Introduction to Deep Learning

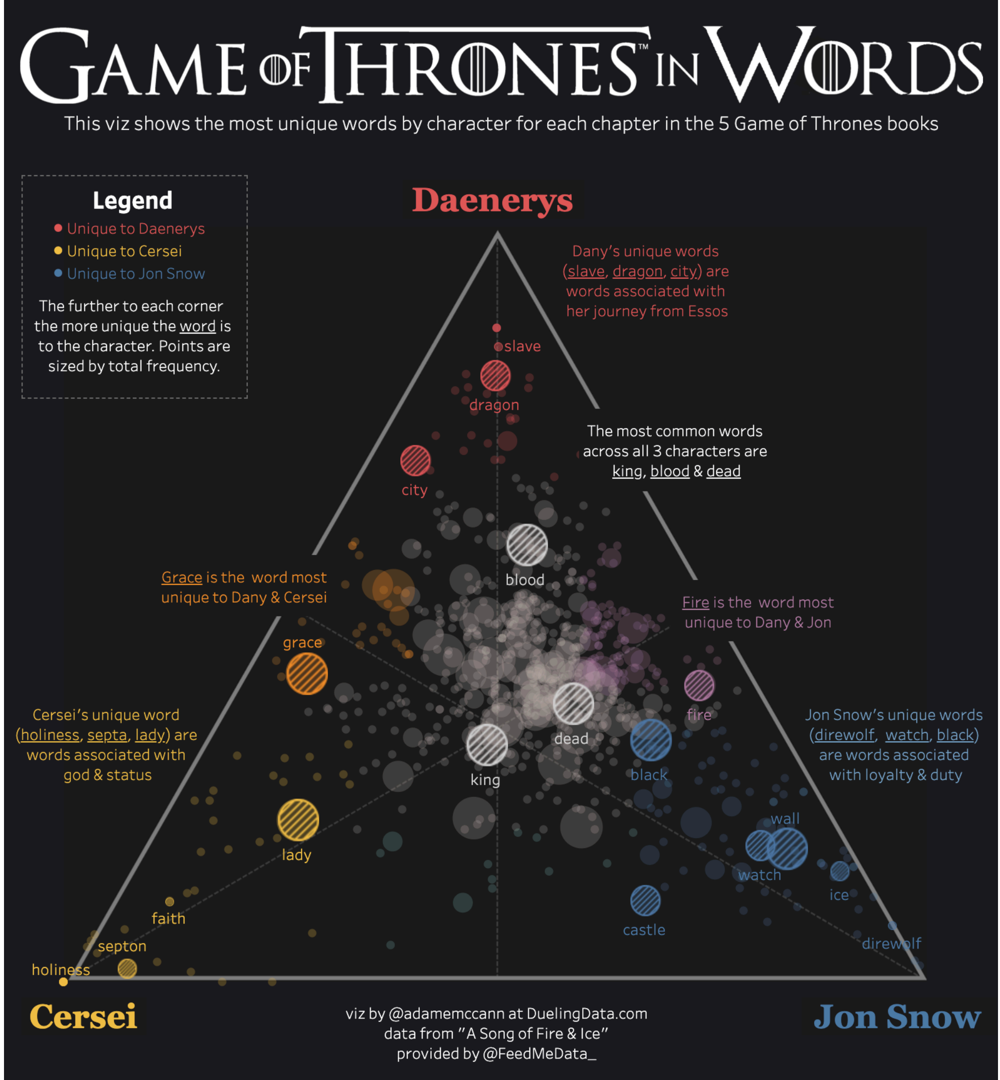
22. Embeddings, Word2wec, fastText, GloVe

