Learning Word Representations with Hierarchical Sparse Coding

Dani Yogatama, Manaal Faruqui, Chris Dyer, Noah A. Smith

Language Technologies Institute
School of Computer Science
Carnegie Mellon University

Contributions

- A word embedding model that respects hierarchical organization of dimensions of word vectors (word meanings)

- Better than `word2vec` (Mikolov et al., 2013) and `glove` (Pennington et al., 2014) for word similarity ranking and when used as features for sentiment analysis; competitive on other tasks

- An optimization method for large-scale sparse coding

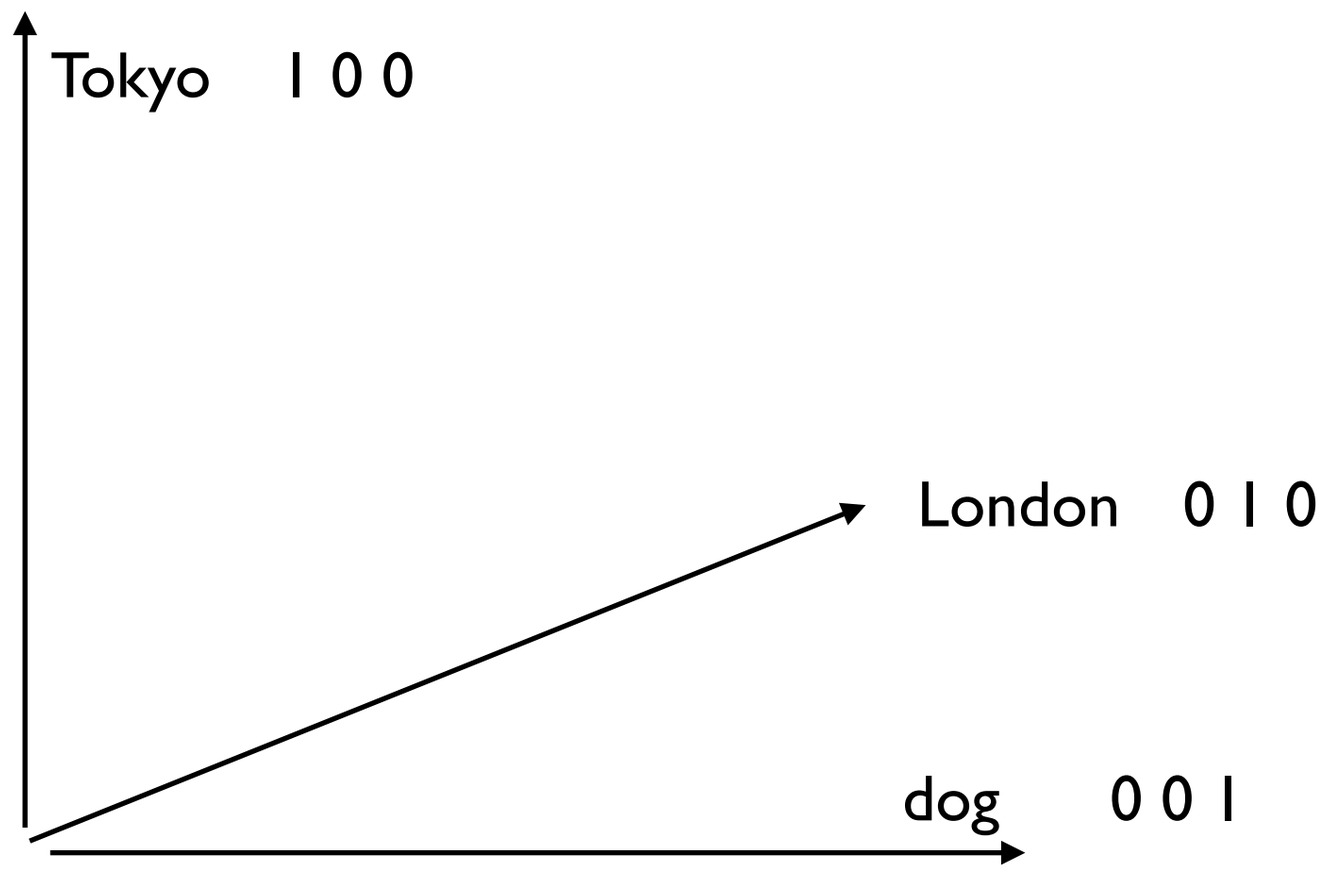
clab

Outline

- Background
- Model
- Learning algorithm
- Experiments
- Summary

Word representations

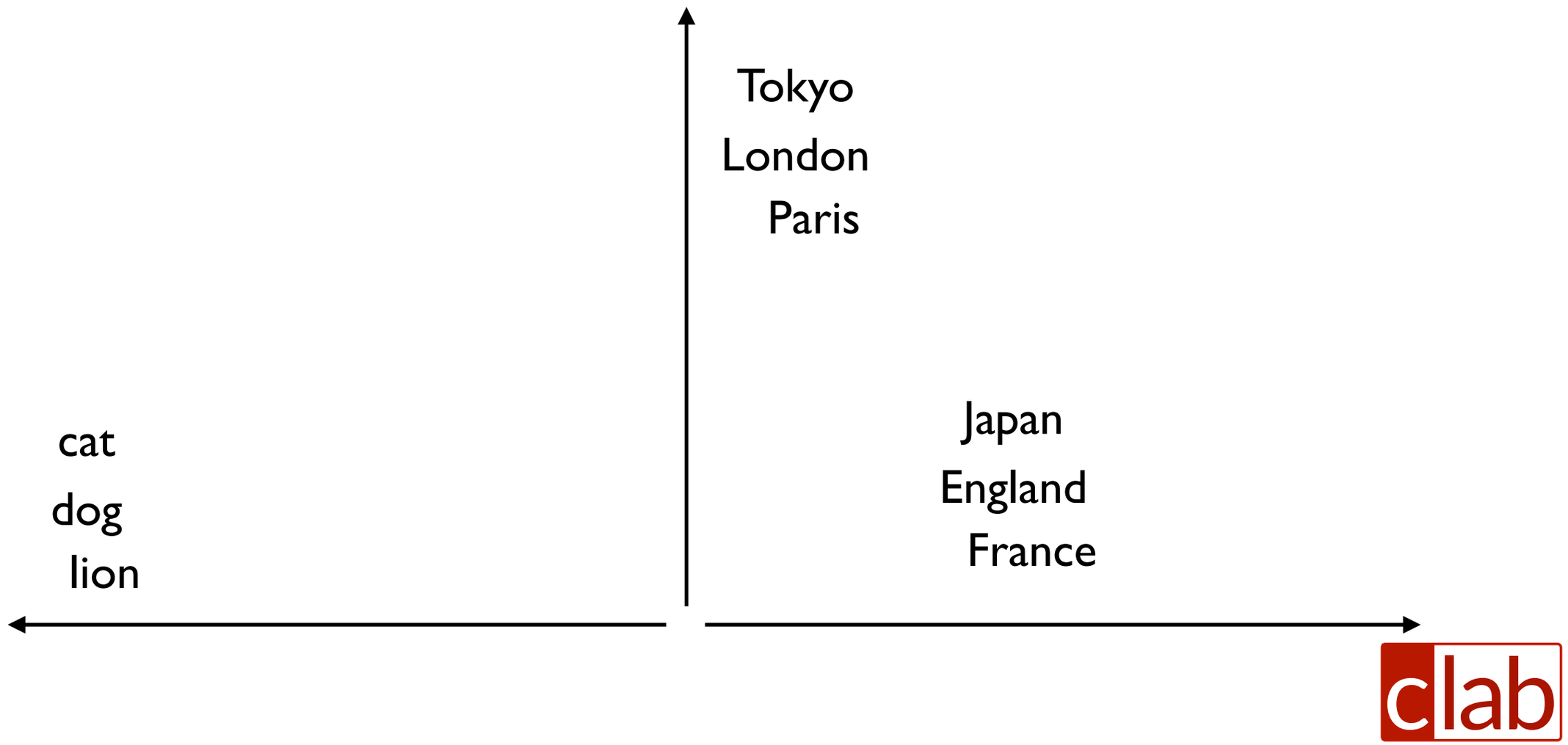
- Classic categorical representation of words as indices does not capture syntactic and semantic similarities



Word representations

- Classic categorical representation of words as indices does not capture syntactic and semantic similarities

Turney and Pantel, 2010, Mikolov et al., 2010, Mnih and Teh, 2012, Huang et al., 2012

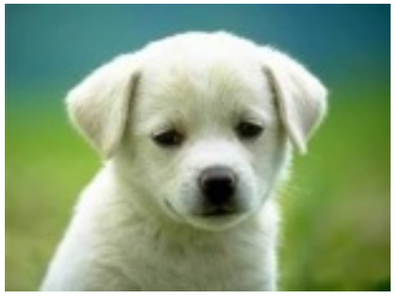


Main ideas

- In lexical semantics, we often capture the relationships between word meaning's in hierarchically-organized lexicons

Main ideas

- In lexical semantics, we often capture the relationships between word meaning's in hierarchically-organized lexicons
  - Example: WordNet (**Miller 1995**)



