Language Technologies Institute

MACHINE LEARNING DEPARTMENT

Language Modeling with Power Low Rank Ensembles




Ankur Parikh *Avneesh Saluja* *Chris Dyer* *Eric Xing*

Overview

Overview

- **Model:** Framework for language modeling using ensembles of low rank matrices and tensors

- **Relations:** Includes existing *n*-gram smoothing techniques as special cases

Overview

MACHINE LEARNING DEPARTMENT

- **Model:** Framework for language modeling using ensembles of low rank matrices and tensors

- **Relations:** Includes existing *n*-gram smoothing techniques as special cases

- **Performance:** Consistently outperforms state-of-the-art Kneser Ney baselines for same context length

- **Speed:** Easily scalable since no partition function required

Outline

- Introduction
- Background on *n*-gram smoothing
- Our Approach
  - Rank
  - Power
  - Constructing the Ensemble
- Experiments

Language Modeling

MACHINE LEARNING DEPARTMENT

- Evaluate probabilities of sentences

Language Modeling

MACHINE LEARNING DEPARTMENT

- Evaluate probabilities of sentences

Linear algebra is awesome

Language Modeling

- Evaluate probabilities of sentences

Linear algebra is awesome $P(w_1, .., w_4) = 0.3648$

Language Modeling

- Evaluate probabilities of sentences

Linear algebra is awesome $P(w_1, .., w_4) = 0.3648$

Linear algebra is boring

Language Modeling

- Evaluate probabilities of sentences

Linear algebra is awesome $P(w_1, .., w_4) = 0.3648$

Linear algebra is boring $P(w_1, .., w_4) = 0.1922$

Language Modeling

- Evaluate probabilities of sentences

Linear algebra is awesome $P(w_1, .., w_4) = 0.3648$

Linear algebra is boring $P(w_1, .., w_4) = 0.1922$

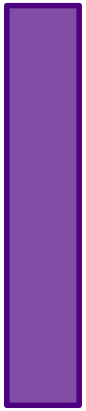
- Very useful in downstream applications such as machine translation and speech recognition.

N-grams

- Predominant approach to language modeling

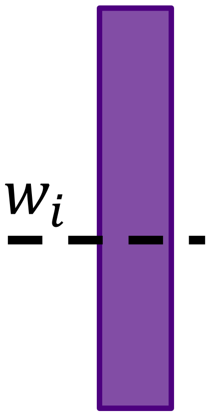
N-grams

- Predominant approach to language modeling



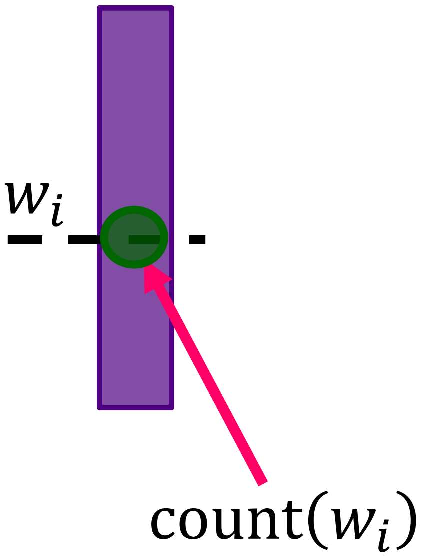
N-grams

- Predominant approach to language modeling



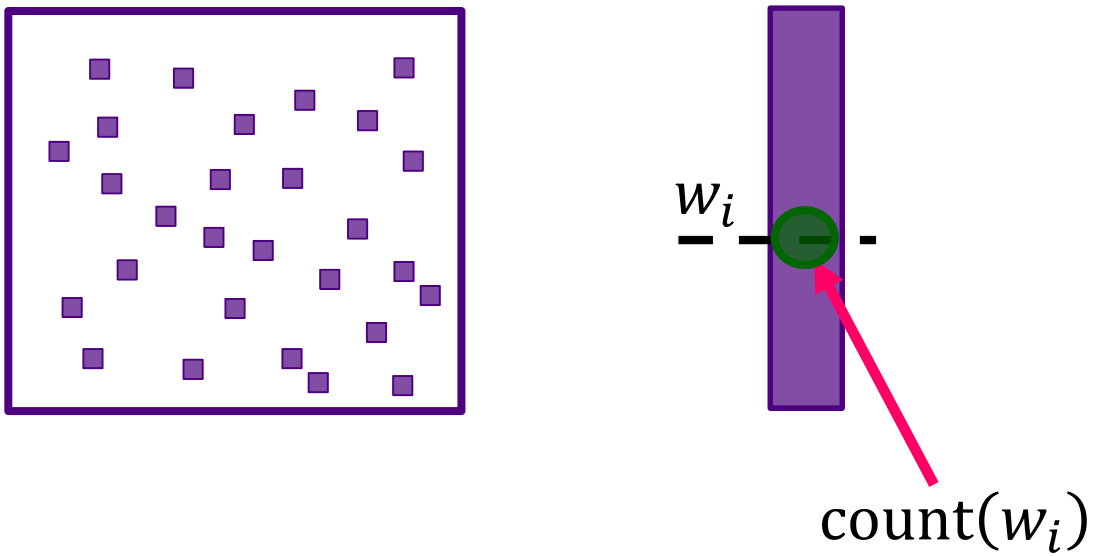
N-grams

- Predominant approach to language modeling



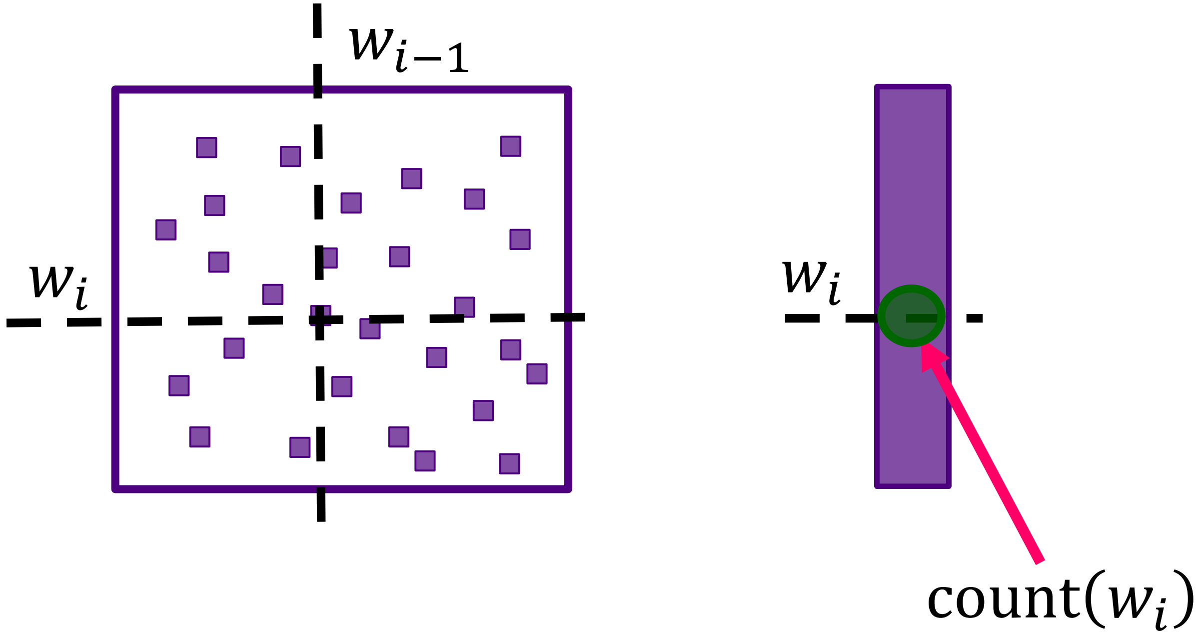
N-grams

- Predominant approach to language modeling



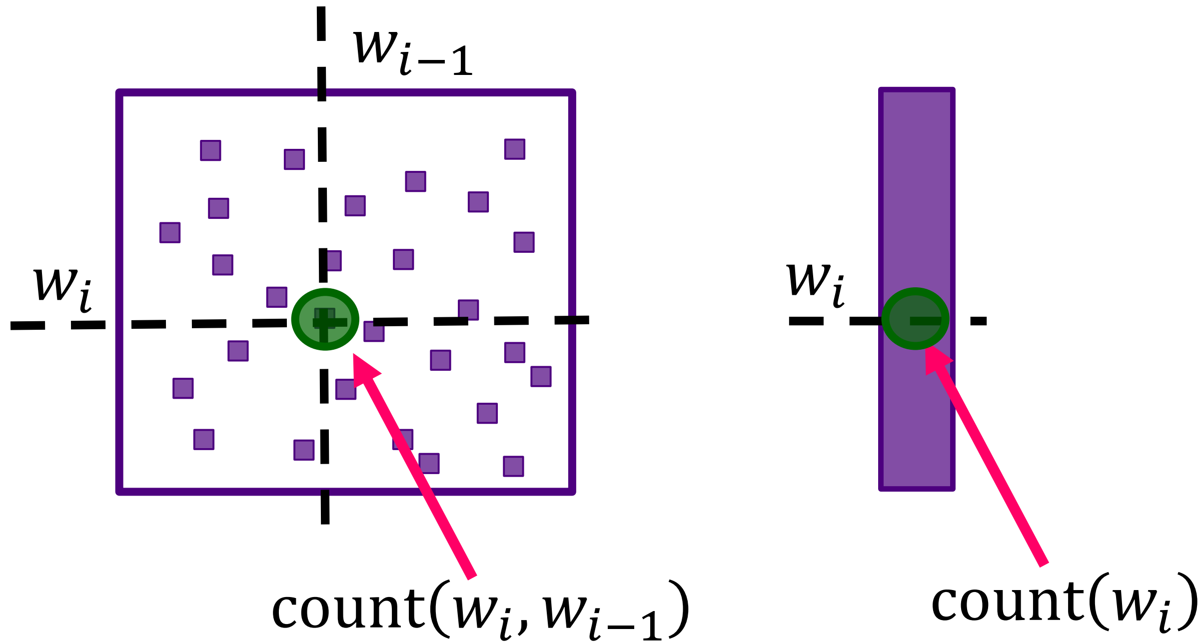
N-grams

- Predominant approach to language modeling



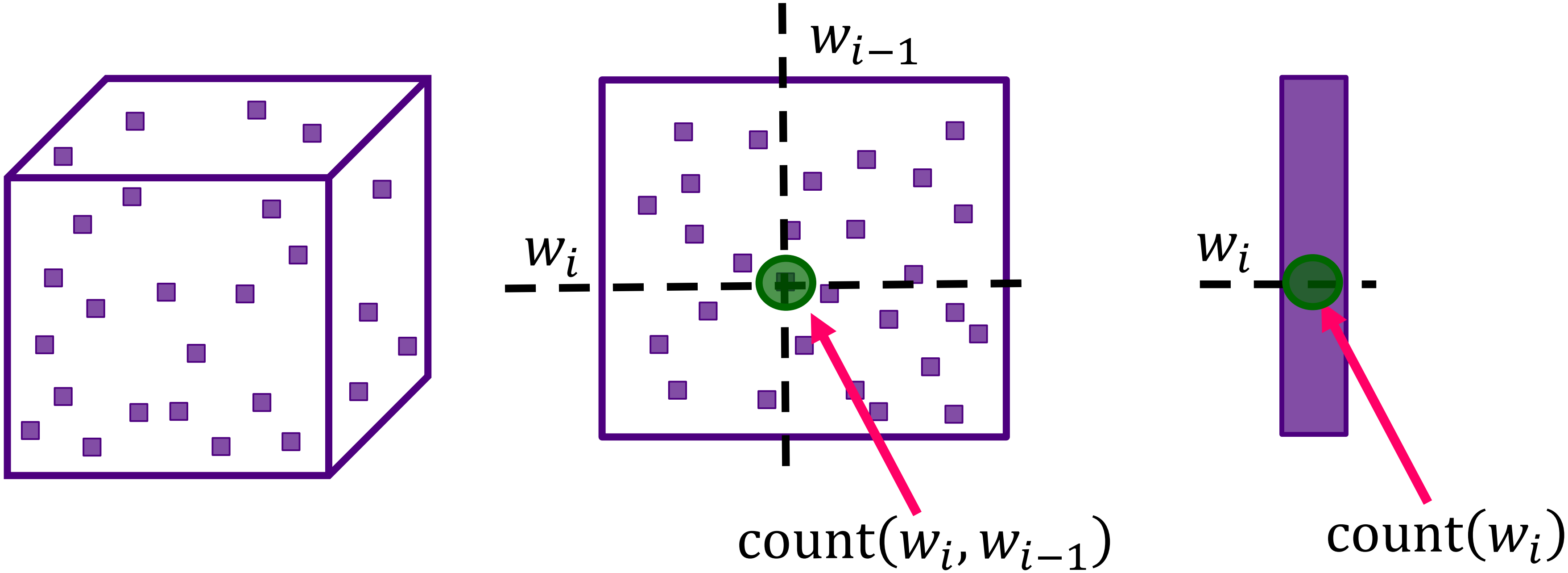
N-grams

- Predominant approach to language modeling



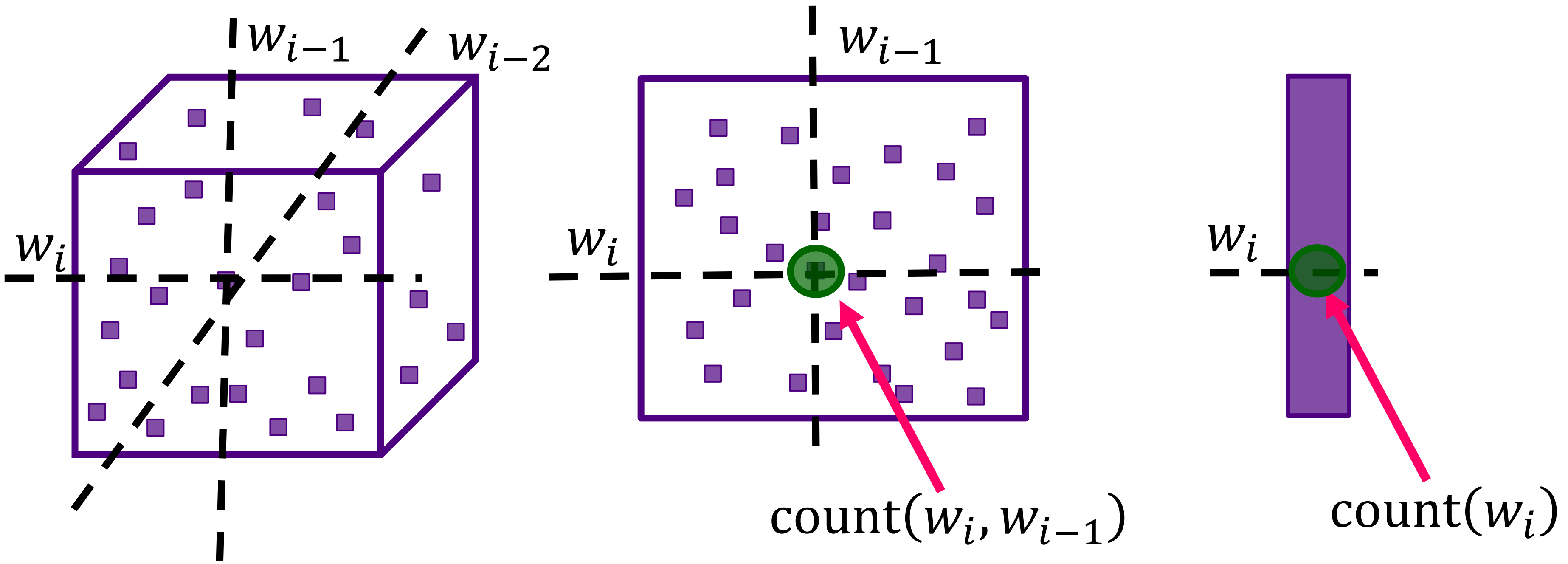
N-grams

- Predominant approach to language modeling



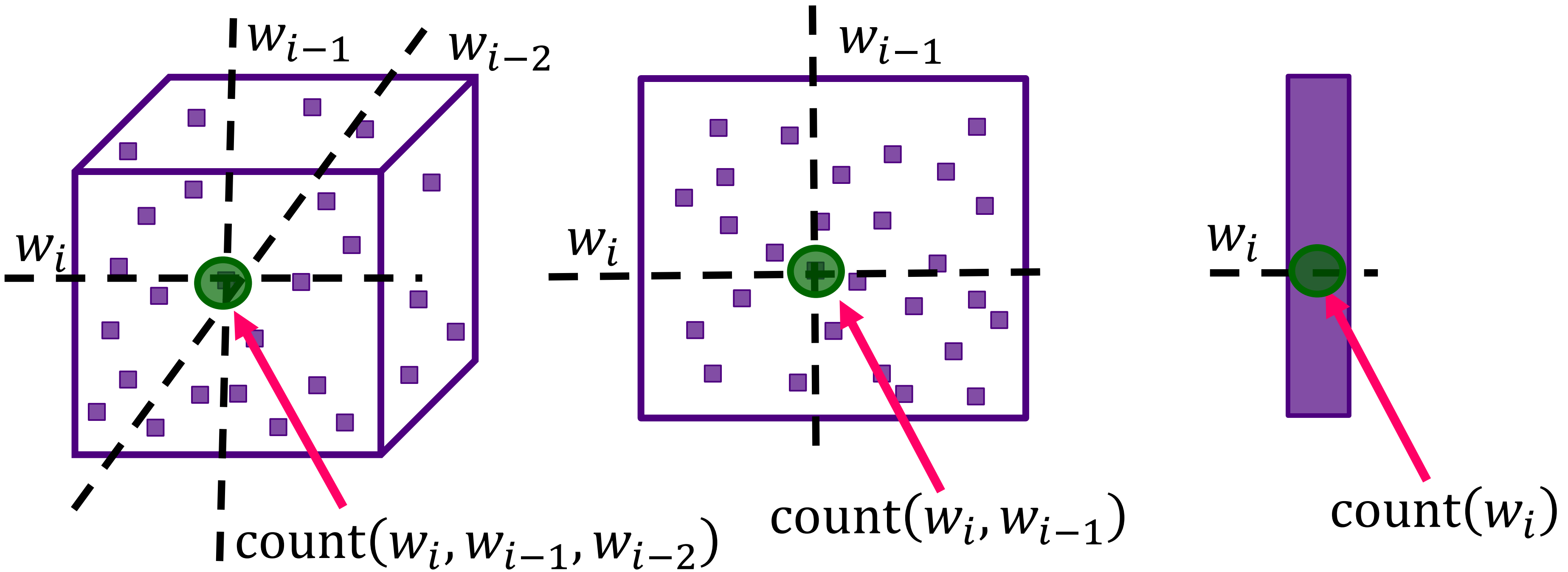
N-grams

- Predominant approach to language modeling



N-grams

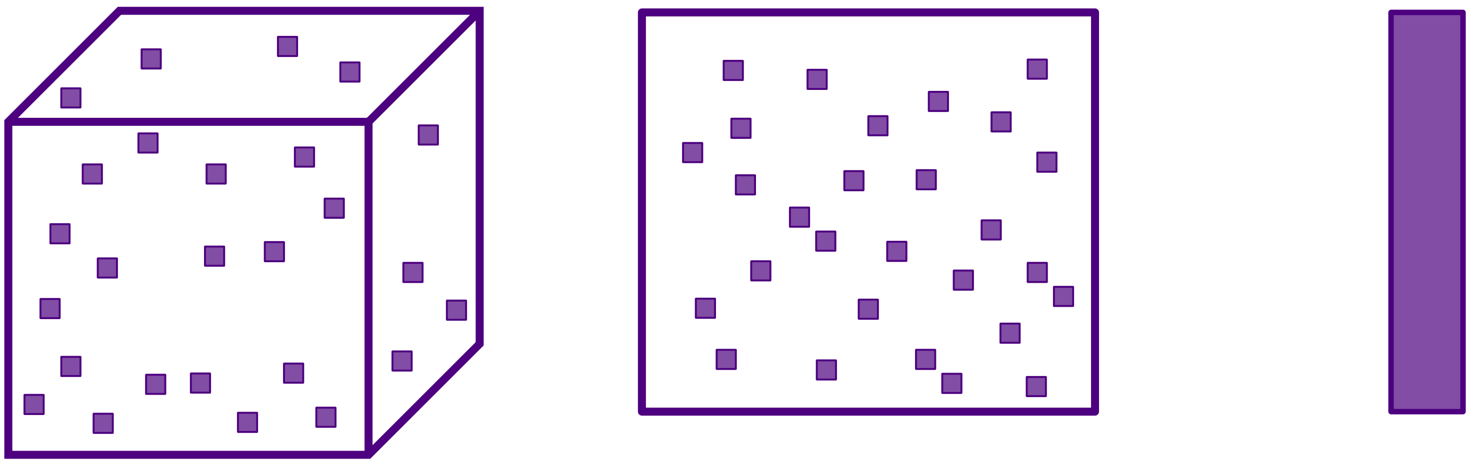
- Predominant approach to language modeling



