word2vec Parameter Learning Explained

Shigeaki ONODA

Graduate School of Information System Engineering Dept
t855005@st.pu-toyama.ac.jp

Tues., 12 January, 2016, Osaka Univ.

Introduction

Motivation

離散オブジェクトを数値として扱うことで和・差等の演算が可能となる.
自然言語をテンソルとして扱うためにはモデルとして以下が考案されている離散オブジェクト:
単語, 概念のような物理的に計測できる量を伴わず, 通常記号を用いて離散的に表現するもの

- n-gram 言語モデル

Agenda

- 言語モデルとは
- word2vec の概要
- 結論

言語モデルと n-gram

言語モデル

人間が扱う自然言語で書かれた文や文書が生成される確率をモデル化したもの.
例えば「おはようございます.　朝会を始めます.」には高い確率を与え,「おはようございます.
懲戒を始めます.」といった出現確率の低い文には低い確率を与える

n-gram モデル

隣り合って出現した n 単語のことを n-gram と呼ぶ.
nurture passes nature =>

- 1-gram: nurture, passes, nature
- 2-gram: nurture-passes, passes-nature
- 3-gram: nurture-passes-nature

このように 2 グラム以上だと前後の文脈を考慮したモデルとなる.

Bag of Words

1-gram の場合の文書ベクトル表現のこと単純に文書中のその単語の頻度を返す.
"nurture or nature? nurture passes nature."
だとその BoW は
$x^d = (nature, nurture, or, passes) = (2, 2, 1, 1)$
この手法でも言語処理ではある程度有効であるため, これを用いる研究も多い

one hot ベクトル

one-hot ベクトル

要素のうち一つだけ 1 のベクトル

$$x = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0.01 & 0.02 & 0.03 & 0.04 \\ 0.06 & 0.01 & 0.08 & 0.10 \\ 0.02 & 0.12 & 0.14 & 0.15 \\ 0.01 & 0.02 & 0.03 & 0.04 \\ 0.21 & 0.22 & 0.23 & 0.24 \end{pmatrix} * \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.02 \\ 0.01 \\ 0.12 \\ 0.02 \\ 0.22 \end{pmatrix} \tag{1}$$

言語処理における one hot

上の式の左行列を埋め込み行列とよび右辺のベクトルを埋め込みベクトルとよぶよって埋め込みベクトルが i 番目の単語の情報を格納している

記号からテンソル, テンソルから記号への変換

単語数と同じ次元ベクトル $\boldsymbol{o}$ を考えるそのベクトル中で最も大きな値になった要素番号（インデックス）を返す近似式は以下となる

$$softmax_a(\boldsymbol{o}) = \frac{1}{\exp(a\boldsymbol{o}) \cdot \mathbf{1}} \exp(a\boldsymbol{o}) \tag{2}$$

ニューラル言語モデル

ニューラル言語モデル

機械学習の権威ヒントンらが考案したニューラルネットを用いた言語モデルの総称 n-gram モデルと比べて言語モデルとしての性能が優れているただし計算量は n-gram に比べて膨大になる
=>GPGPU の発展によりこの問題が解消されつつあるので近年フォーカスされている

ニューラル言語モデルの定式化

単語の位置 t より前に出現した $t-a$ 単語を $\boldsymbol{Y_{[a,t-1]}} = (\boldsymbol{y_a}, \boldsymbol{y_{a+1}}, \cdots, \boldsymbol{y_{t-1}}$ と表しこれが文脈に相当する.
このとき文 Y の生成確率 $P(\mathrm{Y})$ を各単語が生成される条件付き確率の積でモデル化する. ただし Y は文中単語の one hot ベクトルが格納された集合である

$$P_{model}(\boldsymbol{Y}) = P(\boldsymbol{y_0}) \prod_{t-1}^{T+1} P_{model}(\boldsymbol{y_t} | \boldsymbol{Y_{[a,t-1]}}) \tag{3}$$

ニューラル言語モデルの応用

これを応用（ただ学習した埋め込み行列を得る）することで分散表現を獲得できる.
分散表現とは分布仮設（単語の意味は周囲の単語によって決まる）に基づくものであり,
離散オブジェクトを D 次元ベクトルで表現したもの
これにより単語という離散オブジェクトに対して和や差, 距離等のオペレーションが可能となる.

word2vec

word2vec とは

前述したニューラル言語モデルにおける分散表現の獲得は精度が高いが計算量の問題もあった
そこで計算量の問題を克服するため「最も簡単な言語モデル」として開発されたのが word2vec である.
word2vec とは単一の手法を指すわけでなく CBOW と skip-gram を合わせたモデル及びツールを指す. 対数双線形モデルとも呼ばれる.