Main ideas

- In lexical semantics, we often capture the relationships between word meaning's in hierarchically-organized lexicons
  - Example: WordNet (**Miller 1995**)
- In word representations, each (latent) dimension can be seen as a concept
- We are interested in organizing these dimensions in hierarchies
- Our approach is still several steps away from inducing a lexicon such as WordNet, but it still seeks to discover a solution in a similar coarse-to-fine way

clab

Notation

- We represent words as vectors of contexts

Notation

- We represent words as vectors of contexts

$\mathbf{X} =$ context

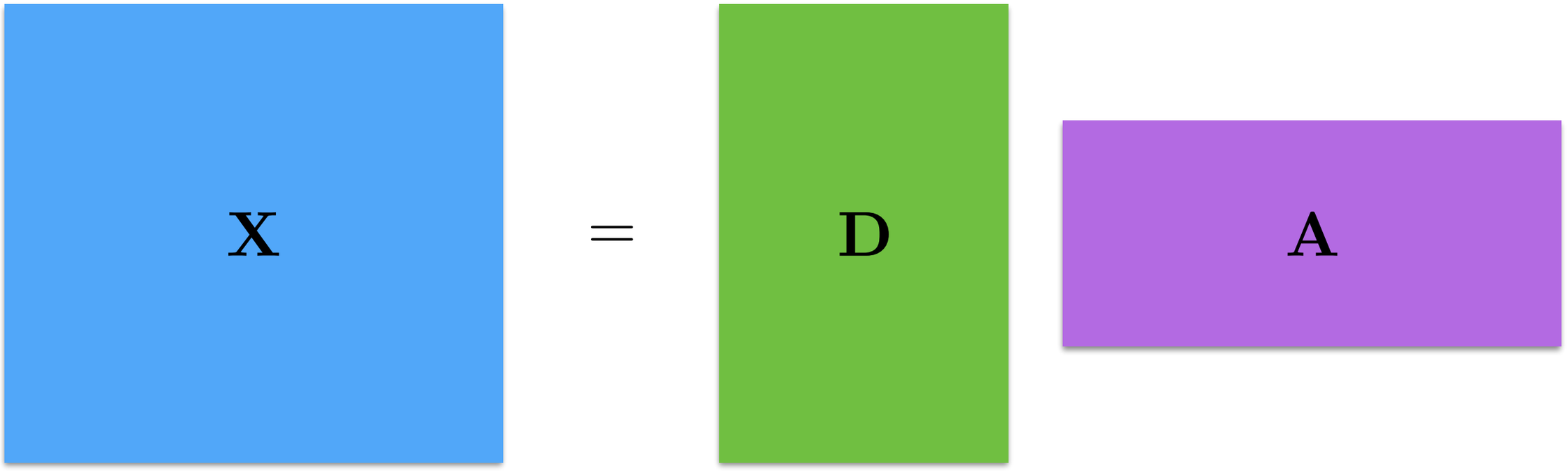
	words		
	tokyo	**london**	**paris**
tokyo	2	6	0
london	6	4	3
paris	0	3	2

Notation

- Hierarchical Sparse Coding
  - Given a word co-occurrence matrix $\mathbf{X}$

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}} \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_2^2 + \lambda\Omega(\mathbf{A})$$

Notation

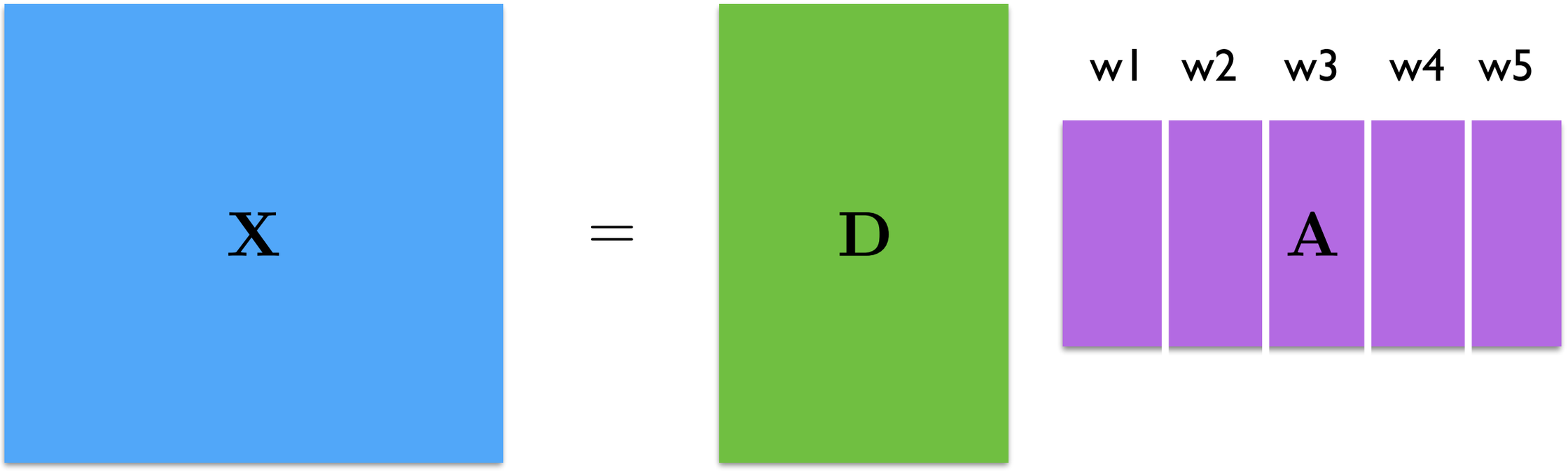


input matrix
context by words

dictionary
context by latent dimensions

word representations
latent dimensions by words

Notation



input matrix
context by words

dictionary
context by latent dimensions

word representations
latent dimensions by words

Notation

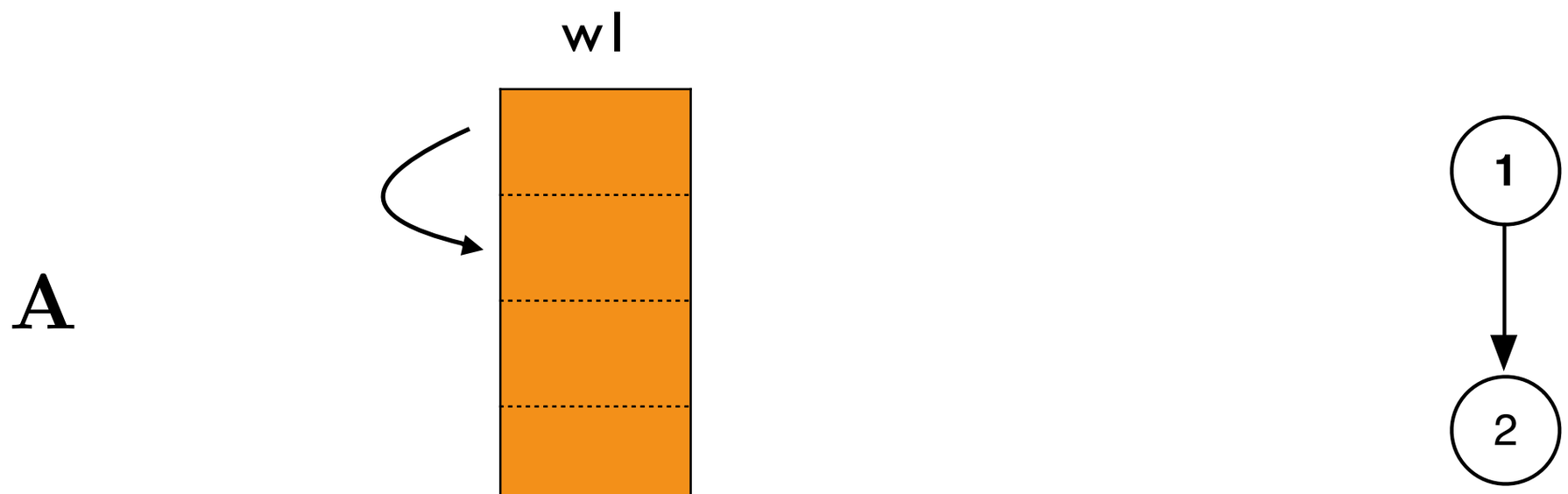
- Hierarchical Sparse Coding
  - Given a word co-occurrence matrix $\mathbf{X}$

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}} \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_2^2 + \lambda\Omega(\mathbf{A})$$

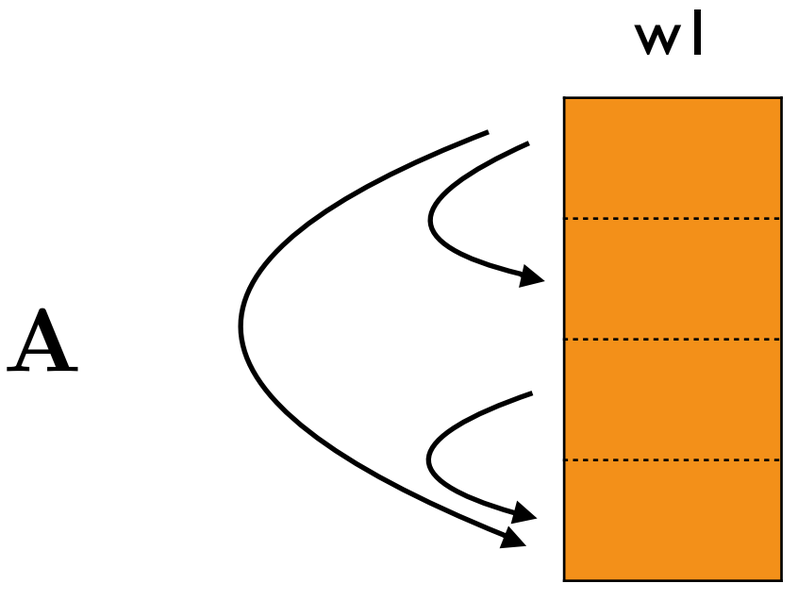
  - Impose hierarchical ordering of the embedding dimensions