N-gram Smoothing

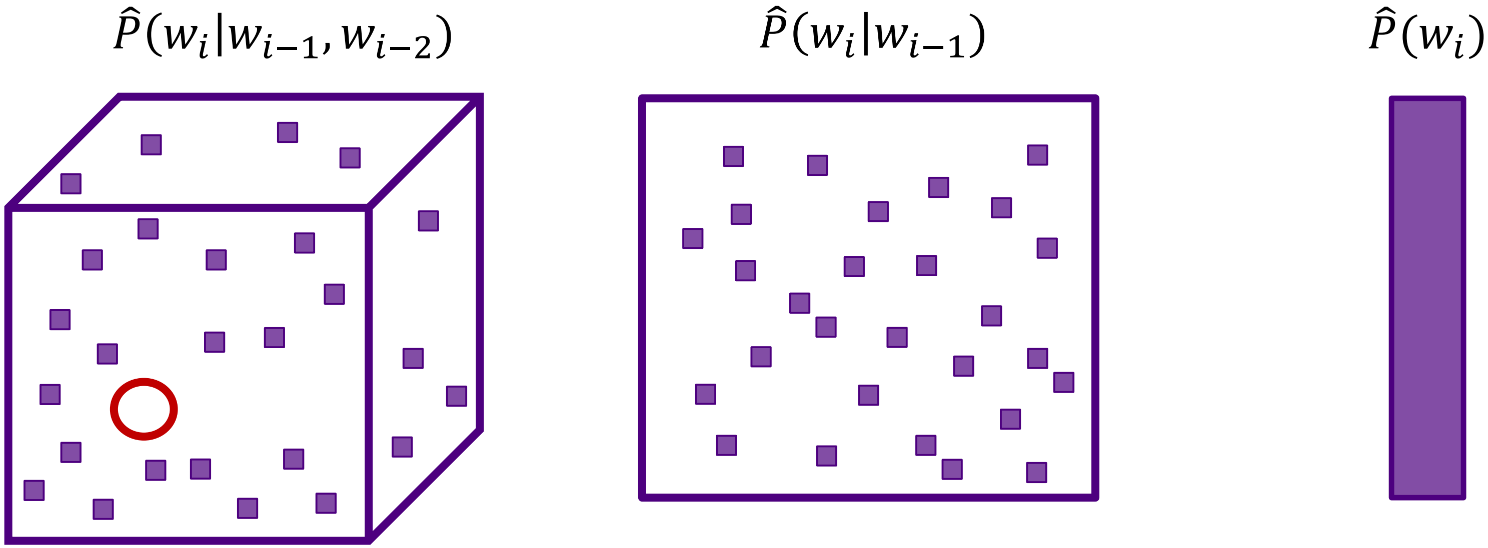
- Alleviate data sparsity problem

$\hat{P}(w_i|w_{i-1}, w_{i-2})$ $\hat{P}(w_i|w_{i-1})$ $\hat{P}(w_i)$



N-gram Smoothing

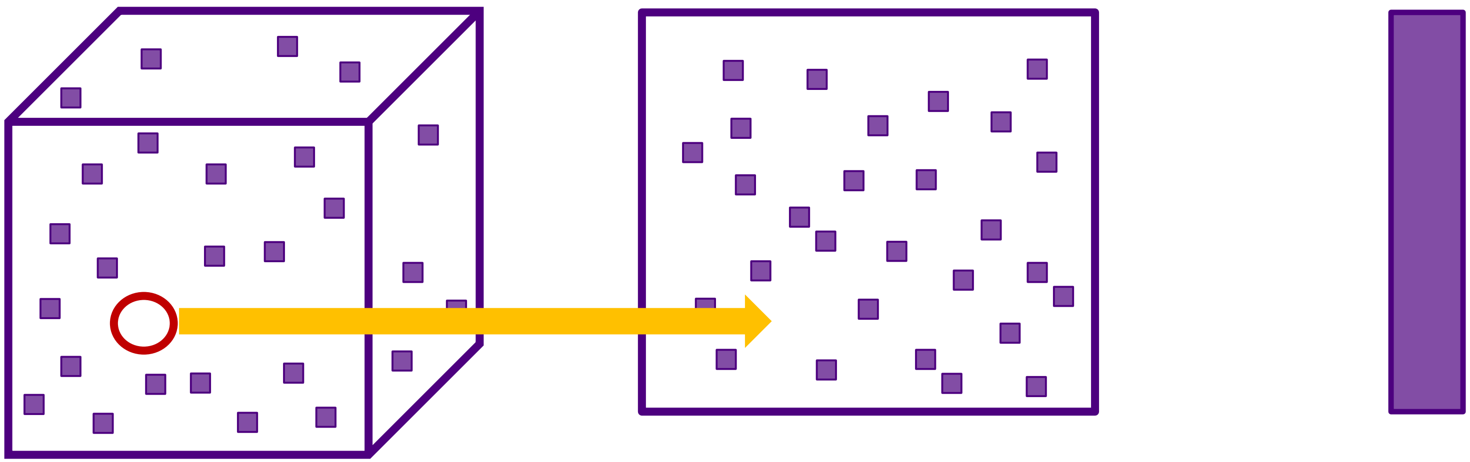
- Alleviate data sparsity problem



N-gram Smoothing

- Alleviate data sparsity problem

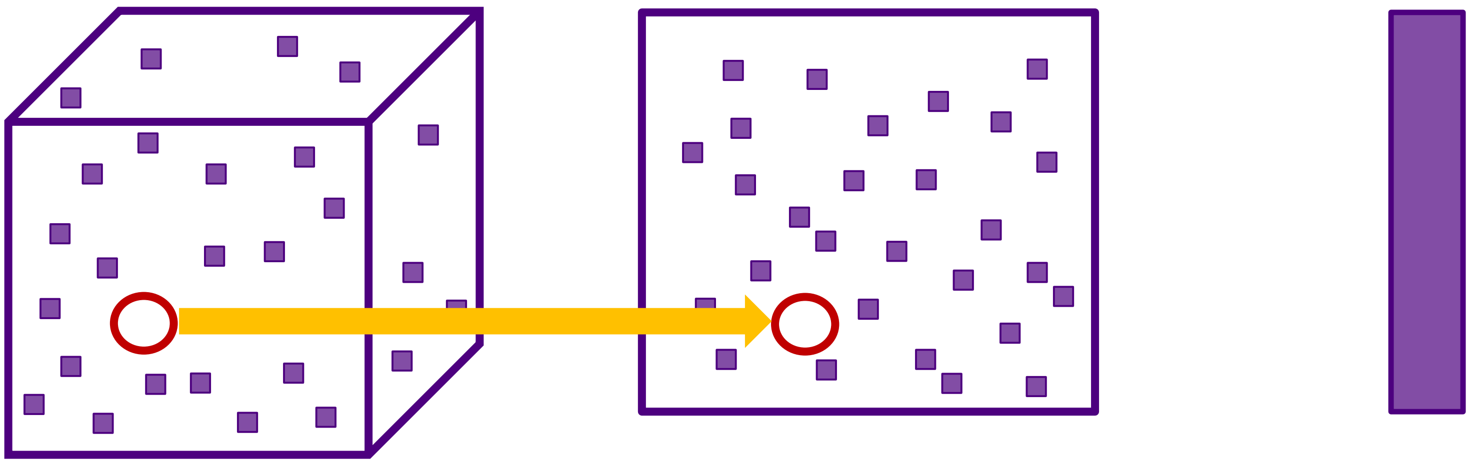
$\hat{P}(w_i|w_{i-1}, w_{i-2})$ $\hat{P}(w_i|w_{i-1})$ $\hat{P}(w_i)$



N-gram Smoothing

- Alleviate data sparsity problem

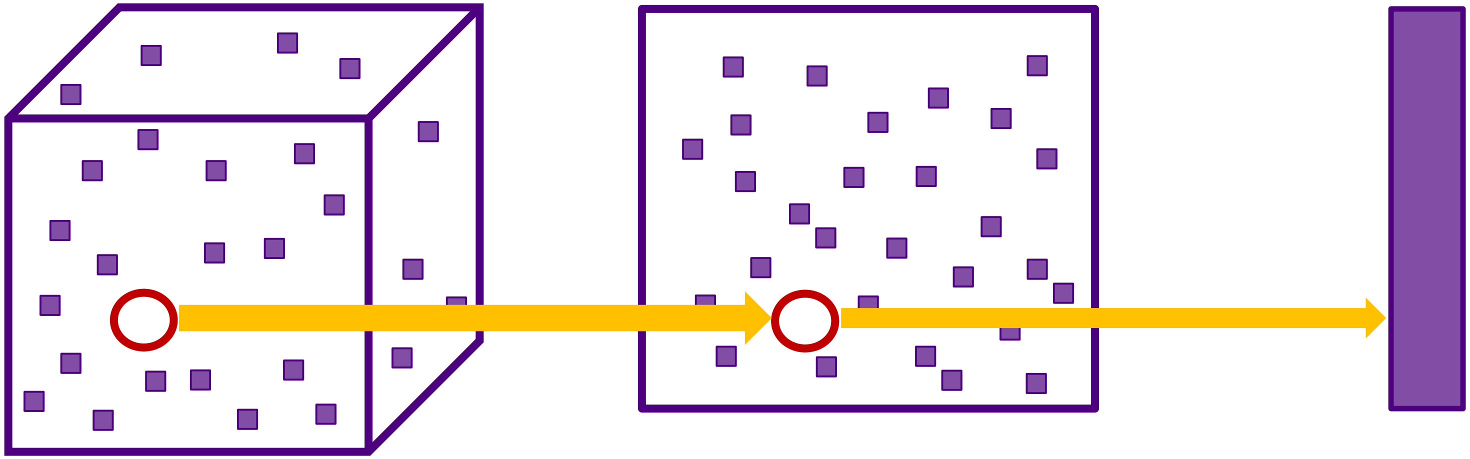
$\hat{P}(w_i|w_{i-1}, w_{i-2})$ $\hat{P}(w_i|w_{i-1})$ $\hat{P}(w_i)$



N-gram Smoothing

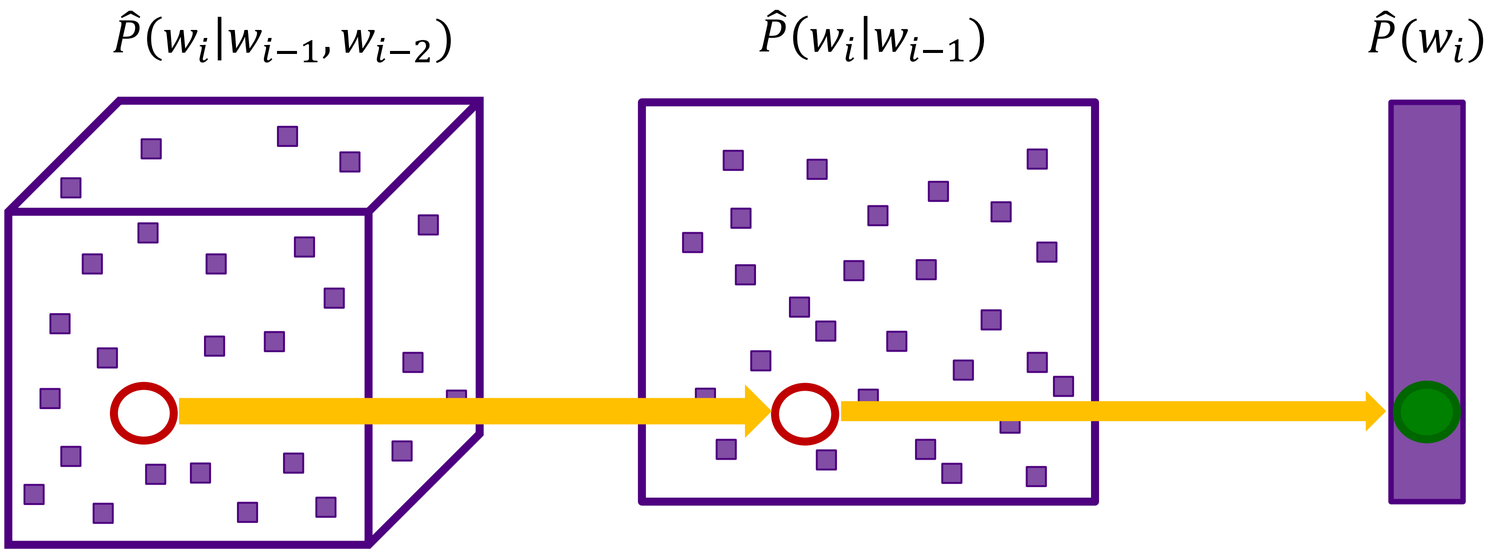
- Alleviate data sparsity problem

$\hat{P}(w_i|w_{i-1}, w_{i-2})$ $\hat{P}(w_i|w_{i-1})$ $\hat{P}(w_i)$



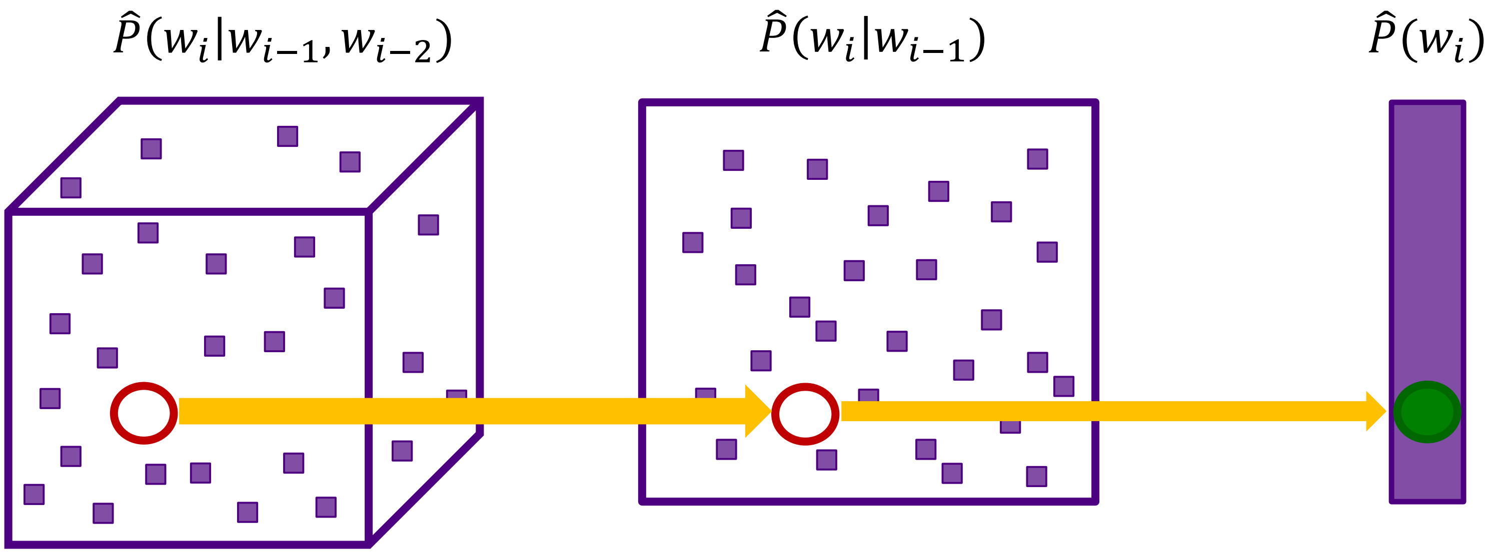
N-gram Smoothing

- Alleviate data sparsity problem



Advantages of *N*-gram Models

- "Fine-to-coarse", captures various levels of dependence

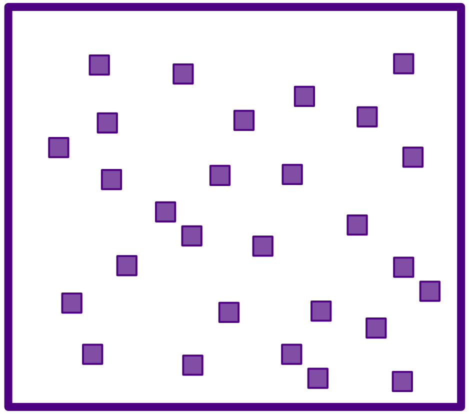


- Very fast
  - O(*N*) test complexity
  - Low context sizes sufficient

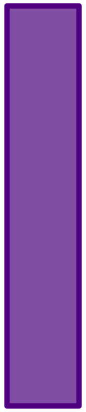
Classic Disadvantage of *N*-gram Models

- No notion of similarity between words

$\hat{P}(w_i|w_{i-1})$



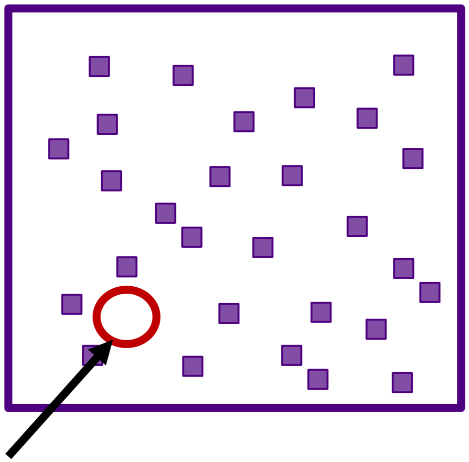
$\hat{P}(w_i)$



Classic Disadvantage of *N*-gram Models

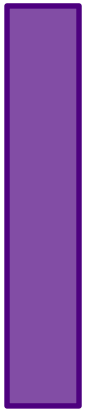
- No notion of similarity between words

$\hat{P}(w_i|w_{i-1})$



(house, decrepit)

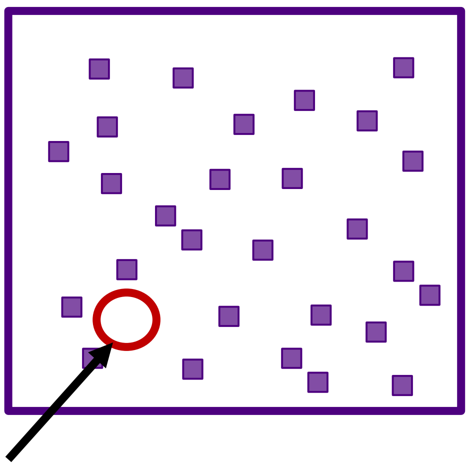
$\hat{P}(w_i)$



Classic Disadvantage of *N*-gram Models

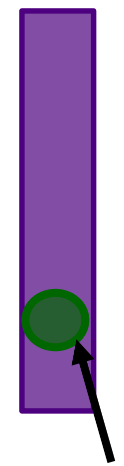
- No notion of similarity between words

$\hat{P}(w_i|w_{i-1})$



(house, decrepit)

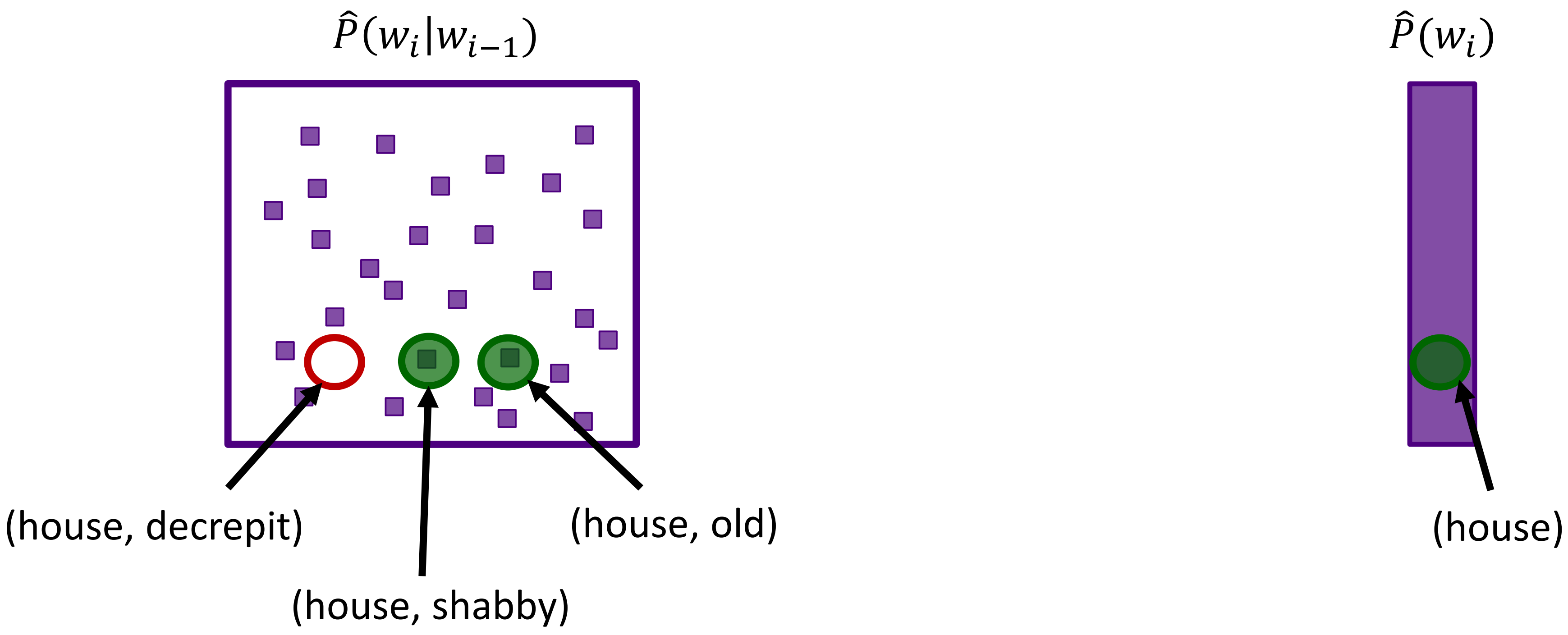
$\hat{P}(w_i)$



(house)

