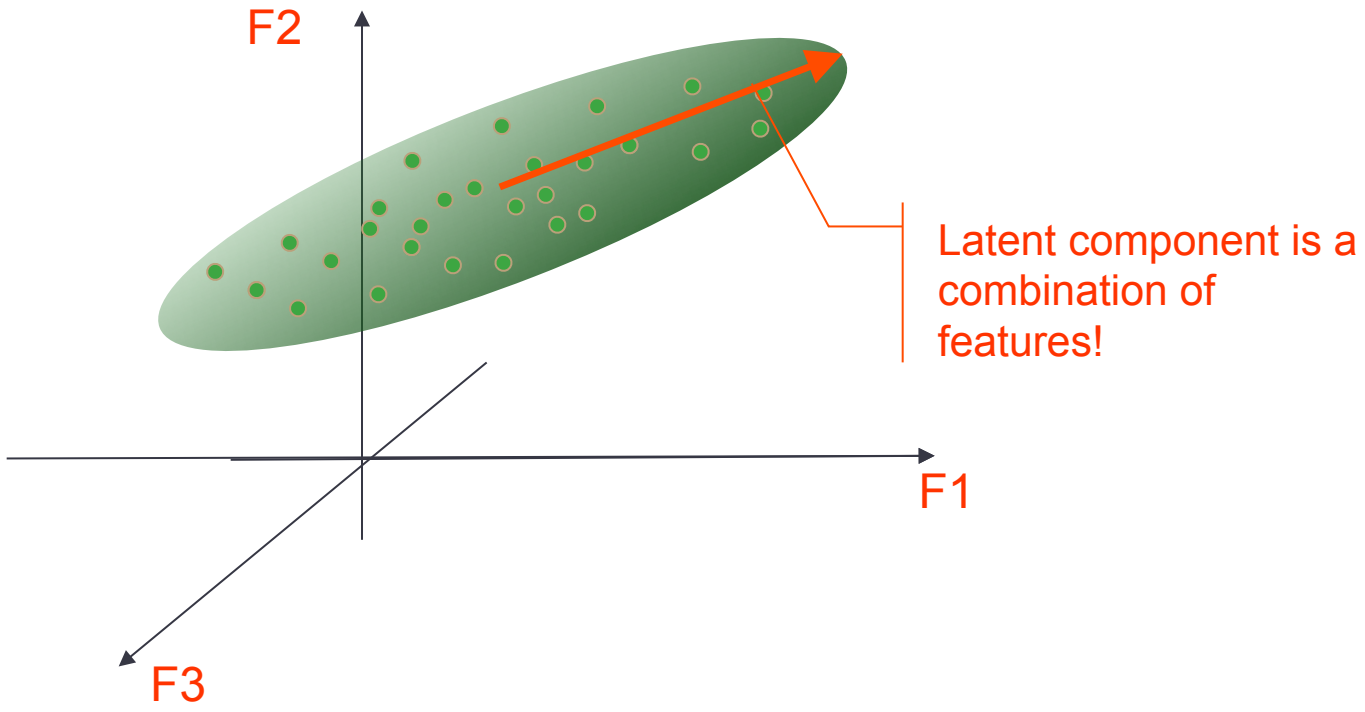




Feature selection…



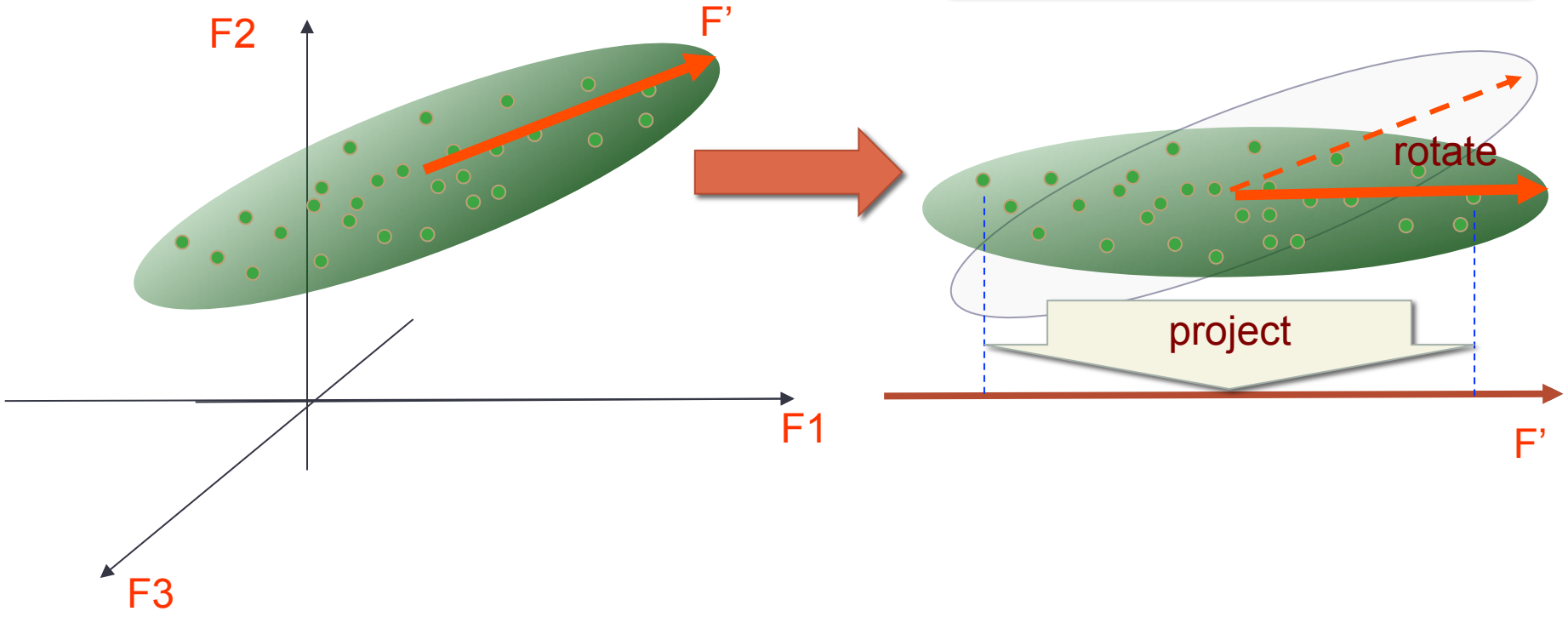
Latent component analysis



Latent feature...

Information is distributed across features

Information is concentrated on one feature



Through an appropriate transformation, the latent component can be used as a “discriminating” feature of the data

Search for latent features

- Given a database,
  - $D = \{o_1, \ldots, o_n\}$ of objects,
  - originally described in terms of a feature set, $F = \{f_1, \ldots, f_m\}$,

  identify
  - a set of "latent" features, $Z = \{z_1, \ldots, z_k\}$,

  such that
  - $k << m$ and
  - the transformed database $D' = \{o'_1, \ldots, o'_n\}$ (described in the terms of latent features Z instead of F) is not very different from the original database D'

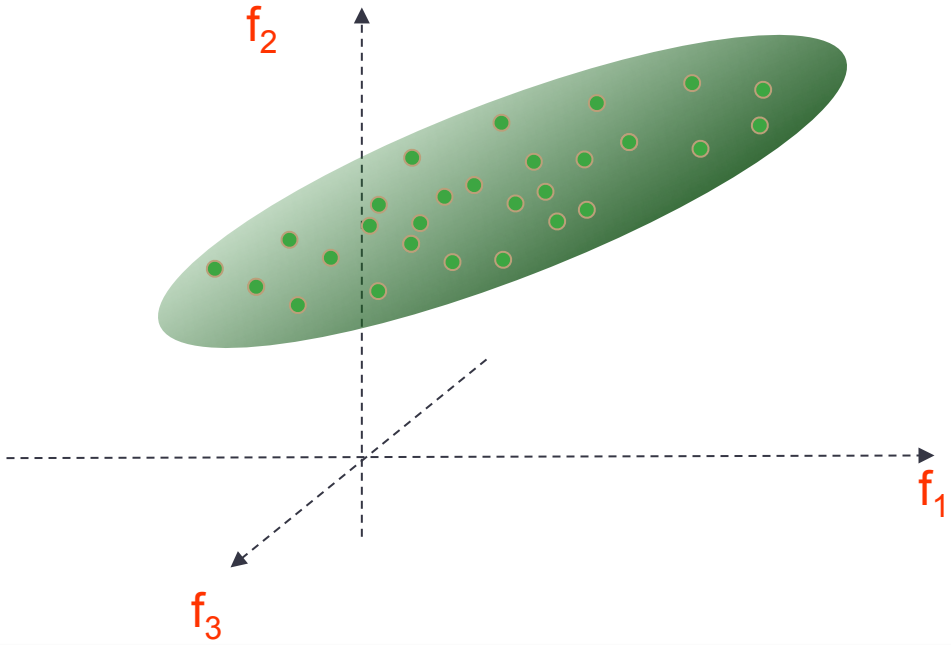
    i.e.,

$$D \sim D'$$

Search for latent features

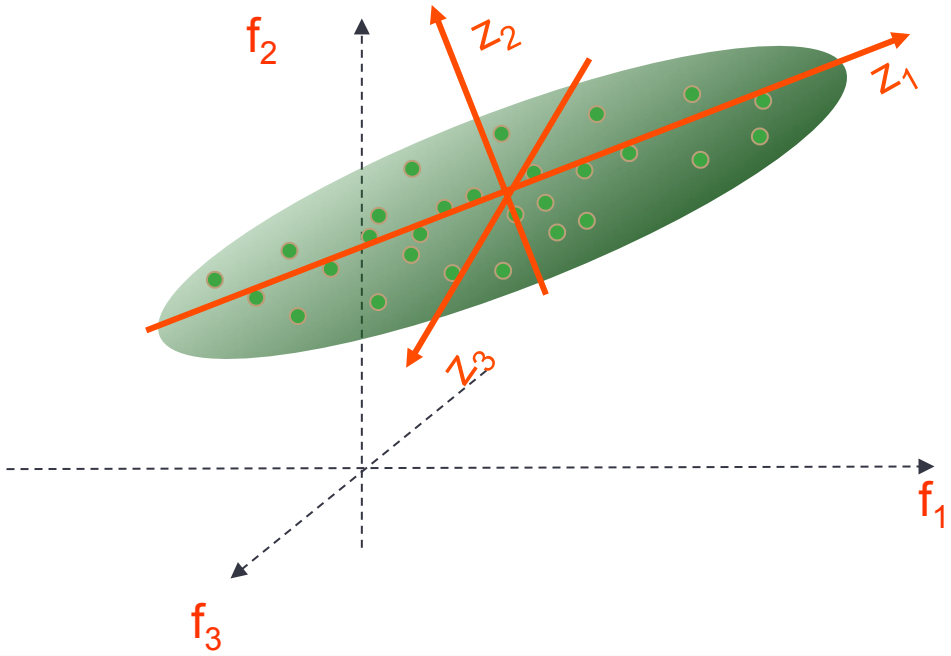
- Alternative #1: Feature selection
  - $Z \subseteq F$
- Alternative #2: Representative object selection
  - $Z \subseteq D$
- Alternative #3: Ortho-normal basis transformation
  - $\forall j \;\; z'_j = \Theta_j(f_1, \ldots, f_m)$, where
    - $\forall j \; () \;\; \Theta_j \; ()$ is a "normal" transformation
    - $\forall i,j \; () \;\; \Theta_i \; ()$ and $\Theta_j()$ are mutually orthogonal
- Alternative #4: Arbitrary transformation

