Векторные представления текстов и графов

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ • 19 февраля 2021

Содержание

Дистрибутивная гипотеза и виды семантической близости слов

«Смысл слова определяется множеством его контекстов»

- Words that occur in the same contexts tend to have similar meanings [Harris, 1954].
- You shall know a word by the company it keeps [Firth, 1957].

Синтагматическая близость слов:
сочетаемость слов в одном контексте
(здание–строитель, кран–вода, функция–точка)

Парадигматическая близость слов:
взаимозаменяемость слов в одном контексте
(здание–дом, кран–смеситель, функция–отображение)

Z.Harris. Distributional structure. 1954.
J.R.Firth. A synopsis of linguistic theory 1930-1955. Oxford, 1957.
P.Turney, P.Pantel. From frequency to meaning: vector space models of semantics. 2010.

Формализация дистрибутивной гипотезы

Дано: текст $(w_1 \dots w_n)$, состоящий из слов словаря W

Найти: векторные представления слов $v_w \in \mathbb{R}^d$, так, чтобы близкие по смыслу слова имели близкие векторы

Модель CBOW (continuous bag-of-words) для вероятности слова w_i в заданном контексте $C_i = (w_{i-k} \dots w_{i-1} w_{i+1} \dots w_{i+k})$:

$$p(w_i = w | C_i) \;=\; \underset{w \in W}{\mathrm{SoftMax}} \langle u_w, v^{-i} \rangle,$$

$v^{-i} = \frac{1}{2k} \sum\limits_{w \in C_i} v_w$ — средний вектор слов из контекста C_i,
v_w — векторы предсказывающих слов,
u_w — вектор предсказываемого слова, в общем случае $u_w \neq v_w$.

Критерий максимума log-правдоподобия, $U, V \in \mathbb{R}^{|W| \times d}$:

$$\sum_{i=1}^{n} \log p(w_i | C_i) \to \max_{U,V}$$

Ещё одна формализация дистрибутивной гипотезы

Дано: текст $(w_1 \dots w_n)$, состоящий из слов словаря W

Найти: векторные представления слов $v_w \in \mathbb{R}^d$, так, чтобы близкие по смыслу слова имели близкие векторы

Модель Skip-gram для предсказания вероятности слов контекста $C_i = (w_{i-k} \dots w_{i-1} w_{i+1} \dots w_{i+k})$ по слову w_i:

$$p(w|w_i) \;=\; \underset{w \in W}{\mathrm{SoftMax}} \langle u_w, v_{w_i} \rangle \;\equiv\; \underset{w \in W}{\mathrm{norm}} \big(\exp\langle u_w, v_{w_i} \rangle\big),$$

v_w — вектор предсказывающего слова,
u_w — вектор предсказываемого слова, в общем случае $u_w \neq v_w$.

Критерий максимума log-правдоподобия, $U, V \in \mathbb{R}^{|W| \times d}$:

$$\sum_{i=1}^{n} \sum_{w \in C_i} \log p(w|w_i) \to \max_{U,V}$$

T.Mikolov et al. Efficient estimation of word representations in vector space, 2013.

Сравнение моделей CBOW и Skip-gram

- Различие — в структуре оптимизационного критерия:

$$\text{CBOW:} \quad \sum_{i=1}^{n} \log \underset{w_i \in W}{\mathrm{SoftMax}} \left(\frac{1}{2k} \sum_{c \in C_i} \langle u_{w_i}, v_c \rangle \right) \to \max_{U,V}$$

$$\text{Skip-gram:} \quad \sum_{i=1}^{n} \sum_{c \in C_i} \log \underset{c \in W}{\mathrm{SoftMax}} \langle u_c, v_{w_i} \rangle \to \max_{U,V}$$

- Skip-gram точнее моделирует вероятности редких слов
- Обе модели можно обучать с помощью SGD
- Обе модели реализованы в программе word2vec [Mikolov]
- Оба критерия трудно оптимизировать из-за SoftMax
- Что делать? Заменять либо SoftMax, либо критерий

T.Mikolov et al. Efficient estimation of word representations in vector space, 2013.