STAT 157, Spring 2019, UC Berkeley

Alex Smola and Mu Li

courses.d2l.ai/berkeley-stat-157

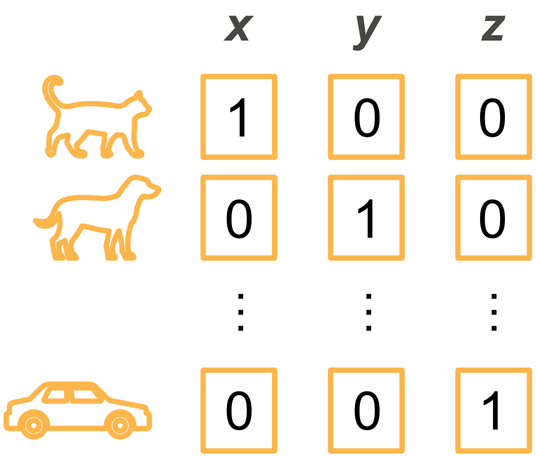
Word2Vec


GAME OF THRONES IN WORDS
This viz shows the most unique words by character for each chapter in the 5 Game of Thrones books
Legend
Unique to Daenerys
Unique to Cersei
Unique to Jon Snow
The further to each corner the more unique the word is to the character. Points are sized by total frequency.
Daenerys
Dany's unique words (slave, dragon, city) are words associated with her journey from Essos
slave
dragon
city
The most common words across all 3 characters are king, blood & dead
blood
Grace is the word most unique to Dany & Cersei
Fire is the word most unique to Dany & Jon
grace
fire
dead
king
black
Cersei's unique word (holiness, septa, lady) are words associated with god & status
Jon Snow's unique words (direwolf, watch, black) are words associated with loyalty & duty
lady
wall
watch
ice
castle
faith
septon
holiness
direwolf
Cersei
Jon Snow
viz by @adamemccann at DuelingData.com
data from "A Song of Fire & Ice"
provided by @FeedMeData_

Motivation

- One-hot vectors map objects/ words into fixed-length vectors
- These vectors only contain the identity information, not semantic meaning, e.g.

$$\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{z}, \mathbf{y} \rangle = 0$$

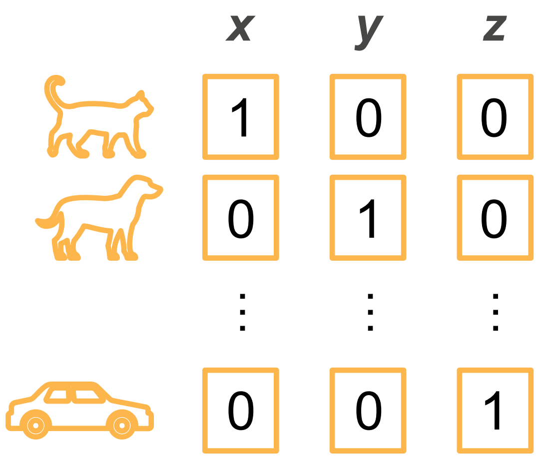


Word2vec

- Learn an embedding vector for each word
- Use $\langle \mathbf{x}, \mathbf{y} \rangle$ to measure the similarity

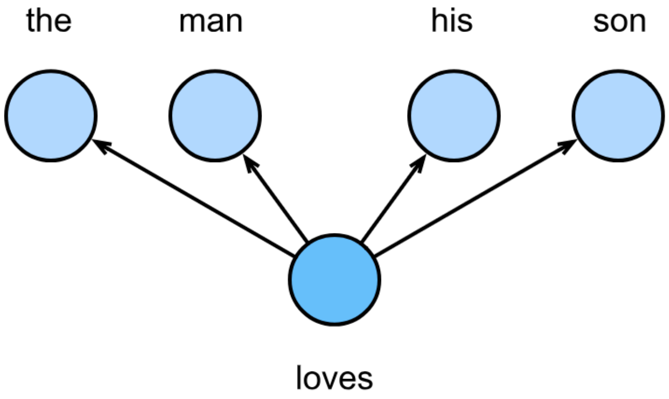
$$\langle \mathbf{x}, \mathbf{y} \rangle > \langle \mathbf{z}, \mathbf{y} \rangle$$

- Build a probability model
- Maximize the likelihood function to learn the model



The Skip-Gram Model

- A word can be used to generate the words surround it
- Given the center word, the context words are generated independently



$$\mathbb{P}(\text{"the"}, \text{"man"}, \text{"his"}, \text{"son"} \mid \text{"loves"})$$

$$= \mathbb{P}(\text{"the"} \mid \text{"loves"}) \cdot \mathbb{P}(\text{"man"} \mid \text{"loves"}) \cdot \mathbb{P}(\text{"his"} \mid \text{"loves"}) \cdot \mathbb{P}(\text{"son"} \mid \text{"loves"})$$

Likelihood Function

Summing over all words is too expensive

	Word	Embedding
Center	w_c	$\mathbf{v}_c \in \mathbb{R}^d$
Context	w_o	$\mathbf{u}_o \in \mathbb{R}^d$

$$\mathbb{P}(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathscr{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