Ortho-normal basis transformations



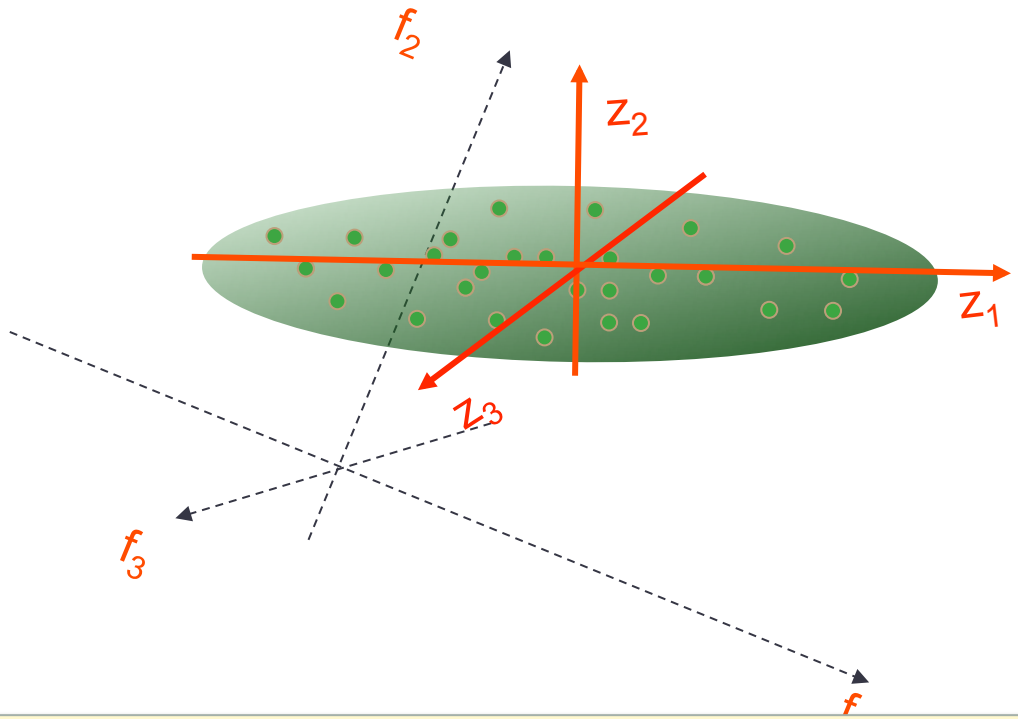
Let “D” denote distances among the objects

Ortho-normal basis transformations



Let “D” denote distances among the objects

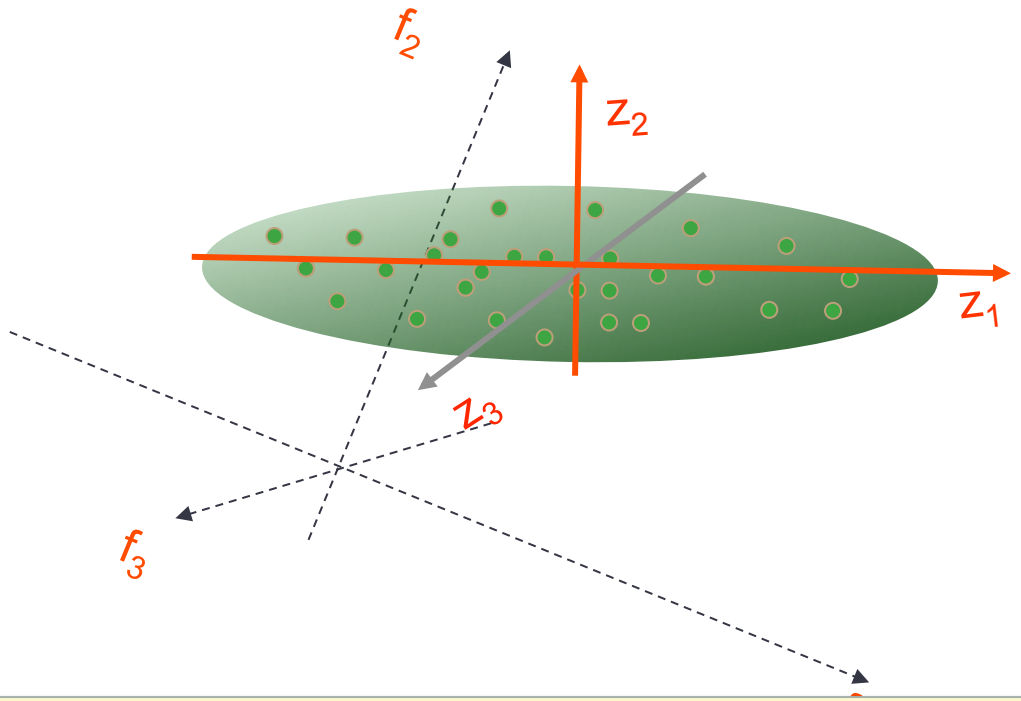
Ortho-normal basis transformations



Distances and angles among the objects are preserved

D'=D

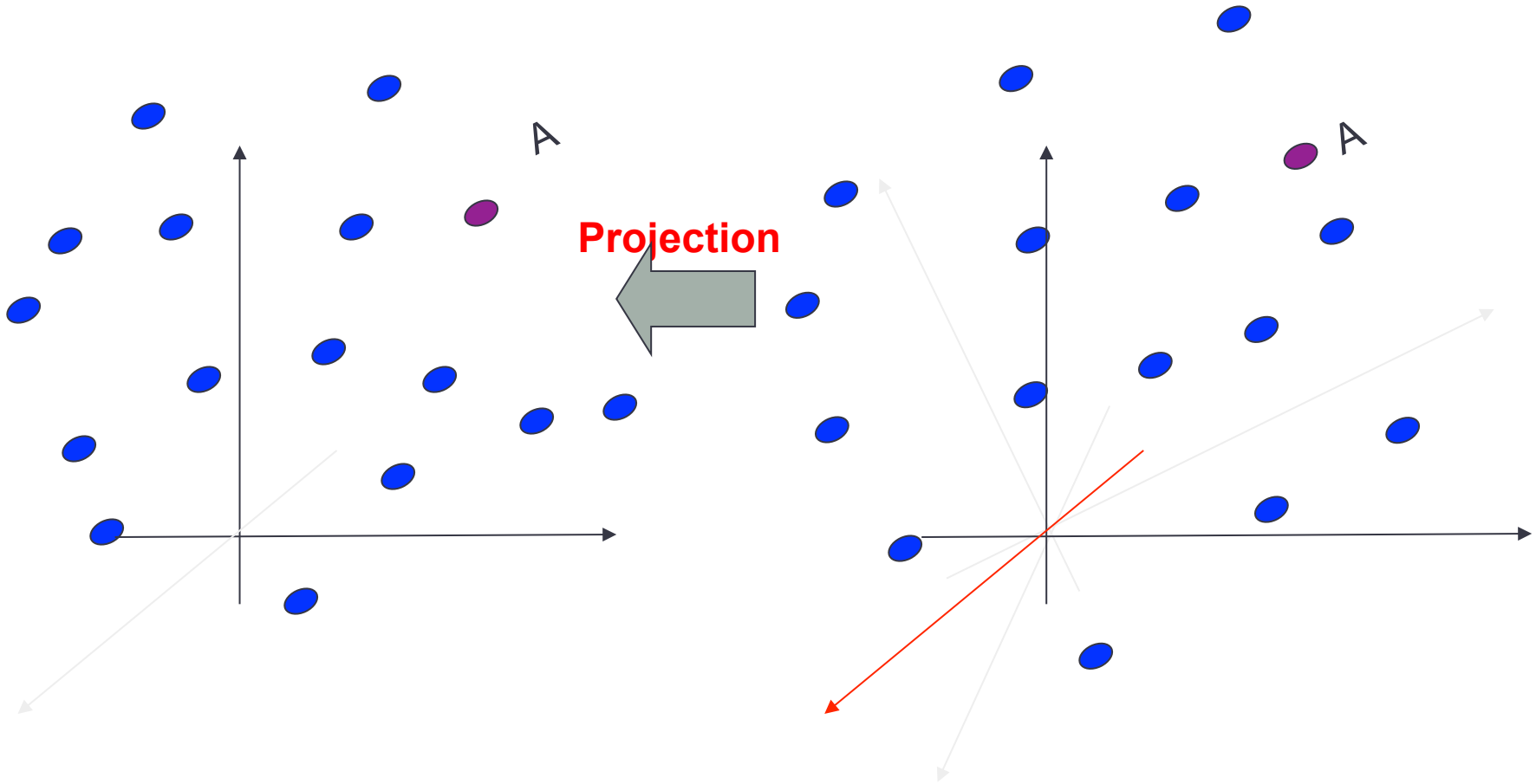
Ortho-normal basis transformations



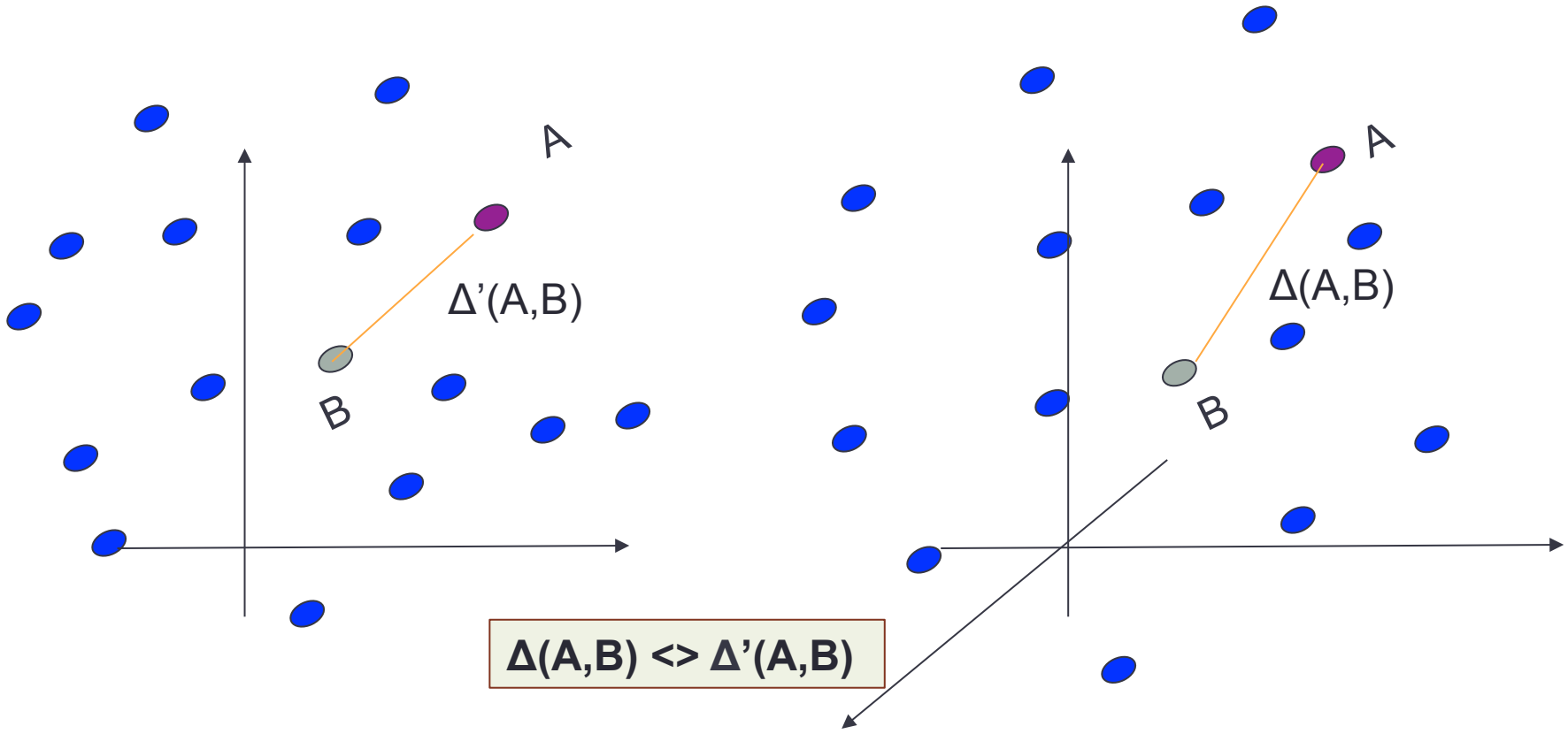
Distances and angles among the objects are "approximately" preserved

