Machine Learning, Artificial Intelligence, and Natural Language Processing

Marcelo C. Medeiros

Department of Economics
The University of Illinois at Urbana-Champaign

ECONOMETRICS LAB
QUANTITATIVE RESEARCH @ UIUC

What is Word Embedding?

- A technique to represent words in vector space
- Captures semantic and syntactic similarity
- Dense, low-dimensional representations of words
- Enables algebraic operations on words

What is Word Embedding?

- Traditional methods (e.g., BoW) fail to capture context
- Sparse and high-dimensional representations
- Word embeddings overcome these issues with dense vectors

What is Word Embedding?

How should we group words?

What is Word Embedding?

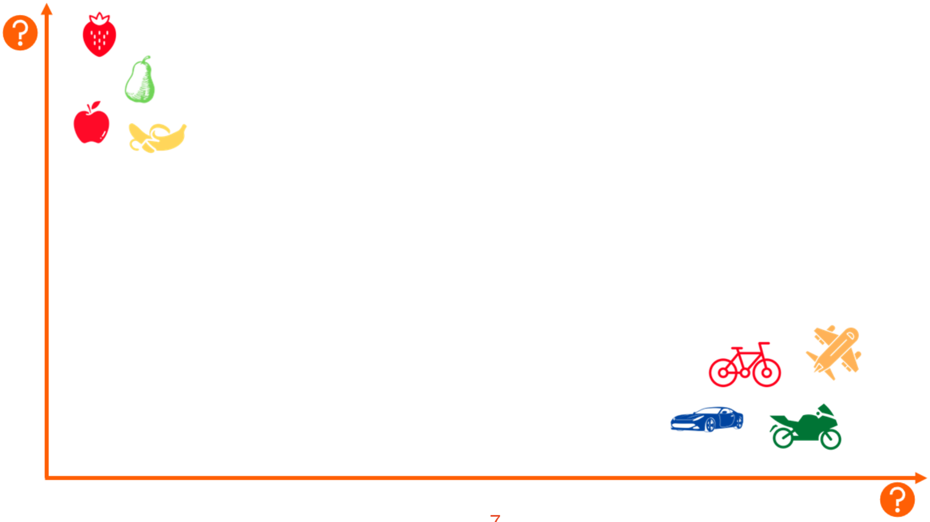
How should we group words?

What is Word Embedding?

How should we group words?

What is Word Embedding?

How should we group words?



What is Word Embedding?

How should we group words?

- In traditional NLP, we regard words as discrete symbols
- Words can be represented by one-hot vectors:

$$\begin{aligned} \text{banana} &= [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ \cdots\ 0\ 0] \\ \text{apple} &= \underbrace{[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \cdots\ 0\ 0]}_{\text{dimension: } V \text{(size of the vocabulary)}} \end{aligned}$$

- Challenge: How to compute similarity of two words?

Distributional Hypothesis

- Words that occur in similar contexts tend to have similar meanings
- J.R. Firth (1957): “You shall know a word by the company it keeps”

...government debt problems turning into **banking** crises as happened in 2009...
...saying that Europe needs unified **banking** regulation to replace the hodgepodge...
...the country has just given its **banking** system a shot in the arm...

These context words will represent **banking**

From Words to Numbers
Distributional hypothesis

caipirinha

C1: A glass of [ ] is on the table

C2: Everybody likes [ ]

C3: Don't have [ ] before you drive

C4: We make [ ] out of cachaça and lime

From Words to Numbers

Distributional hypothesis

Words that occur in similar contexts tend to have similar meanings

	C1	C2	C3	C4
caipirinha	1	1	1	1
loud	0	0	0	0
motor-oil	1	0	0	0
choices	0	1	0	0
wine	1	1	1	0

C1: A glass of [ ] is on the table

C2: Everybody likes [ ]

C3: Don't have [ ] before you drive

C4: We make [ ] out of cachaça and lime

ECONOMETRICS LAB
QUANTITATIVE RESEARCH @ UIUC

Words as Vectors

- Model of meaning focusing on similarity
- Each word is a vector
- Similar words are "nearby in space"
- A first solution: we can just use context vectors to represent the meaning of words
    - word-word co-occurrence matrix

ECONOMETRICS LAB
QUANTITATIVE RESEARCH @ UIUC

Words as Dense Vectors

- The vectors we get from word-word occurrence matrix are sparse (most are 0's) and long (vocabulary size)

- Alternative: we want to represent words as short (50-300 dimensional) and dense (real-valued) vectors

- The basis of all the modern NLP systems

Words as Dense Vectors

$$\text{employees} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 10.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix}$$

Why Dense Vectors?

- Short vectors are easier to use as features in ML systems
- Dense vectors may generalize better than storing explicit counts
- They do better at capturing synonymy
- word2vec and friends: learn the vectors!