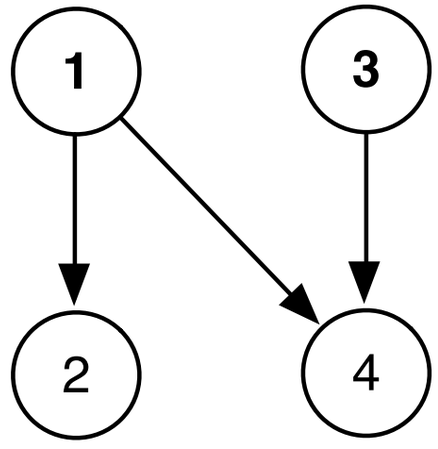
clab

Notation

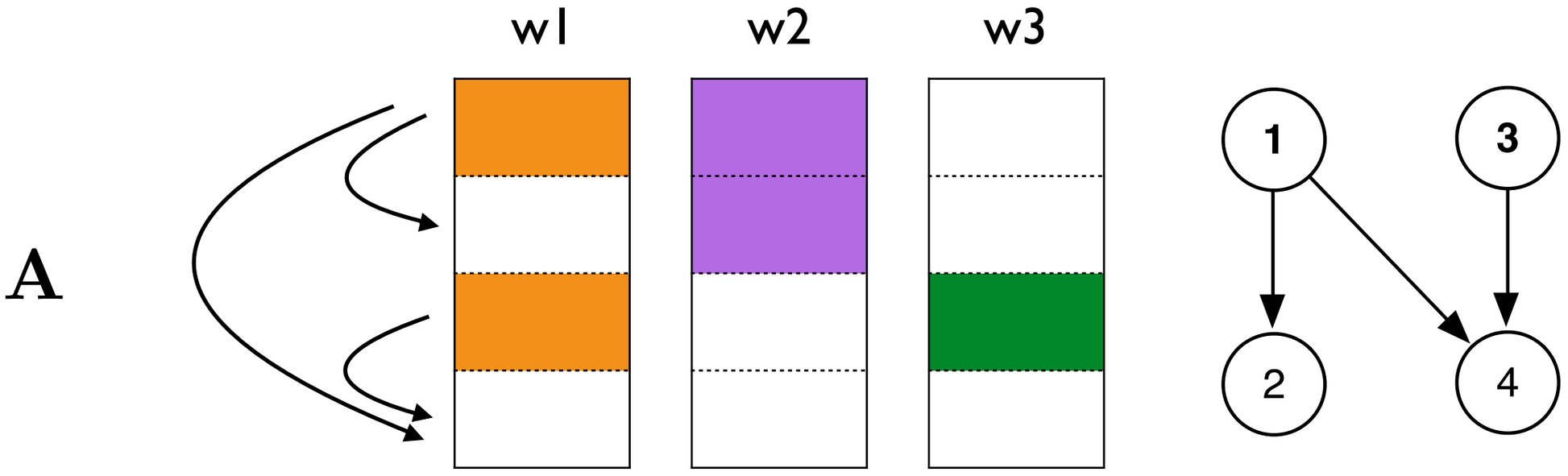


Notation





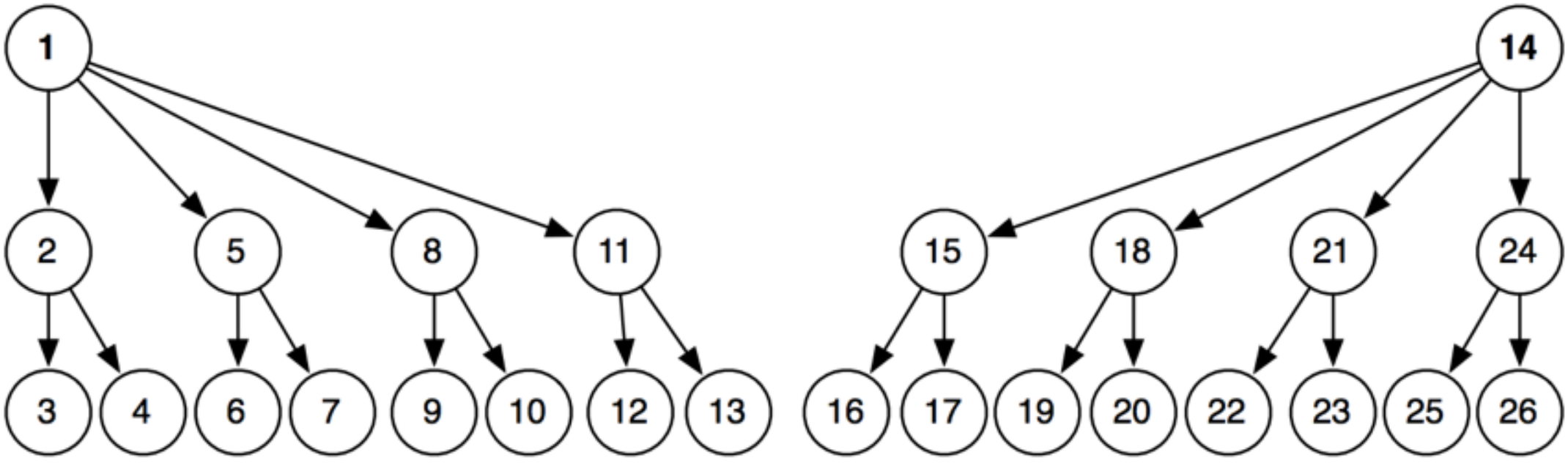
Notation



For each word, the value of the children row (dimension) can be nonzero if and only if the values of all of its ancestor rows (dimensions) are non-zero

Zhao et al., 2009, Jenatton et al., 2011

Tree regularizer

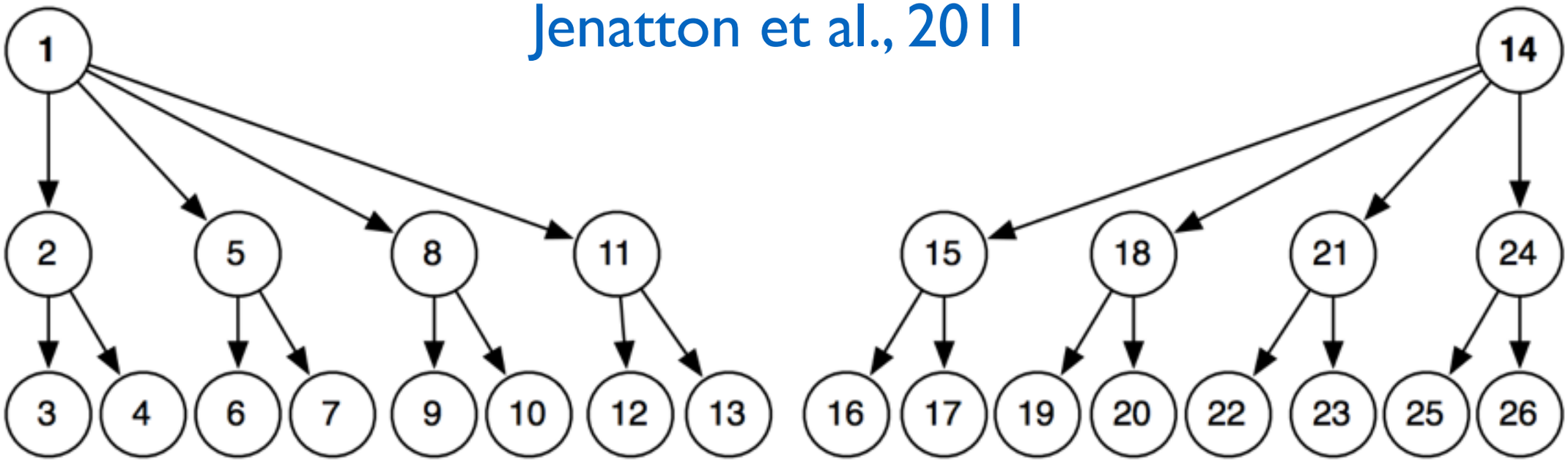


Tree regularizer

Recursively apply group lasso from root to leaves

Each group is a node and all its descendants

Jenatton et al., 2011



$$\Omega(\mathbf{a}_v) = \sum_i \|\langle a_{v,i}, \mathbf{a}_{v,\text{Descendants(i)}}\rangle\|_2$$