D'~D

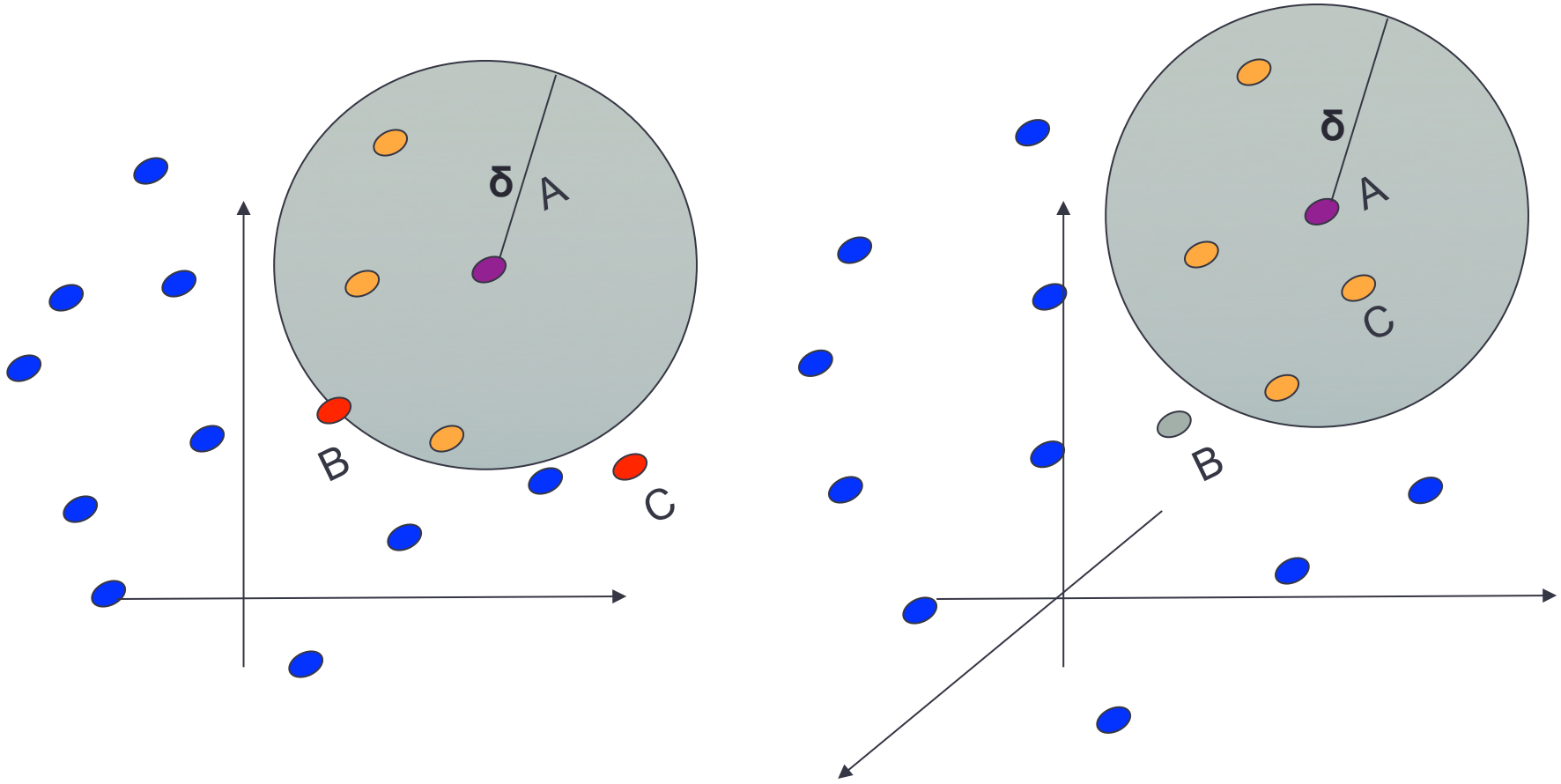
Transform + Projection (Compression or Feature selection)



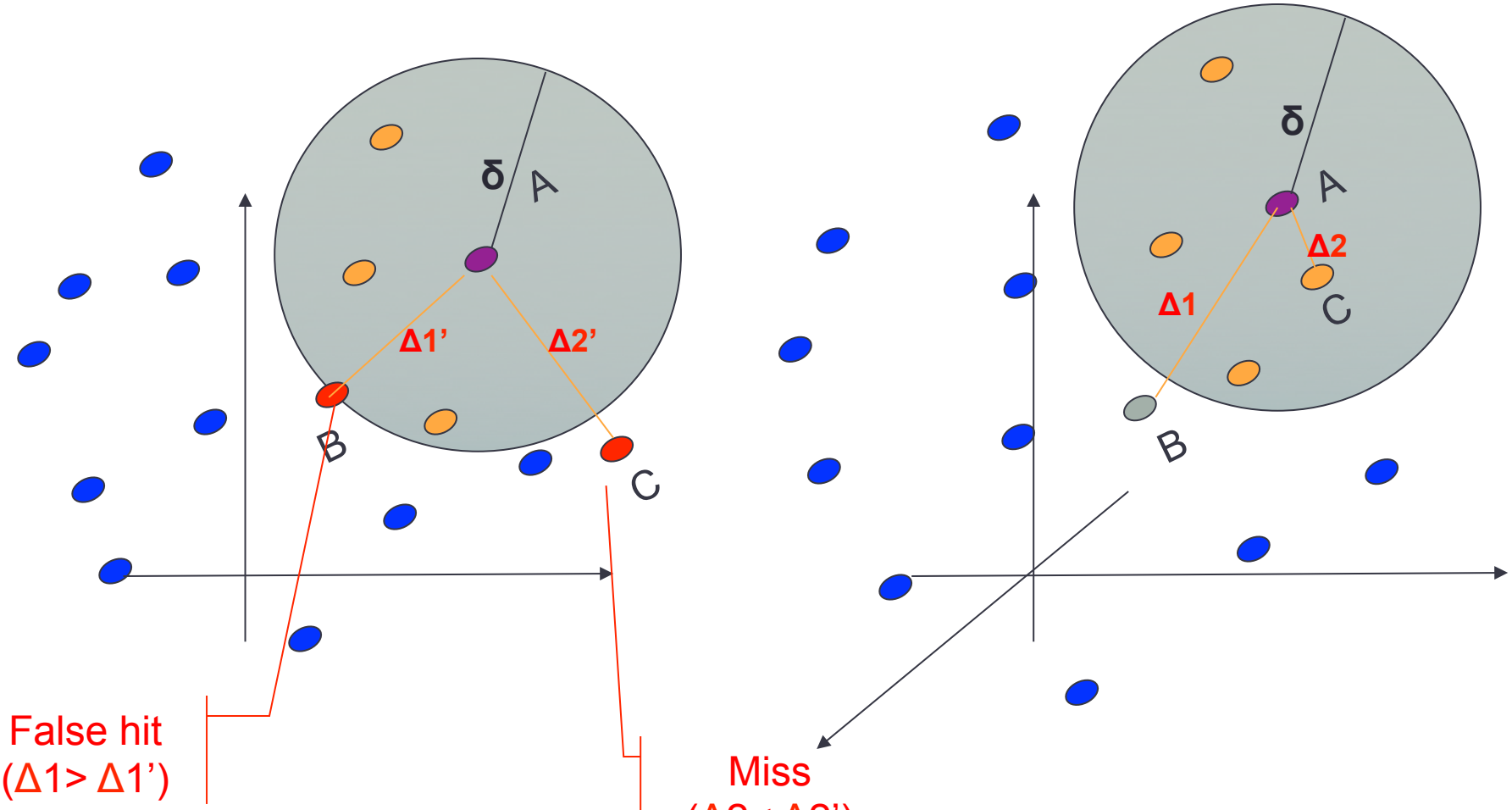
What happens to distances???



What happens to distances???

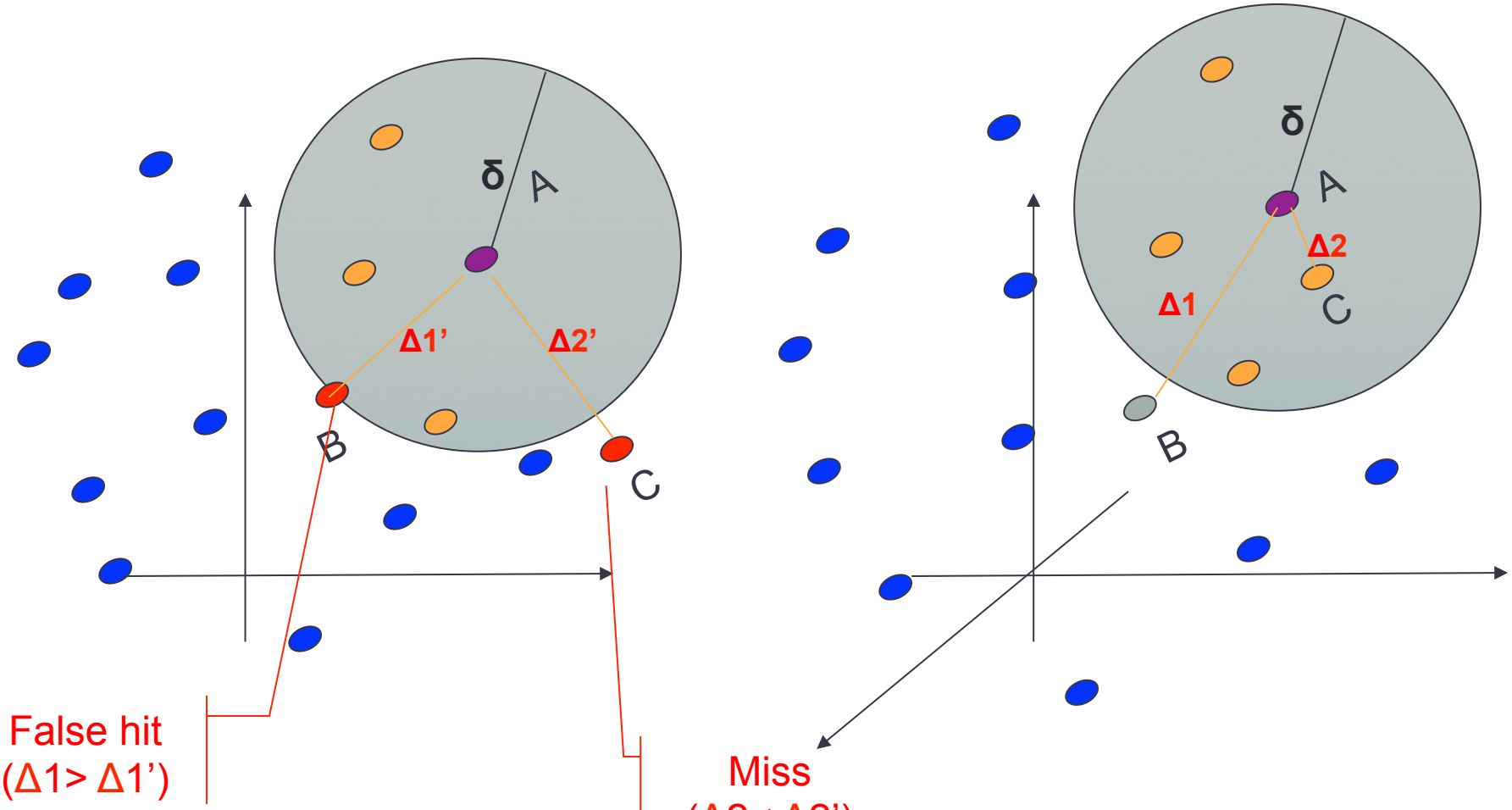


What happens to distances???



