A competitive document representation with provable properties

Modern NLP pipelines combine low-dimensional distributed representations of text with deep learning models like LSTMs. Our goal is to reason formally about these systems using compressed sensing tools.

The Text Classification Pipeline

- **Bag-of-n-Grams (BonGs)**: Simple to implement, strong baseline, high-dimensional word-order, moderate performance, often ignored word-order, slow, can be beaten by BonGs.
- **LSTM Hidden States**: Simple to implement, low dimensional, long range dependencies, strong performance.
- **Past Linear Schemes**: e.g., sum of embeddings, SIF [10], simple to implement, localized word-order, moderate performance.
- **Skip-Thought Vectors**: Can be expressed as a linear combination of BonGs. Since $A$ preserves their inner products and the loss is Lipschitz, the loss of $A\hat{w}_{BonG}$ is thus bounded in terms of that of $\hat{w}_{BonG}$.

How well does our representation do on linear text classification?

**Case 1: Random Word Embeddings:** Using i.i.d. Random word embeddings as input our representations are provably as powerful as Bag-of-n-Grams for linear text classification. This yields a new theoretical result about LSTMs (below).

**Case 2: Pretrained Word Embeddings:** Using GloVe word embeddings our representations achieve state-of-the-art results on several text classification tasks:

**Theorem: LSTMs beat BonGs**

If $\ell$ is a convex Lipschitz loss and $D$ is a distribution on documents of length at most $T$ with optimal linear BonG classifier $w_{BonG}$ then for $\delta, \Omega = \Omega(\log T)$ one can initialize an $O(d)\text{-memory}$ LSTM such that with probability $1 - \delta$ the linear classifier $\hat{w}_{LSTM}$ trained over $m$ documents represented by the LSTM’s last hidden state satisfies

$$
\ell(\hat{w}_{LSTM}) \leq \ell_1(w_{BonG}) + O\left(\frac{\|w_{BonG}\|_1}{m} \log \frac{1}{\delta}\right)
$$

**Proof Sketch:** Using results from compressed sensing we can write $\hat{v}_{document} = A\hat{v}_{BonG}$, where the matrix $A$ preserves inner products of $T$-sparse vectors up to distortion $\varepsilon$ and $v_{BonG}$ is the document’s BonG vector. As $v_{BonG}$ can be computed by a low-memory LSTM, it suffices to show that learning is possible under compression [2].

1. The loss of learned classifier $\hat{w}_{BonG}$ is bounded in terms of that of the optimal classifier $w_{BonG}$.
2. $w_{BonG}$ can be expressed as a linear combination of BonGs. Since $A$ preserves their inner products and the loss is Lipschitz, the loss of $A\hat{w}_{BonG}$ is thus bounded in terms of that of $\hat{w}_{BonG}$.
3. The loss of learned classifier $\hat{w}_{LSTM}$ is bounded in terms of that of $A\hat{w}_{BonG}$.

What information does our representation encode?

**Case 1: Random Word Embeddings:** Guaranteed polynomial-time recovery of the Bag-of-n-Grams vector from our representation using $\ell_1$-minimization. Follows from the compressed sensing properties of random matrices.

**Case 2: Pretrained Word Embeddings:** Standard compressed sensing theory does not apply to GloVe/word2vec. Surprisingly, they encode Bag-of-Words vectors more efficiently than random embeddings, requiring fewer dimensions for recovery.

Empirical Observation

As a result of being trained on a large text corpus, word embeddings satisfy a weak compressed sensing condition that only holds for natural language documents. This leads to highly-efficient BoW recovery.

References