Иерархический SoftMax

Идея: заменить SoftMax на другую функцию потерь, сложность вычисления которой $O(\log|W|)$ вместо $O(|W|)$.

Предварительный этап:

- По словарю частот строится *бинарное дерево Хаффмана*
- Каждая внутренняя вершина n *хранит* вектор $u_n \in \mathbb{R}^d$
- Каждый лист *вычисляет* вектор v_w для слова $w \in \mathbb{R}^d$
- Модель переходов из внутренних вершин дерева:

$$\begin{aligned} \text{направо:} \quad & p(+1|n,w) = \sigma(\langle u_n, v_w \rangle) \\ \text{налево:} \quad & p(-1|n,w) = \sigma(-\langle u_n, v_w \rangle) = 1 - p(+1|n,w) \end{aligned}$$

Обучаются векторы u_n во внутренних вершинах дерева

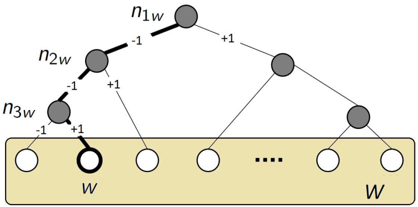
Иерархический SoftMax: обучение модели

Модель $p(w|w_i)$, гарантирующая нормировку $\sum_w p(w|w_i) = 1$:

$$p(w|w_i) = \prod_{j=1}^{\ell(w)} p\bigl(\beta_{jw}|n_{jw}, w_i\bigr) = \prod_{j=1}^{\ell(w)} \sigma\bigl(\beta_{jw}\langle u_{n_{jw}}, v_{w_i}\rangle\bigr)$$

где $\ell(w)$ — длина пути к листу w,
n_{jw} — j-я внутренняя вершина на пути к листу w,
$\beta_{jw} \in \{-1, +1\}$ — поворот из j-й вершины на пути к w.

Пример: $p(w|w_i) = p(-1|n_{1w}, w_i)\, p(-1|n_{2w}, w_i)\, p(+1|n_{3w}, w_i)$



Подмена задачи: классификация пар слов на два класса

Критерий log-loss для SGNS (Skip-gram Negative Sampling):

$$\sum_{i=1}^{n} \sum_{w \in C_i} \Big(\log p(+1|w, w_i) + \log p(-1|\bar{w}, w_i) \Big) \to \max_{U,V}$$

где $p(y|w, w_i) = \sigma\big(y\langle u_w, v_{w_i}\rangle\big)$ — модель классификации, $y = \pm 1$;
$y = +1$, если пара слов (w, w_i) находится в общем контексте;
$y = -1$, если пара слов (w, w_i) не находится в общем контексте;
$\bar{w} \sim p(w)^{3/4}$ сэмплируется из $W \backslash C_i$ в методе SG.

Эвристики и прочие замечания:

- Dynamic window: случайный выбор $k \sim [3..10]$
- Итоговые векторы слов: $\alpha v_w + (1 - \alpha) u_w$
- Приём NS применяют, когда не хватает второго класса
- Что делать со словами, которые встречаются впервые?

Связь word2vec с матричными разложениями

d — размерность векторов слов v_w и u_w
$V = (v_w)_{W \times d}$ — матрица предсказывающих векторов слов
$U = (u_w)_{W \times d}$ — матрица предсказываемых векторов слов

SGNS строит матричное разложение $P \approx UV^{\mathsf{T}}$ матрицы Shifted PMI (Point-wise Mutual Information):

$$P_{ab} = \ln \frac{n_{ab} n}{n_a n_b} - \ln k,$$

n_{ab} — частота пары слов a, b в окне $\pm k$ слов,
n_a, n_b — число пар с участием слова a и b соответственно,
n — число всех пар слов в коллекции.

