CS598JHM: Advanced NLP (Spring 2013)
http://courses.engr.illinois.edu/cs598jhm/

Lecture 6: (Probabilistic) Latent Semantic Analysis

Julia Hockenmaier
juliahmr@illinois.edu
3324 Siebel Center
Office hours: by appointment

Indexing by Latent Semantic Analysis (Deerwester et al., 1990)

Latent Semantic Analysis

The task:

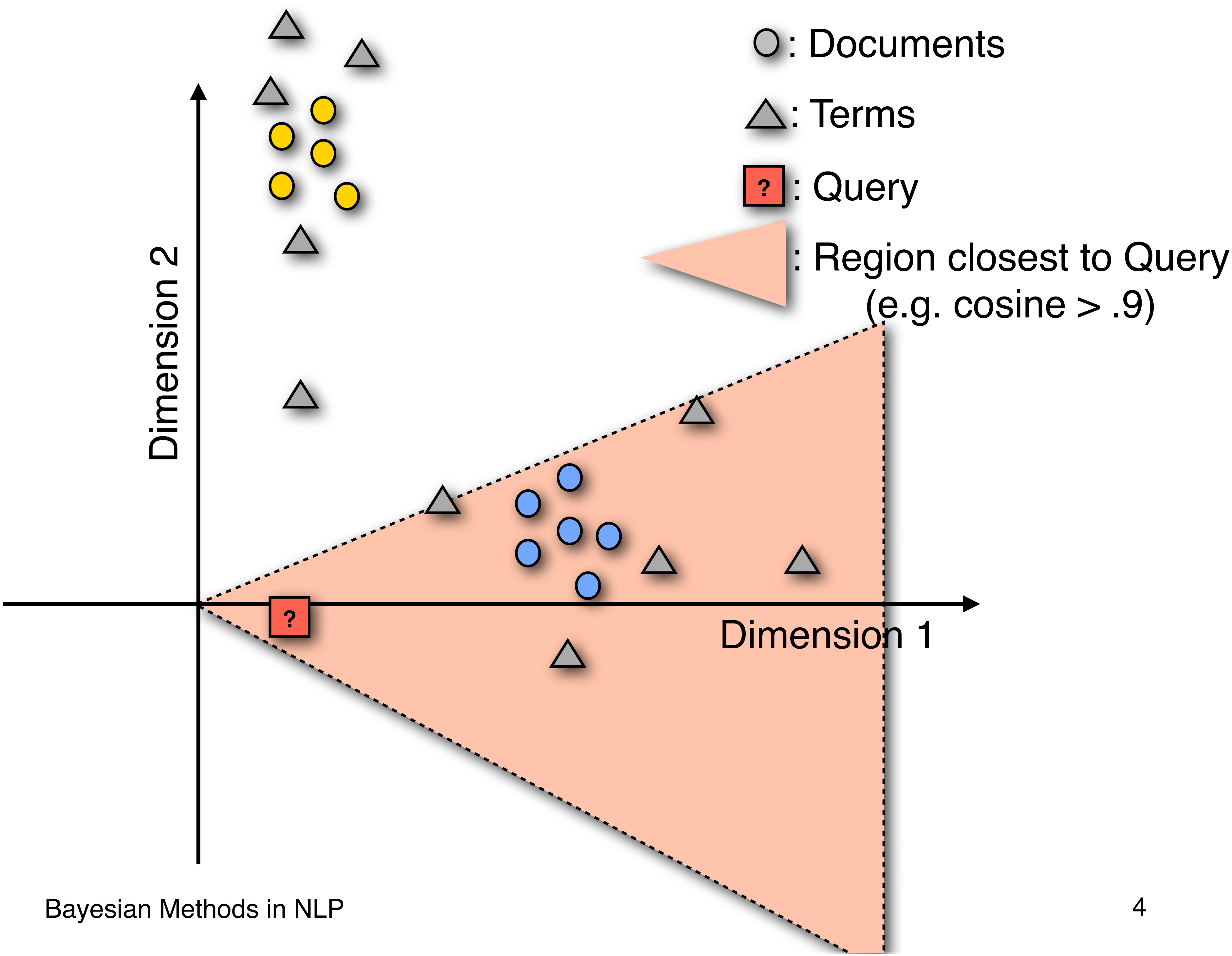
Return *relevant* documents for text queries