CBOW

隠れ層の出力を計算するときに, CBOW は入力文脈ベクトルの平均を取る. 導出手順は以下をとる.

$$\boldsymbol{h} = \frac{1}{C}\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{x_1} + \boldsymbol{x_2} + \ldots + \boldsymbol{x_c}) = \frac{1}{C}(\boldsymbol{v_{w1}} + \boldsymbol{v_{w2}} + \ldots + \boldsymbol{v_{wc}})^{\mathrm{T}} \tag{4}$$

ここで C は文脈中の単語の数, $w1 \ldots wc$ は文脈の単語, $\boldsymbol{v_w}$ は単語 w の入力ベクトルとする損失関数は以下となる

$$E = -\log p(w_O|w_{I,1}, \ldots, w_{I,C}) \tag{5}$$

$$= -u_{j*} + \log \sum_{j=1}^{V} \exp(u_j) \tag{6}$$

$$= -\mathbf{v}_{w_O}^{\mathrm{T}} \cdot \boldsymbol{h} + \log \sum_{j=1}^{V} \exp(\mathbf{v}_{w_j}^{\mathrm{T}} \cdot \boldsymbol{h}) \tag{7}$$

CBOW

word2vec まとめ

基本的には更新式は通常のニューラルネットワークと同じだが文中の単語をまとめる部分だけ通常のネットワークと違う

Model (II)

結論 CBOW を用いると分散表現に特化したベクトル表現を得られる.

Theorem

[Thorem] for identifying parameters w_{hv}, $\boldsymbol{m}_{hv}$ and $\boldsymbol{\Sigma}_{hv}$

Vector $\boldsymbol{z}_h^k$ obeys mixture normal distribution

$$p(\boldsymbol{z}_h^k|\theta_h) = \sum_{v=1}^{V} w_{hv} N_d(\boldsymbol{z}_{hv}^k, \phi_{hv})$$

thus parameters $\theta_h = \{w_{hv}, \phi_{hv}\}$ can be derived by applying Expectation Maximization algorithms.

Because vecotor $\boldsymbol{z}_{hv}^k$ obeys d dimensional normal distribution with mean vector $\boldsymbol{m}_{hv}$, and covariance matrix $\boldsymbol{\Sigma}_{hv}$ as

$$p(\boldsymbol{z}_{hv}^k|\phi_{hv}) = N_d(\boldsymbol{z}_{hv}^k, \phi_{hv}).$$

From the result, each estimation $\hat{\theta}_h$ of parameter θ_h can be derived by maximizing log-liklifood function for incompete data,

$$L(\hat{\theta}_h) = \sum_{k=1}^{K} \log p(\boldsymbol{z}_h^k|\hat{\theta}_h).$$

Update rule of parameter $\hat{\theta}$ is given by EM algorithms maximizing conditional expectation of log-likelihood function for complete data,

$$\begin{aligned} Q(\hat{\theta}_h|\hat{\theta}_h^{(t)}) &= \mathrm{E}[L(\hat{\theta}_h, v)|\boldsymbol{z}_h^k, \hat{\theta}_h^{(t)}] \\ &= \sum_{k=1}^{K}\sum_{v=1}^{V} h_{hv}^{(t)}(\boldsymbol{z}_h^k) \log p(\boldsymbol{z}_h^k, v|\hat{\theta}_l). \end{aligned}$$