В качестве эвристики используют также Shifted Positive PMI:

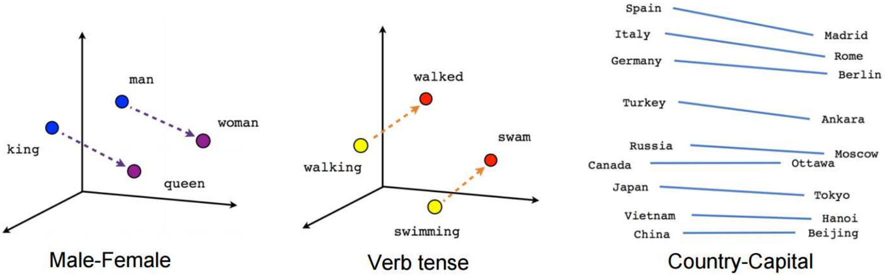
$$P_{ab}^{+} = \left(\ln \frac{n_{ab} n}{n_a n_b} - \ln k \right)_{+}.$$

O.Levy, Y.Goldberg. Neural word embedding as implicit matrix factorization. 2014.

Проверка на задачах семантической близости и аналогии слов

Задача семантической близости слов:
по выборке пар слов (a, b) оценивается корреляция Спирмена между $\cos(v_a, v_b)$ и экспертными оценками близости $y(a, b)$

Задача семантической аналогии слов:
по трём словам угадать четвёртое



Модель векторных представлений FastText

Идея: векторное представление слова w определяется как сумма векторов всех его буквенных n-грамм $G(w)$:

$$u_w = \sum_{g \in G(w)} u_g$$

В Skip-gram вместо векторов слов u_w обучаются векторы u_g

Пример: $G(\text{дармолюб}) = \{\text{<да}, \text{арм}, \text{рмо}, \text{мол}, \text{олю}, \text{люб}, \text{юб>}\}$

Преимущества:

- Это решает проблемы новых слов и слов с опечатками
- Подходит для обработки текстов социальных медиа
- Словарь 2- и 3-грамм обычно меньше словаря W
- Существует много предобученных моделей

Bojanowski et al. Enriching word vectors with subword information. 2016.

Модели векторных представлений для текстов и графов

word2vec: *эмбединги* (векторные представления) слов
T.Mikolov et al. Efficient estimation of word representations in vector space. 2013.

paragraph2vec: эмбединги фрагментов или документов
Q.Le, T.Mikolov. Distributed representations of sentences and documents. 2014.

sent2vec: эмбединги предложений
M.Pagliardini et al. Unsupervised learning of sentence embeddings using compositional n-gram features. 2017.

FastText: эмбединги символьных n-грамм
`https://github.com/facebookresearch/fastText`

node2vec: эмбединги вершин графа
A.Grover, J.Leskovec. Node2vec: scalable feature learning for networks. 2016.

graph2vec: более общие эмбединги на графах
A.Narayanan et al. Graph2vec: learning distributed representations of graphs. 2017.

StarSpace: эмбединги чего угодно от Facebook AI Research
L.Wu, A.Fisch, S.Chopra, K.Adams, A.B.J.Weston. StarSpace: embed all the things! 2018.

BERT: эмбединги фраз и предложений от Google AI Language
J.Devlin et al. BERT: pre-training of deep bidirectional transformers for language understanding. 2018.

GPT-3: эмбединги, предобученные по 570Gb текстов от OpenAI
T.B.Brown et al. Language Models are Few-Shot Learners. 2020.

Многомерное шкалирование (multidimensional scaling, MDS)

Дано: $(i, j) \in E$ — выборка рёбер графа $\langle V, E\rangle$,
R_{ij} — расстояния между вершинами ребра (i, j).
Например, в IsoMAP R_{ij} — длина кратчайшего пути по графу.

Найти: векторные представления вершин $z_i \in \mathbb{R}^d$, так, чтобы близкие (по графу) вершины имели близкие векторы.