Misses are not desirable!
Can not be eliminated with post-processing

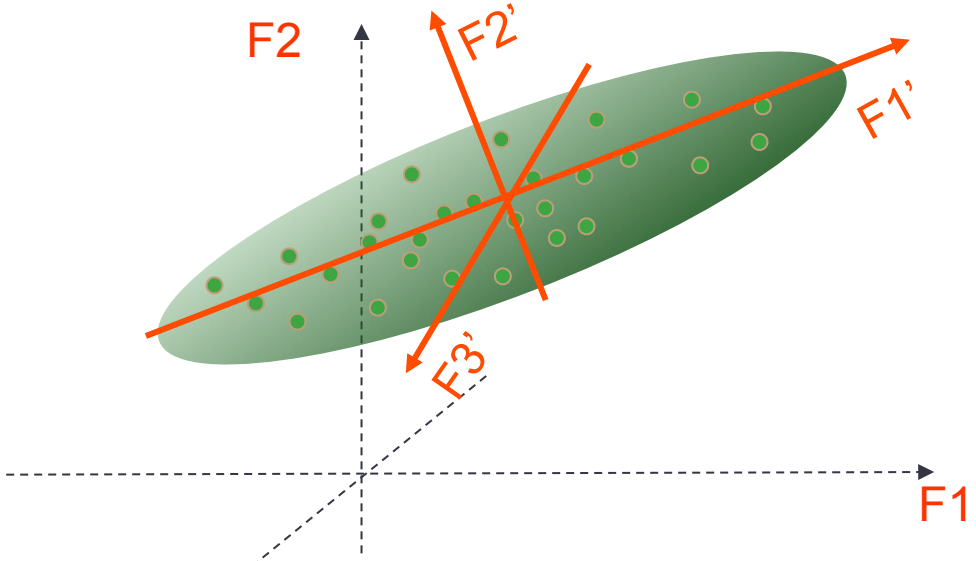
What happens to distances???



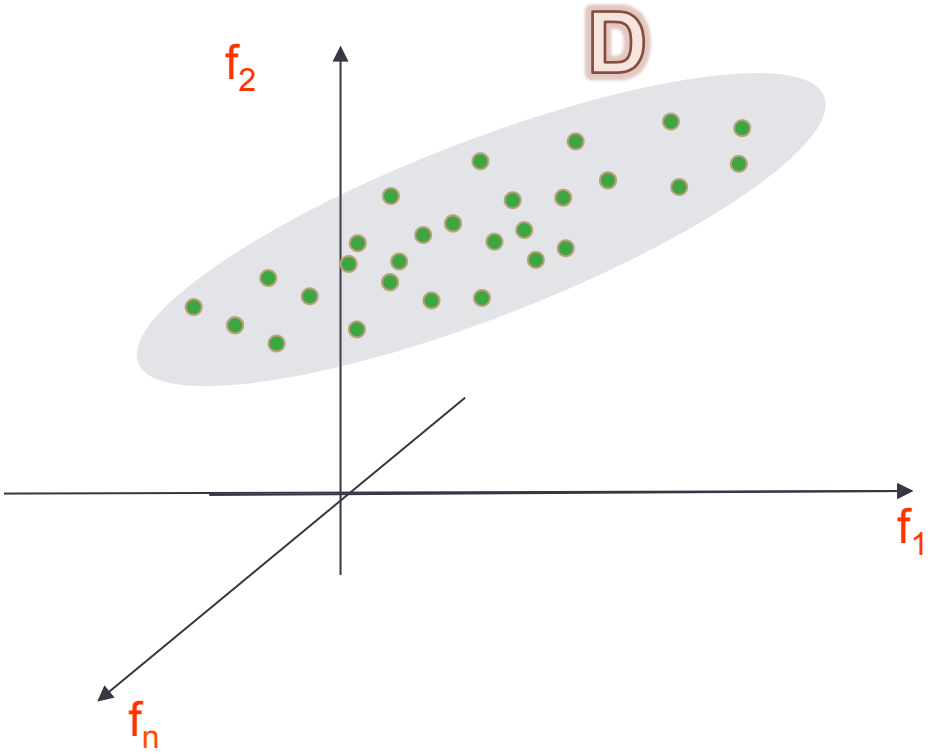
So we need data transforms where

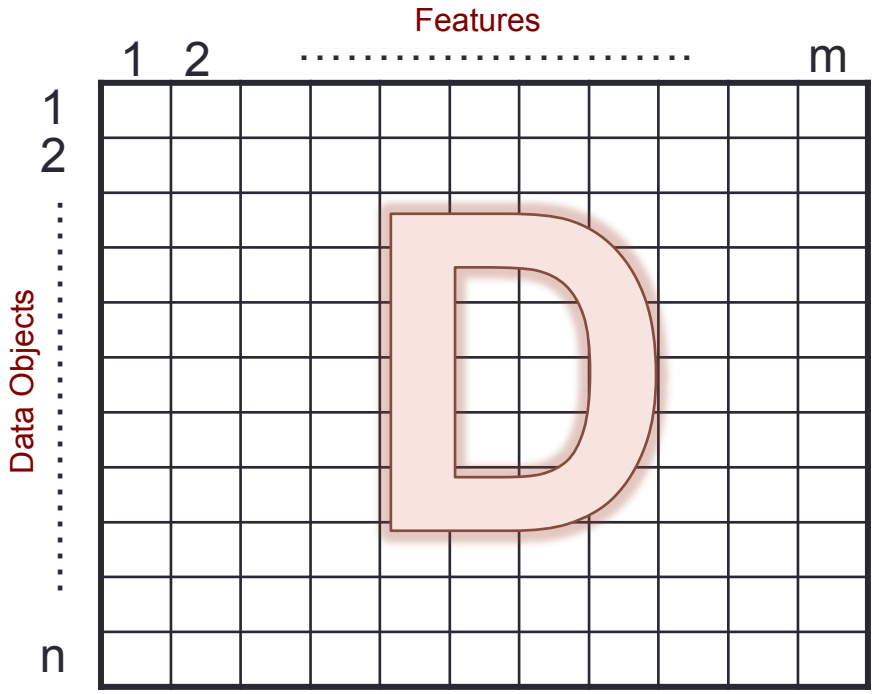
....

- distances and angles are preserved (e.g. linear, orthonormal transforms).
- the discriminatory power is associated with a few dimensions of the new space (the dimensions are as little correlated as possible)

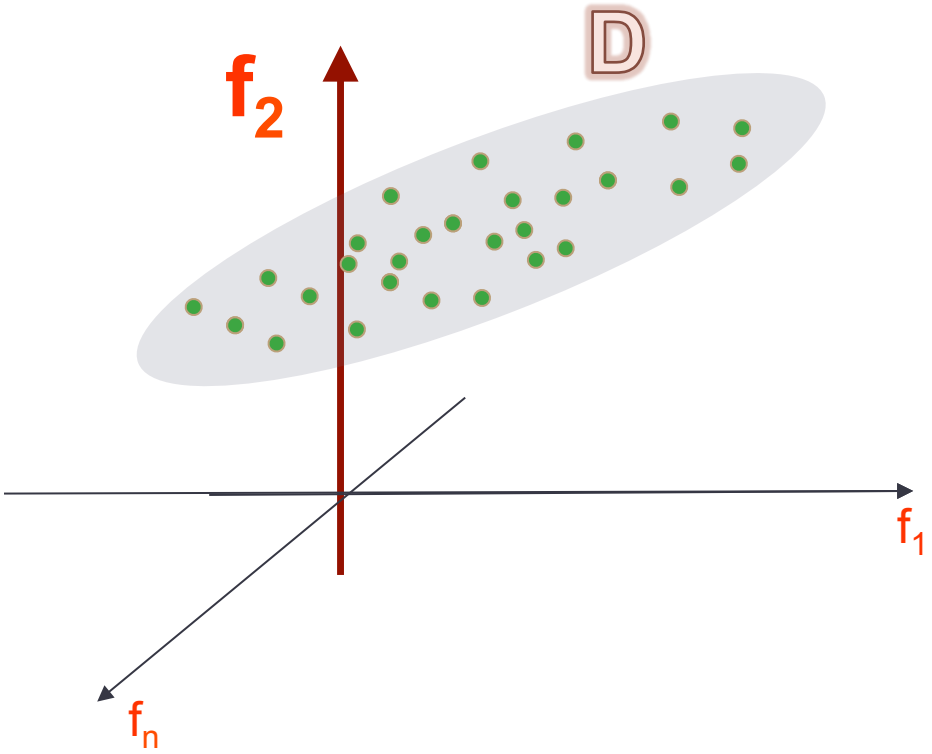


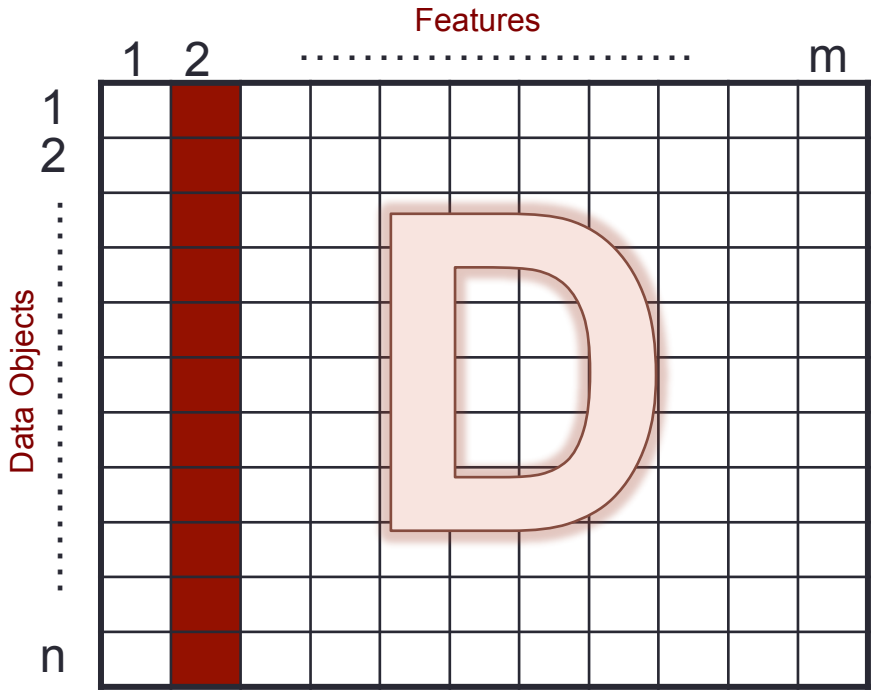
Singular valued decomposition (SVD)