ARK

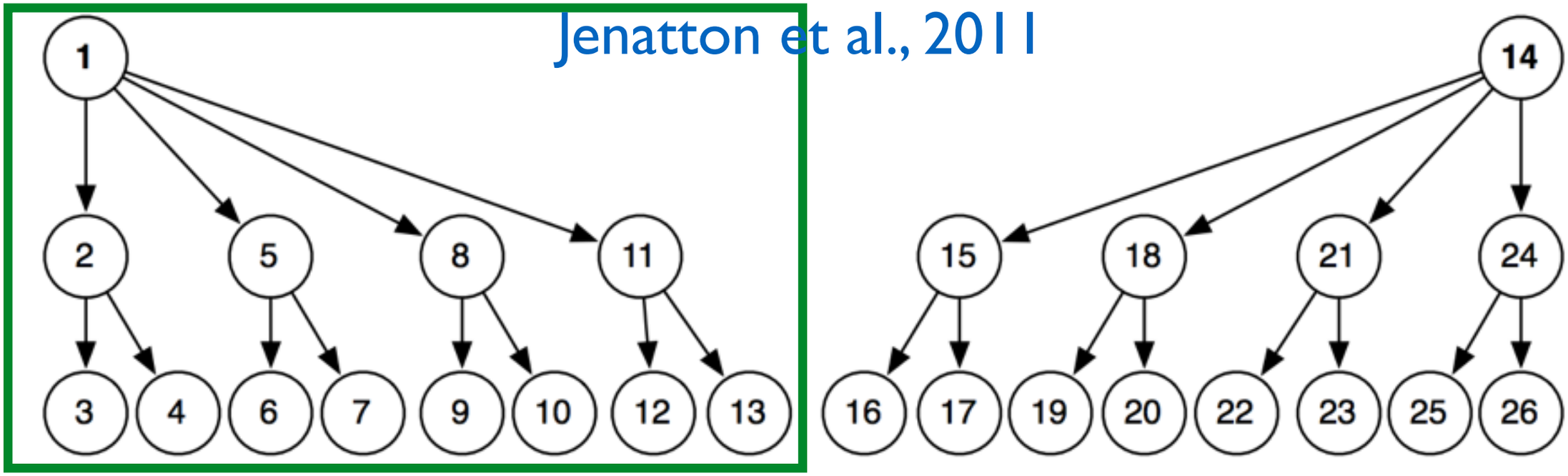
clab

Tree regularizer

Recursively apply group lasso from root to leaves

Each group is a node and all its descendants

Jenatton et al., 2011



$$\Omega(\mathbf{a}_v) = \sum_i \|\langle a_{v,i}, \mathbf{a}_{v,\text{Descendants(i)}}\rangle\|_2$$

ARK

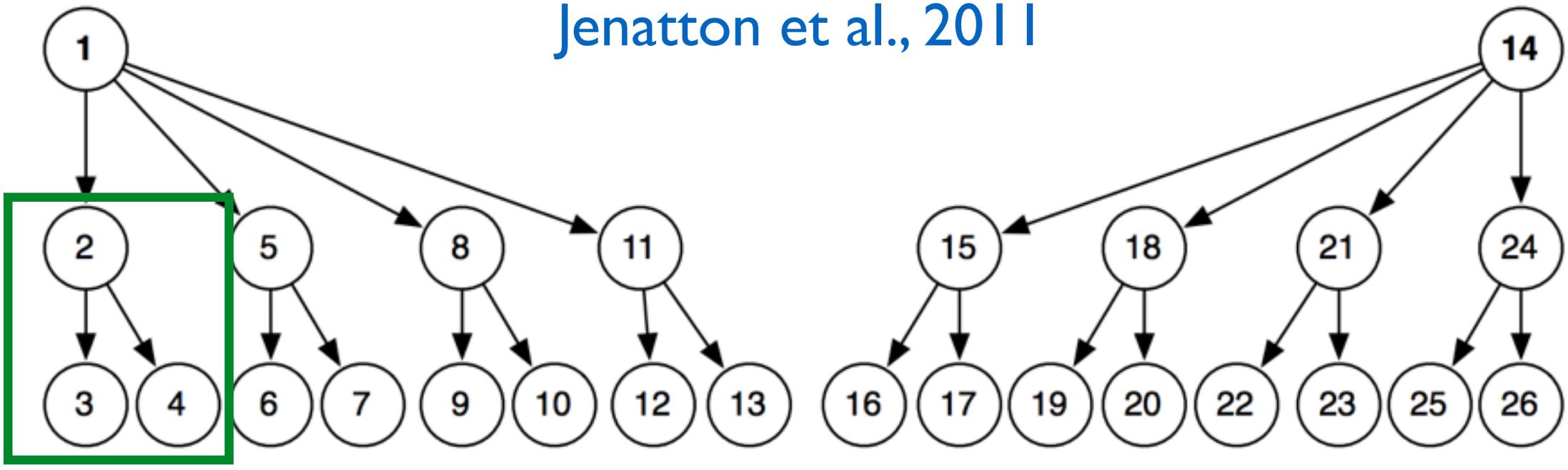
clab

Tree regularizer

Recursively apply group lasso from root to leaves

Each group is a node and all its descendants

Jenatton et al., 2011



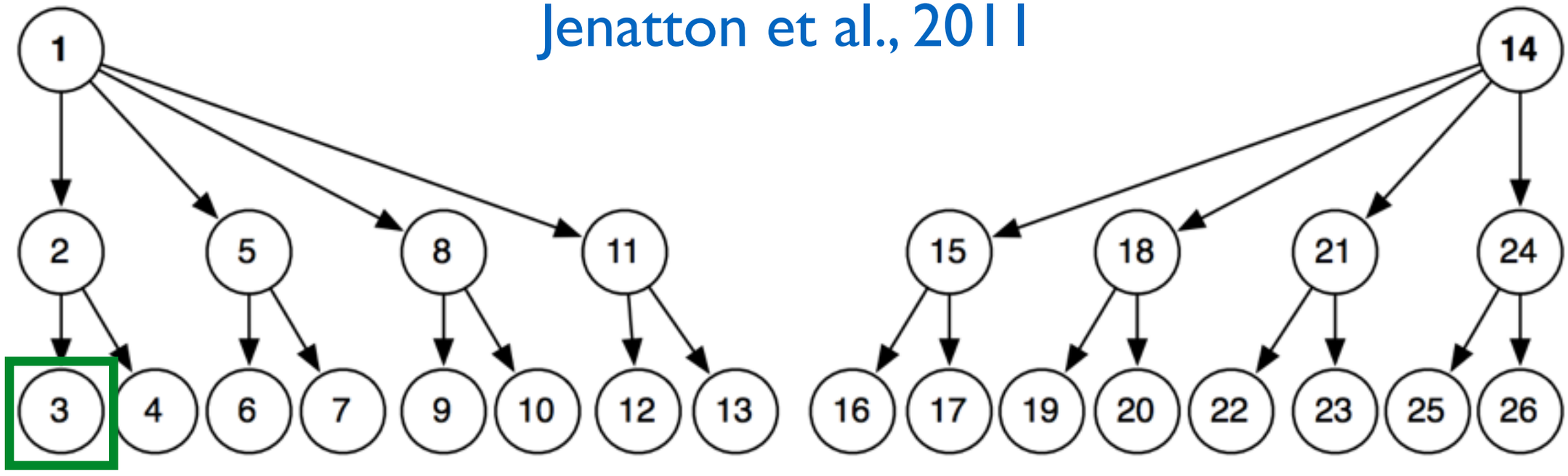
$$\Omega(\mathbf{a}_v) = \sum_i \|\langle a_{v,i}, \mathbf{a}_{v,\text{Descendants(i)}}\rangle\|_2$$

Tree regularizer

Recursively apply group lasso from root to leaves

Each group is a node and all its descendants

Jenatton et al., 2011



$$\Omega(\mathbf{a}_v) = \sum_i \|\langle a_{v,i}, \mathbf{a}_{v,\text{Descendants(i)}}\rangle\|_2$$

ARK

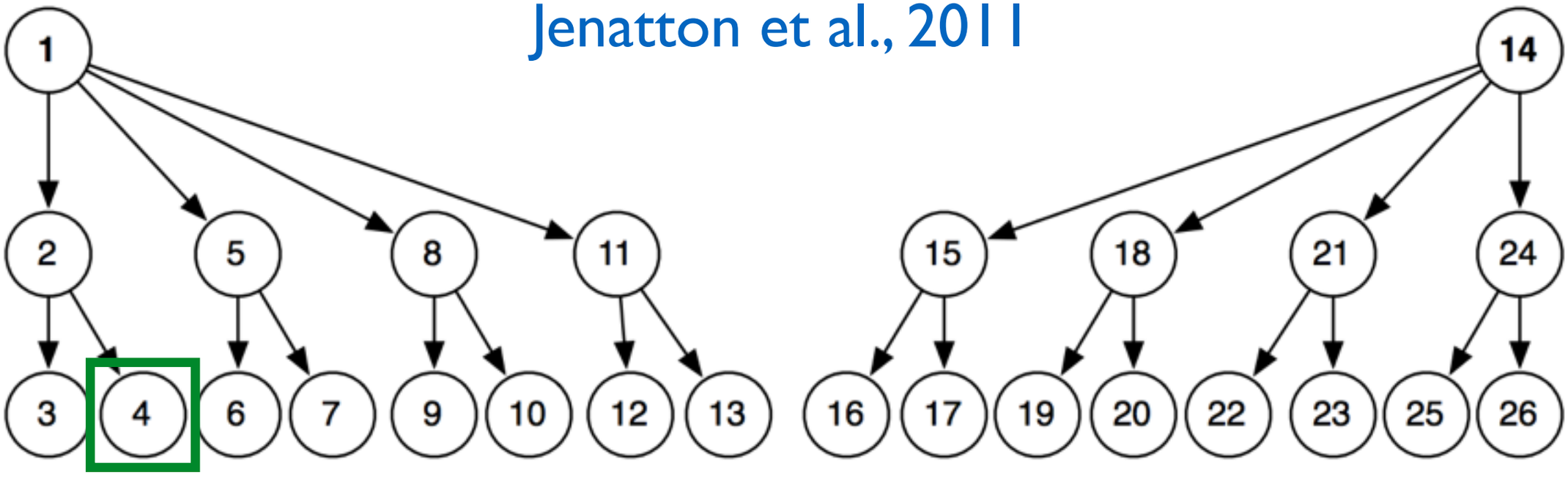
clab

Tree regularizer

Recursively apply group lasso from root to leaves

Each group is a node and all its descendants

Jenatton et al., 2011



$$\Omega(\mathbf{a}_v) = \sum_i \|\langle a_{v,i}, \mathbf{a}_{v,\text{Descendants(i)}}\rangle\|_2$$