Критерий стресса (stress):

$$\sum_{(i,j)\in E} w(R_{ij})\bigl(\rho(z_i, z_j) - R_{ij}\bigr)^2 \to \min_Z, \quad Z \in \mathbb{R}^{V\times d},$$

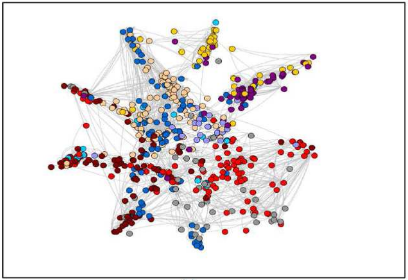
где $\rho(z_i, z_j) = \|z_i - z_j\|$ — обычно евклидово расстояние,
$w(R_{ij})$ — веса (какие расстояния важнее, большие или малые).

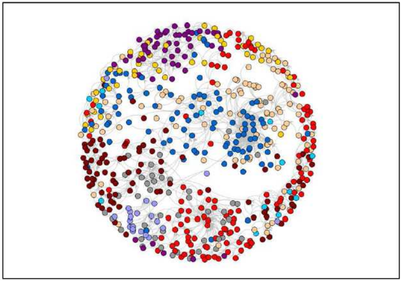
Обычно решается методом стохастического градиента (SG).

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

Многомерное шкалирование для визуализации данных

При $d = 2$ осуществляется проекция выборки на плоскость





- Используется для визуализации кластерных структур
- Форму облака точек можно настраивать весами и метрикой
- Недостаток — искажения неизбежны
- Наиболее популярный метод для визуализации — t-SNE

Laurens van der Maaten, Geoffrey Hinton. Visualizing data using t-SNE. 2008

Метод векторного представления соседства (Stochastic Neighbor Embedding, SNE)

Дано: исходные точки $x_i \in \mathbb{R}^n$, $i = 1, \dots, \ell$
Найти: точки на карте-проекции $y_i \in \mathbb{R}^d$, $i = 1, \dots, \ell$, $d \ll n$
Критерий: расстояния $\|y_i - y_j\|$ близки к исходным $\|x_i - x_j\|$

Вероятностная модель события «j является соседом i»
на основе перенормированных гауссовских распределений:

$$p(j|i) = \underset{j \neq i}{\mathrm{norm}} \exp\left(-\frac{1}{2\sigma_i^2}\|x_i - x_j\|^2\right)$$ — в исходном пространстве;

$$q(j|i) = \underset{j \neq i}{\mathrm{norm}} \exp\left(-\|y_i - y_j\|^2\right)$$ — в пространстве проекции;

где $p(j) = \underset{j}{\mathrm{norm}}(z_j) = \frac{z_j}{\sum_k z_k}$ — операция нормировки вектора.

Максимизация правдоподобия (стохастическим градиентом):

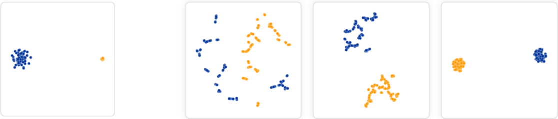
$$\sum_i \sum_{j \neq i} p(j|i) \ln q(j|i) \;\to\; \max_{\{y_i\}}$$

Преимущества метода SNE

- Преобразование расстояний в вероятности устраняет дисбалансы между большими и малыми расстояниями
- Дисбаланс между точками с большой и малой плотностью соседей выравнивается настройкой σ_i по перплексии

$H(i) = -\sum_j p(j|i) \log_2 p(j|i)$ — энтропия распределения $p(j|i)$;
$2^{H(i)}$ — перплексия = «эффективное число соседей у x_i»
(если $p(j|i) = \frac{1}{k}$, то $2^{H(i)} = k$); обычно перплексия = 5..50.

Выбор перплексии может существенно влиять на вид проекции:



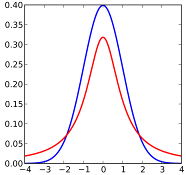
Original Perplexity: 2 Perplexity: 5 Perplexity: 30

G.E.Hinton, S.T.Roweis. Stochastic Neighbor Embedding. 2002.