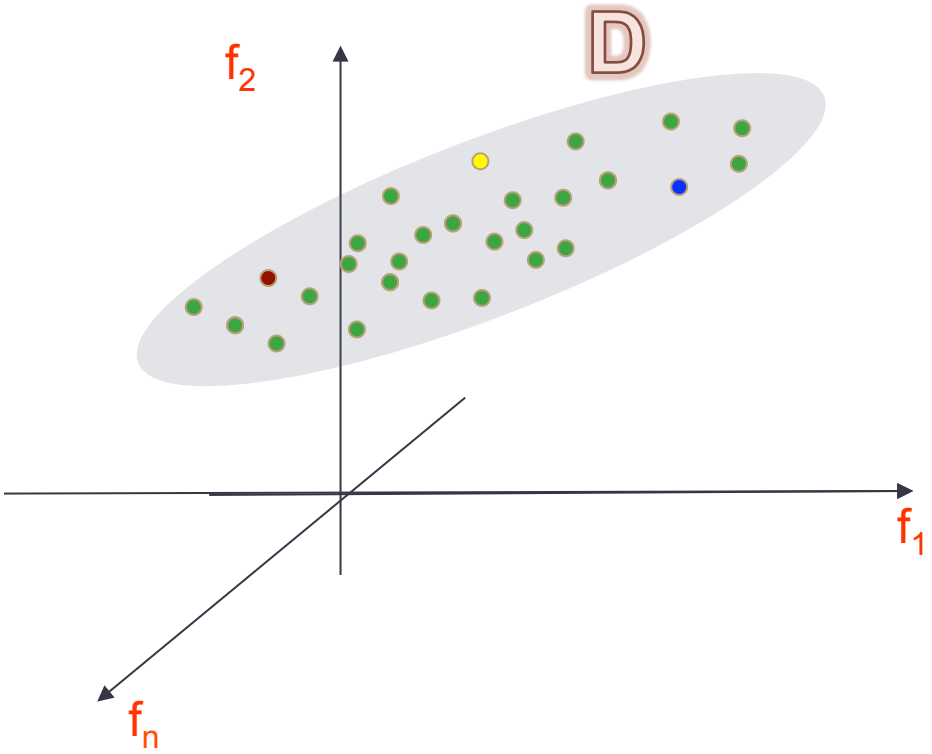


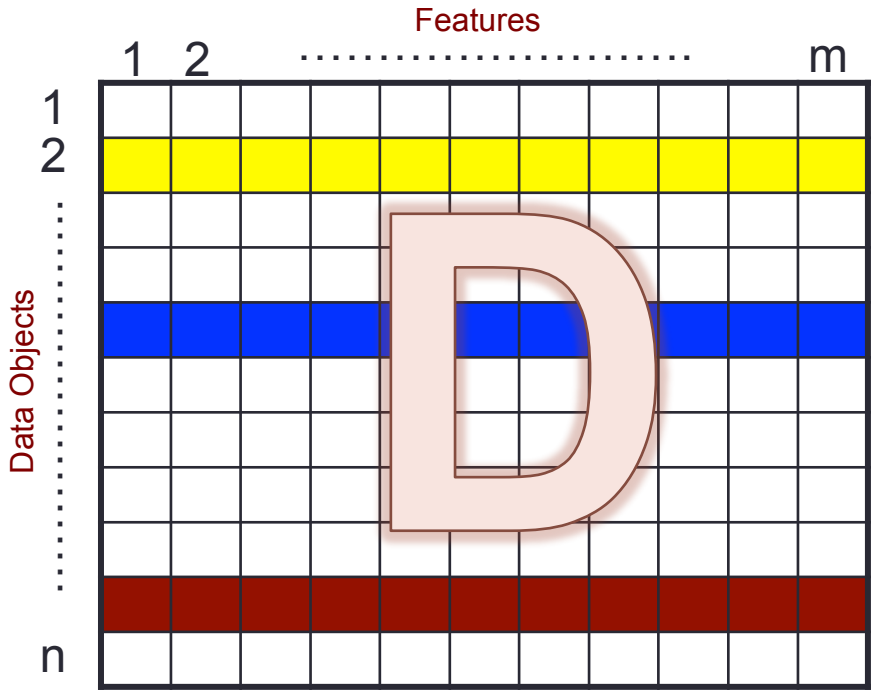
Singular valued decomposition (SVD)



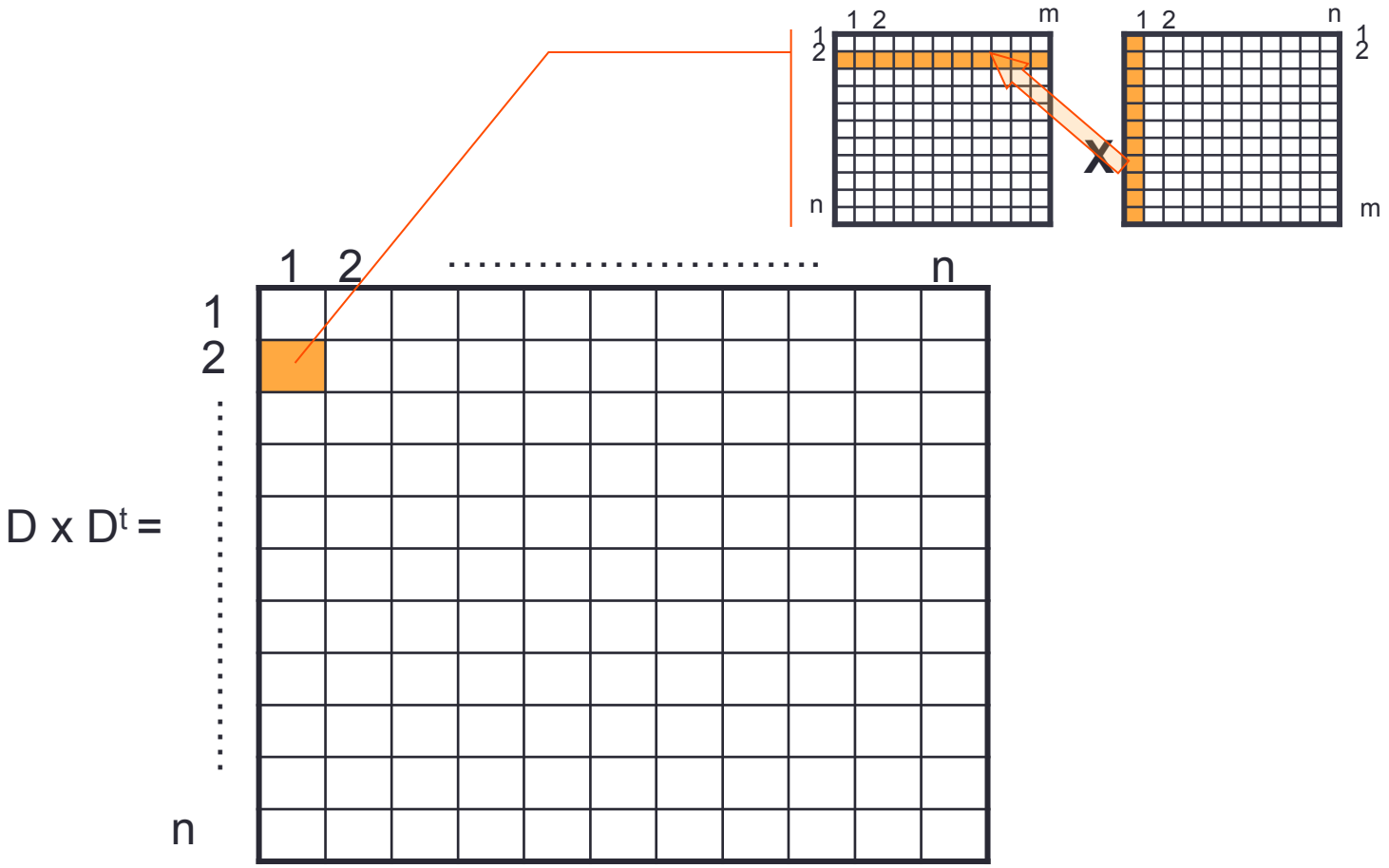


Singular valued decomposition (SVD)

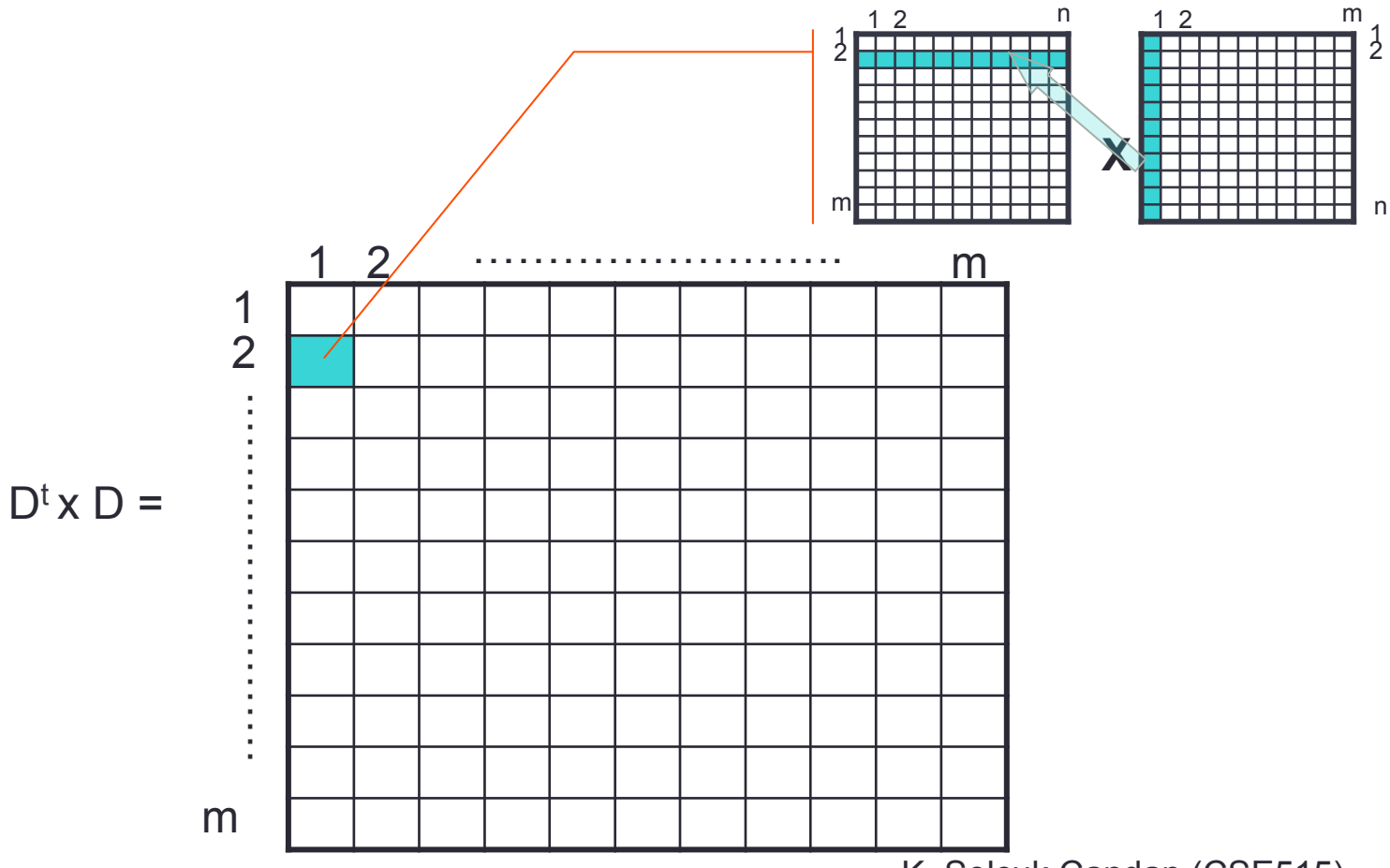




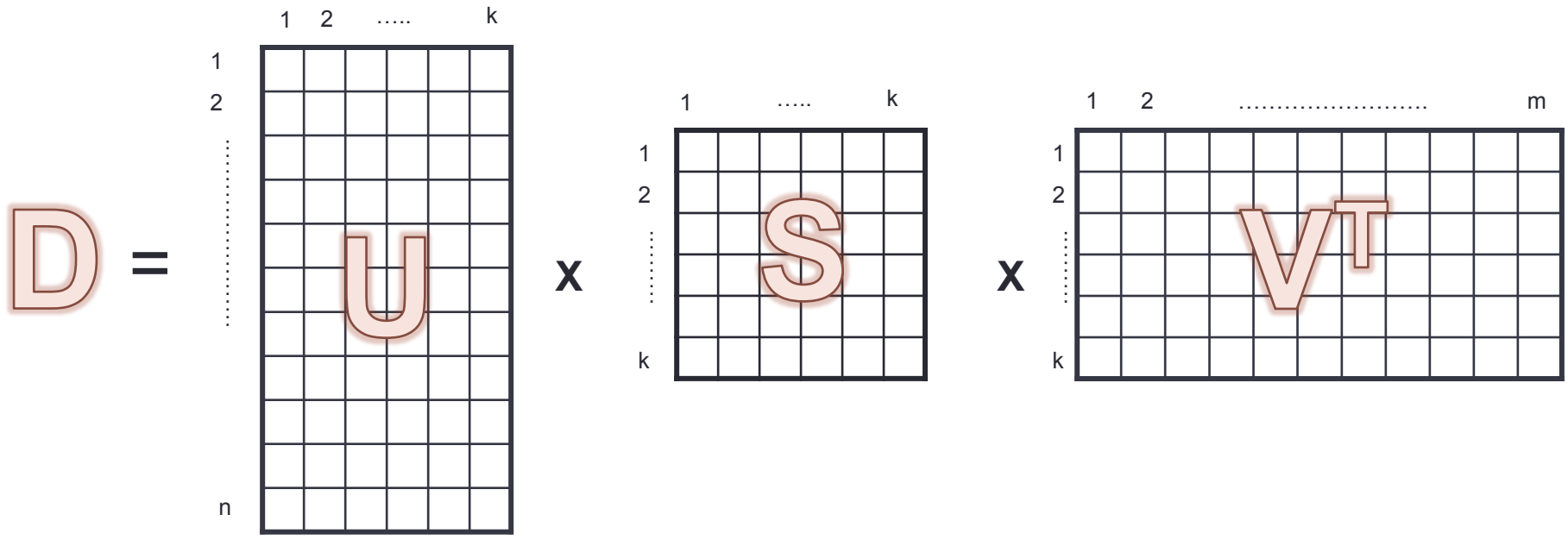
Obj-obj similarity matrix!!!!