ARK

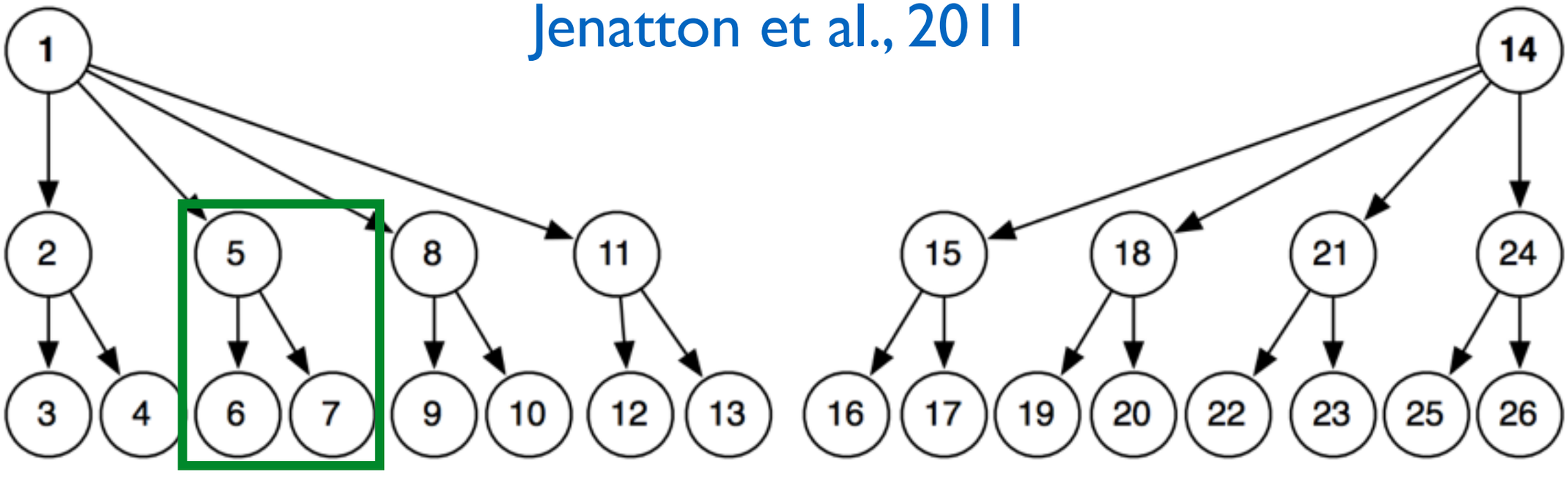
clab

Tree regularizer

Recursively apply group lasso from root to leaves

Each group is a node and all its descendants

Jenatton et al., 2011



$$\Omega(\mathbf{a}_v) = \sum_i \|\langle a_{v,i}, \mathbf{a}_{v,\text{Descendants(i)}}\rangle\|_2$$

Learning

- Optimization problem

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}} \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_2^2 + \lambda\Omega(\mathbf{A})$$

- For learning word representations, $\mathbf{X}$ is a huge matrix
- We have billions of parameters to estimate
- If the input matrix is not too big, a popular method is the online dictionary learning algorithm of **Mairal et al., 2010**

clab

Learning

- Optimization problem

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}} \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_2^2 + \lambda\Omega(\mathbf{A})$$

- For learning word representations, $\mathbf{X}$ is a huge matrix
- We have billions of parameters to estimate
- Rewrite

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}} \sum_{c,v}(x_{c,v}-\mathbf{d}_c\mathbf{a}_v)^2 + \lambda\sum_v \Omega(\mathbf{a}_v)$$

clab

Learning

- Optimization problem

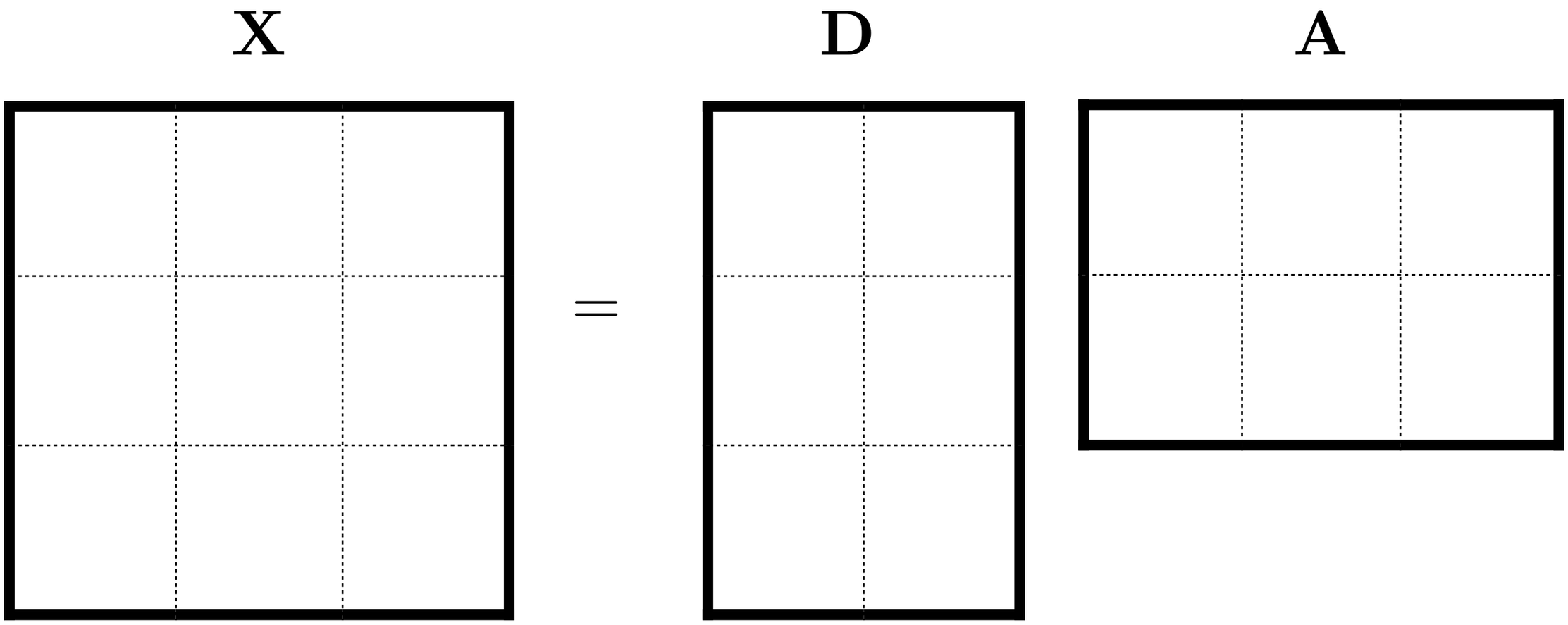
$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}} \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_2^2 + \lambda\Omega(\mathbf{A})$$

- For learning word representations, $\mathbf{X}$ is a huge matrix
- We have billions of parameters to estimate
- Rewrite

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}} \sum_{c,v}(x_{c,v}-\mathbf{d}_c\mathbf{a}_v)^2 + \lambda\sum_v \Omega(\mathbf{a}_v)$$