- A text is a sequence of words
- For example, “this course is boring” has the following words:

w_1: this

w_2: course

w_3: is

w_4: boring

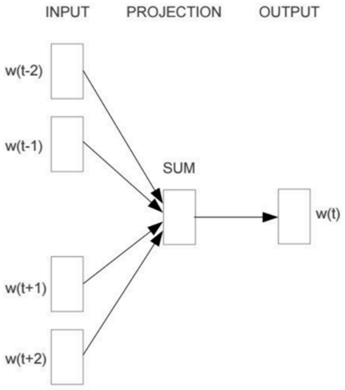
Word2Vec
Introduction

- For a given corpus of text, we define target and context words
- Consider the text

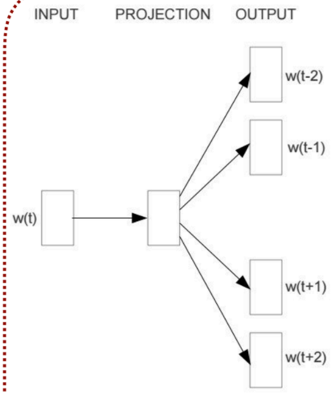
  … problems turning into banking crises as …
- In this example, into is a target word, and problems, turning, banking, and crises are context words
- Idea: learn context from target or vice-versa

Word2Vec

Architectures



Continuous Bag of Words (CBOW)



Skip-grams

- **CBOW (Continuous Bag of Words)**: Predicts target word from context
- **Skip-gram**: Predicts context words from the target word

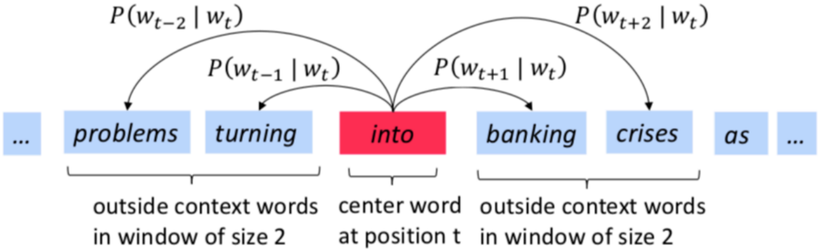
CBOW vs Skip-gram

Feature	CBOW	Skip-gram
Training speed	Faster	Slower
Accuracy on infrequent words	Lower	Higher
Input	Context words	Target word
Output	Target word	Context words

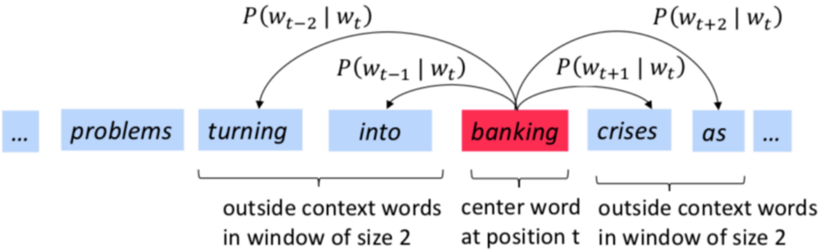
ECONOMETRICS LAB
QUANTITATIVE RESEARCH @ UIUC

Skip-Gram

- The idea: we want to use words to predict their context words
- Context: a fixed window of size $2m$



Skip-Gram



$P(w_{t-2} \mid w_t)$
$P(w_{t-1} \mid w_t)$
$P(w_{t+1} \mid w_t)$
$P(w_{t+2} \mid w_t)$
... problems turning into banking crises as ...
outside context words in window of size 2
center word at position t
outside context words in window of size 2

Skip-gram: objective function

- For each position $t = 1, 2, \ldots T$, predict context words within context size m, given center word w_j

$$\mathcal{L}(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} \mathsf{P}\left(w_{t+j} \mid w_t; \boldsymbol{\theta}\right)$$

where $\boldsymbol{\theta}$ is a vector of parameters defining the probability model

- The objective function $J(\theta)$ is the negative log-likelihood:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log \mathsf{P}\left(w_{t+j} \mid w_t; \boldsymbol{\theta}\right)$$

Skip-gram: Objective unction

How should we define $P\left(w_{t+j} \mid w_t; \theta\right)$?

- We have two sets of vectors for each word in the vocabulary

$$\boldsymbol{u}_i \in \mathbb{R}^d: \text{ embedding for target word } i$$
$$\boldsymbol{v}_j \in \mathbb{R}^d: \text{ embedding for context word } j$$

- Use inner product $\boldsymbol{u}_i' \boldsymbol{v}_j$ to measure how likely word i appears with context word j; the larger the better

$$\mathsf{P}\left(w_{t+j} \mid w_t; \boldsymbol{v}, \boldsymbol{u}\right) = \frac{\exp\left(\boldsymbol{u}_{w_t}' \boldsymbol{v}_{w_{t+j}}\right)}{\sum_{k \in V} \exp\left(\boldsymbol{u}_{w_t}' \boldsymbol{v}_k\right)}$$