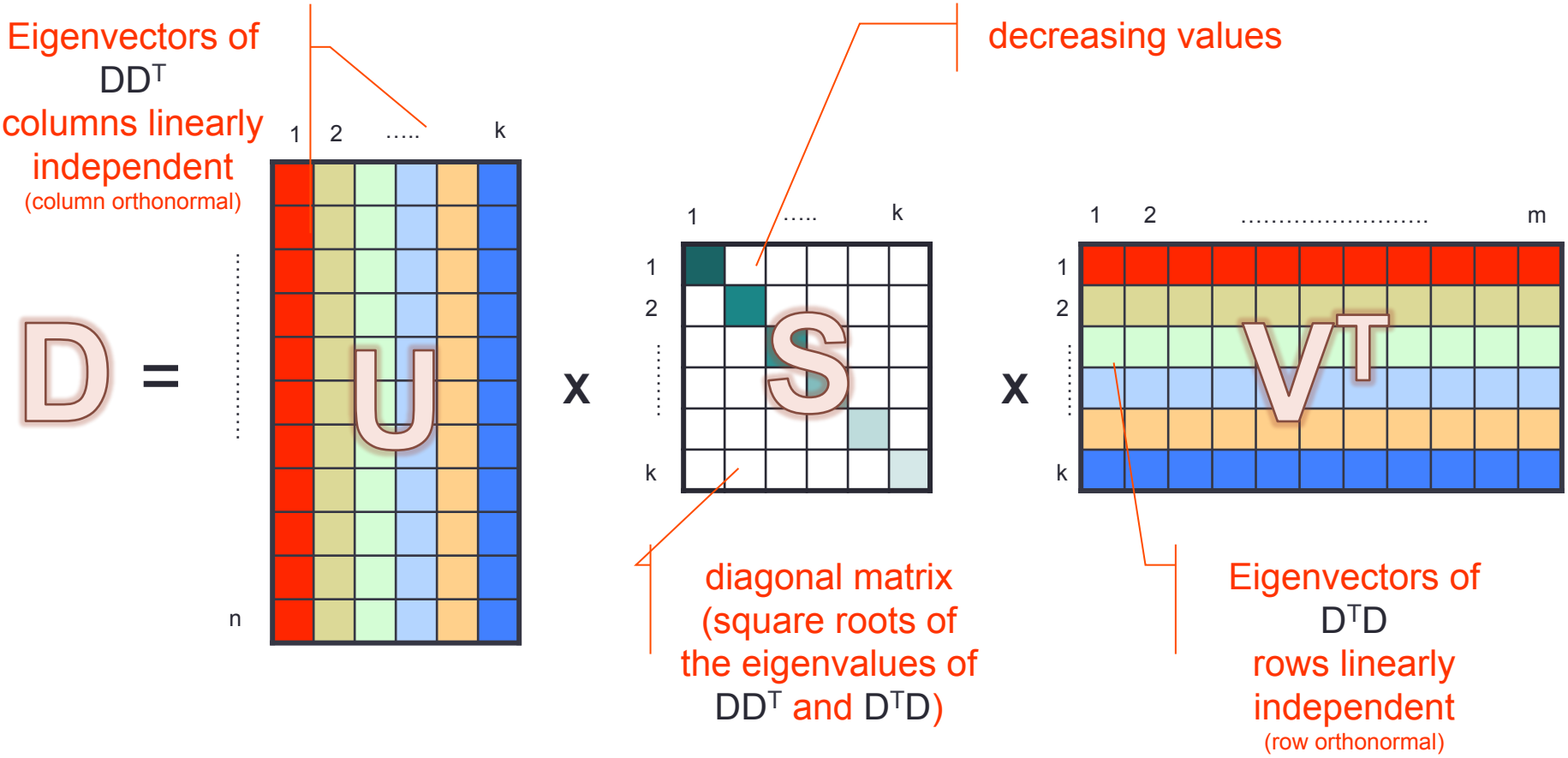
Feature-feature similarity matrix!!!!



Singular valued decomposition



Singular valued decomposition



…reminder

- Eigenvalue and eigenvector
- Given a matrix A, let *c* (scalar) and *x* (vector) be such that

$$c\vec{x} = A\vec{x}$$

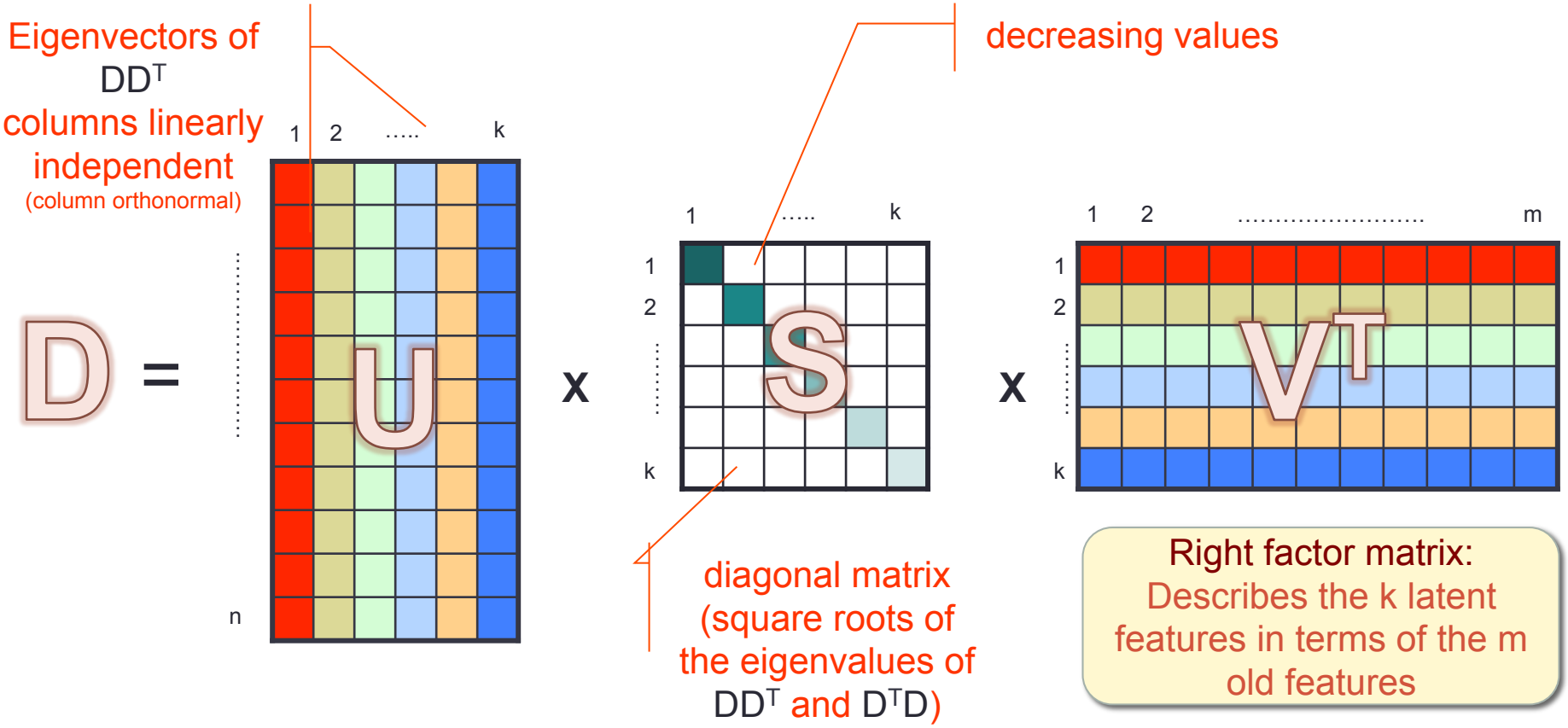
Eigenvalue

Eigenvector

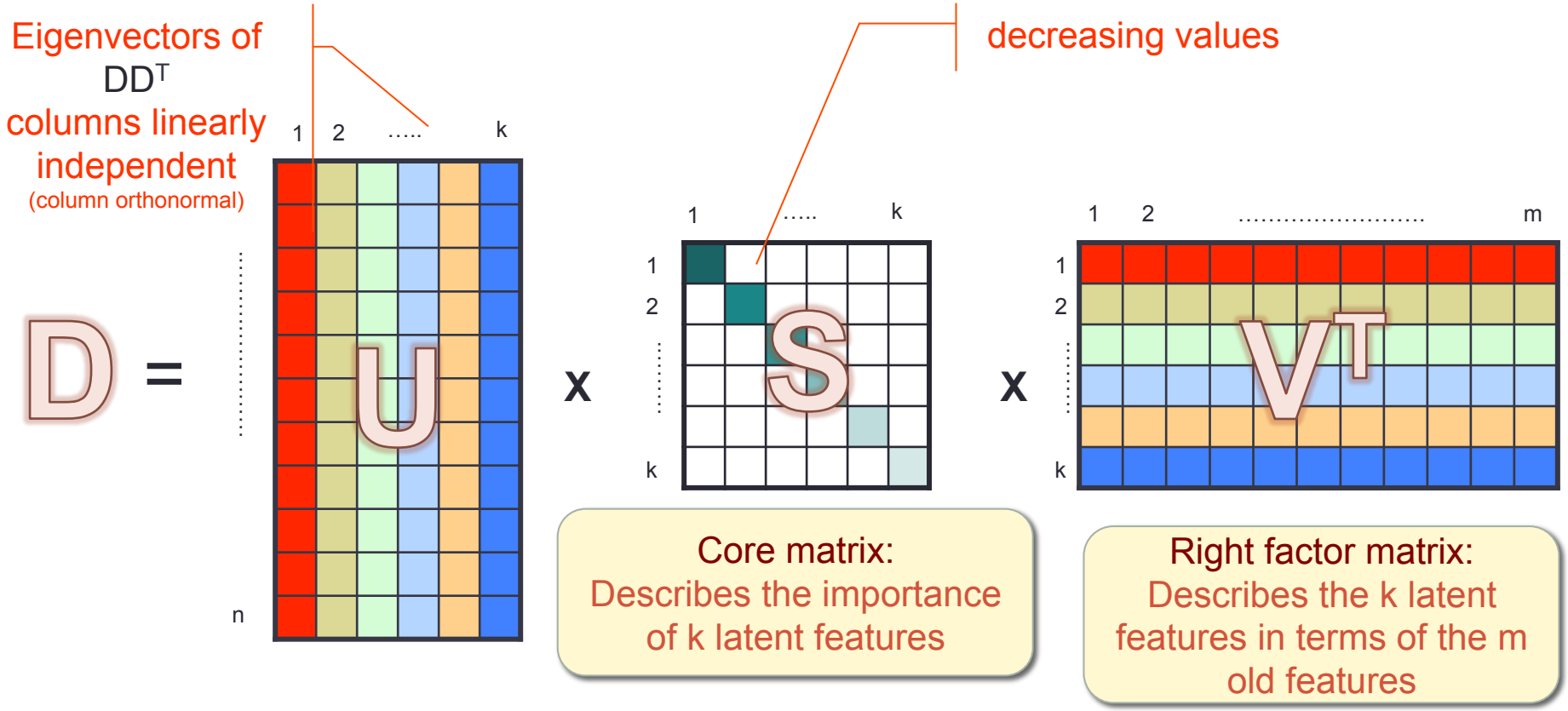
Properties of Eigenvectors

- Suppose that A is and n × n square matrix
  - if the eigenvalues, c1....ck, are distinct, then eigenvectors v1,...vk are a set of k linearly independent vectors.
    - thus they can be used as the basis of the space!!!
  - The value of ci describes the contribution of vi in A.