The problem: relevance is conceptual/semantic

- The index of relevant documents may not contain all query terms (**synonymy** and missing information)
- The query terms may be ambiguous (**polysemy**)

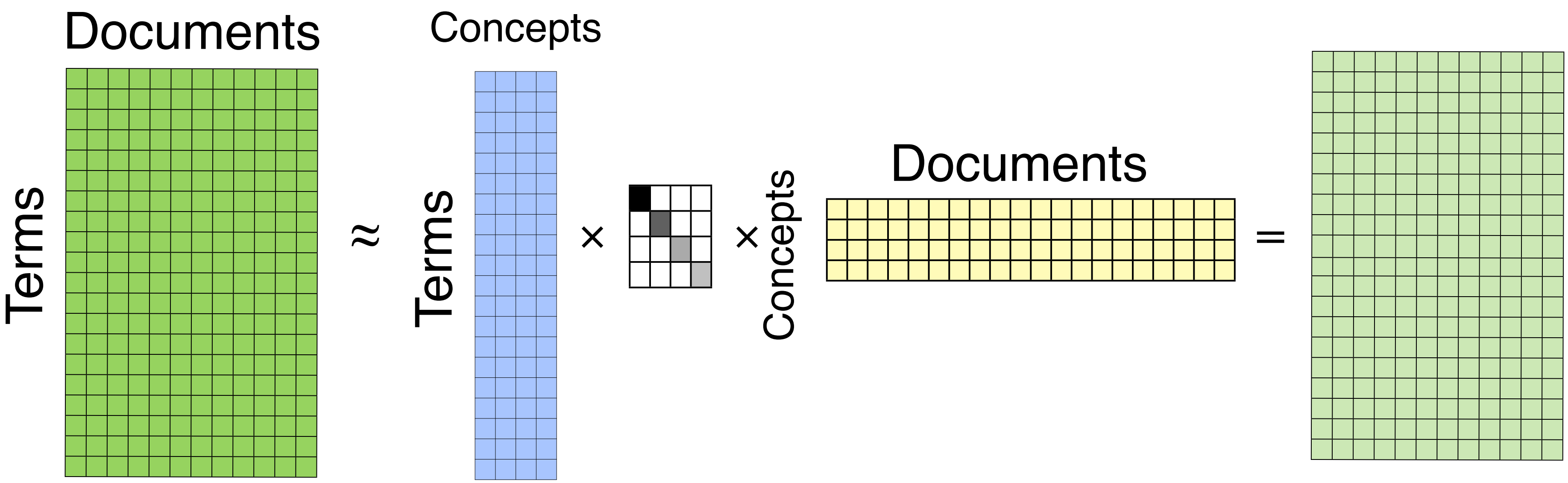
Indexing by Latent Semantic Analysis

- Map queries and documents into a new vector space whose *k* dimensions correspond to independent concepts
- In this space, queries will be near semantically close documents


: Documents
: Terms
?
: Query
: Region closest to Query
(e.g. cosine > .9)
Dimension 2
Dimension 1
?

Latent Semantic Analysis

Low-rank approximation of Singular Value Decomposition (SVD):



$$\mathbf{X} \approx \mathbf{T_0} \times \mathbf{S_0} \times \mathbf{D_0}' = \dot{\mathbf{X}}$$

$\mathbf{X}$: Term-document matrix (=data): $\mathbf{X}_{ij}$ = freq of w_i in D_j

$\dot{\mathbf{X}} = \mathbf{T_0}\,\mathbf{S_0}\mathbf{D_0}'$ (k-rank approximation of $\mathbf{X}$)