Proof of Theorem

They are iterative calculation form like

$$\begin{aligned}
\hat{w}_{hv}^{(t+1)} &= \frac{1}{K}\sum_{k=1}^{K} h_{hv}^{(t)}(\boldsymbol{z}_h^k), \\
\hat{\boldsymbol{m}}_{hv}^{(t+1)} &= \sum_{k=1}^{K} \gamma_{hv}^{c(t)} \boldsymbol{z}_h^k, \\
\hat{\boldsymbol{\Sigma}}_{hv}^{(t+1)} &= \sum_{k=1}^{K} \gamma_{hv}^{k(t)} (\boldsymbol{z}_h^k - \hat{\boldsymbol{m}}_{hv}^{(t+1)})(\boldsymbol{z}_h^k - \hat{\boldsymbol{m}}_{hv}^{(t+1)})^{\mathrm{T}}
\end{aligned}$$

where

$$\begin{aligned}
h_{hv}^{(t)}(\boldsymbol{z}_h^k) &= \frac{\hat{w}_{hv}^{(t)} N_d(\boldsymbol{z}_{hv}^k, \hat{\phi}_{hv}^{(t)})}{\sum_{v=1}^{V} \hat{w}_{hv}^{(t)} N_d(\boldsymbol{z}_h^k, \hat{\phi}_{hv}^{(t)})}, \\
\gamma_{hv}^{k(t)} &= \frac{h_{hv}^{(t)}(\boldsymbol{z}_h^k)}{\sum_{k=1}^{K} h_{hv}^{(t)}(\boldsymbol{z}_h^k)}.
\end{aligned}$$

□

Summary of Parameter Identification

- Observed data vector $\boldsymbol{z}_h^k$ consists of faculty vector $\boldsymbol{x}^k$ and choice result y_h^{k*}.
- Utility u_{hv}^k is given by conditional expectation $\mathrm{E}[y_{hv}^k | \boldsymbol{x}^k]$ at value factor level.
- Probability of observed data obeys contaminated normal distribution with θ_h.
- Model parameters θ_h can be identified by EM algorithms.

Algorithms

Algorithms of Parameter Estimation for Modeling Decision Making

1. Input data vector $\boldsymbol{z}_h^k$ which consists of faculty vector $\boldsymbol{x}^k$ and choice result y_h^{k*}.
2. Give the initial values of parameters $\theta_h^{(0)}$ that are set of $\{w_{hv}^{(0)}, \phi_{hv}^{(0)}\}$ where $\phi_{hv}^{(0)}$ is set of $\{\boldsymbol{m}_{hv}^{(0)}, \boldsymbol{\Sigma}_{hv}^{(0)}\}$.
3. Calculate $N_d(\boldsymbol{z}_h^k, \phi_{hv}^{(t)})$, $h_{hv}^{(t)}(\boldsymbol{z}_h^k)$ and $\gamma_{hv}^{k(t)}$.
4. Update parameters $\theta_h^{(t+1)}$.
5. If the change, $|\theta_h^{(t+1)} - \theta_h^{(t)}|$, becomes smaller than given threshold ϵ, then the procedure is terminated, otherwise return to $\boldsymbol{S}$tep 3.
6. From identified parameters θ_h, decision making model parameters $\boldsymbol{\beta}_{hv}^k$ for deriving utility can be estimated.

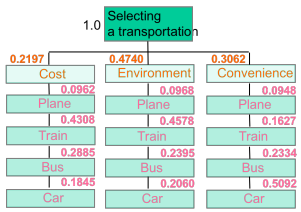

Fig. 9: Example of AHP
(Case in (K,V,R)=(4, 1, 3))

By incorporating similar concept of AHP (Analytic Hierarchy Process), structure of discrete choice model will be more clearly.

Conclusions

Model mapping from characteristic to selection

$$\mathrm{E}[y_h^k|\boldsymbol{x}^k] = \sum_{v=1}^{V} \lambda_{hv}(\boldsymbol{x}^k) u_{hv}^k \quad \text{where} \quad \lambda_{hv}(\boldsymbol{x}^k) = \frac{w_{hv} p(\boldsymbol{x}^k|\phi_{hv}^x)}{\sum_{v'=1}^{V} w_{hv'} p(\boldsymbol{x}^k|\phi_{hv'}^x)}$$

Consider ambiguity to decision making

$$u_{hv}^k = \boldsymbol{\beta}_{hv}^{k\,\mathrm{T}} \boldsymbol{x}'^k \quad \text{where} \quad \beta_{hv}^{kr} = f(\theta_h)$$

Algorithms of parameter estimation for modeling discrete choice

$\theta_h = \{w_{hv}, \phi_{hv}\}$ can be estimated by Expectation Maximization algorithms

Future plans

Apply the concept of discrete choice model.

- Development of attractive product and service on limited budget,
- Consensus building among stakeholder,
- Objective decision making and support mental training, etc.