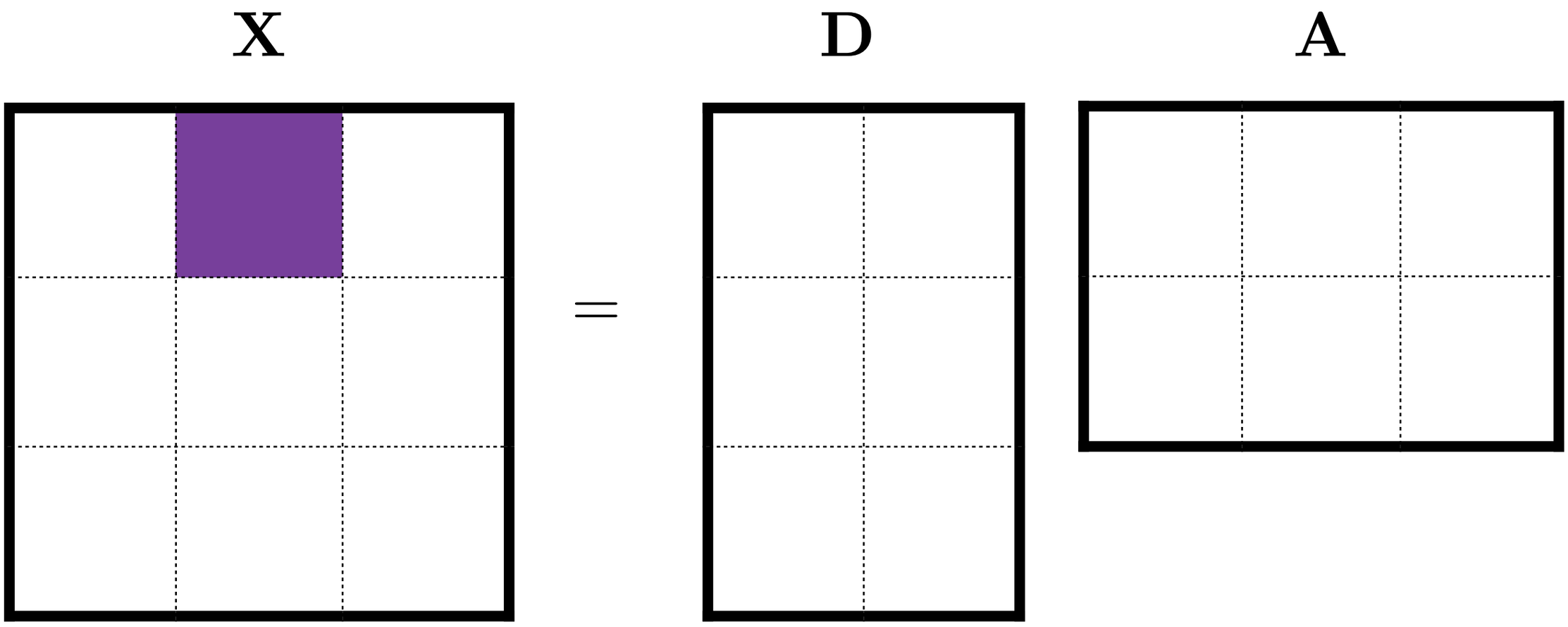
$$\min_{\mathbf{D},\mathbf{A}} \sum_{c,v}(x_{c,v}-\mathbf{d}_c\mathbf{a}_v)^2 + \lambda_1\Omega(\mathbf{a}_v) + \lambda_2\|\mathbf{d}_m\|_2^2$$

Learning



$$\min_{\mathbf{D},\mathbf{A}} \sum_{c,v} (x_{c,v} - \mathbf{d}_c \mathbf{a}_v)^2 + \lambda_1 \Omega(\mathbf{a}_v) + \lambda_2 \|\mathbf{d}_m\|_2^2$$

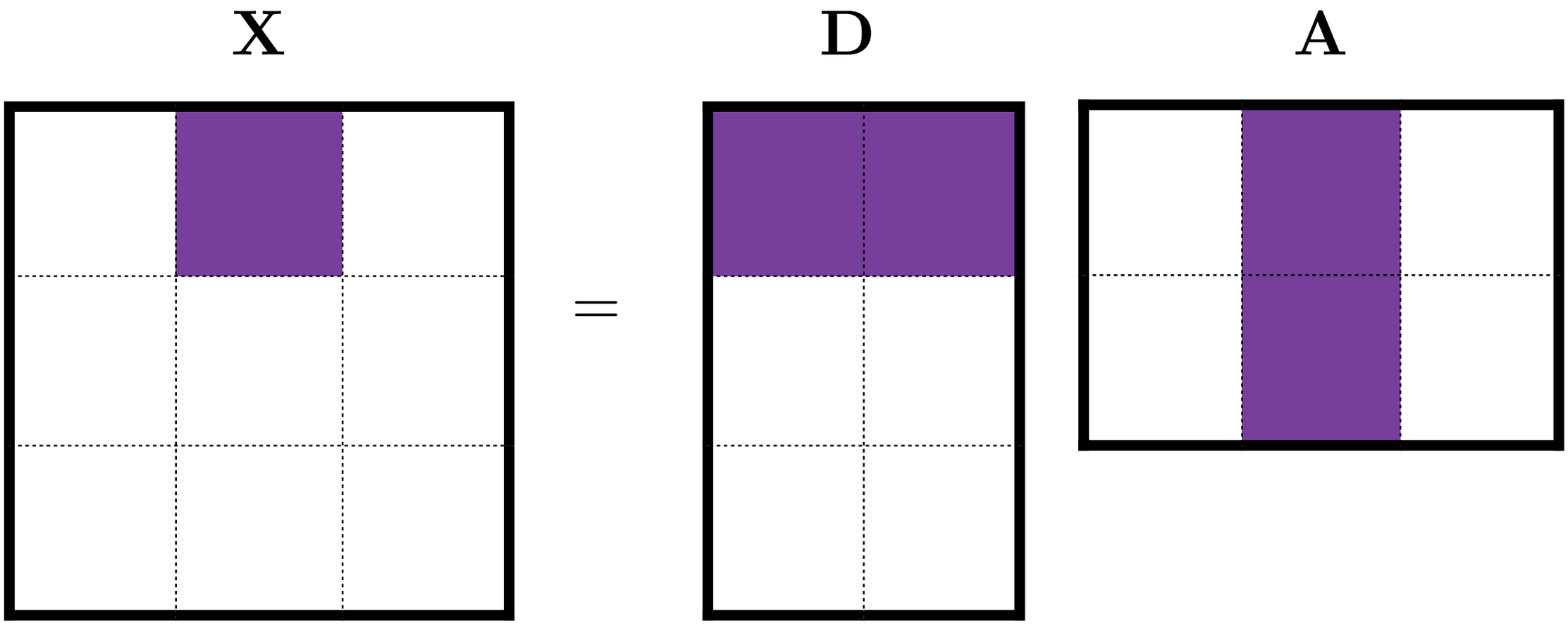
Learning



Sample an element from the input matrix

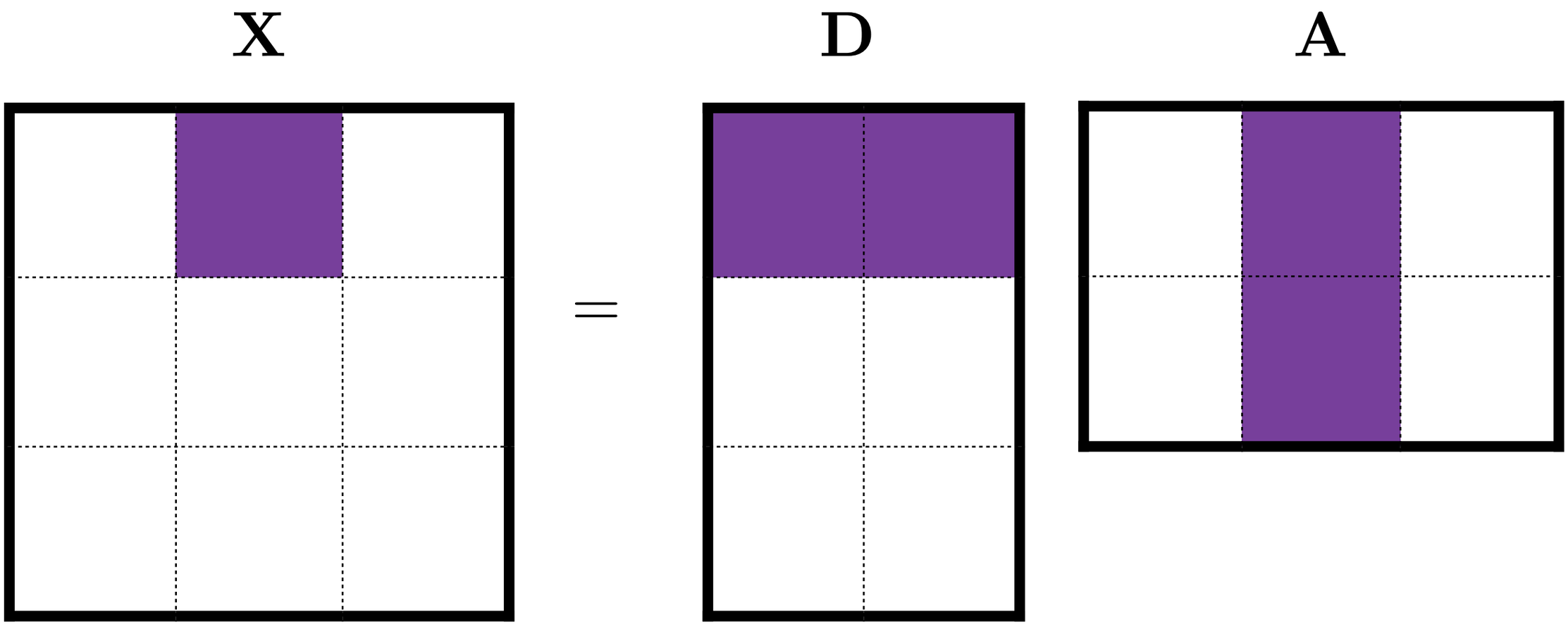
$$\min_{\mathbf{D},\mathbf{A}} \sum_{c,v} (x_{c,v} - \mathbf{d}_c \mathbf{a}_v)^2 + \lambda_1 \Omega(\mathbf{a}_v) + \lambda_2 \|\mathbf{d}_m\|_2^2$$