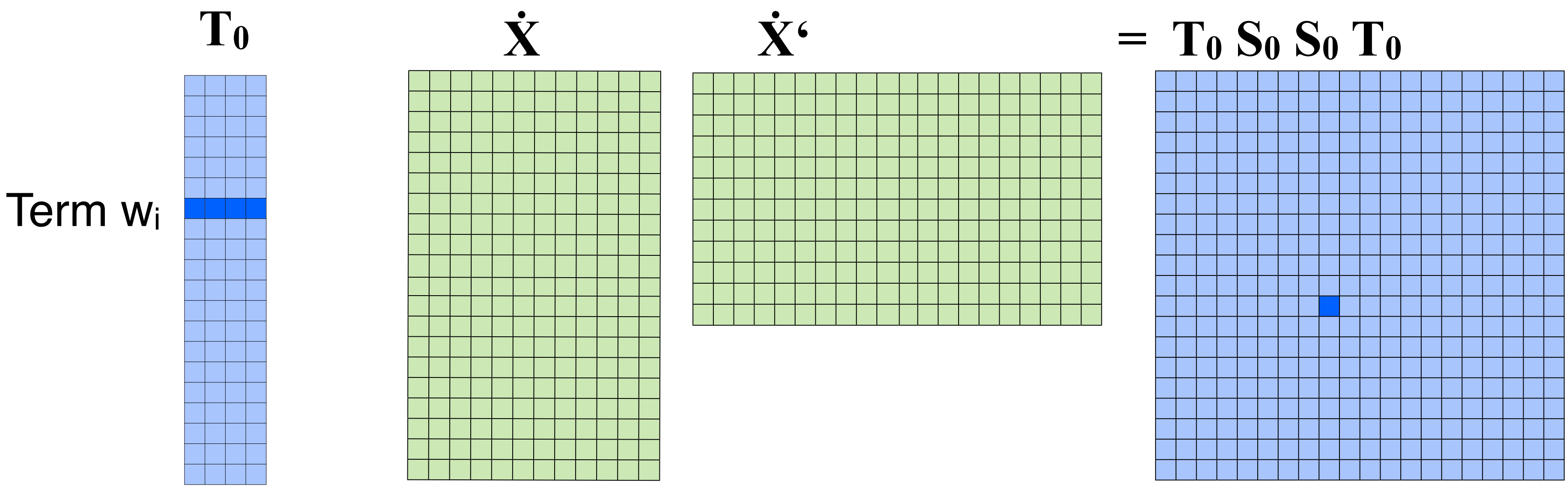
$\mathbf{T_0}$: Columns are orthogonal and unit-length $\mathbf{T_0}'\mathbf{T_0} = \mathbf{I}$

$\mathbf{S_0}$: Diagonal matrix of the k largest singular values

$\mathbf{D_0}$: Columns are orthogonal and unit-length $\mathbf{D_0}'\mathbf{D_0} = \mathbf{I}$