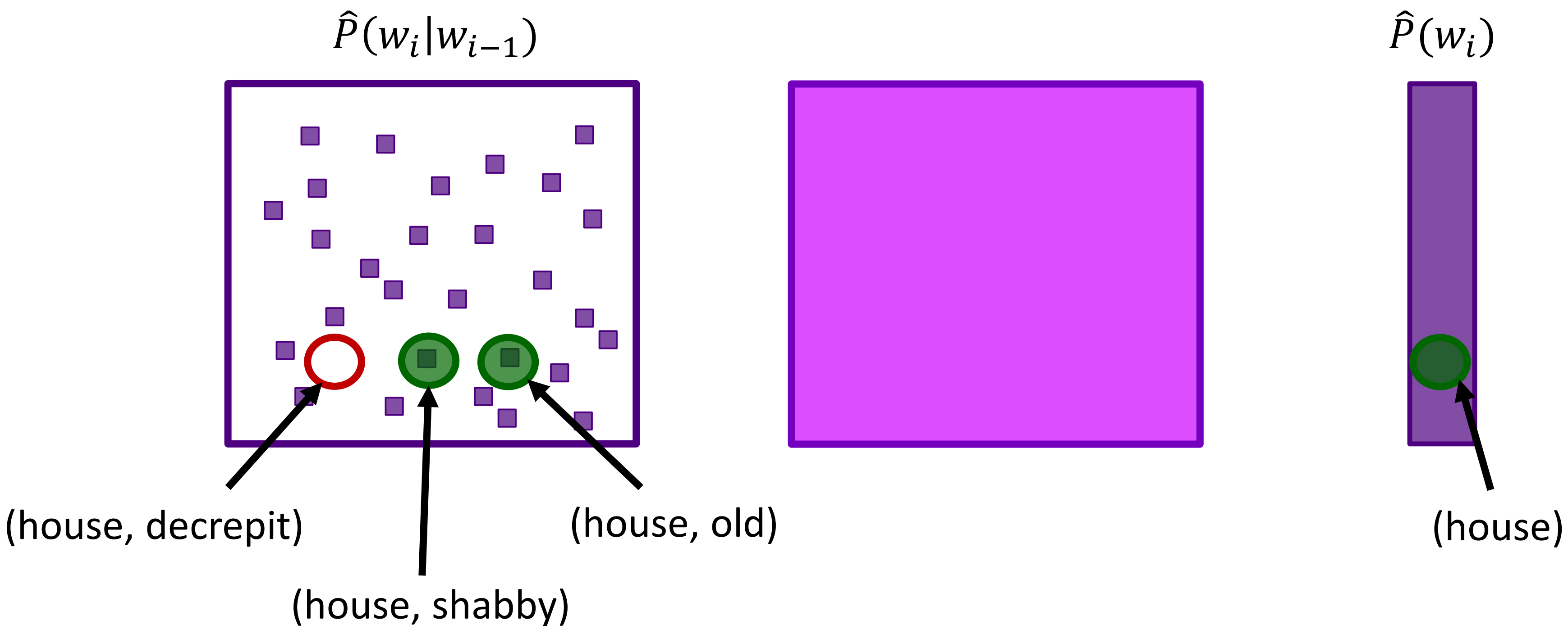
Classic Disadvantage of *N*-gram Models

- No notion of similarity between words



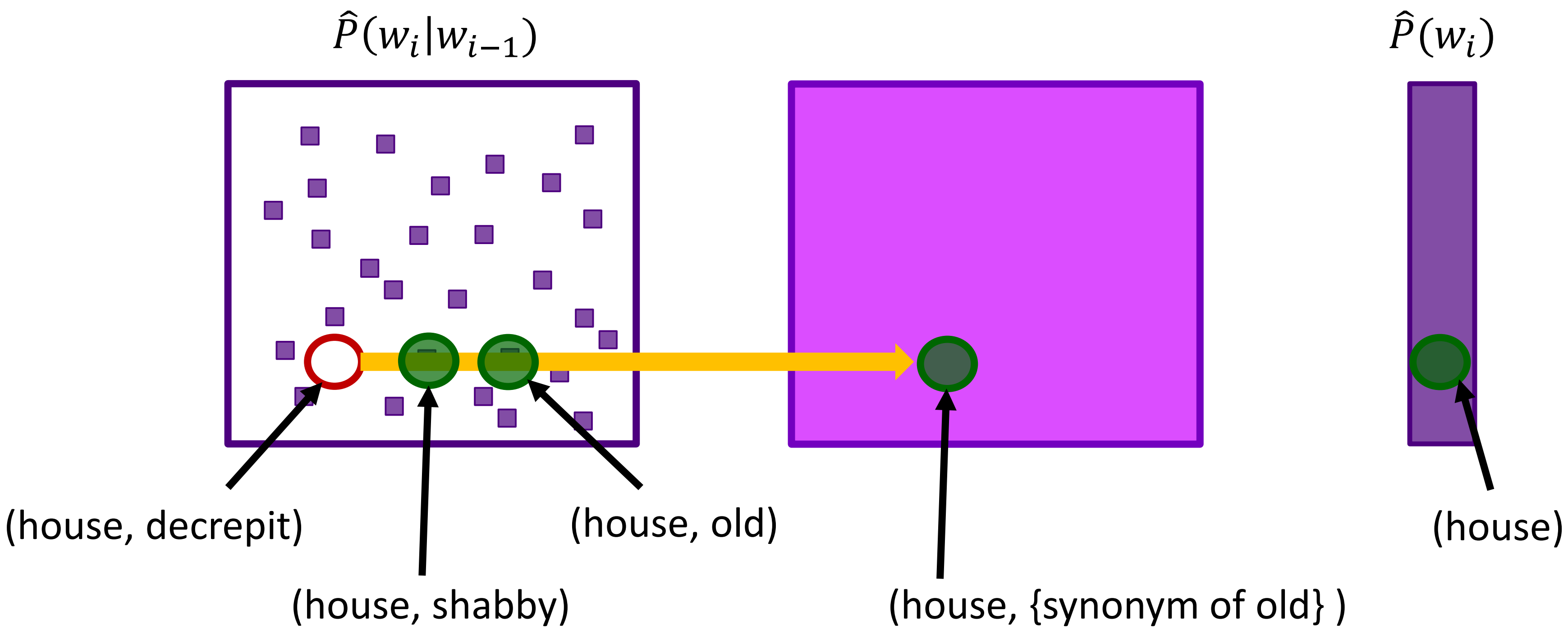
Classic Disadvantage of *N*-gram Models

- No notion of similarity between words



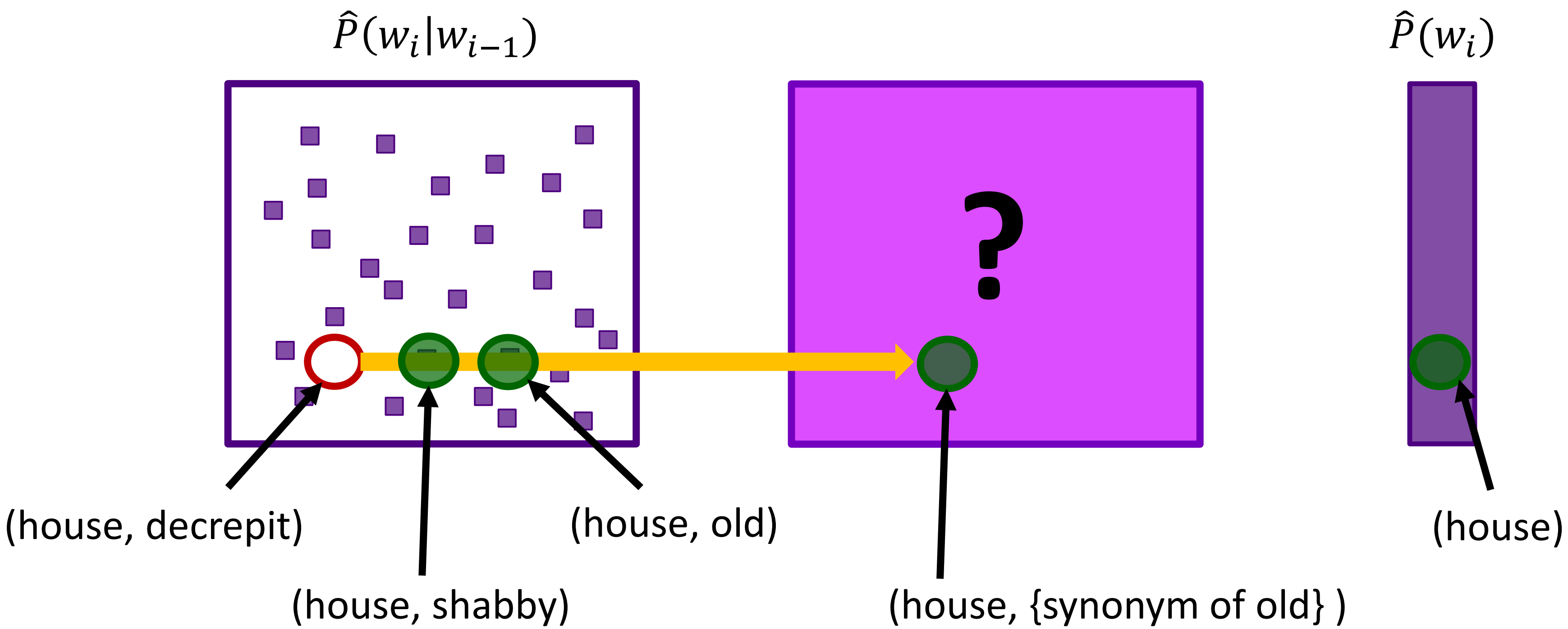
Classic Disadvantage of *N*-gram Models

- No notion of similarity between words



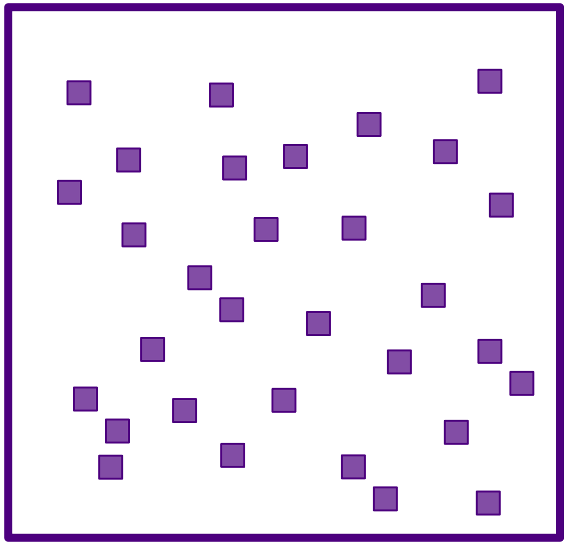
Classic Disadvantage of *N*-gram Models

- No notion of similarity between words



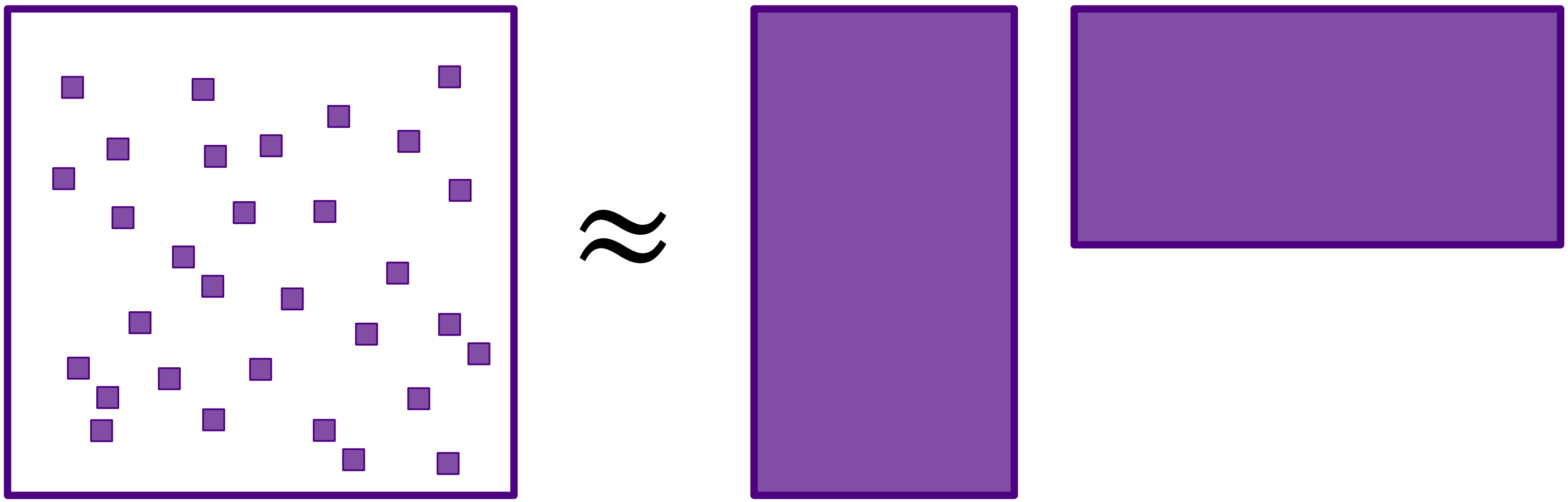
Motivation For Low Rank Methods

- Project words to lower-dimensional space



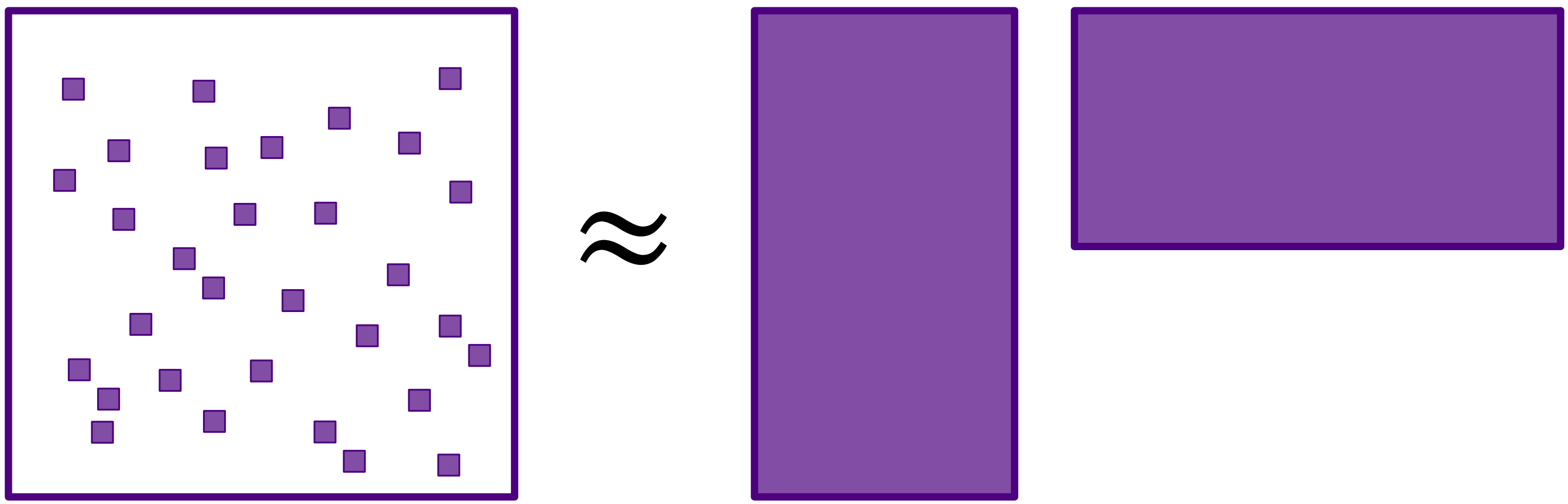
Motivation For Low Rank Methods

- Project words to lower-dimensional space



Motivation For Low Rank Methods

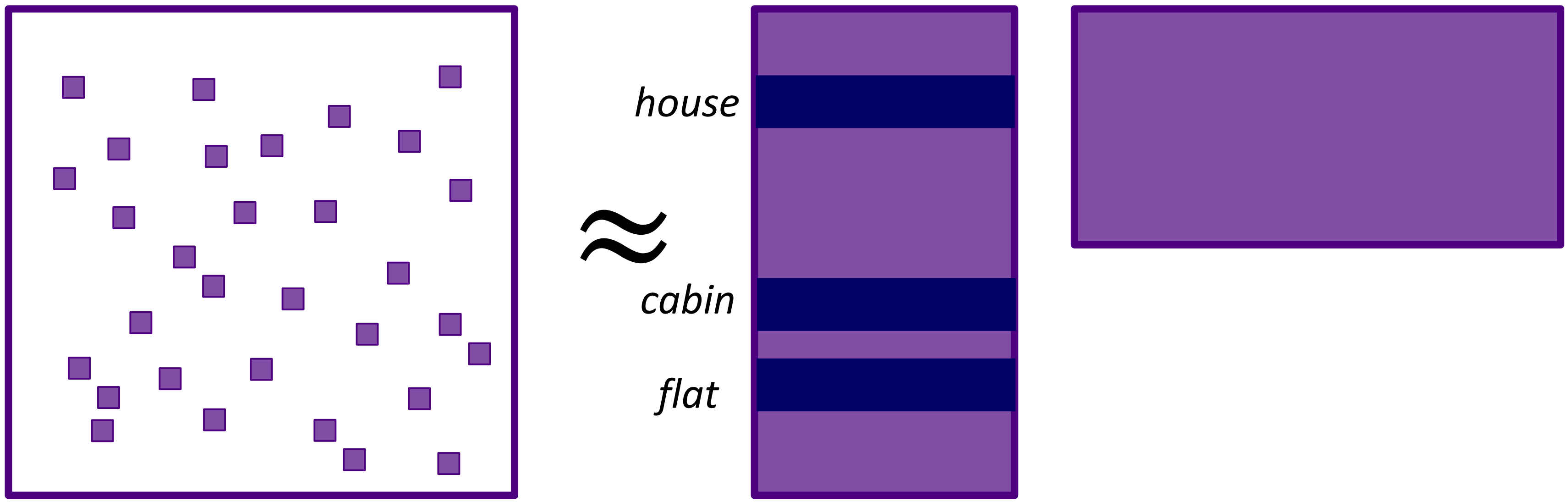
- Project words to lower-dimensional space



- Words with similar contexts will have similar projections

Motivation For Low Rank Methods

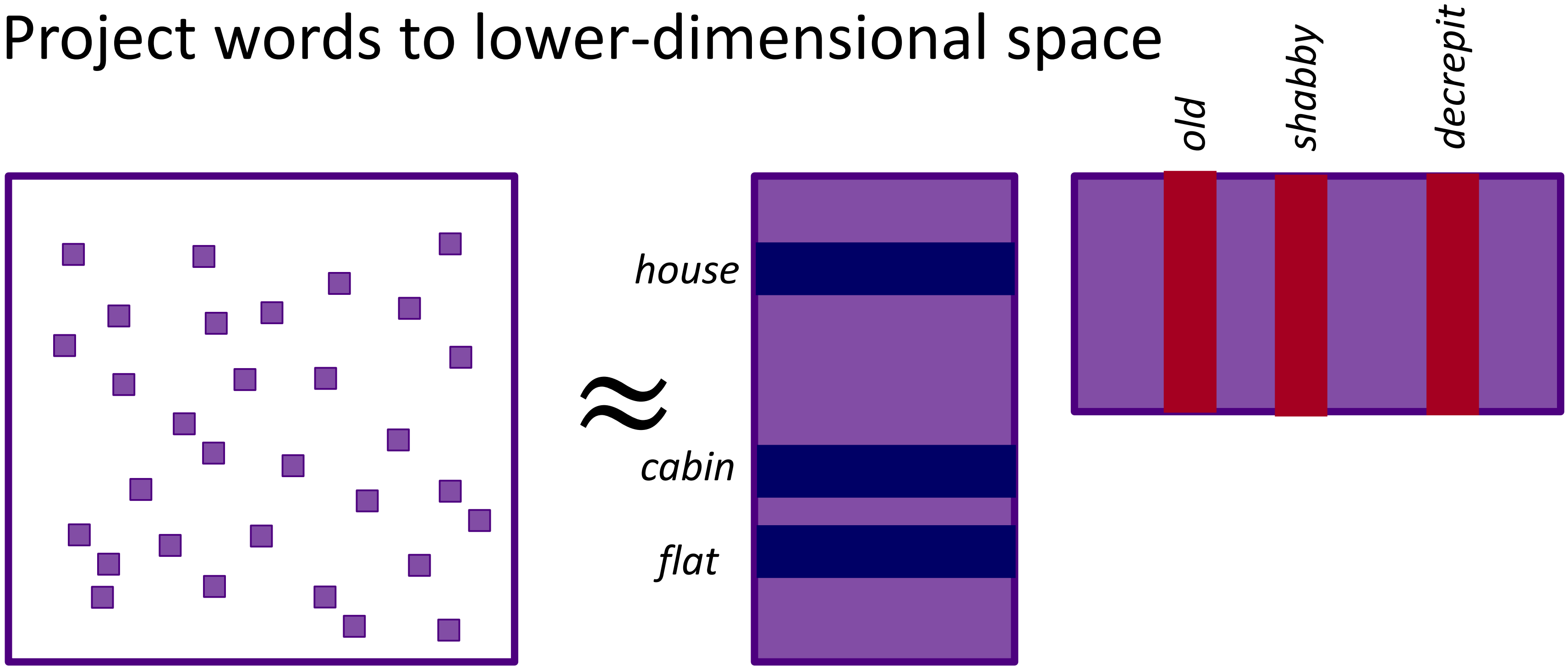
- Project words to lower-dimensional space