$\mathscr{V}$: all context words

- Given length *T* sequence, context window *m*, the likelihood function:

$$\prod_{t=1}^{T} \prod_{-m \leq j \leq m,\ j \neq 0} \mathbb{P}(w^{(t+j)} \mid w^{(t)})$$

Negative Sampling

- Treat a center word and a context word appear in the same context window as an event

$$\mathbb{P}\left(D=1 \mid w_c, w_o\right)=\sigma\left(\mathbf{u}_c^T \mathbf{v}_o\right) \qquad \sigma(x)=\frac{1}{1+\exp(-x)}$$

- Change the likelihood function from $\prod_{t=1}^{T} \prod_{-m \leq j \leq m,\ j \neq 0} \mathbb{P}(w^{(t+j)} \mid w^{(t)})$ to

$$\prod_{t=1}^{T} \prod_{-m \leq j \leq m,\ j \neq 0} \mathbb{P}(D=1 \mid w^{(t)}, w^{(t+j)})$$

Naive solution: infinity

Negative Sampling

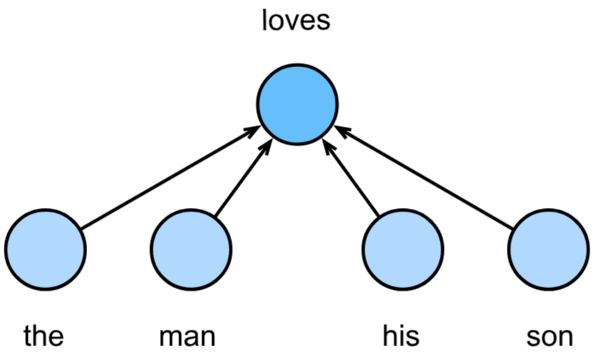
- Sample noise word w_n that doesn't appear in the window

$$\mathbb{P}\left(D=0 \mid w_c, w_n\right)=1-\sigma\left(\mathbf{u}_n^T \mathbf{v}_c\right)$$

- Add into the likelihood function as well
- Maximizing the likelihood equals to solve a binary classification problem with a binary logistic regression loss

Continuous Bag Of Words (CBOW)

- The center word is generated based on the context words



$$\mathbb{P}(\text{"loves"} \mid \text{"the"}, \text{"man"}, \text{"his"}, \text{"son"})$$

Likelihood Function

- Compute the probability

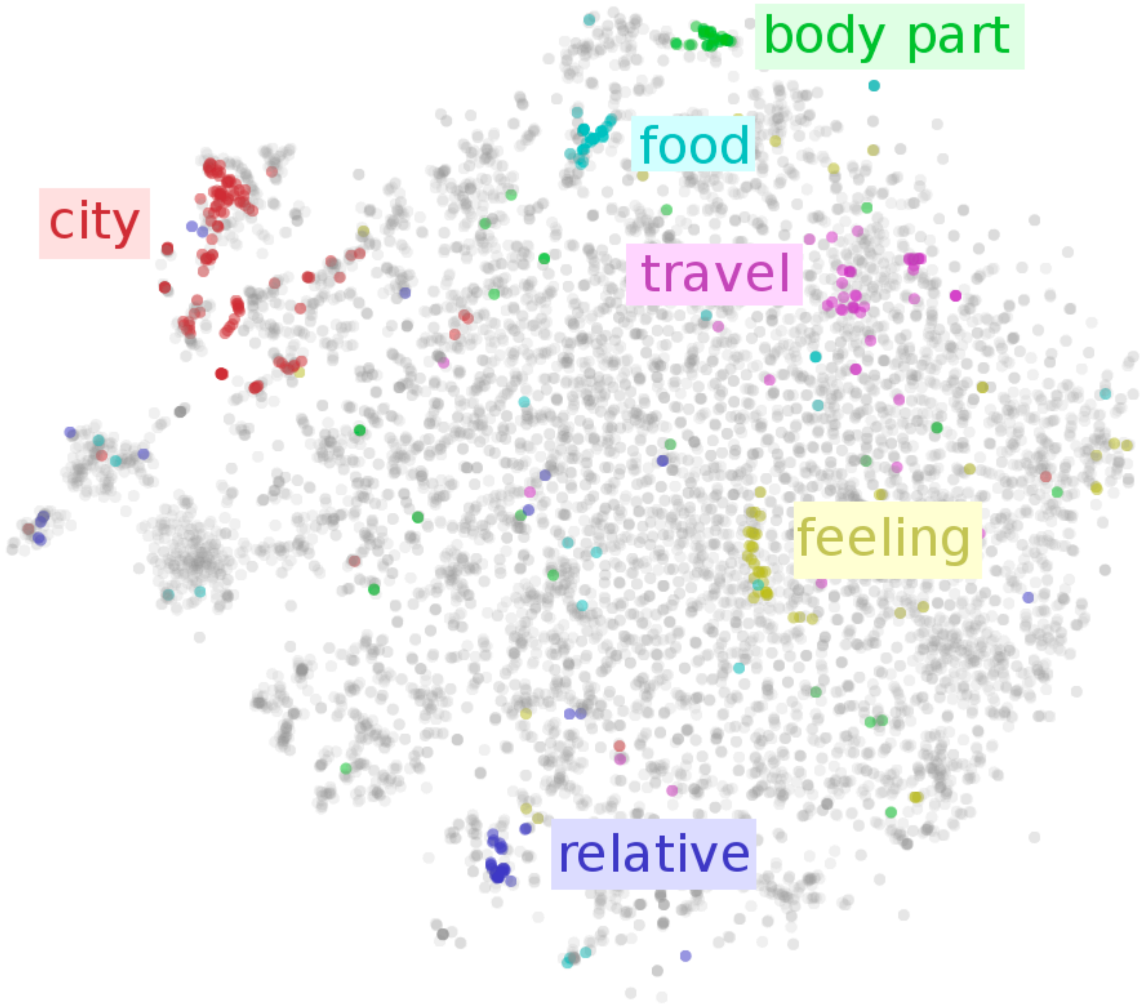
$$\mathbb{P}(w_c \mid w_{o_1}, \ldots, w_{o_{2m}}) = \frac{\exp\left(\frac{1}{2m}\mathbf{u}_c^\top(\mathbf{v}_{o_1} + \ldots + \mathbf{v}_{o_{2m}})\right)}{\sum_{i \in \mathcal{V}} \exp\left(\frac{1}{2m}\mathbf{u}_i^\top(\mathbf{v}_{o_1} + \ldots + \mathbf{v}_{o_{2m}})\right)}$$

