PAC-Bayes Analysis of Transferred Sentence Vectors

Kento Nozawa and Issei Sato
The University of Tokyo / RIKEN AIP

Overview

- Simple heuristic methods (i.e., averaging) are strong baselines, but they are not well understood theoretically.
- We formulate learning sentence vectors from pre-trained word vectors as transfer learning, then we analyze them with PAC-Bayes.

Learning word vectors

Goal: Finding maps from a word w to a vector $\mathbf{h}_w$ given word sequences $[w_1, \dots, w_T]$. For example, skip-gram with negative sampling minimizes the loss function defined by

$$L_{SG} = -\sum_{t=1}^{T} \Big[\sum_{w_c \in \mathcal{C}_t} \ln \sigma\big(\mathbf{h}_{w_t}^\top \mathbf{h}'_{w_c}\big) + \sum_{w_n \in \mathcal{NS}} \ln \sigma\big(-\mathbf{h}_{w_t}^\top \mathbf{h}'_{w_n}\big) \Big].$$

Note: $\mathcal{NS}$ is negative words, $\mathcal{C}$ is positive words, and $\mathbf{h}'$ is an output word vector.

Sentence vectors from pre-trained word vectors

Goal: Finding maps from a sentence $\mathcal{S}$ to a vector $\mathbf{h}_\mathcal{S}$ given sentences $\{\mathcal{S}_1, \dots, \mathcal{S}_N\}$ and pre-trained word vectors $\{\hat{\mathbf{h}}_w, \hat{\mathbf{h}}'_w \mid w \in \mathcal{V}\}$. The simple heuristic way is to average pre-trained word vectors of words appearing in the sentence $\mathcal{S}$, e.g.,

$$\mathbf{h}_\mathcal{S} = \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \hat{\mathbf{h}}_w \qquad \text{or} \qquad \mathbf{h}_\mathcal{S} = \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \frac{\hat{\mathbf{h}}_w + \hat{\mathbf{h}}'_w}{2}.$$

PAC-Bayes bound for transferred sentence vectors

We formulate learning word vectors and sentence vectors in terms of transfer learning;

- Source: minimizing the loss function of skip-gram with negative sampling by updating word vectors $\mathbf{h}$ and $\mathbf{h}'$.
- Target: minimizing a loss function by updating sentence vector $\mathbf{h}_\mathcal{S}$ with the fixed pre-trained word vectors $\hat{\mathbf{h}}$ and $\hat{\mathbf{h}}'$ given $\mathcal{S}$.

This formulation enables us to analyze generalization errors in learning sentence vector with PAC-Bayesian theory, which can consider transferability to a target hypothesis from a learned source hypothesis through prior knowledge.

Theorem 1. *Given a sentence $\mathcal{S}$, $\forall \lambda > 0$, with probability at least $1 - \delta$ over training samples $\mathcal{D}_\mathcal{S}$, $\forall h \in \mathcal{Q}_\mathcal{S}$,*

$$R_\mathcal{D}(\mathcal{Q}_\mathcal{S}) \le \mathbb{E}_{h \sim \mathcal{Q}_\mathcal{S}} \frac{1}{|\mathcal{D}_\mathcal{S}|} \sum_{i=1}^{|\mathcal{D}_\mathcal{S}|} l(\mathbf{x}_i, y_i, h) + \frac{\lambda}{2\sigma^2} \Big\| \mathbf{h}_\mathcal{S} - \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \hat{\mathbf{h}}'_w \Big\|_2^2 + C, \tag{1}$$

where C is a constant term that does not depend sentence vector $\mathbf{h}_\mathcal{S}$, and σ^2 is a variance parameter of prior and posterior.

Sentence vectors via L2 loss

- Output space: $\mathcal{Y} = \mathbb{R}^d$
- Hypothesis: $h = \mathbf{h}_\mathcal{S}$
- Loss function: $l(y = \hat{\mathbf{h}}_w, h) = \frac{1}{2} \|\mathbf{h}_\mathcal{S} - \hat{\mathbf{h}}_w\|_2^2$
- Hyper-parameter of PAC-Bayes bound: $\alpha = \lambda/\sigma^2$.

We obtain the closed form of $\mathbf{h}_\mathcal{S}$ from Eq. (1);

$$\mathbf{h}_\mathcal{S} = \frac{1}{(1+\alpha)|\mathcal{S}|} \sum_{w \in \mathcal{S}} \Big(\hat{\mathbf{h}}_w + \alpha \hat{\mathbf{h}}'_w \Big). \tag{2}$$

Corollary 1 (Average of Input Word Vectors). *The sentence vector $\mathbf{h}_\mathcal{S}$ estimated by minimizing Eq. (1) with $\alpha = 0$ is equivalent to $1/|\mathcal{S}| \sum_{w \in \mathcal{S}} \hat{\mathbf{h}}_w$.*

Corollary 2 (Average of Input and Output Word Vectors). *The averaged vector of the input and output word vectors, $(\hat{\mathbf{h}} + \hat{\mathbf{h}}')/2$, can improve the performance of downstream tasks [1]. This operation corresponds to the solution of Eq. (2) with $\alpha = 1$.*

We also derive another heuristic vector weighed by IDF, $\mathbf{h}_\mathcal{S} = \frac{1}{(1+\lambda) \sum_{w \in \mathcal{S}} \mathrm{IDF}(w)} \sum_{w \in \mathcal{S}} \mathrm{IDF}(w)(\hat{\mathbf{h}}_w + \lambda \hat{\mathbf{h}}'_w)$.