- Words with similar contexts will have similar projections

Motivation For Low Rank Methods

- Project words to lower-dimensional space



- Words with similar contexts will have similar projections

Low Rank Approaches

Low Rank Approaches

- Low rank approximation successful in many ML applications
  - *Collaborate filtering (Netflix)*
  - *Matrix completion*

Low Rank Approaches

- Low rank approximation successful in many ML applications
  - *Collaborate filtering (Netflix)*
  - *Matrix completion*

- These solutions have been attempted in language modeling
  - *Saul and Pereira 1997*
  - *Hutchinson et al. 2011*

Low Rank Approaches

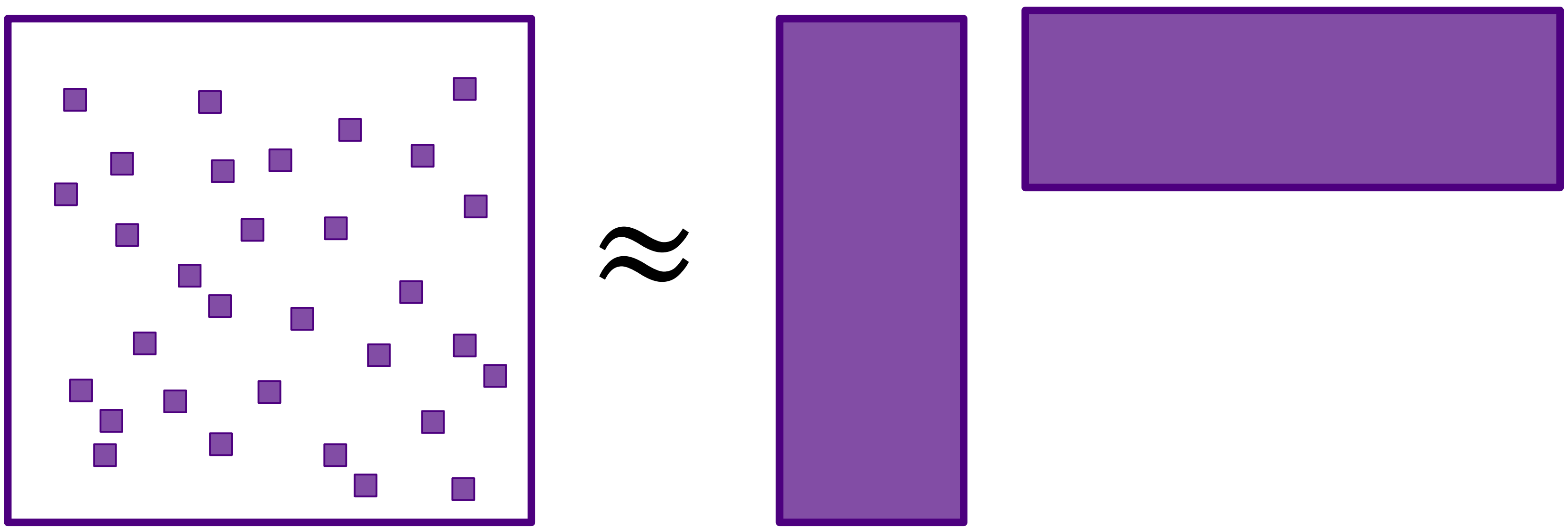
- Low rank approximation successful in many ML applications
  - *Collaborate filtering (Netflix)*
  - *Matrix completion*

- These solutions have been attempted in language modeling
  - *Saul and Pereira 1997*
  - *Hutchinson et al. 2011*

- Unfortunately, not generally competitive with Kneser Ney

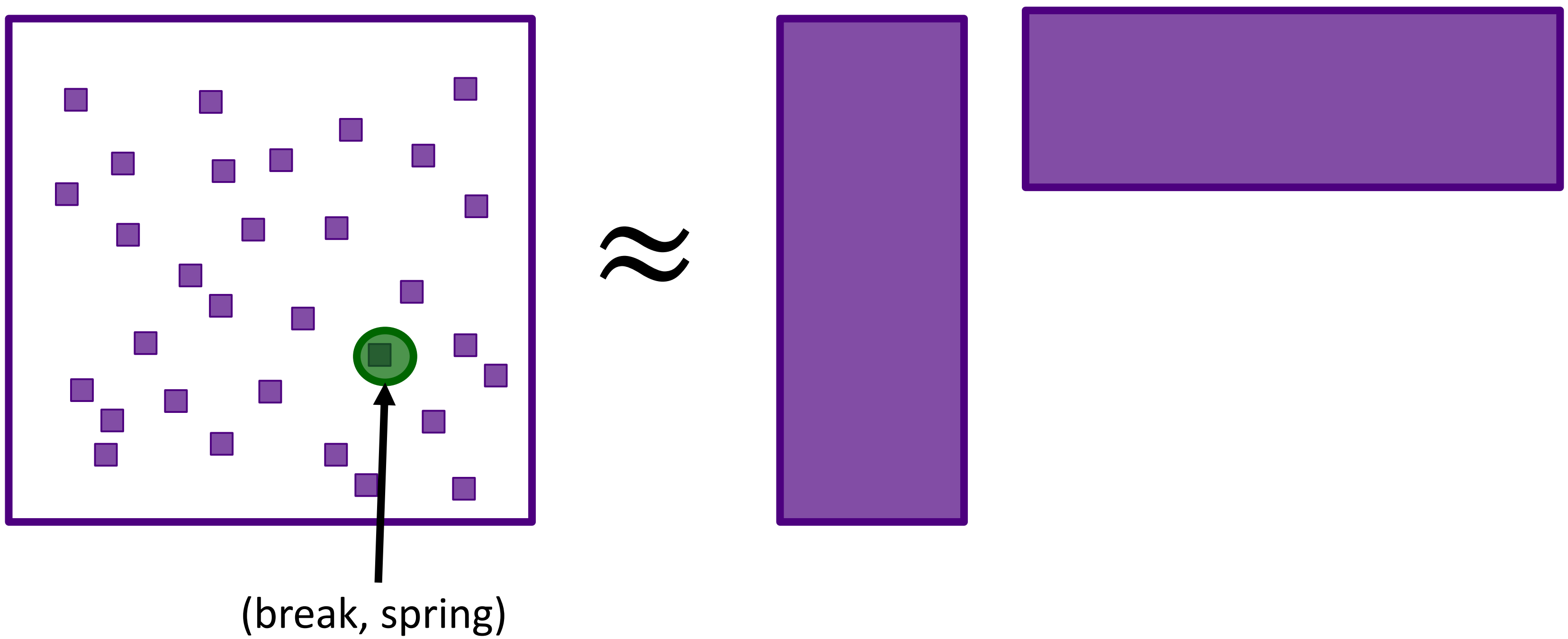
Problem: Low Rank Methods Operate at Fixed Granularity

If rank is too small……



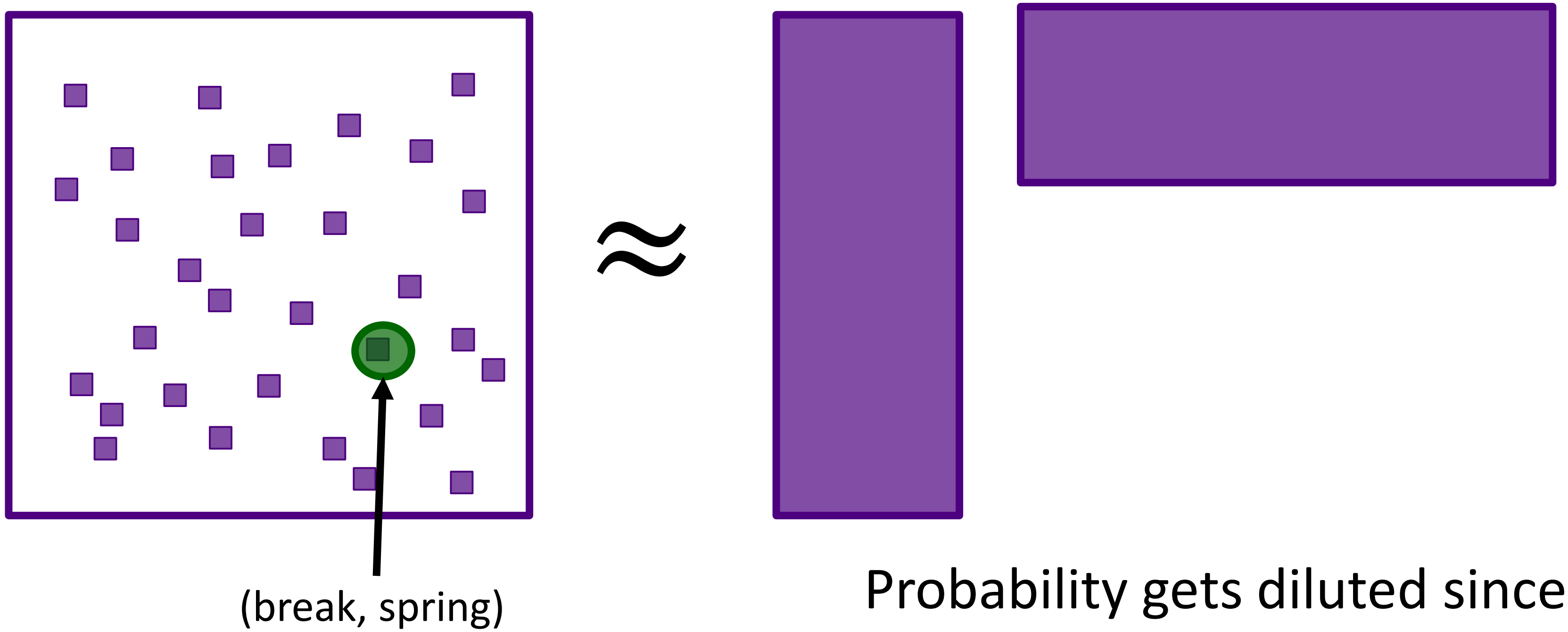
Problem: Low Rank Methods Operate at Fixed Granularity

If rank is too small……



Problem: Low Rank Methods Operate at Fixed Granularity

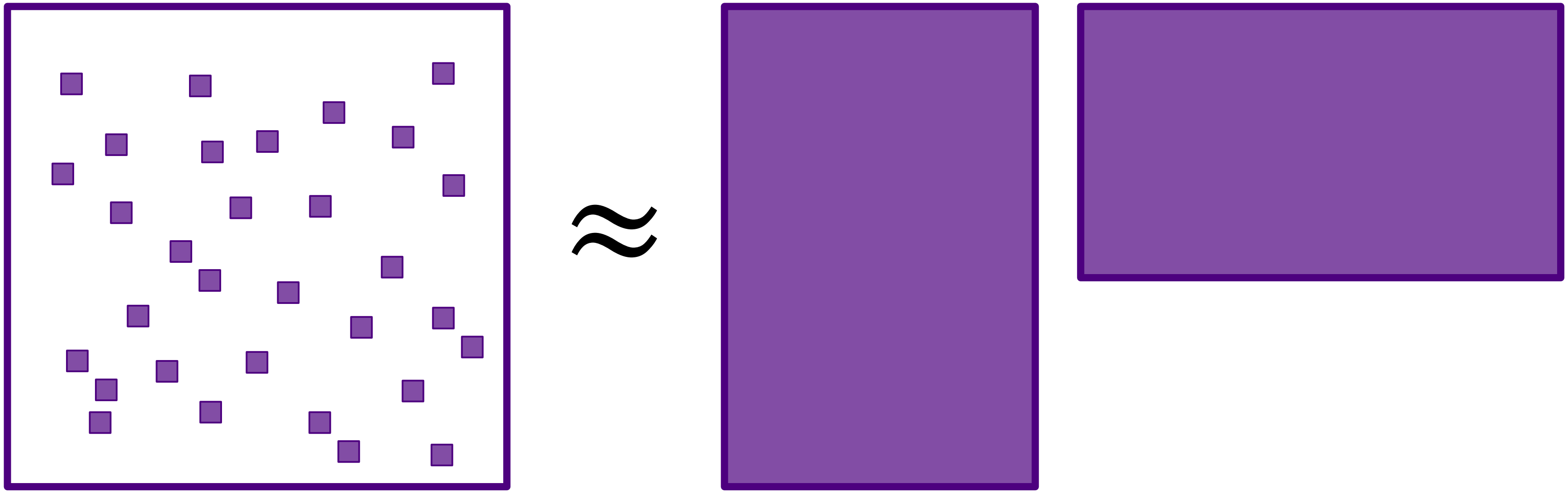
If rank is too small……



Probability gets diluted since “break” has many synonyms

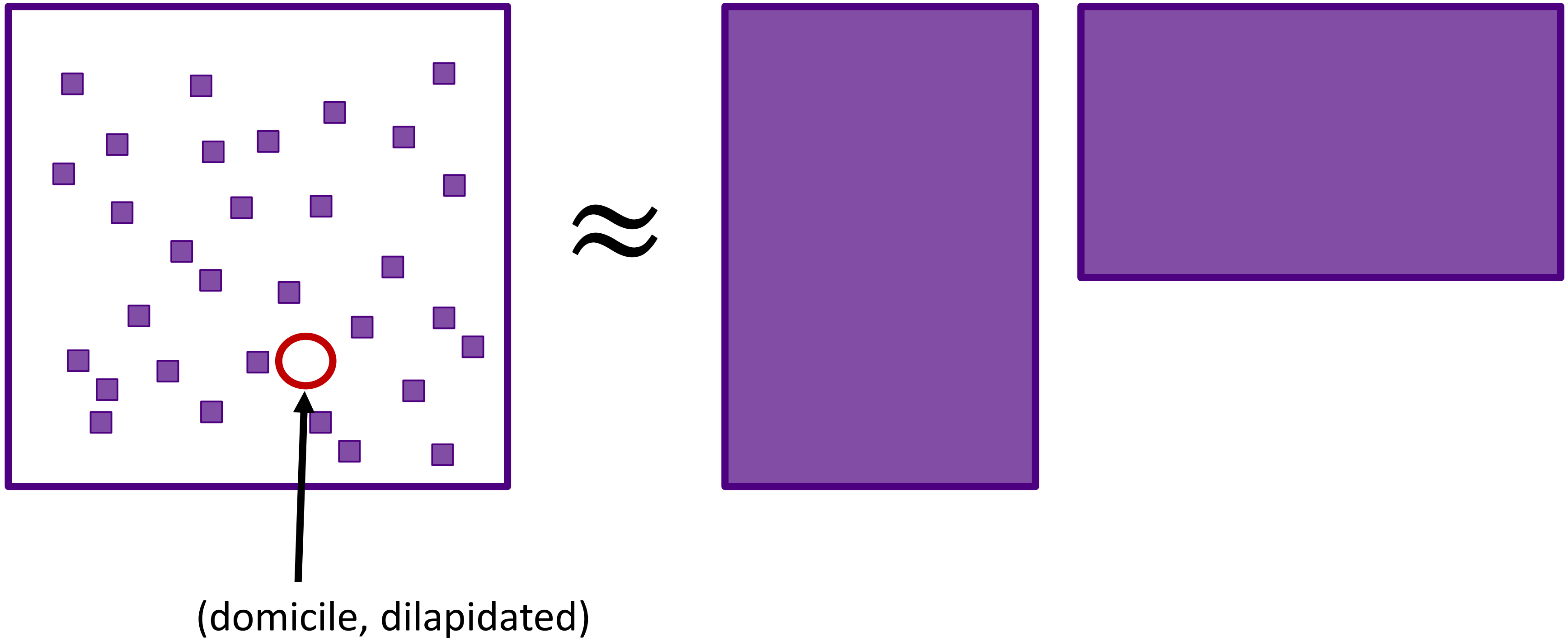
Problem: Low Rank Methods Operate at Fixed Granularity

If rank is too large....



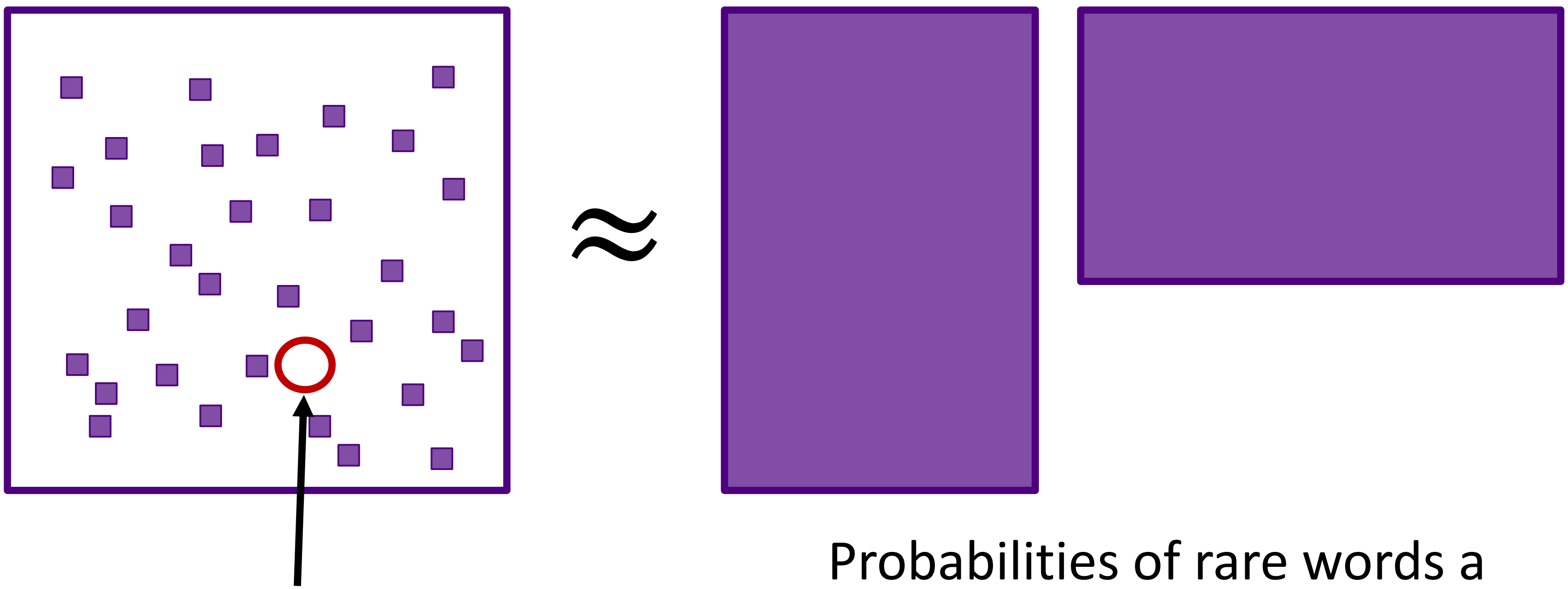
Problem: Low Rank Methods Operate at Fixed Granularity

If rank is too large....



Problem: Low Rank Methods Operate at Fixed Granularity

If rank is too large....