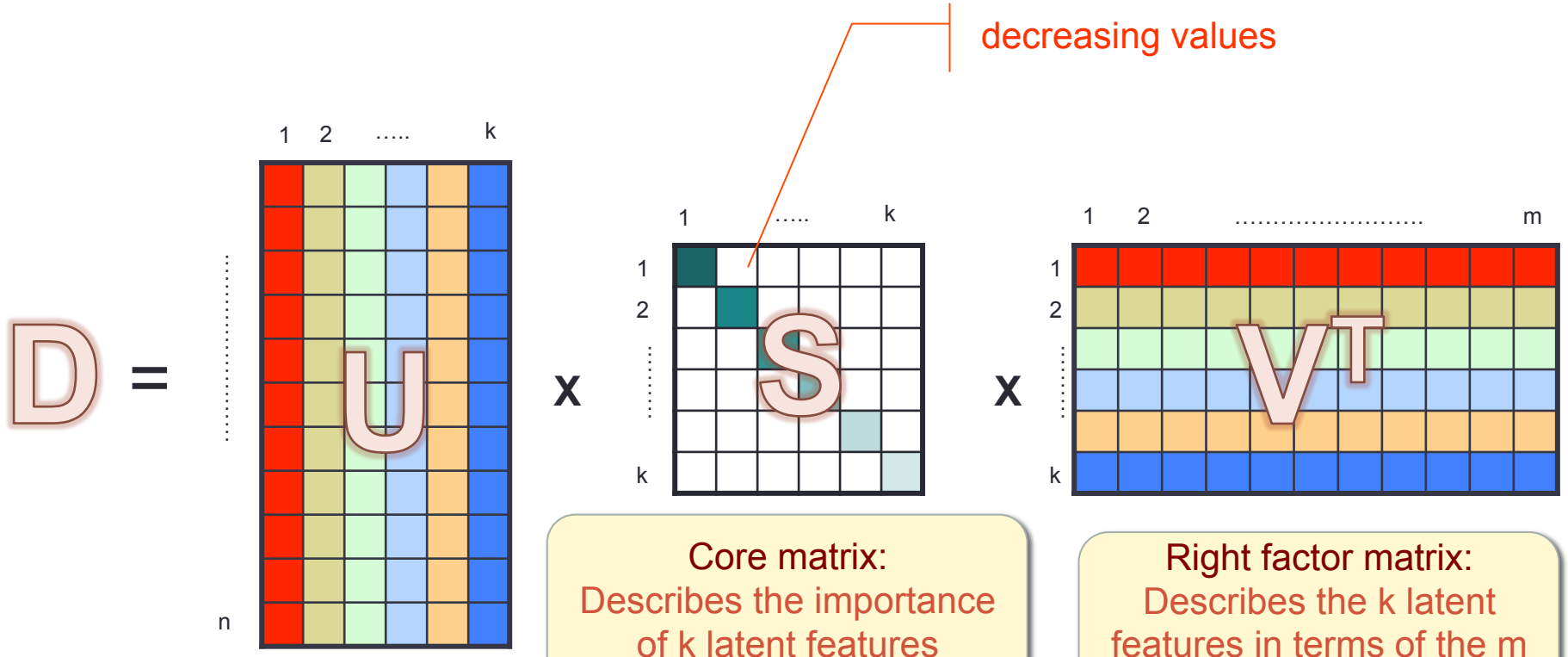
Singular valued decomposition



Singular valued decomposition



Singular valued decomposition

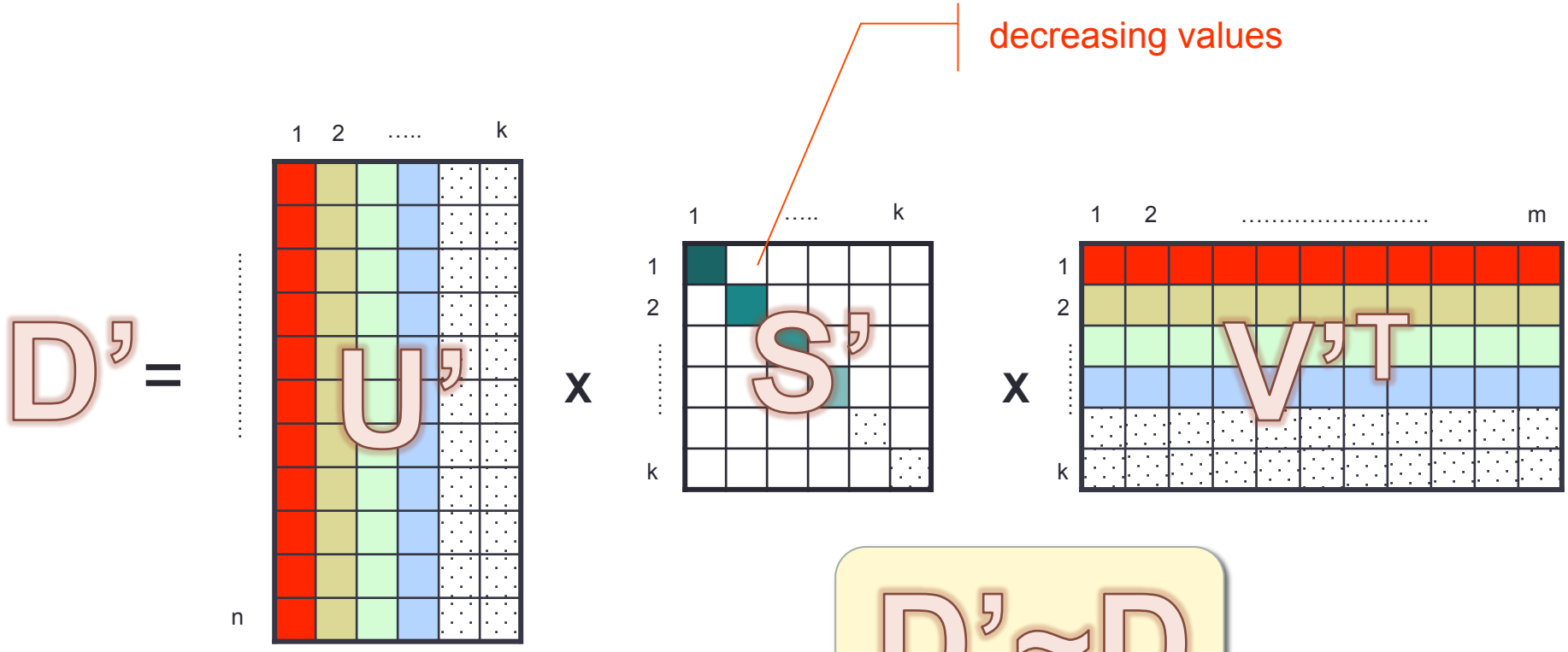


Left factor matrix: Describes the n objects in terms of the k latent features

Core matrix: Describes the importance of k latent features

Right factor matrix: Describes the k latent features in terms of the m old features

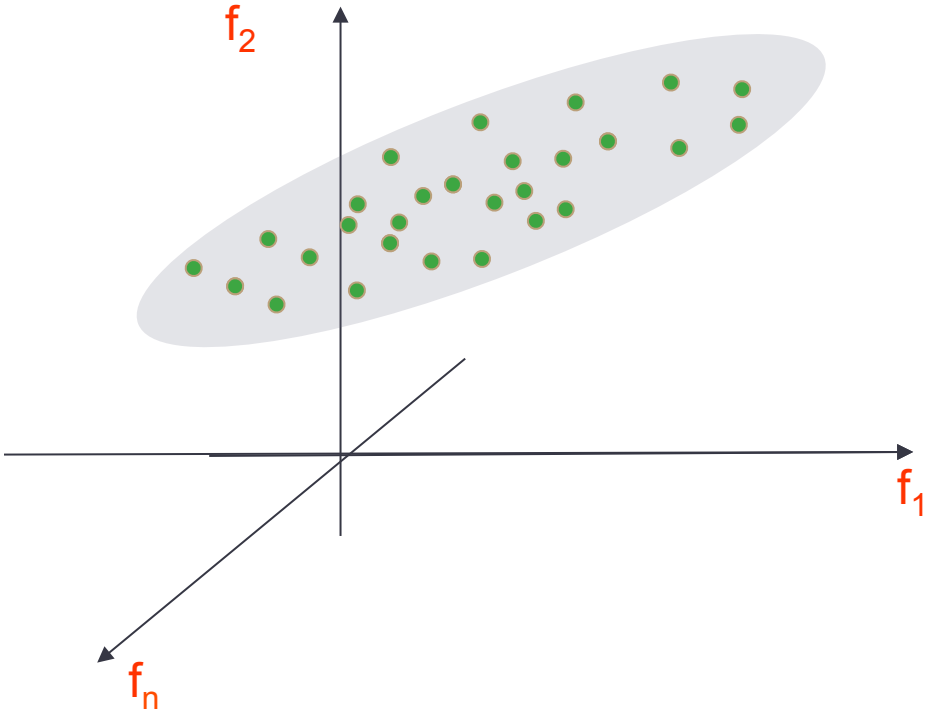
Singular valued decomposition



D'~D

data is preserved

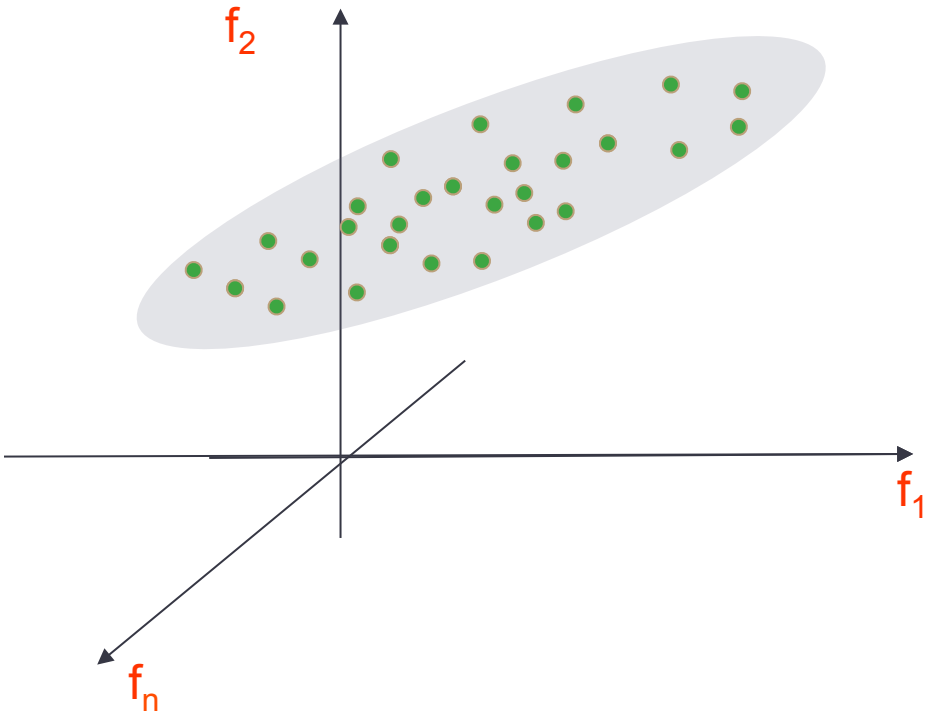
Principle Component Analysis (PCA)

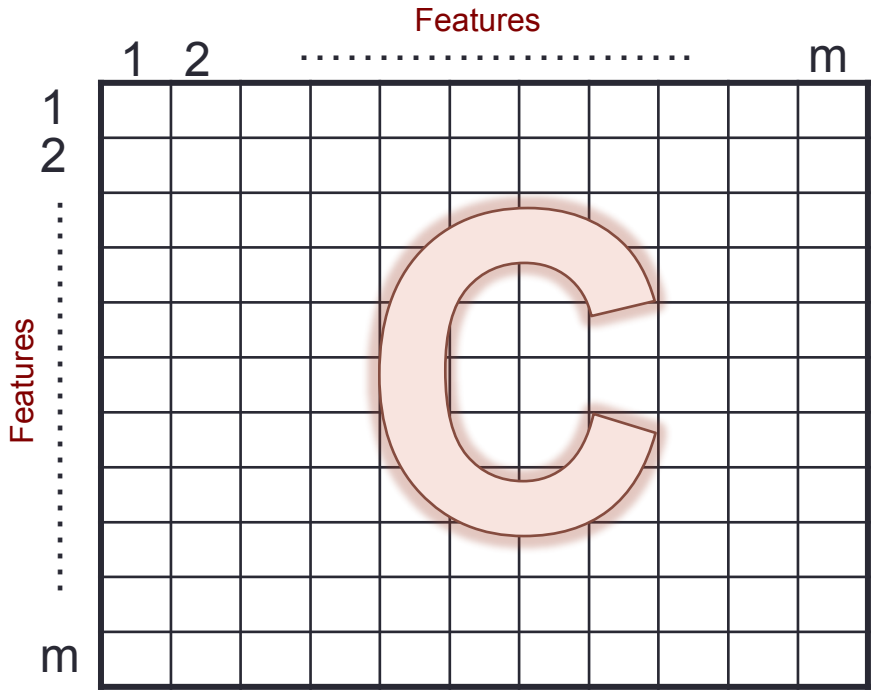


Reminder: The higher the variance is, the better (features are more discriminating)

Principle Component Analysis (PCA)

- ..also known as Karhunen-Loeve Transfrom
  - ..a linear transform that optimally de-correlates the input.