Sentence vectors via 0–1 loss

We define a new target task that is similar to the source task: Predicting whether sentence $\mathcal{S}$ contains word w_t.

- Output space: $\mathcal{Y} = \{-1, 1\}$
- Hypothesis: $h(\mathbf{x}_w) = \mathrm{sign}((\hat{\mathbf{H}} \mathbf{x}_w)^\top \mathbf{h}_\mathcal{S})$
- Loss function: $l(\mathbf{x}, y, h) = \mathbb{I}[h(\mathbf{x}) \ne y]$

In practice, we minimize the negative sampling based surrogate loss;

$$L = -\frac{1}{|\mathcal{S}|} \Big[\sum_{w \in \mathcal{S}} \ln \sigma \left(\hat{\mathbf{h}}_w^\top \mathbf{h}_\mathcal{S} \right) + \sum_{w_n \in \mathcal{NS}} \ln \sigma \left(-\hat{\mathbf{h}}_{w_n}^\top \mathbf{h}_\mathcal{S} \right) \Big] + \frac{\lambda}{2\sigma^2} \Big\| \mathbf{h}_\mathcal{S} - \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \hat{\mathbf{h}}'_w \Big\|_2^2. \tag{3}$$

Corollary 3 (Relationship to Paragraph Vector Models). *PV-DBoW [2] is the same to Eq. (3) with $\lambda = 0$ and without pre-trained word vectors.*

Table 1: Comparison of source tasks and target tasks in our transfer learning.

Model	Input x	Output y	Hypothesis h	Loss l
`Skip-gram`	w_t, w_c	Binary	$\sigma(\mathbf{h}_{w_t}^\top \mathbf{h}'_{w_c})$	Negative log
`Avg.`	**1**	$\mathbb{R}^d$	$\mathbf{h}_\mathcal{S}$	Root L2
`IDF-Avg.`	**1**	$\mathbb{R}^d$	$\mathbf{h}_\mathcal{S}$	Weighted Root L2
`Input-trans.`	$\mathbf{x}_w$	Binary	$\mathrm{sign}(\hat{\mathbf{h}}_w^\top \mathbf{h}_\mathcal{S})$	Zero-one
`Output-trans.`	$\mathbf{x}_w$	Binary	$\mathrm{sign}(\mathbf{h}_\mathcal{S}^\top \hat{\mathbf{h}}'_w)$	Zero-one

Experiments: Sentence classification

Settings:

- Classifier: Logistic regression
- Word vector models: `Skip-gram` and `CBoW`
- Source data: English Wikipedia
- Hyper-parameters:
  - $\sigma^2 = 1$
  - λ: searched in $\{10^{-2}, 10^{-1}, 1, 10\}$

Models of target tasks:

- `Avg.`: Eq. (2).
- `IDF-Avg.`: Averaging with IDF.
- `Output-trans.`: Minimize Eq. (3).
- `Input-trans.`: Swap roles of $\hat{\mathbf{h}}$ and $\hat{\mathbf{h}}'$ in Eq. (3).

Table 2: Test accuracy of sentence classification (averaged over three times).

Method	Source model	20news	IMDb	SUBJ
`Avg.` ($\alpha = 0$)	`Skip-gram`	**0.749** ± **0.000**	0.841 ± 0.000	**0.907** ± **0.001**
`Avg.` ($\alpha = 1$)	`Skip-gram`	**0.747** ± **0.002**	0.838 ± 0.000	0.905 ± 0.000
`Avg.` ($\alpha = 0$)	`CBoW`	0.733 ± 0.000	0.838 ± 0.000	0.905 ± 0.000
`Avg.` ($\alpha = 1$)	`CBoW`	0.737 ± 0.001	0.840 ± 0.000	0.904 ± 0.000
`IDF-Avg.` ($\lambda = 0$)	`Skip-gram`	0.735 ± 0.000	0.823 ± 0.001	**0.908** ± **0.001**
`IDF-Avg.` ($\lambda = 1$)	`Skip-gram`	0.732 ± 0.000	0.821 ± 0.000	0.905 ± 0.001
`IDF-Avg.` ($\lambda = 0$)	`CBoW`	0.726 ± 0.001	0.826 ± 0.000	0.903 ± 0.000
`IDF-Avg.` ($\lambda = 1$)	`CBoW`	0.723 ± 0.000	0.826 ± 0.001	0.904 ± 0.000
`Input-trans.`	`Skip-gram`	**0.749** ± **0.002**	**0.842** ± **0.000**	**0.908** ± **0.002**
`Input-trans.`	`CBoW`	0.717 ± 0.000	0.817 ± 0.001	**0.907** ± **0.001**
`Output-trans.`	`Skip-gram`	**0.749** ± **0.000**	**0.842** ± **0.000**	**0.908** ± **0.000**
`Output-trans.`	`CBoW`	0.734 ± 0.003	**0.841** ± **0.002**	**0.910** ± **0.003**

References

[1] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with LessonsLearned from Word Embeddings. *TACL*, 3:211–225, 2015.
[2] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *ICML*, 2015.