Learning



$$\min_{\mathbf{D},\mathbf{A}} \sum_{c,v} (x_{c,v} - \mathbf{d}_c \mathbf{a}_v)^2 + \lambda_1 \Omega(\mathbf{a}_v) + \lambda_2 \|\mathbf{d}_m\|_2^2$$

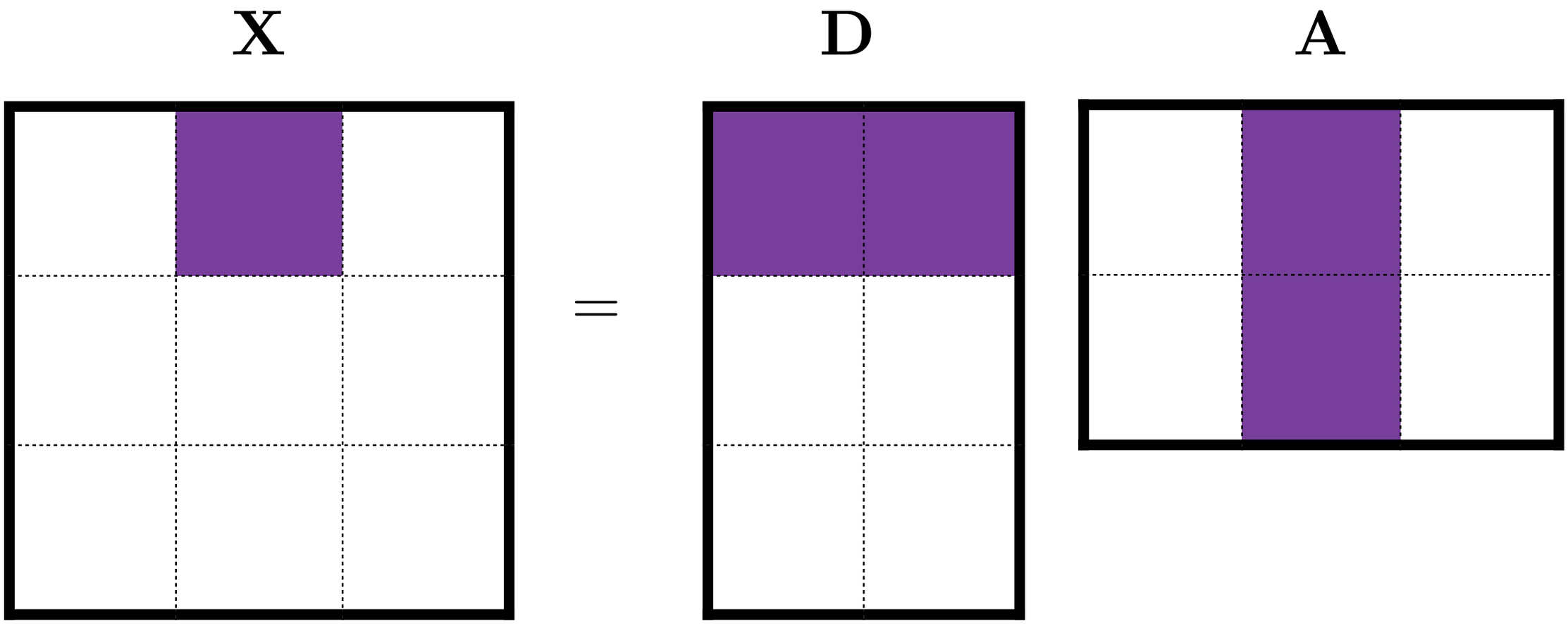
Learning



Take a gradient step and update $\mathbf{D}$

$$\min_{\mathbf{D},\mathbf{A}} \sum_{c,v} (x_{c,v} - \mathbf{d}_c \mathbf{a}_v)^2 + \lambda_1 \Omega(\mathbf{a}_v) + \lambda_2 \|\mathbf{d}_m\|_2^2$$

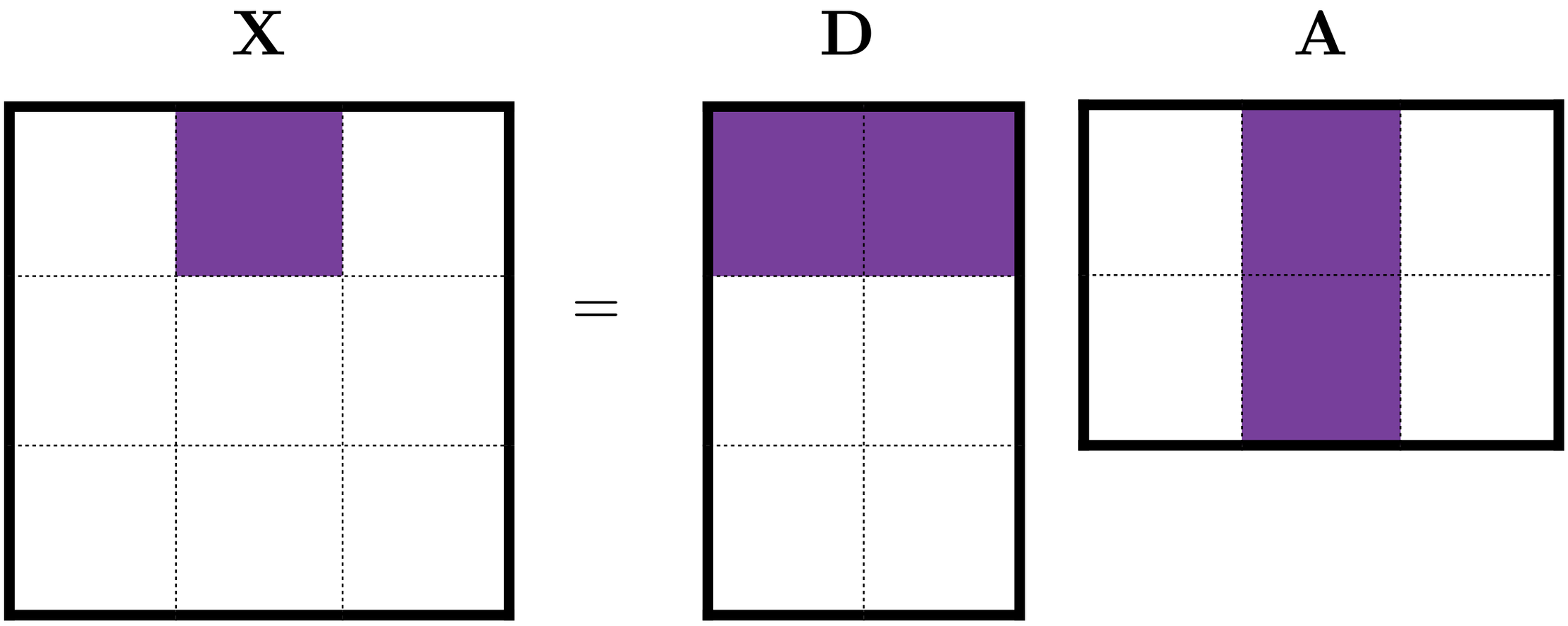
Learning



Take a gradient step and update $\mathbf{A}$

$$\min_{\mathbf{D},\mathbf{A}} \sum_{c,v} (x_{c,v} - \mathbf{d}_c \mathbf{a}_v)^2 + \lambda_1 \Omega(\mathbf{a}_v) + \lambda_2 \|\mathbf{d}_m\|_2^2$$