(domicile, dilapidated)

Probabilities of rare words a problem, since representation is too fine grained

Our Approach

Our Approach

MACHINE LEARNING DEPARTMENT

- Construct ensembles of low rank matrices/tensors to model language at multiple granularities

Our Approach

- Construct ensembles of low rank matrices/tensors to model language at multiple granularities

- Includes existing *n*-gram techniques as special cases
  - Absolute discounting
  - Jelinek Mercer (deleted-interpolation)
  - Kneser Ney

Our Approach

- Construct ensembles of low rank matrices/tensors to model language at multiple granularities

- Includes existing *n*-gram techniques as special cases
  - Absolute discounting
  - Jelinek Mercer (deleted-interpolation)
  - Kneser Ney

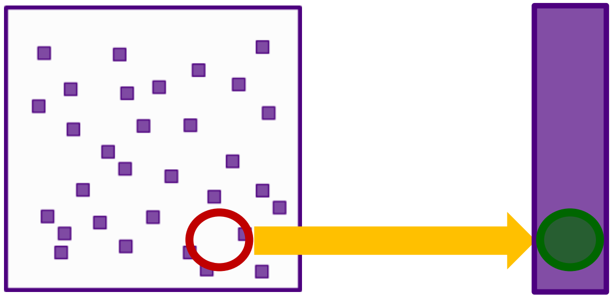
- Preserves advantages of standard *n*-gram approaches
  - Effective for short context lengths
  - Fast evaluation at test time

Outline

- Introduction
- Background on *Kneser Ney* smoothing
- Our Approach
  - Rank
  - Power
  - Constructing the Ensemble
- Experiments

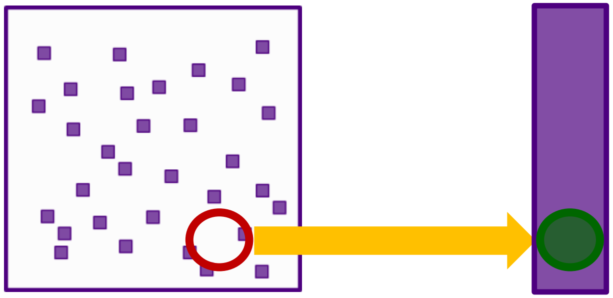
Kneser Ney - Intuition

- Lower order distribution should be altered



Kneser Ney - Intuition

- Lower order distribution should be altered

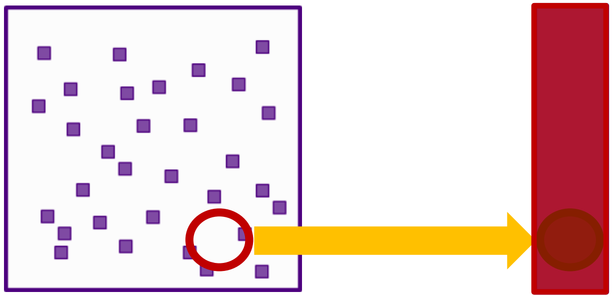


- Consider two words, *York* and *door*
  - *York only follows very few words i.e. New York*
  - *Door can follow many words i.e. “the door”, “red door”, “my door” etc.*

$$P(w_i = \text{door} \mid \text{backed} - \text{off on } w_{i-1})$$
$$> P(w_i = \text{York} \mid \text{backed} - \text{off on } w_{i-1})$$

Kneser Ney - Intuition

- Lower order distribution should be altered



- Consider two words, *York* and *door*
  - *York only follows very few words i.e. New York*
  - *Door can follow many words i.e. "the door", "red door", "my door" etc.*

$$P(w_i = \text{door} \mid \text{backed} - \text{off on } w_{i-1}) > P(w_i = \text{York} \mid \text{backed} - \text{off on } w_{i-1})$$

Kneser Ney Unigram Distribution

MACHINE LEARNING DEPARTMENT

$$N_{-}(w_i) = |\{w : c(w_i, w) > 0\}|$$

Diversity of $w_i's$ history

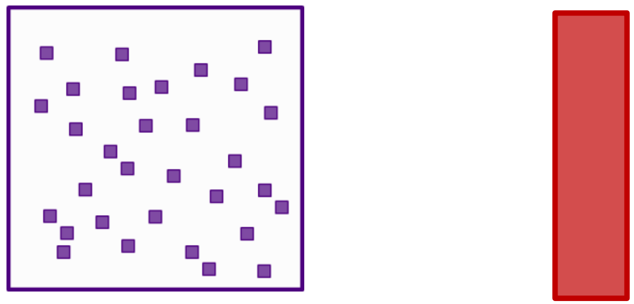
Kneser Ney Unigram Distribution

$$N_{-}(w_i) = |\{w : c(w_i, w) > 0\}|$$

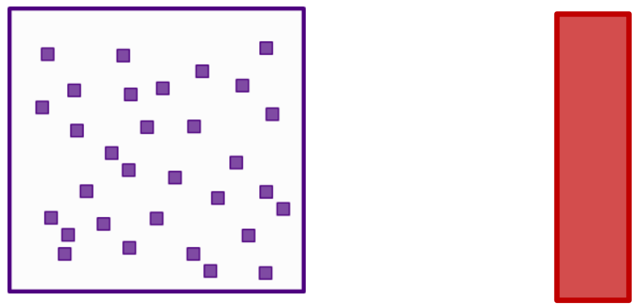
Diversity of $w_i's$ history

$$\hat{P}_{kn-uni}(w_i) = \frac{N_{-}(w_i)}{\sum_w N_{-}(w)}$$

Discounting

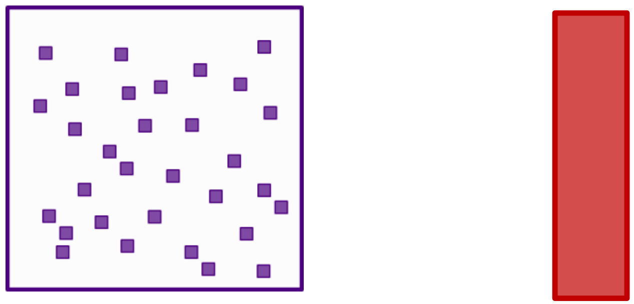


Discounting



$$\hat{P}_d(w_i|w_{i-1}) = \frac{\max(c(w_i, w_{i-1}) - d, 0)}{\sum_w c(w, w_{i-1})}$$

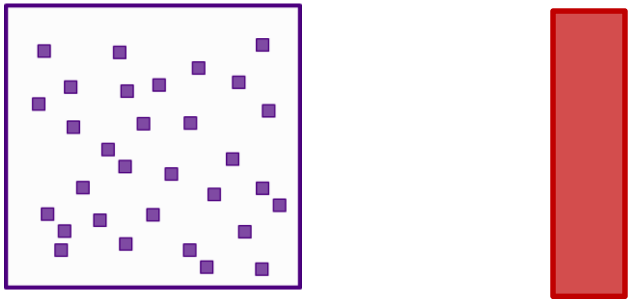
Discounting



$$\hat{P}_d(w_i|w_{i-1}) = \frac{\max(c(w_i, w_{i-1}) - d, 0)}{\sum_w c(w, w_{i-1})}$$

$$\hat{P}_{kney}(w_i|w_{i-1}) = \hat{P}_d(w_i|w_{i-1}) + \gamma(w_{i-1})\hat{P}_{kn-uni}(w_i)$$

Discounting



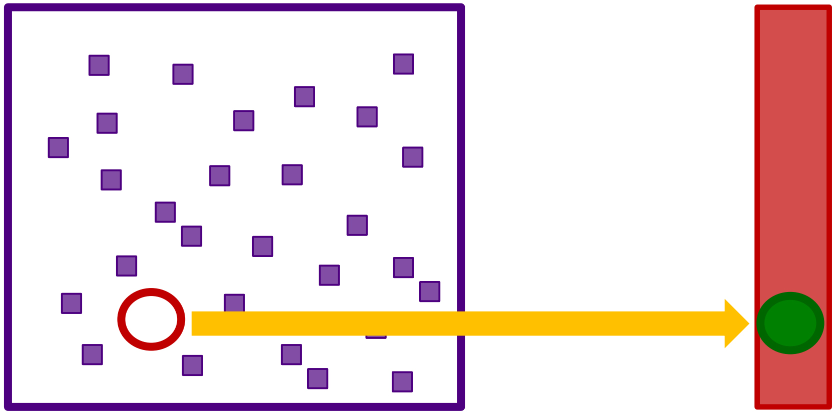
$$\hat{P}_d(w_i|w_{i-1}) = \frac{\max(c(w_i, w_{i-1}) - d, 0)}{\sum_w c(w, w_{i-1})}$$

$$\hat{P}_{kney}(w_i|w_{i-1}) = \hat{P}_d(w_i|w_{i-1}) + \gamma(w_{i-1})\hat{P}_{kn-uni}(w_i)$$

Where $\gamma(w_{i-1})$ is the leftover probability

Lower Order Marginal Aligns!

$$\hat{P}(w_i) = \sum_{w_{i-1}} \hat{P}_{kney}(w_i | w_{i-1}) \hat{P}(w_{i-1})$$



Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

Kneser Ney

Power Low Rank Ensembles

Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

Kneser Ney

- Ensemble composed of unsmoothed *n*-grams

Power Low Rank Ensembles

Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

Kneser Ney

- Ensemble composed of unsmoothed *n*-grams

- Alter lower order distributions by using count of unique histories

Power Low Rank Ensembles

Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

Kneser Ney

- Ensemble composed of unsmoothed *n*-grams
- Alter lower order distributions by using count of unique histories
- Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint

Power Low Rank Ensembles

Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

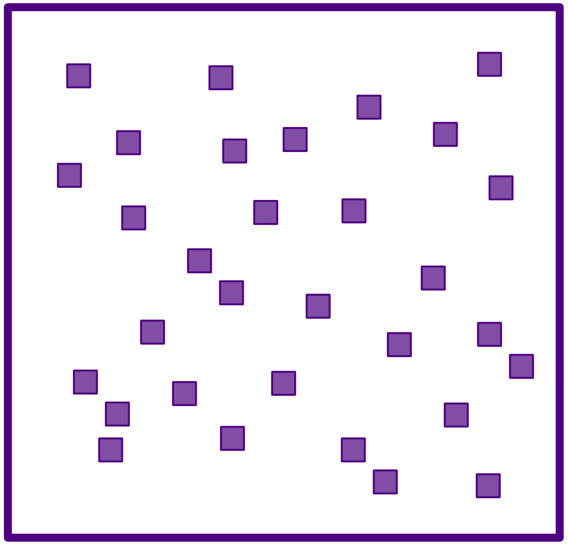
Kneser Ney	*Power Low Rank Ensembles*
Ensemble composed of unsmoothed *n*-grams	?
Alter lower order distributions by using count of unique histories	?
Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint	?

Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

Kneser Ney	*Power Low Rank Ensembles*
Ensemble composed of unsmoothed *n*-grams	?
Alter lower order distributions by using count of unique histories	?
Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint	?

In General, Bigram is Full Rank



Independence = Rank 1

MACHINE LEARNING DEPARTMENT

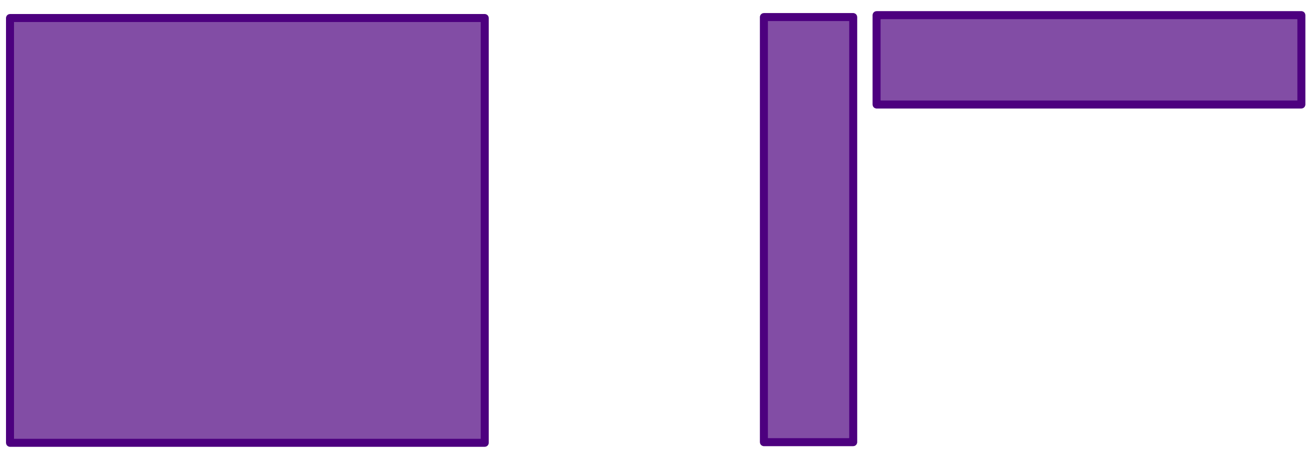
- If w_i and w_{i-1} are independent

$$P(w_i, w_{i-1}) = P(w_i)P(w_{i-1})$$

Independence = Rank 1

- If w_i and w_{i-1} are independent

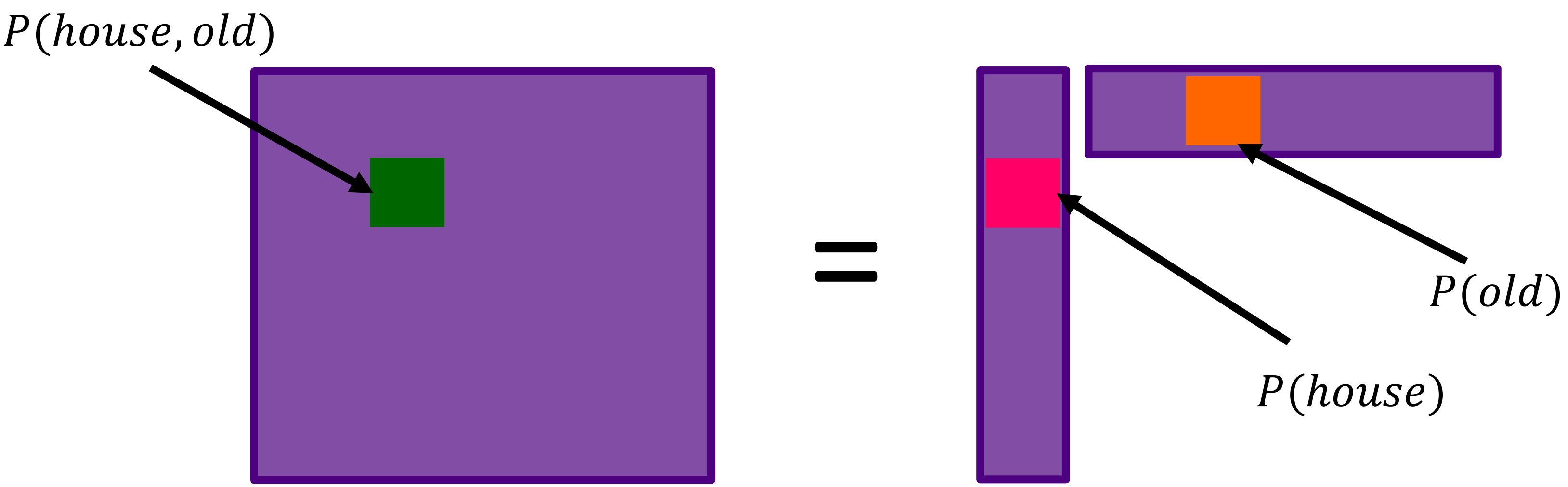
$$P(w_i, w_{i-1}) = P(w_i)P(w_{i-1})$$



Independence = Rank 1

- If w_i and w_{i-1} are independent

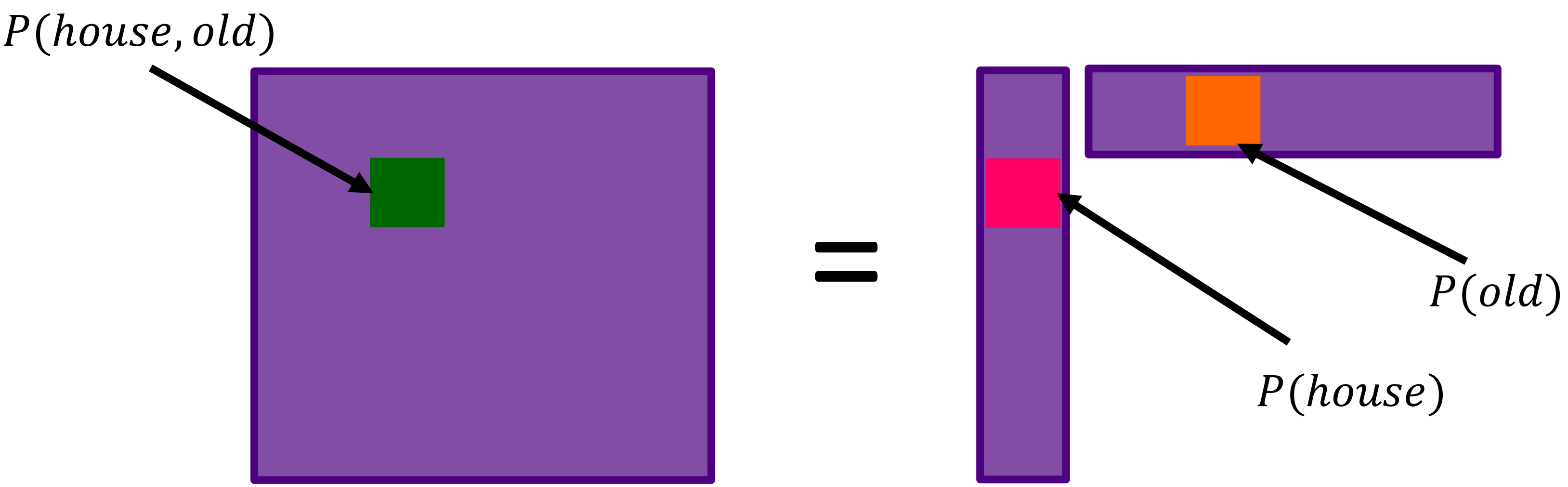
$$P(w_i, w_{i-1}) = P(w_i)P(w_{i-1})$$



Independence = Rank 1

- If w_i and w_{i-1} are independent

$$P(w_i, w_{i-1}) = P(w_i)P(w_{i-1})$$



- But what if w_i and w_{i-1} are not independent? What does the ***best*** rank 1 approximation give?

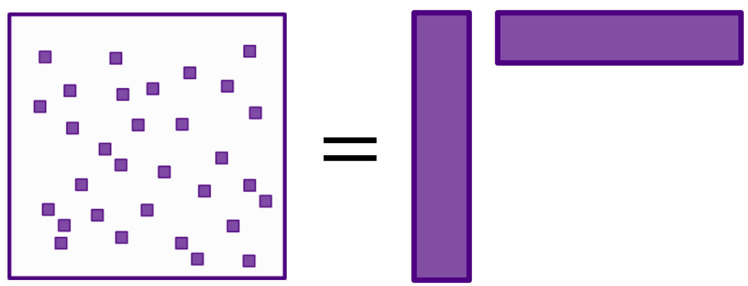
Rank

- Let $\boldsymbol{B}$ be the matrix such that

$$\boldsymbol{B}(w_i, w_{i-1}) = c(w_i, w_{i-1})$$

- Let

$$\boldsymbol{M}_1 = min_{\boldsymbol{M}:\boldsymbol{M}\geq 0, rank(\boldsymbol{M})=1} \|\boldsymbol{B} - \boldsymbol{M}\|_{KL}$$



Generalized KL
[*Lee and Seung 2001*]

- Then

$$\boldsymbol{M}_1(w_i, w_{i-1}) \propto \hat{P}(w_i)\hat{P}(w_{i-1})$$

Rank

- MLE unigram is normalized rank 1 approx. of MLE bigram under KL:

$$\hat{P}(w_i) = \frac{\boldsymbol{M}_1(w_i, w_{i-1})}{\sum_{w_i} \boldsymbol{M}_1(w_i, w_{i-1})}$$

Rank

- MLE unigram is normalized rank 1 approx. of MLE bigram under KL:

$$\hat{P}(w_i) = \frac{\boldsymbol{M}_1(w_i, w_{i-1})}{\sum_{w_i} \boldsymbol{M}_1(w_i, w_{i-1})}$$

- Vary rank to obtain quantities between bigram and unigram

Rank

- MLE unigram is normalized rank 1 approx. of MLE bigram under KL:

$$\hat{P}(w_i) = \frac{\boldsymbol{M}_1(w_i, w_{i-1})}{\sum_{w_i} \boldsymbol{M}_1(w_i, w_{i-1})}$$

- Vary rank to obtain quantities between bigram and unigram

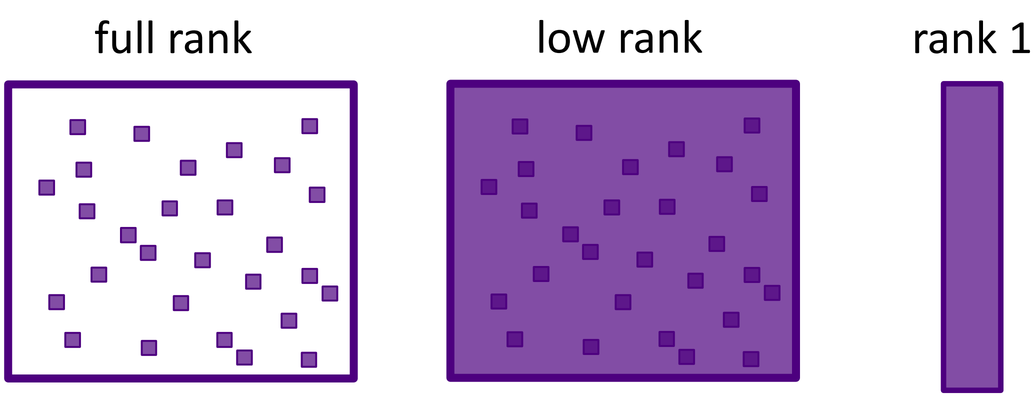
full rank   rank 1

Rank

- MLE unigram is normalized rank 1 approx. of MLE bigram under KL:

$$\hat{P}(w_i) = \frac{\boldsymbol{M}_1(w_i, w_{i-1})}{\sum_{w_i} \boldsymbol{M}_1(w_i, w_{i-1})}$$

- Vary rank to obtain quantities between bigram and unigram



Generalizing KN to PLRE

Kneser Ney	*Power Low Rank Ensembles*
• Ensemble composed of unsmoothed *n*-grams	• Ensemble composed of unsmoothed *n*-grams plus other low rank matrices/tensors
• Alter lower order distributions by using count of unique histories	?
• Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint	?

Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

Kneser Ney

- Ensemble composed of unsmoothed *n*-grams
- Alter lower order distributions by using count of unique histories
- Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint

Power Low Rank Ensembles

- Ensemble composed of unsmoothed *n*-grams plus other low rank matrices/tensors

?

?

Consider Elementwise Power

Consider Elementwise Power

$$\boldsymbol{B}$$

$$\begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{0} & \mathbf{5} & \mathbf{0} \\ \mathbf{2} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Consider Elementwise Power

$$\boldsymbol{B}$$

$$\begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{0} & \mathbf{5} & \mathbf{0} \\ \mathbf{2} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

row sum

$$\begin{bmatrix} \mathbf{4} \\ \mathbf{5} \\ \mathbf{2} \end{bmatrix}$$

Consider Elementwise Power

$$\boldsymbol{B}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 5 & 0 \\ 2 & 0 & 0 \end{bmatrix}$$

row sum

$$\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix}$$

Consider Elementwise Power

$$\boldsymbol{B} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 5 & 0 \\ 2 & 0 & 0 \end{bmatrix} \rightarrow \boldsymbol{B}^{0.5} = \begin{bmatrix} 1 & 1.4 & 1 \\ 0 & 2.2 & 0 \\ 1.4 & 0 & 0 \end{bmatrix}$$

row sum

$$\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix}$$

Consider Elementwise Power

$$\boldsymbol{B} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 5 & 0 \\ 2 & 0 & 0 \end{bmatrix} \rightarrow \boldsymbol{B}^{0.5} = \begin{bmatrix} 1 & 1.4 & 1 \\ 0 & 2.2 & 0 \\ 1.4 & 0 & 0 \end{bmatrix}$$

row sum ↓ $\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix}$

row sum ↓ $\begin{bmatrix} 3.4 \\ 2.2 \\ 1.4 \end{bmatrix}$

Consider Elementwise Power

$$\boldsymbol{B} \qquad\qquad\qquad \boldsymbol{B}^{0.5}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 5 & 0 \\ 2 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1.4 & 1 \\ 0 & 2.2 & 0 \\ 1.4 & 0 & 0 \end{bmatrix} \rightarrow$$

row sum

row sum

$$\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix} \qquad\qquad \begin{bmatrix} 3.4 \\ 2.2 \\ 1.4 \end{bmatrix}$$

Consider Elementwise Power

$$\boldsymbol{B} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 5 & 0 \\ 2 & 0 & 0 \end{bmatrix} \rightarrow \boldsymbol{B}^{0.5} = \begin{bmatrix} 1 & 1.4 & 1 \\ 0 & 2.2 & 0 \\ 1.4 & 0 & 0 \end{bmatrix} \rightarrow \boldsymbol{B}^{0} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

row sum

$$\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix}$$

row sum

$$\begin{bmatrix} 3.4 \\ 2.2 \\ 1.4 \end{bmatrix}$$

Consider Elementwise Power

$$\boldsymbol{B} \qquad\qquad \boldsymbol{B}^{0.5} \qquad\qquad \boldsymbol{B}^{0}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 5 & 0 \\ 2 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1.4 & 1 \\ 0 & 2.2 & 0 \\ 1.4 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

row sum   row sum   row sum

$$\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix} \qquad \begin{bmatrix} 3.4 \\ 2.2 \\ 1.4 \end{bmatrix} \qquad \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

Consider Elementwise Power

MACHINE LEARNING DEPARTMENT

$$B \qquad\qquad B^{0.5} \qquad\qquad B^{0}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 5 & 0 \\ 2 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1.4 & 1 \\ 0 & 2.2 & 0 \\ 1.4 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

row sum ↓   row sum ↓   row sum ↓

$$\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix} \qquad \begin{bmatrix} 3.4 \\ 2.2 \\ 1.4 \end{bmatrix} \qquad \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

emphasis on diversity →

Consider Elementwise Power

Consider Elementwise Power

$$\boldsymbol{M}_1^0 = min_{\boldsymbol{M}:\boldsymbol{M}\geq 0, rank(\boldsymbol{M})=1} \left\| \boldsymbol{B^0} - \boldsymbol{M} \right\|_{KL}$$

Consider Elementwise Power

$$\boldsymbol{M}_1^0 = min_{\boldsymbol{M}:\boldsymbol{M}\geq 0, rank(\boldsymbol{M})=1} \left\| \boldsymbol{B}^{\boldsymbol{0}} - \boldsymbol{M} \right\|_{KL}$$

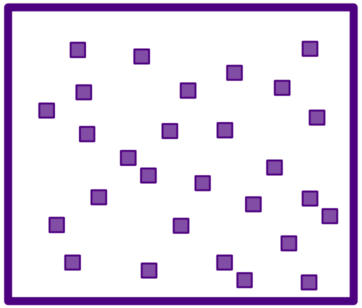
$$\widehat{P}_{kn-uni}(w_i) = \frac{\boldsymbol{M}_1^0(w_i, w_{i-1})}{\sum_w \boldsymbol{M}_1^0(w, w_{i-1})}$$

Consider Elementwise Power

$$\boldsymbol{M}_1^0 = min_{\boldsymbol{M}:\boldsymbol{M}\geq 0, rank(\boldsymbol{M})=1} \left\| \boldsymbol{B}^{\boldsymbol{0}} - \boldsymbol{M} \right\|_{KL}$$

$$\hat{P}_{kn-uni}(w_i) = \frac{\boldsymbol{M}_1^0(w_i, w_{i-1})}{\sum_w \boldsymbol{M}_1^0(w, w_{i-1})}$$

power = 1
full rank



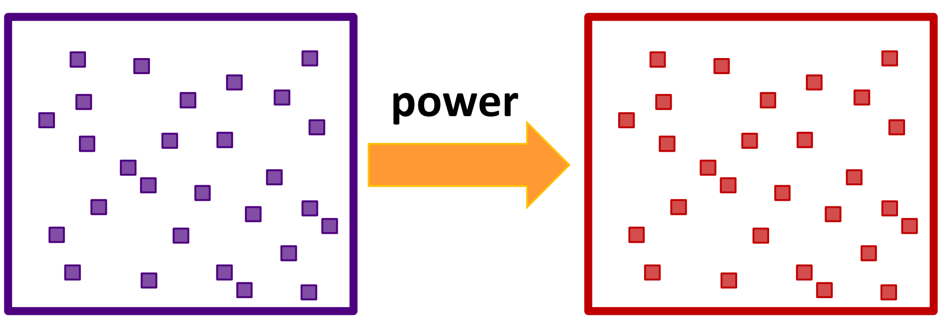
Consider Elementwise Power

$$\boldsymbol{M}_1^0 = min_{\boldsymbol{M}:\boldsymbol{M}\geq 0, rank(\boldsymbol{M})=1} \left\| \boldsymbol{B}^{\boldsymbol{0}} - \boldsymbol{M} \right\|_{KL}$$

$$\hat{P}_{kn-uni}(w_i) = \frac{\boldsymbol{M}_1^0(w_i, w_{i-1})}{\sum_w \boldsymbol{M}_1^0(w, w_{i-1})}$$

power = 1
full rank

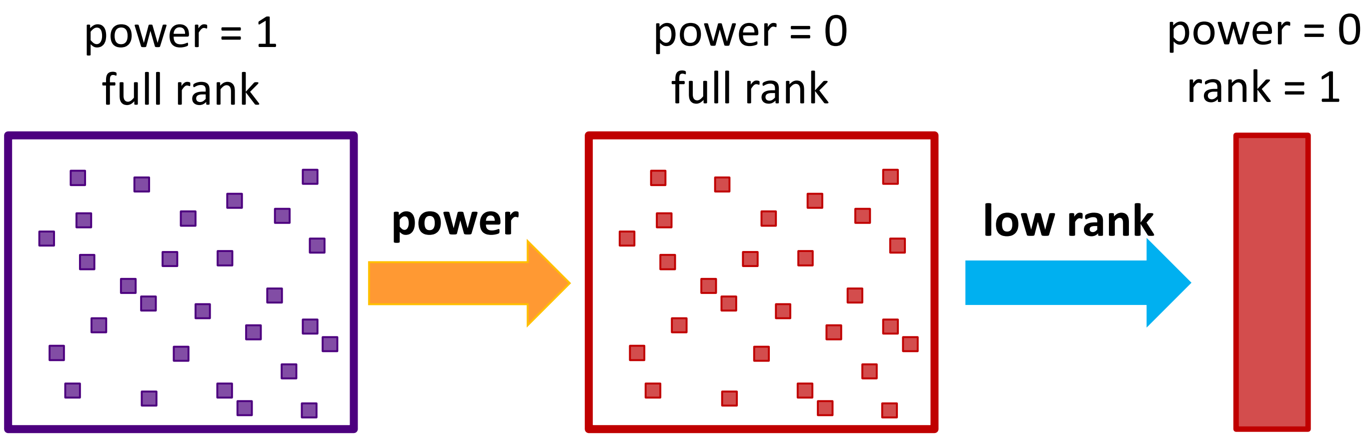
power = 0
full rank



Consider Elementwise Power

$$M_1^0 = min_{\boldsymbol{M}:\boldsymbol{M}\geq 0, rank(\boldsymbol{M})=1} \left\| \boldsymbol{B}^{\boldsymbol{0}} - \boldsymbol{M} \right\|_{KL}$$

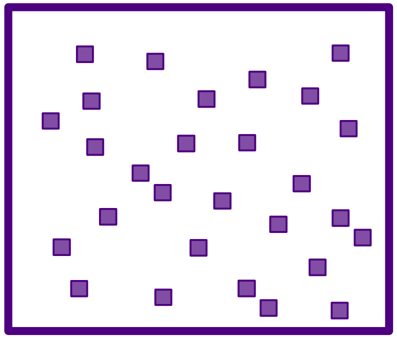
$$\hat{P}_{kn-uni}(w_i) = \frac{\boldsymbol{M}_1^0(w_i, w_{i-1})}{\sum_w \boldsymbol{M}_1^0(w, w_{i-1})}$$



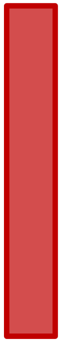
Varying Rank and Power

- Construct matrices of varying rank and power

power = 1
full rank

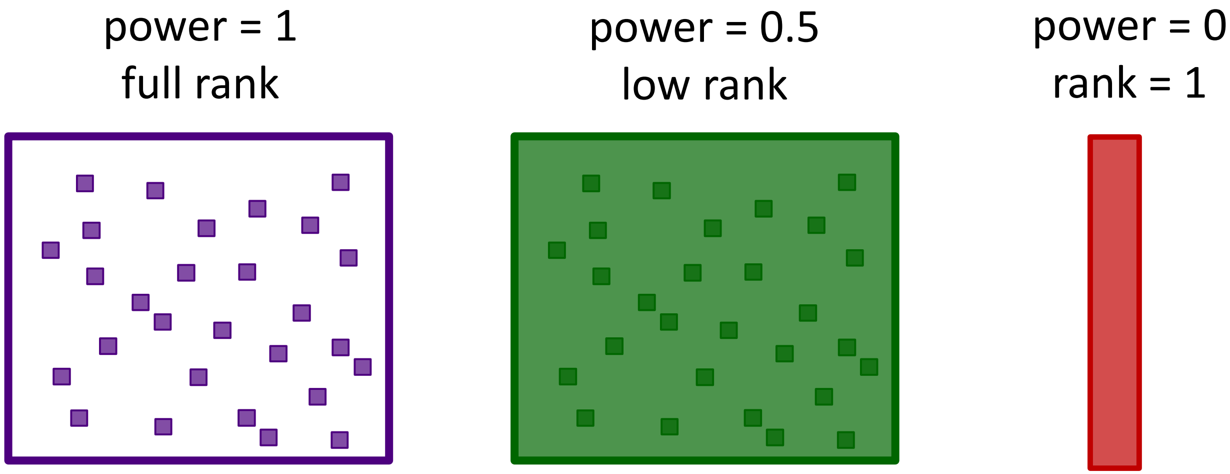


power = 0
rank = 1



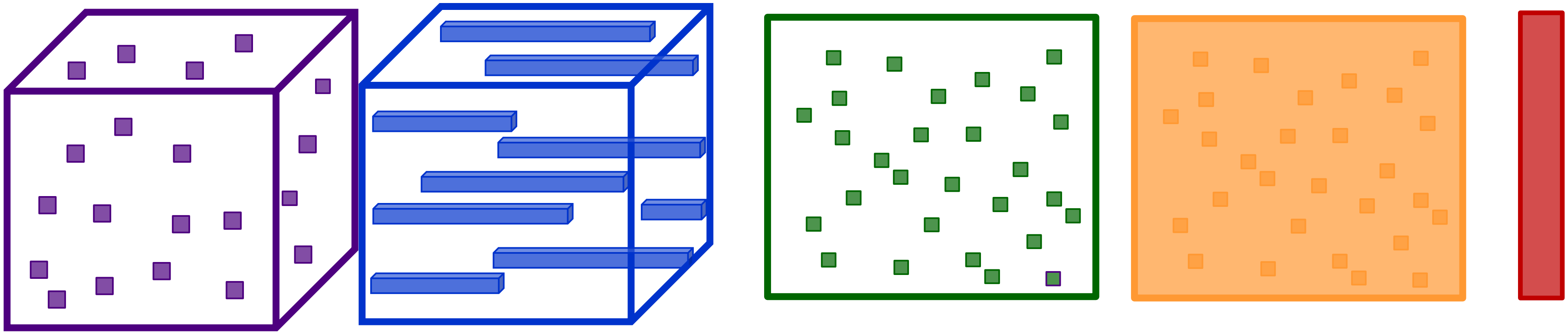
Varying Rank and Power

- Construct matrices of varying rank and power



Varying Rank and Power

- Generalizes to higher orders



Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

Kneser Ney	*Power Low Rank Ensembles*
• Ensemble composed of unsmoothed *n*-grams	• Ensemble composed of unsmoothed *n*-grams plus other low rank matrices/tensors
• Alter lower order distributions by using count of unique histories	• Alter lower order distributions by elementwise power
• Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint	?

Generalizing KN to PLRE

<u>*Kneser Ney*</u>

- Ensemble composed of unsmoothed *n*-grams
- Alter lower order distributions by using count of unique histories
- Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint

<u>*Power Low Rank Ensembles*</u>

- Ensemble composed of unsmoothed *n*-grams plus other low rank matrices/tensors
- Alter lower order distributions by elementwise power

?

Key Requirements

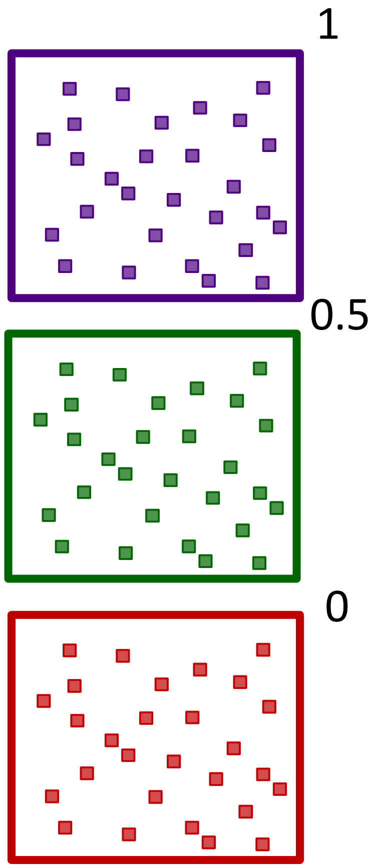
- Marginal constraint must hold:

$$\hat{P}(w_i) = \sum_{w_{i-1}} \hat{P}_{sm}(w_i | w_{i-1}) \hat{P}(w_{i-1})$$

- Evaluation of conditional probabilities must be fast

Our Approach: Two Step Procedure

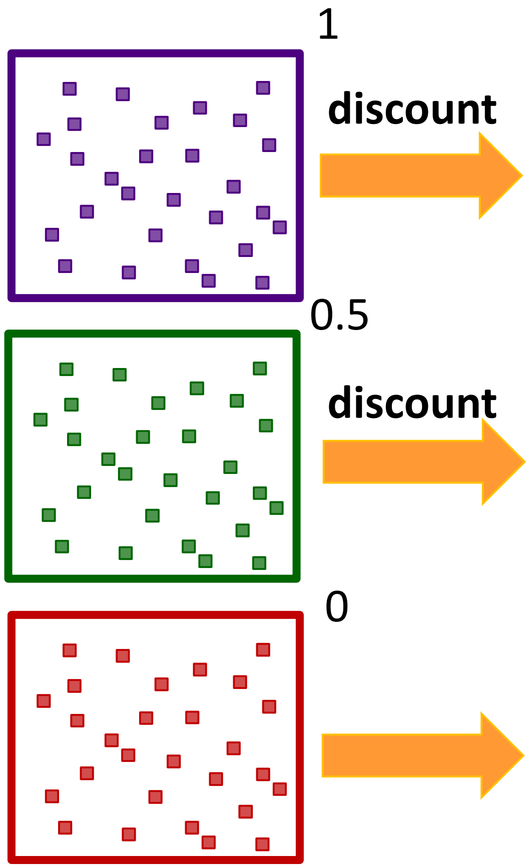
- **Step 1:** Compute discounts on powered counts such that marginal constraint holds. Each count gets a ***different*** discount



Our Approach: Two Step Procedure

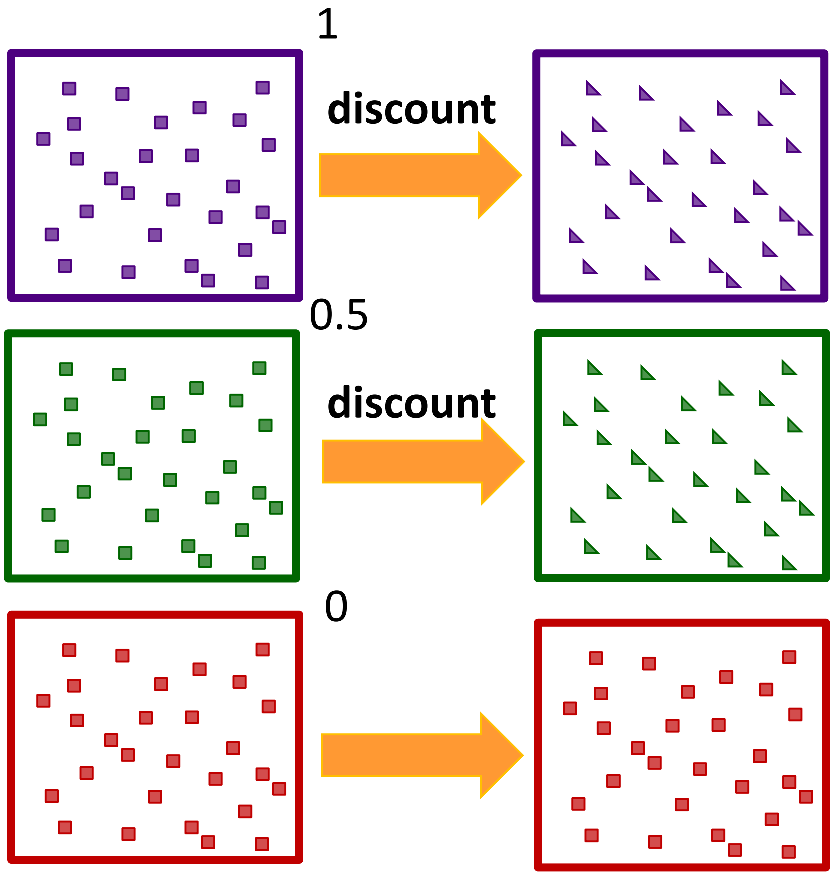
MACHINE LEARNING DEPARTMENT

- **Step 1:** Compute discounts on powered counts such that marginal constraint holds. Each count gets a ***different*** discount



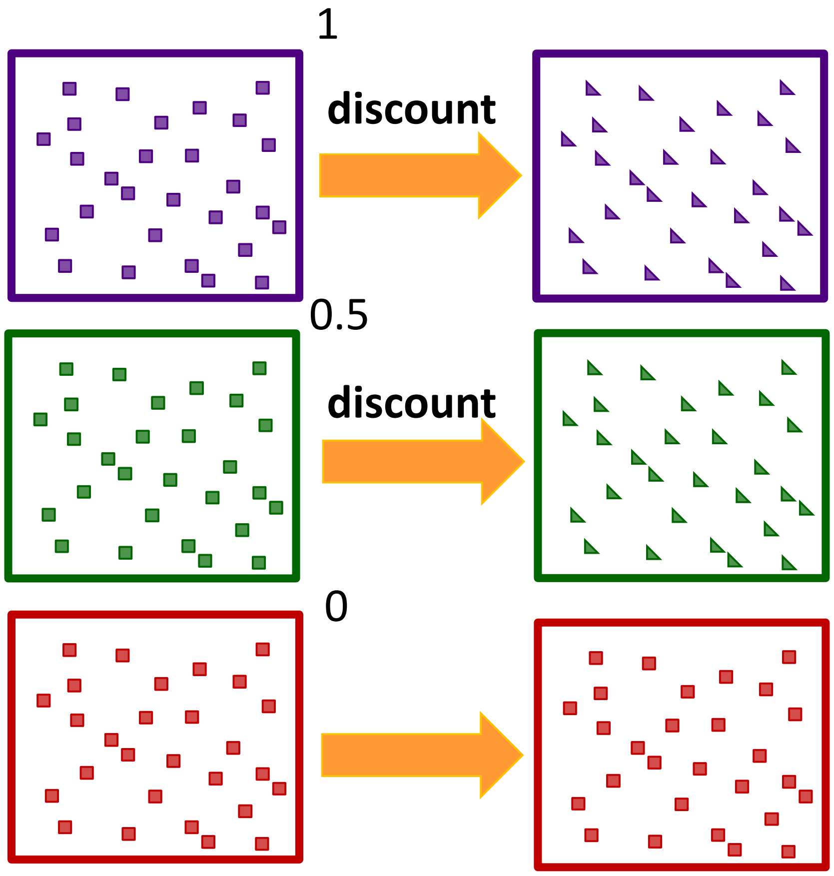
Our Approach: Two Step Procedure

- **Step 1:** Compute discounts on powered counts such that marginal constraint holds. Each count gets a ***different*** discount