this should really be $\hat{x}$

LSA: term similarity



dot product of w_i, w_j in the new space

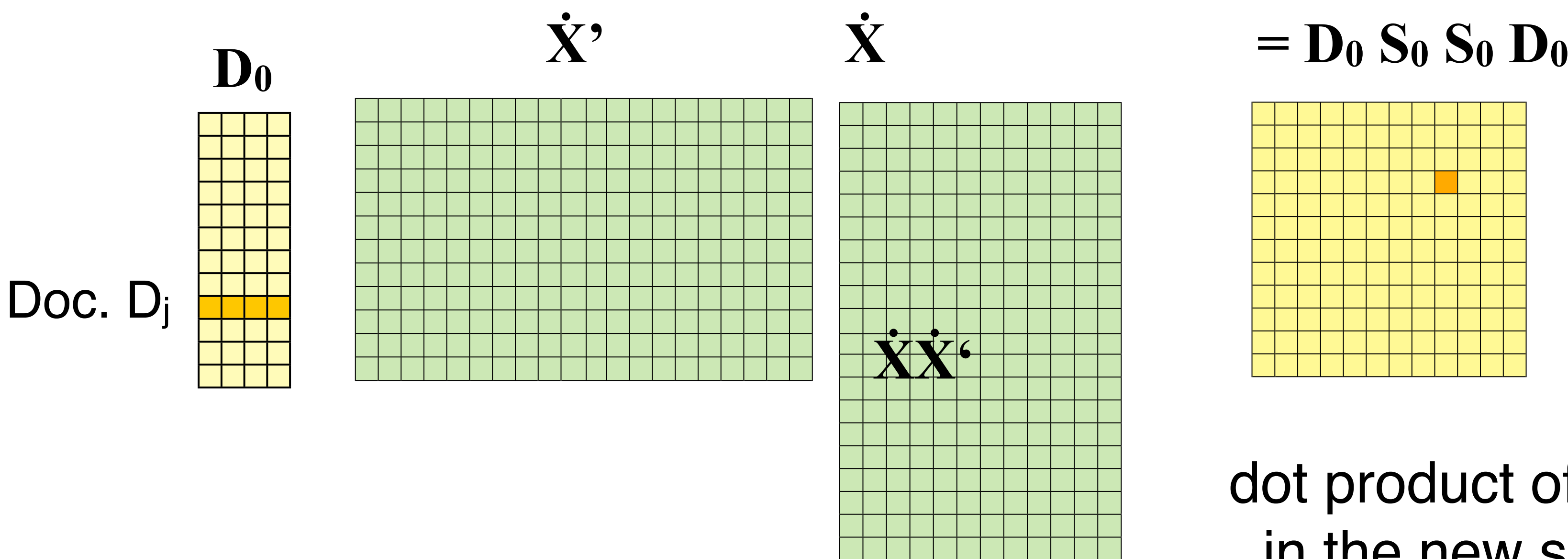
$T_0 S_0$

$\dot{X}\dot{X}' = T_0 S_0 S_0 T_0$

(**D** cancels out because **S** is diagonal and **D** orthonormal)

Similarity of terms w_i, w_j in the new space: $(\dot{X}\dot{X}')_{ij}$

LSA: document similarity



dot product of D_i, D_j
in the new space
$D_0 S_0$

$\dot{X}'\dot{X} = D_0 S_0 S_0 D_0$

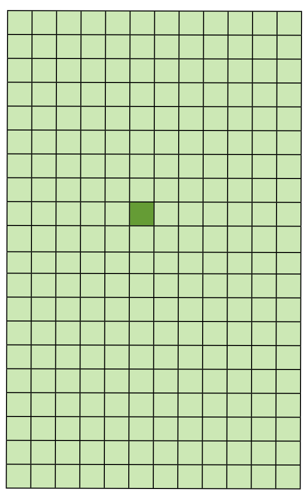
(**T** cancels out because **S** is diagonal and **T** orthonormal)

Similarity of documents d_i, d_j in the new space: $(\dot{X}'\dot{X})_{ij}$

LSA: term-document similarity

The elements of $\dot{\mathbf{X}}$ give the similarity of terms and documents.

Now, terms are projected to $\mathbf{TS}^{1/2}$, documents to $\mathbf{DS}^{1/2}$



LSA: query-document similarity

Queries q are 'pseudo-documents':
they don't appear in $\mathbf{X}$

Construct their term vector $\mathbf{X}_{\mathrm{q}}$
Define their document vector $\mathbf{D}_{\mathrm{q}} = \mathbf{X'}_{\mathrm{q}}\,\mathbf{TS}^{-1}$

Probabilistic Latent Semantic Indexing

(Hofmann 1999)

The aspect model

Observations are document-word pairs (d, w)

Assume there are *k* aspects $z_1 \ldots z_k$
Each observation is associated with a hidden aspect z