Learning

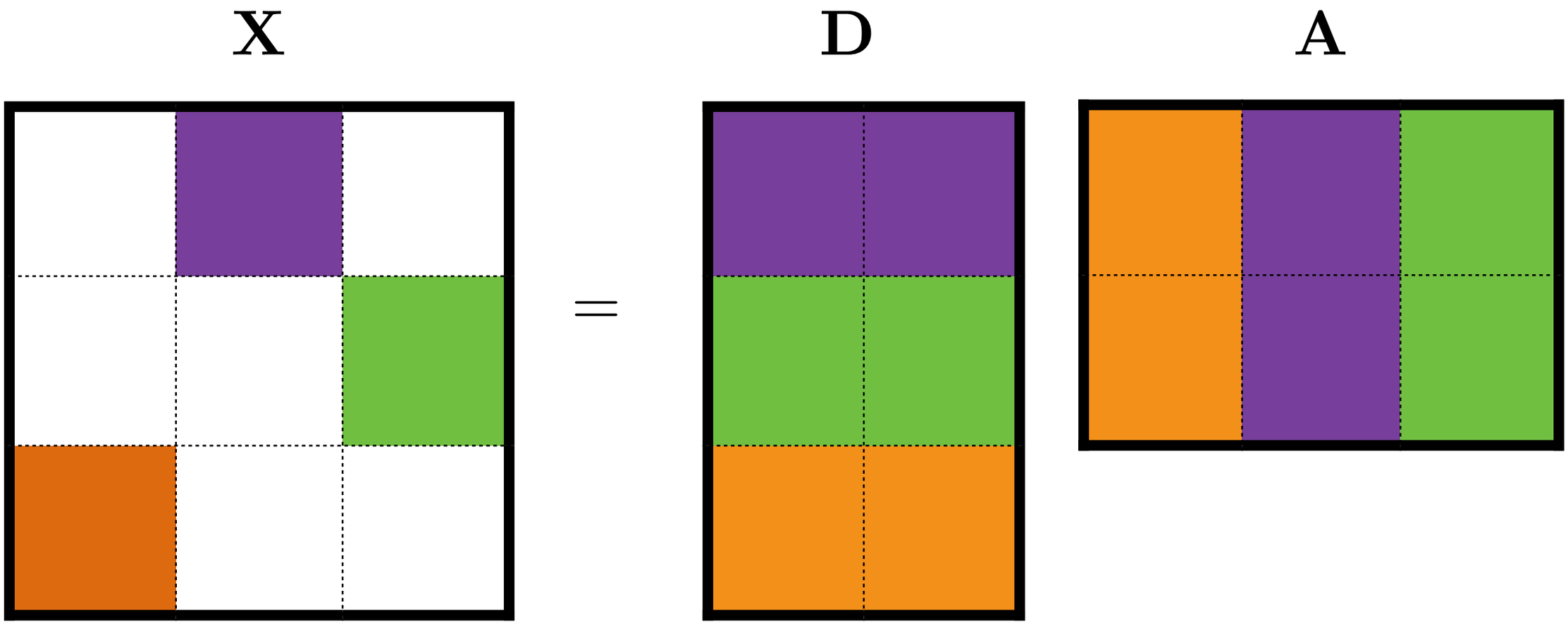


Apply proximal operators associated with the tree regularizer

Jenatton et al., 2011

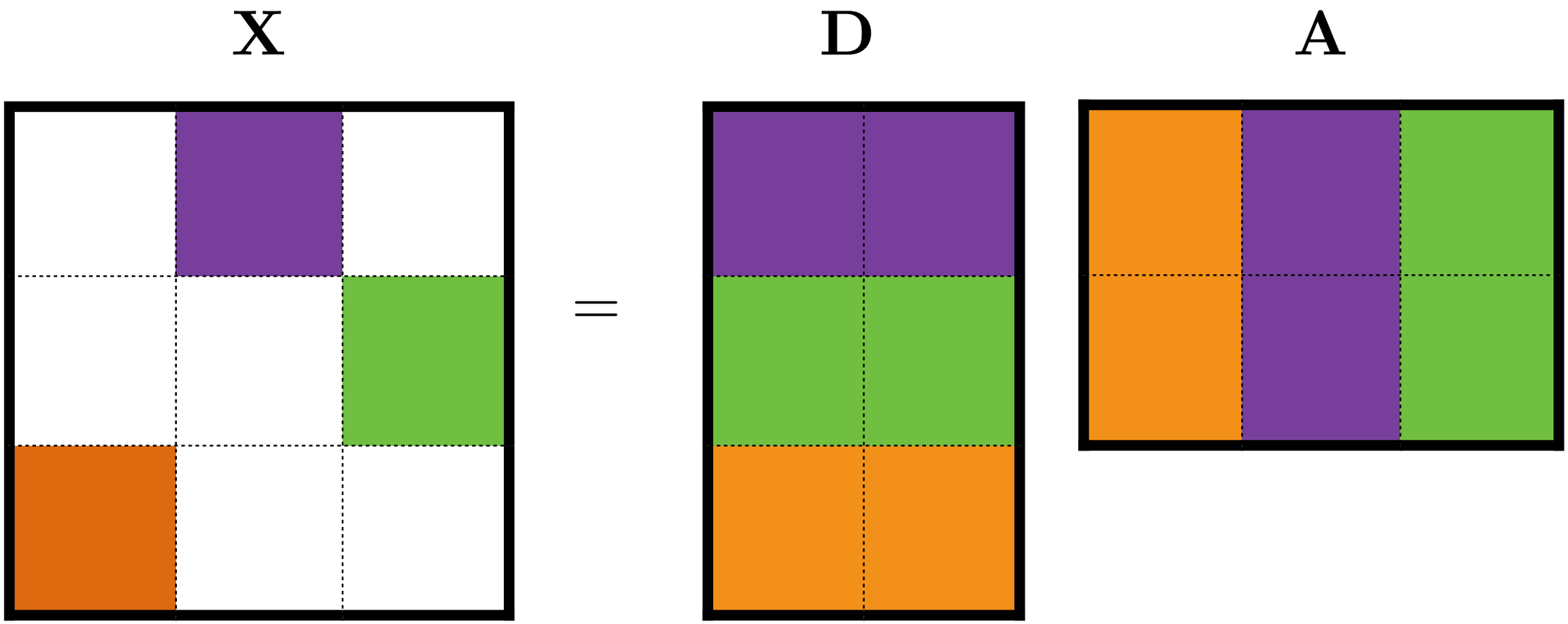
$$\min_{\mathbf{D},\mathbf{A}} \sum_{c,v} (x_{c,v} - \mathbf{d}_c \mathbf{a}_v)^2 + \boxed{\lambda_1 \Omega(\mathbf{a}_v)} + \lambda_2 \|\mathbf{d}_m\|_2^2$$

Learning



Parallelize by sampling more elements

Learning



Converge to a stationary point (non-convex problem)

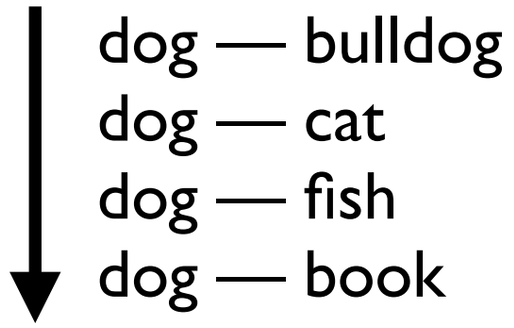
Experiments

- WMT-2011 English news corpus + Wikipedia as our training data
- Baselines
  - Principal Component Analysis (**Turney and Pantel, 2010**)
  - Recursive Neural Network (**Mikolov et al., 2010**)
  - Log Bilinear Model (**Mnih and Teh, 2012**)
  - Continuous Bag-of-Words (**Mikolov et al., 2013**)
  - Skip Gram (**Mikolov et al., 2013**)
  - Glove (**Pennington et al., 2014**)

clab

Experiments

- Word similarity ranking



#dims	PCA	Skip Gram	Glove	Sparse Coding	Our method
52	0.39	0.49	0.43	0.49	**0.52**
520	0.50	0.58	0.51	0.58	**0.66**

Spearman's correlation coefficient, higher is better

clab

Experiments

- Sentiment analysis of movie reviews (Socher et al., 2013)

#dims	PCA	Skip Gram	Glove	Sparse Coding	Our method
52	74.5	68.5	72.6	75.5	**75.8**
520	81.7	79.5	79.4	78.2	**81.9**

classification accuracy, higher is better

Experiments

- Analogies (Mikolov et al., 2013)

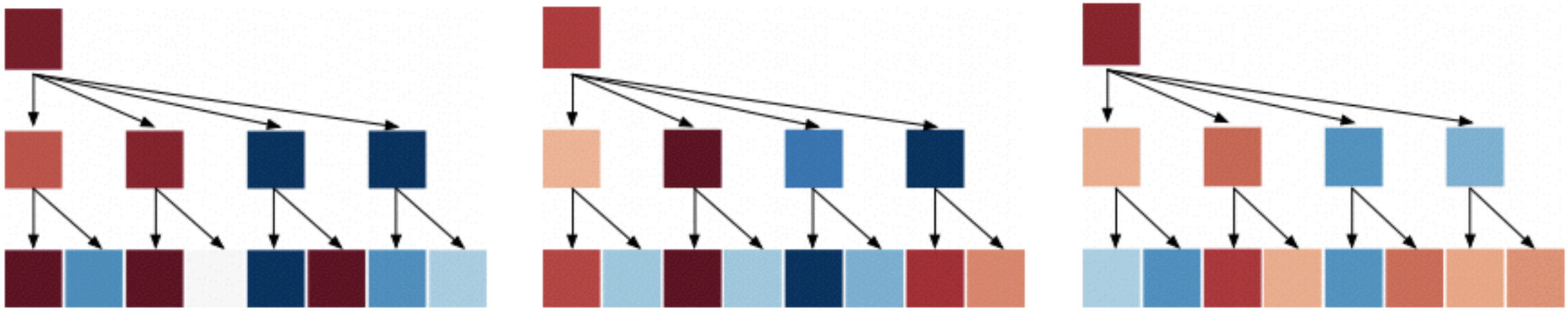
Paris : France :: London : ?
Answer: England

write : writes :: sleep : ?
Answer: sleeps

Task	CBOW	Skip Gram	Glove	Our method
Syntactic	61.4	63.6	65.56	**65.63**
Semantic	23.1	54.5	**74.4**	52.9

classification accuracy, higher is better

Tree visualizations



(a) animal (b) horse (c) elephant

Each box is a word dimension

Red indicates negative values

Blue indicates positive values

The darker the color, the more extreme the value is

Summary

- Structured sparsity to encode linguistic knowledge (hierarchical organization of word meanings) into a word embedding model

- A first step towards a more interpretable embedding dimensions

- An optimization method for large-scale sparse coding

clab

Thanks!