Skip-gram with Negative Sampling (SGNS)

- Idea: recast the problem as binary classification
- Target word is a positive example
- All words not in context are negative
- To compute loss, pick K random words as negative examples

Skip-gram with Negative Sampling (SGNS)

positive examples

t	c	Output (Y)
apricot	tablespoon	1
apricot	of	1
apricot	jam	1
apricot	a	1

negative examples

t	c	Output (Y)
apricot	aardvark	0
apricot	my	0
apricot	where	0
apricot	coaxial	0
apricot	seven	0
apricot	forever	0
apricot	dear	0
apricot	if	0

Skip-gram with Negative Sampling (SGNS)

- Probability model:

$$\mathsf{P}(Y_{tc} = 1 | t, c) = \sigma(\boldsymbol{u}_t' \boldsymbol{v}_c)$$

where

$$\sigma(\boldsymbol{u}_t' \boldsymbol{v}_c) = \frac{1}{1 + \exp(-\boldsymbol{u}_t' \boldsymbol{v}_c)}$$

- The vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are the ones that minimizes

$$J(\boldsymbol{u}, \boldsymbol{v}) = -\sum_{t=1}^{T} \Bigg[\sum_{c \in \text{context}} \log \mathsf{P}(Y_{tc} = 1 | t, c) + \sum_{c \notin \text{context}} \log(1 - \mathsf{P}(Y_{tc} = 1 | t, c)) \Bigg]$$

ECONOMETRICS LAB
QUANTITATIVE RESEARCH @ UIUC

Word2Prices

embedding central bank communications for inflation prediction

- BIS Working Papers No 1253 by Araujo and co-authors
- Use word embeddings from ECB press conference statements to predict core inflation
- Compare different embedding techniques: Word2Vec, BERT, OpenAI
- Evaluate performance vs sentiment and placebo methods

- ECB monetary policy introductory statements (2002Q1 to 2023Q2)
- Core inflation measure excluding food and energy
- Forecasting horizon: 1 to 4 quarters ahead

$$\boldsymbol{Y}_t = \begin{bmatrix} \pi_t \\ m_t \end{bmatrix} = \boldsymbol{A} + \boldsymbol{B}(L)\boldsymbol{Y}_{t-1} + \boldsymbol{\eta}_t$$

- π_t: quarter-on-quarter inflation
- m_t: text-based embedding measure
- Estimated with Bayesian techniques

- Normal-Inverse Wishart priors
- Four lags: VAR(4)
- Recursive out-of-sample forecasting scheme
- Parameters updated each quarter

- Predicts target word using surrounding context
- Embedding dimension: 100
- Trained only on data available up to each forecast date
- No pre-cleaning; CBOW architecture

Forecast Accuracy: Full Sample

MSFE ratios vs AR benchmark (lower is better)

	$H = 1$	$H = 2$	$H = 3$	$H = 4$
Language Models				
Word2Vec	0.9685	0.9687	0.8593	0.8318
Bert	0.8075	0.7728	0.6756	0.6440
OpenAI	0.7746	0.7479	0.6714	0.7425
Placebo				
Count Inflation	1.0336	1.0835	1.1016	1.1091
Statement length	1.0157	1.0195	1.0049	1.0030
Sentiment				
Sent. Inflation	0.9408	0.9639	0.9389	0.9627
Sent. GC	0.9820	0.9805	0.9621	0.9695

Forecast Accuracy: Pre-COVID Period

MSFE ratios vs AR benchmark (lower is better)

	$H = 1$	$H = 2$	$H = 3$	$H = 4$
Language Models				
Word2Vec	0.9012	0.7698	0.7304	0.6969
Bert	0.8799	0.8781	0.8905	0.9098
OpenAI	0.8623	0.7950	0.7637	0.7595
Placebo				
Count Inflation	1.0280	1.0583	1.0916	1.1345
Statement length	1.0238	1.0597	1.0764	1.1048
Sentiment				
Sent. Inflation	0.9527	0.8862	0.8901	0.8987
Sent. GC	0.9799	0.9259	0.9251	0.9378

BMPE Forecast Encompassing Test

$$\pi_{t+h}^{\text{opt}} = \theta + \lambda \pi_{t+h}^{m} + (1-\lambda)\pi_{t+h}^{\text{BMPE}}$$

- Tests whether embedding-based forecasts add value beyond BMPE projections.
- $\lambda \approx 0$: embeddings redundant. $\lambda > 0$: embeddings add info.

Optimal Weights in Forecast Combination

Sample	H=1	H=2	H=3	H=4
Pre-COVID	0.37	0.55	0.56	0.52
Full Sample	0.22	0.04	0.26	0.68

Table Estimated λ values: weight on text forecast

Conclusions

- Word embeddings can enhance inflation forecasting from central bank text
- Word2Vec performs well, even vs state-of-the-art LLMs
- Text adds info beyond sentiment or length metrics
- Potential to complement internal projections like BMPE