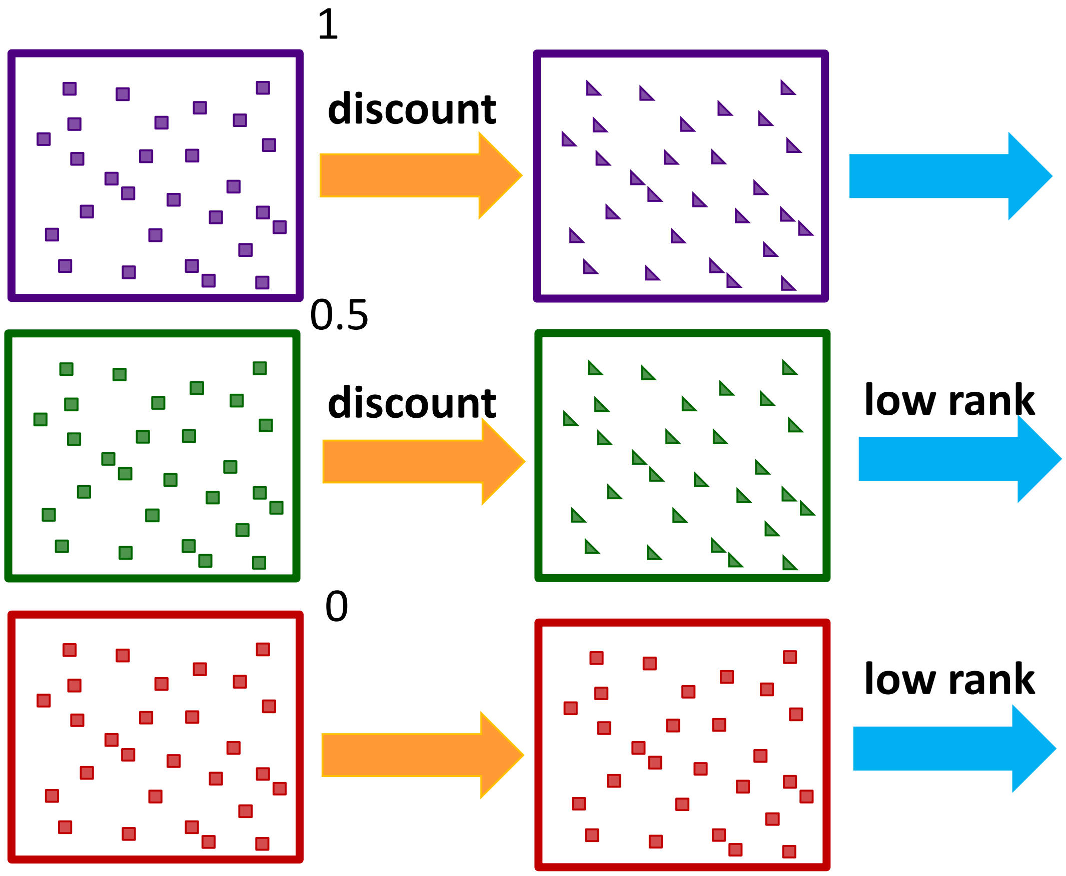
Our Approach: Two Step Procedure

- **Step 2:** Take low rank approximation of discounted quantities such that marginal constraint still holds



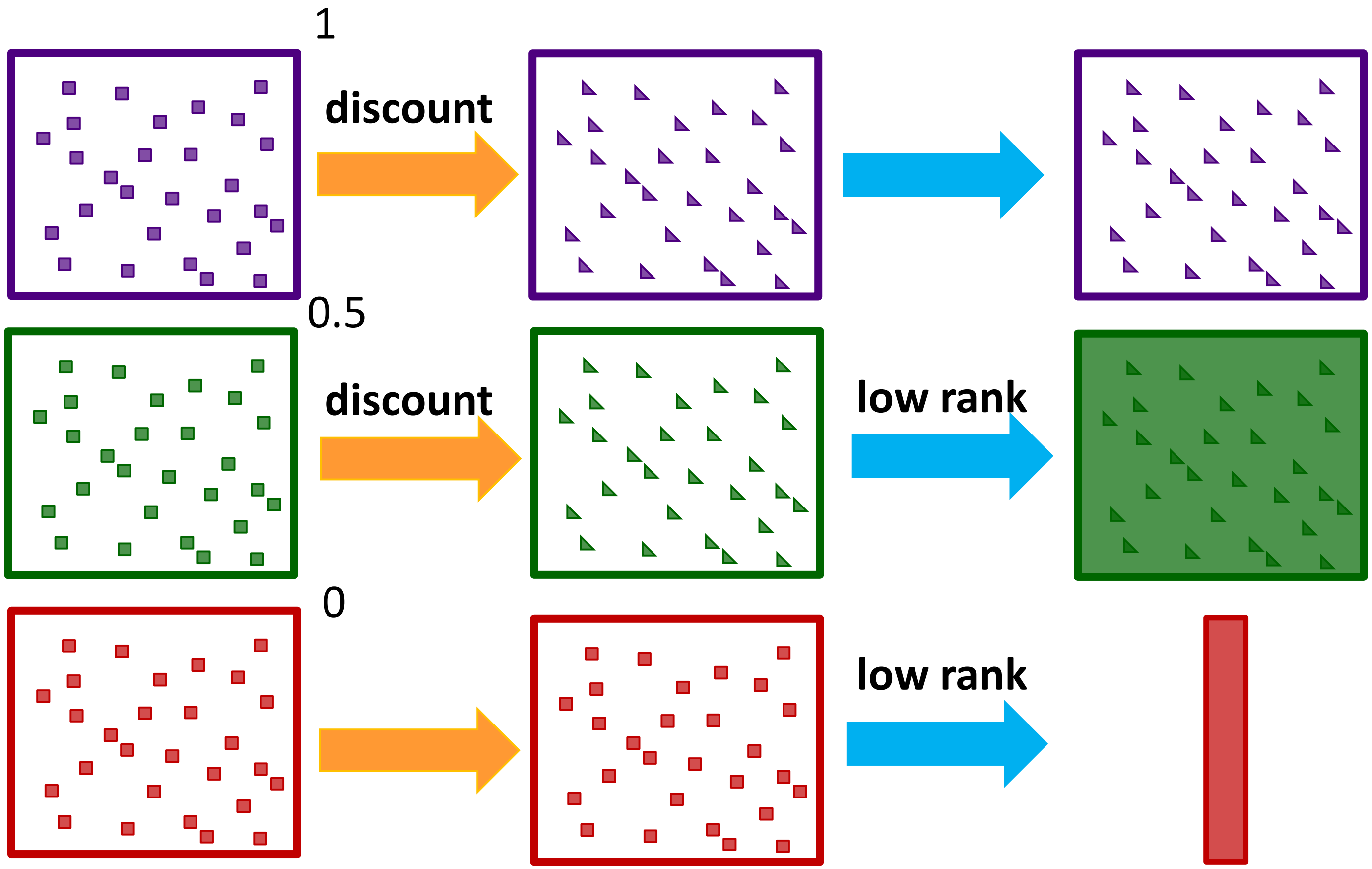
Our Approach: Two Step Procedure

- **Step 2:** Take low rank approximation of discounted quantities such that marginal constraint still holds



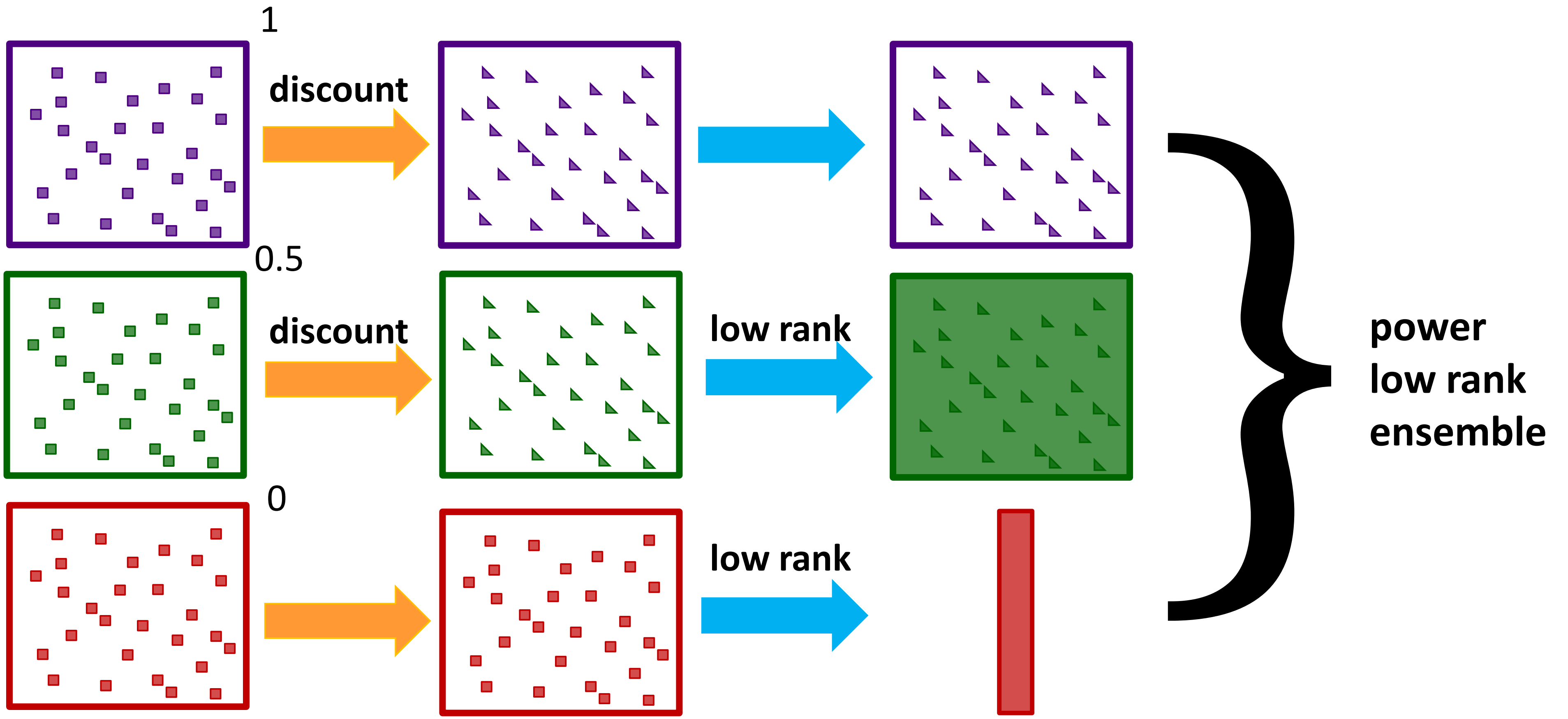
Our Approach: Two Step Procedure

- **Step 2:** Take low rank approximation of discounted quantities such that marginal constraint still holds



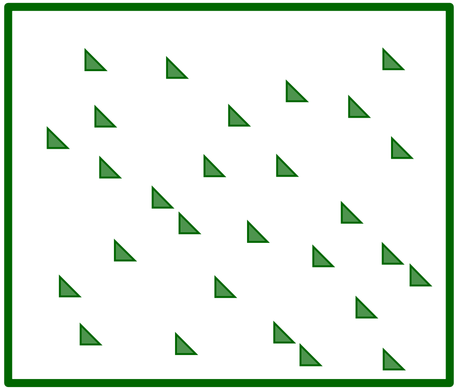
Our Approach: Two Step Procedure

- **Step 2:** Take low rank approximation of discounted quantities such that marginal constraint still holds



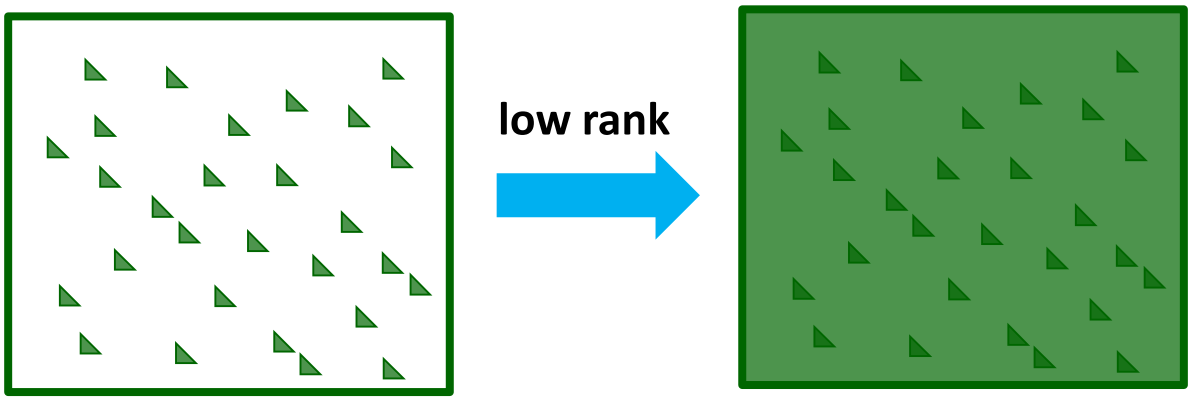
Why It Works

- Low rank approximations with respect to *KL* **preserve row/column sums**



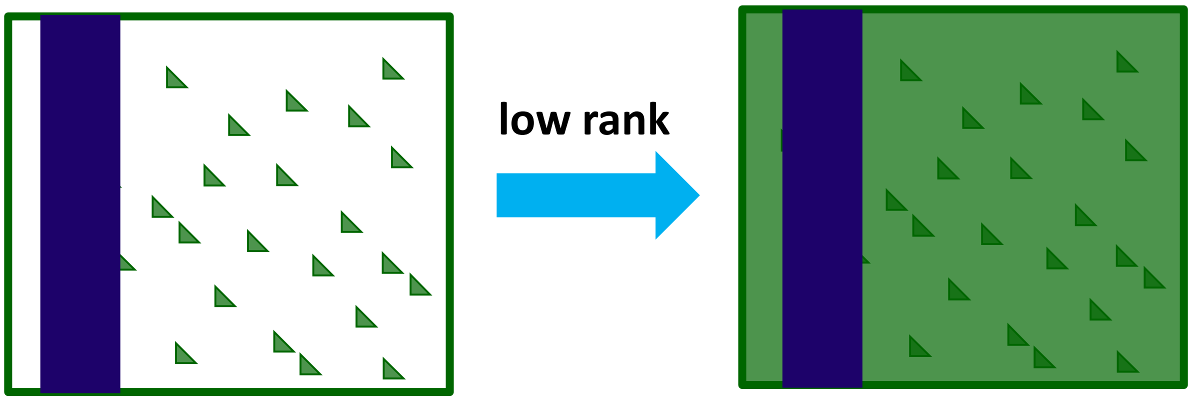
Why It Works

- Low rank approximations with respect to *KL* **preserve row/column sums**



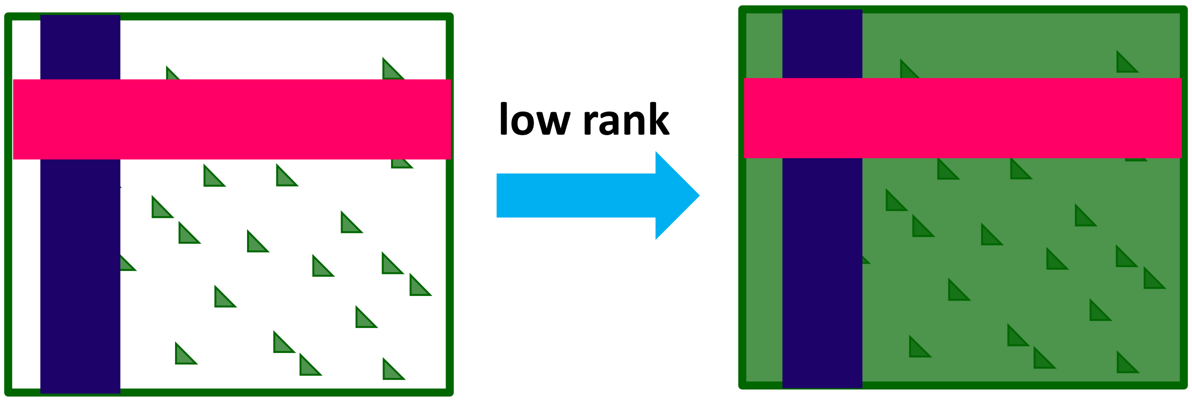
Why It Works

- Low rank approximations with respect to *KL* **preserve row/column sums**



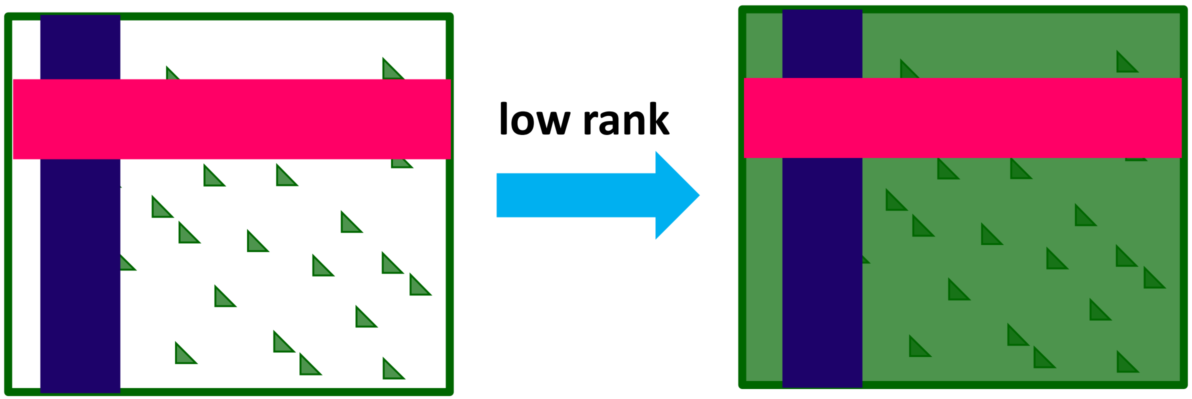
Why It Works

- Low rank approximations with respect to *KL* **preserve row/column sums**



Why It Works

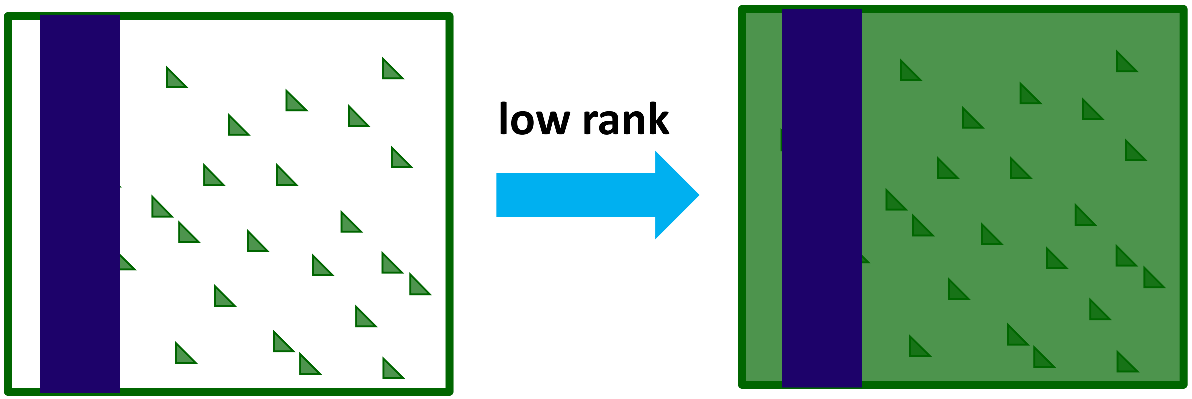
- Low rank approximations with respect to *KL* **preserve row/column sums**



- Therefore, discounting / leftover weight are preserved under the low rank approximation

