L90 Practical: Part II

Helen Yannakoudakis[1]

November 29, 2017

[1] Adapted from Simone Teufel's slides

Procedure/Timeline

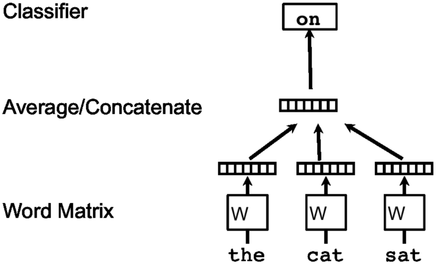
- Today:
  - How to develop the extension system (doc2vec)
  - How to write the final report
- Jan 17: Submit 4,000-word report on the extension system
  - With word count and pointer to running code please!

Doc2vec for Sentiment Analysis

- word2vec: learning neural word embeddings (Mikolov et al., 2013)
- doc2vec (Le and Mikolov, 2014):[2] embeddings for *sequences* of words
- Agnostic to granularity: sentence, paragraph, document
- Learned 'document' vector effective for various/some tasks, including sentiment analysis

[2] Or paragraph vectors, or document vectors ...

Task: predict the next word given the context



Optimisation objective:

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},\ldots,w_{t+k})$$

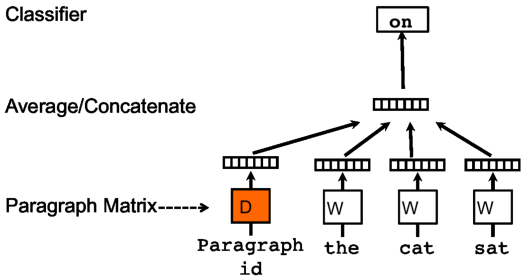
Softmax output layer:

$$p(w_t|w_{t-k},\ldots,w_{t+k}) = \frac{\exp y_{w_t}}{\sum_i \exp y_i}$$

$$y = b + U\,h(w_{t-k},\ldots,w_{t+k};W)$$

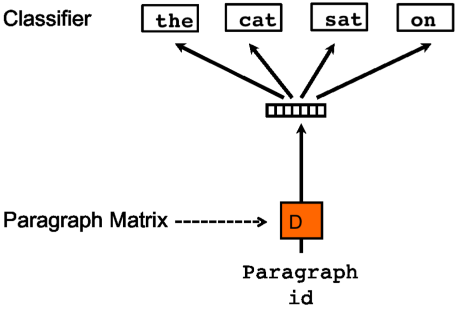
Images and formulas from paper though note inaccuracies...

Doc2vec: distributed memory (dm) architecture



- Add paragraph token: each paragraph mapped to a unique vector
- Paragraph vector now also contributes to the prediction task
  - Shared across all contexts from the same paragraph
- Works as a "memory" of context / topic

Doc2vec: distributed bag of words (dbow) architecture



Alternatively, train paragraph vector to predict words in a window (no word order); similar to Skip-gram model.

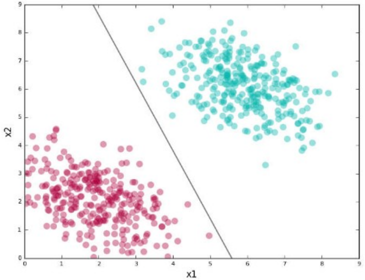
Doc2vec

- A number of available tools (e.g., gensim python library)
- Our level of granularity: document / review
- Parameters:
  - Training algorithm (dm, dbow)
  - The size of the feature vectors (e.g., 100 dimensions good enough for us)
  - Number of iterations / epochs (e.g., 10 or 20)
  - Context window
  - Hierarchical softmax (faster version) ...
- Lau and Baldwin (2016) have a careful investigation of doc2vec
- We provide pre-trained doc2vec models to help you verify your implementation: `/usr/groups/mphil/L90/models/`

Doc2vec: how can we use it for sentiment analysis?

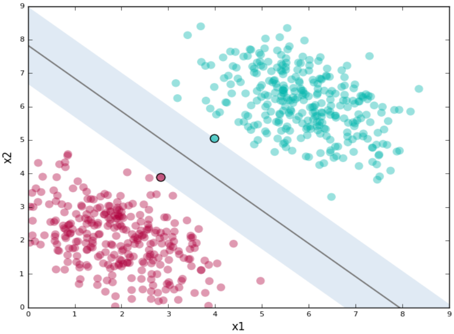
- Training: word vectors, weights, paragraph vectors (seen paragraphs)
- Testing: paragraph vectors inferred by gradient descending while keeping all else fixed (word vectors, weights)
- Vectors can then be used as features within a typical supervised machine learning framework (that's what we are doing here)
- For this practical, we will use Support Vector Machines

Support Vector Machines



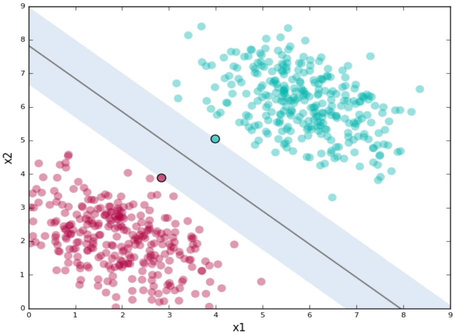
Images from: https://blog.statsbot.co/support-vector-machines-tutorial-c1618e635e93

Support Vector Machines



Images from: https://blog.statsbot.co/support-vector-machines-tutorial-c1618e635e93

Support Vector Machines



- SVM-Light: implementation of Support Vector Machines (Joachims, 1999)
  - Easy to use (just download the binaries and convert to SVM-Light format)
  - Make sure you normalise your input vectors!
    e.g., numpy.linalg.norm(vector)

Images from: https://blog.statsbot.co/support-vector-machines-tutorial-c1618e635e93

Goal of this Practical – “good science”

- Getting high numerical results isn’t everything – neither in this practical nor in science in general
- Good science means:
  - An interesting research question
  - Sound methodology
  - Insightful analysis (something non-obvious)

Goal of this Practical – “good science”

- Getting high numerical results isn’t everything – neither in this practical nor in science in general
- Good science means:
    - An interesting research question
    - Sound methodology
    - **Insightful analysis (something non-obvious)**

Insightful analysis

- Finding out what the model is really doing (visualisation, selected / targeted experimentation ...)
- E.g., see Lau and Baldwin (2016), and Li et al. (2015):
  - Are meaningfully similar documents close to each other?
  - Are document embeddings close in space to their most critical content words?
  - Do inferred embeddings (at a finer level of granularity perhaps) capture local compositionality?

Writing tips

- Only talk about doc2vec approach to sentiment analysis
- Make it look like a paper, including formatting (similar to first report)
- Use scientific language
  - Eradicate all forms of colloquial language
  - Mimic the author's voice in published papers

Writing tips

- Introduction: pretend this is not an L90 assignment but your own idea
- Reader has no pre-knowledge
- Describe your data / datasets
- Describe your methodology appropriately
  - Not too detailed (otherwise you look like a beginner)
  - Enough detail to allow somebody to reimplement your solution
  - Technical terms: use them – define them first
- Describe your numerical results (after your methods, clearly separated)
- Analyse your numerical results: what is the model really doing / not doing?