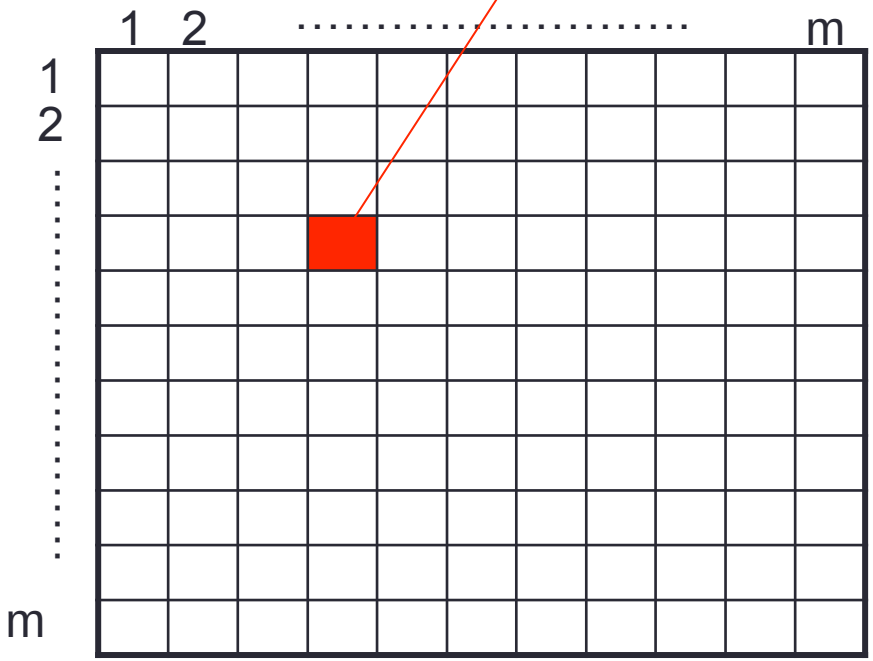


Covariance Matrix
(square symmetric)

$$C[i,j] = Cov(i,j) = E\left((i-\mu_i)(j-\mu_j)\right)$$

Eigen decomposition

C =

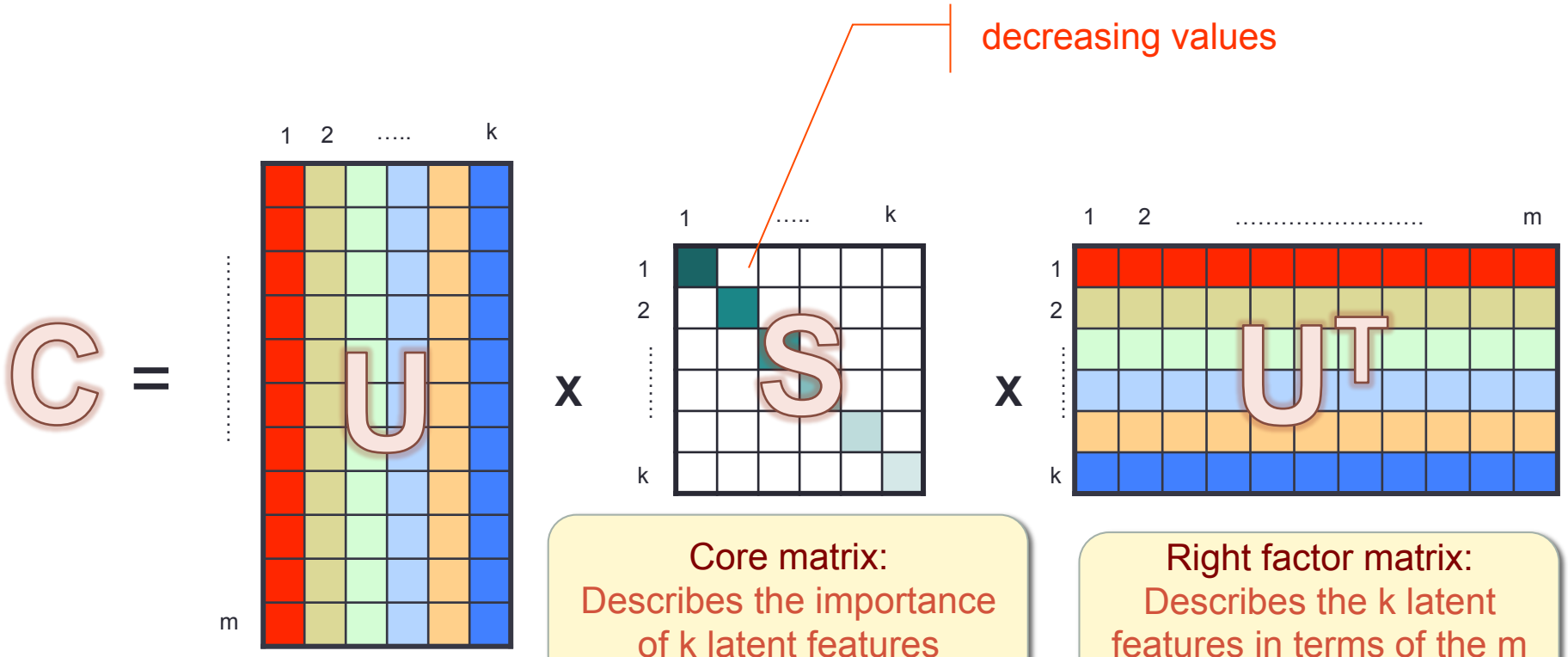


Covariance matrix

Reminder: Properties of Eigenvectors

- If C is and n × n square covariance matrix
  - if the eigenvalues, c1....ck, are distinct, then eigenvectors v1,...vk are a set of k linearly independent vectors.
    - thus they can be used as the basis of the space!!!
  - The value of ci describes the contribution of vi in A. Thus
    - if we pick a C that describes the variation of data,
    - vi will describe the directions along which variance is high

Eigen decomposition of a symmetric covariance matrix

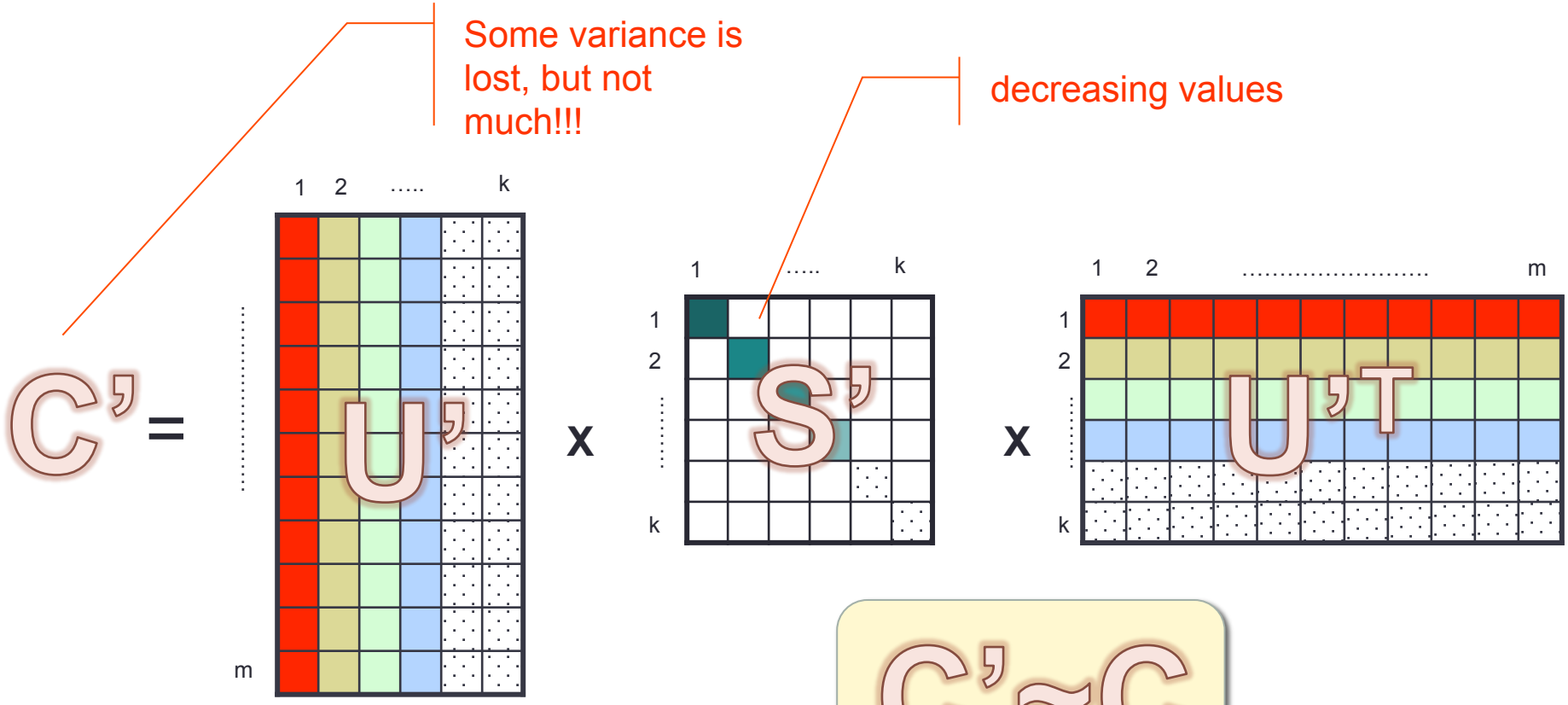


Left factor matrix: Describes the m features in terms of the k latent features

Core matrix: Describes the importance of k latent features

Right factor matrix: Describes the k latent features in terms of the m old features

Principle Component Analysis (PCA)



variance (or discrimination power of the basis) is preserved

How many eigenvectors shall we maintain?

- **Mean eigenvalue**: use only the dimensions whose eigenvalues are greater than or equal to the the mean eigenvalue.
- **Kaiser rule:** keep only those eigenvectors whose eigenvalues are greater than 1 (applied when A is the correlation matrix)
  - eigenvalue is the amount of variance explained by one or more latent semantics,
  - don't add a semantics that explains less variance than is contained in one dimension when if all eigenvalues of a correlation matrix are exactly one
- **Parallel analysis:**
  - analyze a random covariance matrix,
  - Plot cumulative eigenvalues for both random and intended matrices;
  - Find where the two curves intersect.
- **Scree test:** plot the successive eigenvalues to find a point where the plot levels off.
- **Variance explained:** keep enough dimensions to account for 95\% of the initial variance