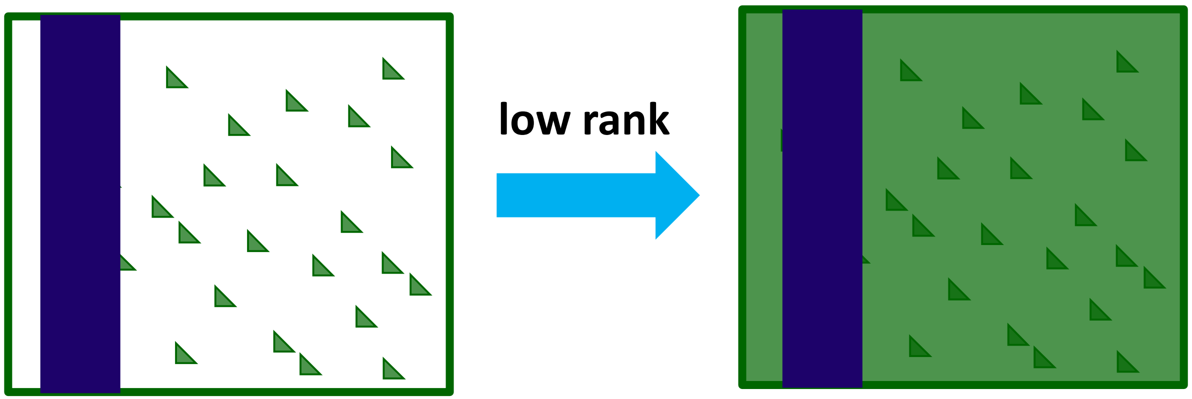
Normalizer can be Precomputed

- Low rank approximations with respect to *KL* **preserve row/column sums**



Normalizer can be Precomputed

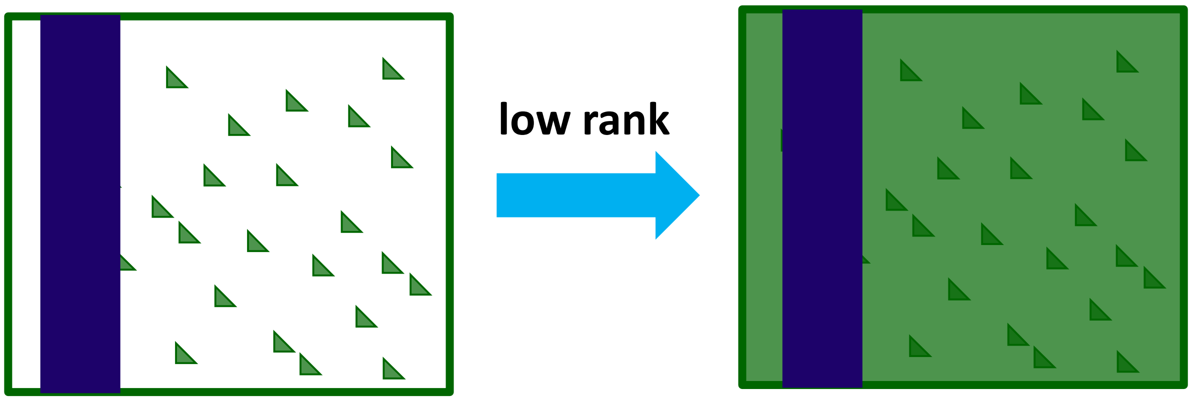
- Low rank approximations with respect to *KL* **preserve row/column sums**



- Compute normalizers on sparse counts

Normalizer can be Precomputed

- Low rank approximations with respect to *KL* **preserve row/column sums**



- Compute normalizers on sparse counts

- **No partition functions!**

Marginal Constraint Holds

MACHINE LEARNING DEPARTMENT

$$\hat{P}(w_i) = \sum_{w_{i-1}} \hat{P}_{plre}(w_i|w_{i-1})\hat{P}(w_{i-1})$$

Generalizing KN to PLRE

MACHINE LEARNING DEPARTMENT

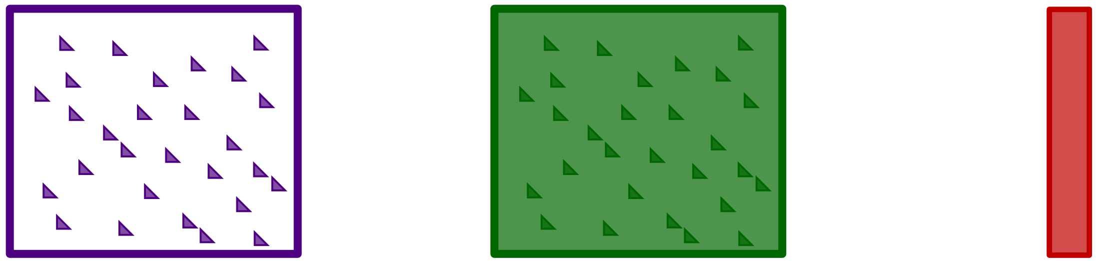
Kneser Ney

- Ensemble composed of unsmoothed *n*-grams
- Alter lower order distributions by using count of unique histories
- Use absolute discounting to interpolate different *n*-grams and preserve lower order marginal constraint

Power Low Rank Ensembles

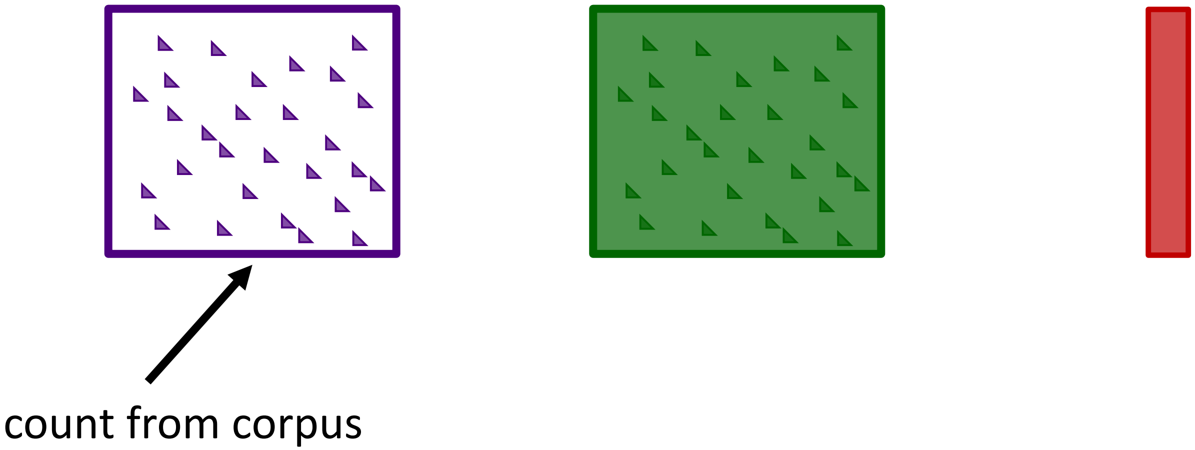
- Ensemble composed of unsmoothed *n*-grams plus other low rank matrices/tensors
- Alter lower order distributions by elementwise power
- Generalized discounting scheme: First compute discounts on powered counts, then take low rank approximation

Training Procedure

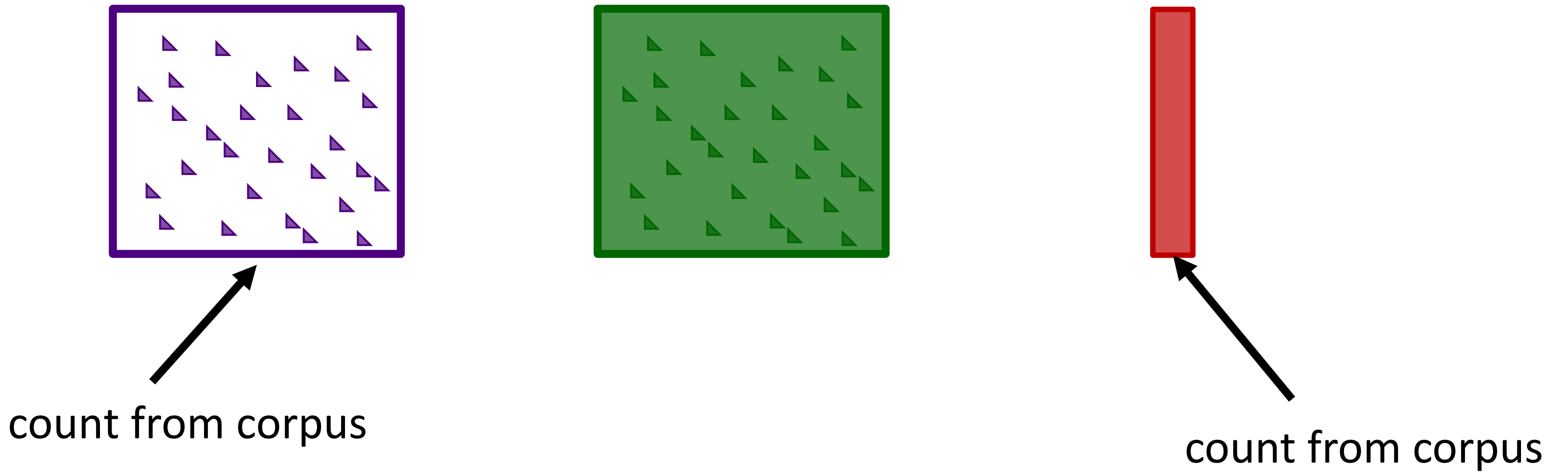


Training Procedure

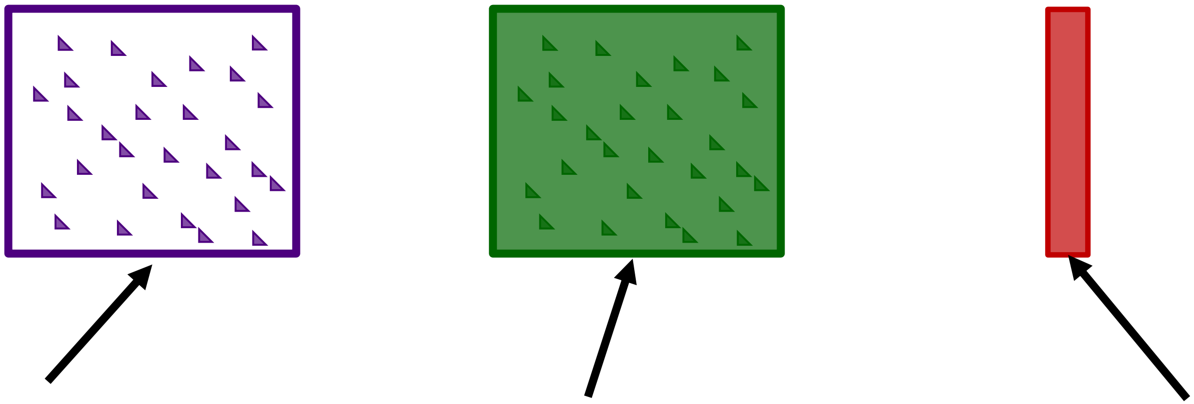
MACHINE LEARNING DEPARTMENT



Training Procedure



Training Procedure

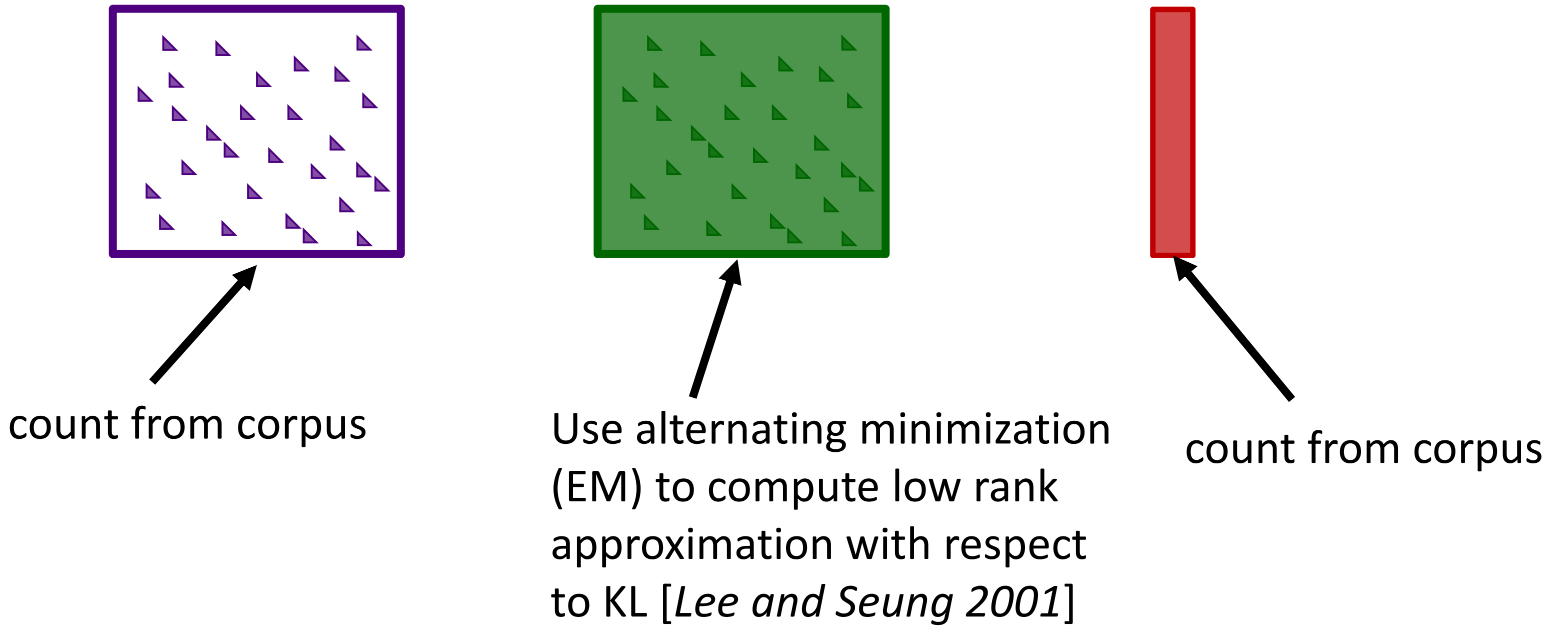


count from corpus

Use alternating minimization (EM) to compute low rank approximation with respect to KL [*Lee and Seung 2001*]

count from corpus

Training Procedure



- Because of ensemble representation, required rank is only about 100, even for billion word datasets

Test Time

MACHINE LEARNING DEPARTMENT

KN Test Complexity: $O(n)$

$$n = order, K = rank$$

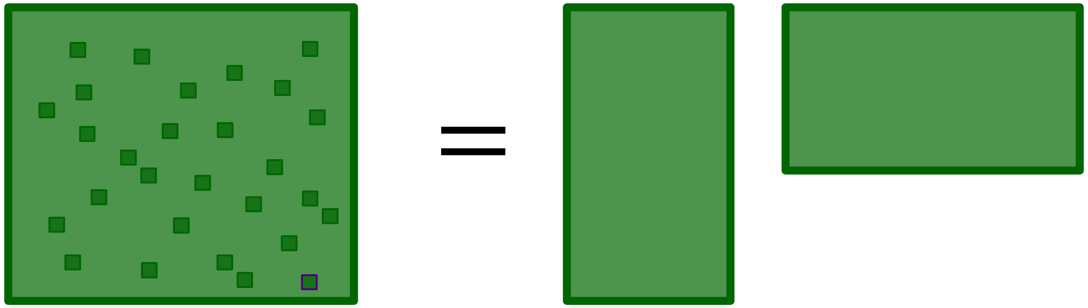
PLRE Test Complexity: $O(nK)$

Test Time

KN Test Complexity: $O(n)$

$n = order, K = rank$

PLRE Test Complexity: $O(nK)$

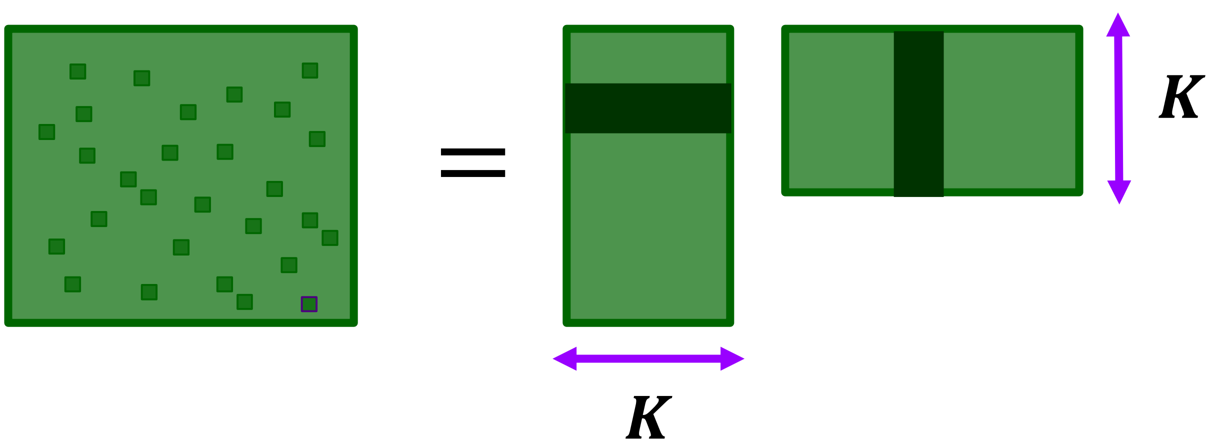


Test Time

KN Test Complexity: $O(n)$

$$n = order, K = rank$$

PLRE Test Complexity: $O(nK)$



Outline

- Introduction
- Background on *n*-gram smoothing
- Our Approach
  - Rank
  - Power
  - Constructing the Ensemble
- Experiments

Experiments

- Evaluate on English and Russian

- Baselines
  - **modKN** – Modified Kneser Ney (back-off)
  - **modint-KN**- Modified Interpolated Kneser Ney
  - **Other comparisons:** Class-based models, Neural Networks, Hierarchical Pitman Yor

Small Datasets - Perplexity

- English-Small [Bengio et al. 2003]
  - 20K vocabulary
  - 14 million tokens

- Russian-Small
  - 77K vocabulary
  - 3.5 million tokens

Small Datasets - Perplexity

- English-Small [Bengio et al. 2003]
  - 20K vocabulary
  - 14 million tokens
- Russian-Small
  - 77K vocabulary
  - 3.5 million tokens

	class KN	mod-KN	modint-KN	PLRE
English-Small	119.7	104.55	100.07	**95.15**
Russian-Small	284.09	283.7	260.19	**238.96**

Small English Comparisons

Small English Comparisons

Model	Context Size	Perplexity
mod-KN(4)	3	128
modint-KN(4)	3	116.6
infinity-gram HPYP [*Wood et al. 2009*]	infinity	111.8

Small English Comparisons

Model	Context Size	Perplexity
mod-KN(4)	3	128
modint-KN(4)	3	116.6
infinity-gram HPYP [*Wood et al. 2009*]	infinity	111.8
PLRE(4)	3	108.7

Small English Comparisons

Model	Context Size	Perplexity
mod-KN(4)	3	128
modint-KN(4)	3	116.6
infinity-gram HPYP [*Wood et al. 2009*]	infinity	111.8
PLRE(4)	3	108.7
LBL [*Mnih and Hinton 2007*]	5	117
LBL [*Mnih and Hinton 2007*]	10	107.8
RNN-ME [*Mikolov et al. 2012*]	infinity	82.1

Large Datasets - Perplexity

- English-Large
  - 836,000 types
  - 837 million tokens

- Russian-Large
  - 1.3 million types
  - 521 million tokens

- On 8 cores, PLRE (with optimal parameter settings) completes training on English-Large in **3.2 hrs** and Russian-Large in **7.7 hours**

Large Datasets - Perplexity

- English-Large
  - 836,000 types
  - 837 million tokens
- Russian-Large
  - 1.3 million types
  - 521 million tokens

	modint-KN	PLRE
English-Large	77.90 +/- 0.20	**75.66 +/- 0.19**
Russian-Large	289.6 +/-6.82	**264.59 +/- 5.84**

- On 8 cores, PLRE (with optimal parameter settings) completes training on English-Large in **3.2 hrs** and Russian-Large in **7.7 hours**

Machine Translation Task

- English to Russian translation task (Language model is used as a feature in the translation system)

- Unlike other recent works, we use PLRE ***instead*** of modint-KN (not both)

- To deal with the non-determinism, the model is only trained once, using modint-KN. The same feature weights are then used for both PLRE and modint-KN

Machine Translation Task

- English to Russian translation task (Language model is used as a feature in the translation system)
- Unlike other recent works, we use PLRE ***instead*** of modint-KN (not both)
- To deal with the non-determinism, the model is only trained once, using modint-KN. The same feature weights are then used for both PLRE and modint-KN

Method	BLEU
modint-KN	17.63 +/- 0.11
PLRE	**17.79 +/- 0.07**
Smallest Diff	PLRE+0.05
Largest Diff	PLRE+0.29

Conclusion

- We presented a novel technique for language modeling called power low rank ensembles
- Consistently outperforms state-of-the-art Kneser Ney baselines
  - Effective for small context sizes
  - No partition function required
- Part of broader theme of exploiting connection between linear algebra and probability to develop new solutions for NLP

Language Technologies Institute

MACHINE LEARNING DEPARTMENT

Thanks!

Code/data available at **http://www.cs.cmu.edu/~apparikh/plre**