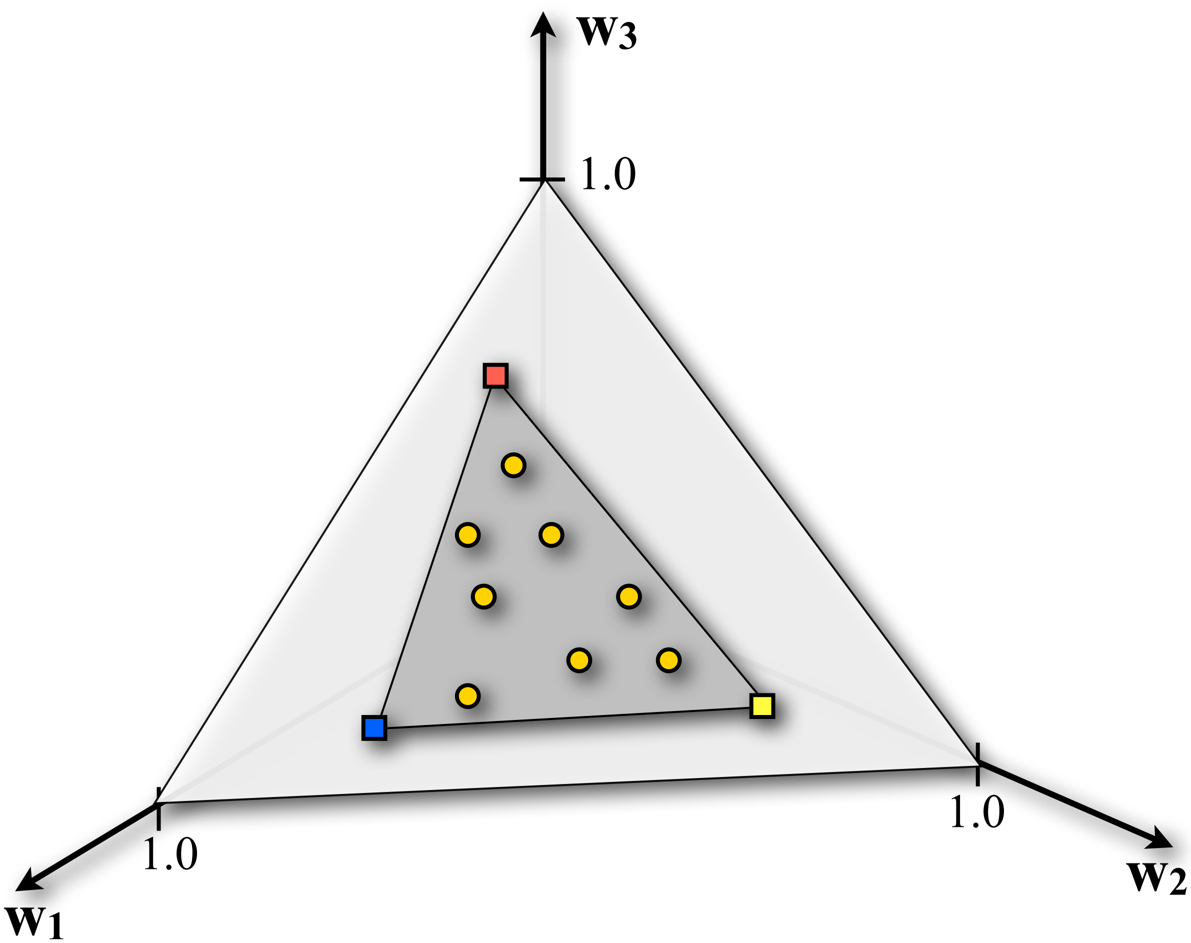
$$P(d, w) = P(d)P(w \mid d)$$

with $$P(w \mid d) = \sum_{z \in Z} P(w \mid z)P(z \mid d)$$

Or, equivalently:

$$P(d, w) = \sum_{z \in Z} P(z)P(d \mid z)P(w \mid z)$$

A geometric interpretation



Word simplex
Any point in this simplex defines a multinomial over words

Documents $P(w \mid d)$
Each document corresponds to one multinomial over words

Topics $P(w \mid z)$
Each topic is a multinomial over words

Topic simplex
The topics define the corners of a (sub)simplex.
All training documents lie inside this topic simplex.

$$P(w \mid d) = \lambda_1 P(w \mid z_1) + \lambda_2 P(w \mid z_2) + \lambda_3 P(w \mid z_3)$$
$$= P(z_1 \mid d)P(w \mid z_1) + P(z_2 \mid d)P(w \mid z_2) + P(z_3 \mid d)P(w \mid z_3)$$

PLSA is a mixture model

Mixture models:

- K mixture components and N observations $x_{1 \dots} x_N$
- Mixing weights $(\theta_1 \dots \theta_K)$: $P(k) = \theta_K$
- Each observation x_n is generated by mixture component z_n
  $P(x_n) = P(z_n)\, P(x_n \mid z_n)$

PLSI:

- Mixture components = topics
- Mixing weights are specific to each document $\theta_d = (\theta_{d1} \dots \theta_{dK})$
- Each observation (word) $w_{d,n}$ is a sample
  from the document-specific mixture model.
  It is drawn from one of the components $z_{d,n}$
  $P(w_{d,n}) = P(z_{d,n} \mid \theta_d)\, P(w_{d,n} \mid z_{d,n})$

Estimation: EM algorithm

E-step: Recompute

$P(z \mid d, w) = P(z, d, w) / \sum_{z'} P(z', d, w)$

with $P(z, d, w) = P(z)P(d \mid z)P(w \mid z)$

M-step: Recompute

$P(w \mid z) \propto \sum_{d} \text{freq}(d, w)\, P(z \mid d, w)$

$P(d \mid z) \propto \sum_{w} \text{freq}(d, w)\, P(z \mid d, w)$

$P(z) \propto \sum_{d} \sum_{w} \text{freq}(d, w)\, P(z \mid d, w)$