- Likelihood

$$\prod_{t=1}^{T} \mathbb{P}(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)})$$

Code…

More Embedding Models



FastText

- English words usually have internal structures and formation methods
  - dog, dogs, dogcatcher
- Each center word is represented as a set of subwords
  - “where” -> “<where>” -> *n*-gram
  - n=3: “<wh", "whe", "her", "ere", “re>”
- Useful for long but infrequent words
  - e.g. pneumonoultramicroscopicsilicovolcanoconiosis

FastText

- For word *w*, $\mathcal{G}_w$ is the union of subwords with length from 3 to 6
- The center vector is then

$$\mathbf{u}_w = \sum_{g \in \mathcal{G}_w} \mathbf{u}_g$$

- The rest model is same as skip-gram

Word Embedding with Global Vectors (GloVe)

- Denote by $q_{ij} = \dfrac{\exp(\mathbf{u}_j^\top \mathbf{v}_i)}{\sum_{k \in \mathcal{V}} \exp(\mathbf{u}_k^\top \mathbf{v}_i)}$

- Rewrite the negative log-likelihood function of skip-gram

$$-\log \left[\prod_{t=1}^{T} \prod_{-m \leq j \leq m,\ j \neq 0} \mathbb{P}(w^{(t+j)} \mid w^{(t)}) \right]$$

- as $-\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij} \log q_{ij}$ with proper counts x_{ij}

Glove

- Further rewrite

$$-\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} x_{ij}\log\, q_{ij} = -\sum_{i\in\mathcal{V}} x_i \sum_{j\in\mathcal{V}} p_{ij}\log\, q_{ij}$$

Cross entropy

with $x_i = \sum_j x_{ij}, \quad p_{ij} = x_i/x_{ij}$

Glove

- Replace the cross entropy with a log square loss

$$\sum_{j \in \mathcal{V}} p_{ij} \log\, q_{ij} \to \sum_{j \in \mathcal{V}} (\log p_{ij} - \log q'_{ij})^2$$

with an easy to compute $q'_{ij} = \exp(\mathbf{u}_j^\top \mathbf{v}_i)$

- Add bias term for center and context words
- Replace the weights x_i with a monotone increasing function in [0,1]

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} h(x_{ij}) \left(\mathbf{u}_j^\top \mathbf{v}_i + b_i + c_j - \log x_{ij} \right)^2$$

Code…