Вероятностная модель t-SNE: два усовершенствования SNE

Проблема скученности в SNE: окрестность вмещает гораздо больше точек в n-мерном пространстве, чем в d-мерном

- Использование t-распределения Стьюдента с более тяжёлым хвостом и симметричного совместного распределения $q(i,j)$:



$$q(i,j) = \underset{(i,j):\, i \neq j}{\text{norm}} \left(1 + \|y_i - y_j\|^2\right)^{-1}$$

- Использование совместного распределения $p(i,j)$:

$$p(i,j) = \frac{1}{2\ell}\bigl(p(j|i) + p(i|j)\bigr)$$

Максимизация правдоподобия (стохастическим градиентом):

$$\sum_{(i,j):\, j \neq i} p(i,j) \ln q(i,j) \to \max_{\{y_i\}}$$

L.J.P. van der Maaten, G.Hinton. Visualizing data using t-SNE. 2008

Матричные разложения (graph factorization)

Дано: $(i,j) \in E$ — выборка рёбер графа $\langle V, E\rangle$,
S_{ij} — близость между вершинами ребра (i,j).
Например, $S_{ij} = [(i,j) \in E]$ — матрица смежности вершин.

Найти: векторные представления вершин, так, чтобы близкие (по графу) вершины имели близкие векторы.

Критерий для неориентированного графа (S симметрична):

$$\sum_{(i,j)\in E} \bigl(\langle z_i, z_j\rangle - S_{ij}\bigr)^2 \to \min_{Z}, \quad Z \in \mathbb{R}^{V\times d}$$

Критерий для ориентированного графа (S несимметрична):

$$\sum_{(i,j)\in E} \bigl(\langle \varphi_i, \theta_j\rangle - S_{ij}\bigr)^2 \to \min_{\Phi,\Theta}, \quad \Phi, \Theta \in \mathbb{R}^{V\times d}$$

Обычно решается методом стохастического градиента (SG).

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

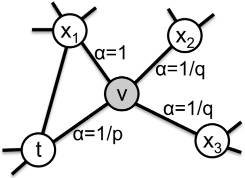
Модель случайных блужданий

Обобщение Skip-gram (текст ведь тоже граф, простая цепь):

$$\sum_{i\in V}\Bigl(\sum_{j\in C_i}\log\sigma(\langle\varphi_i,\theta_j\rangle)+\sum_{j\in\bar{C}_i}\log\sigma(-\langle\varphi_i,\theta_j\rangle)\Bigr)\to\max_{\Phi,\Theta},$$

C_i — окрестность («контекст») вершины i, сэмплируемая случайным блужданием длины k (DeepWalk, node2vec),
$\bar{C}_i$ — вершины, далёкие от i, сэмплируемые $j\sim p(j)^{3/4}$

Параметризация случайных блужданий:
вероятность $p(v\to w)$ после перехода $t\to v$
$p\downarrow\ q\uparrow$ — ближе к поиску в ширину (BFS)
$p\uparrow\ q\downarrow$ — ближе к поиску в глубину (DFS)



B.Perozzi et al. DeepWalk: online learning of social representations. SIGKDD-2014.
A.Grover, J.Leskovec. Node2vec: scalable feature learning for networks. SIGKDD-2016.

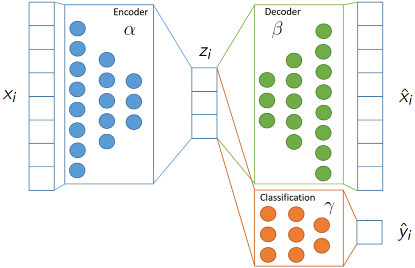
Напоминание. Автокодировщики для обучения с учителем

Данные: неразмеченные $(x_i)_{i=1}^{\ell}$, размеченные $(x_i, y_i)_{i=\ell+1}^{\ell+k}$

Совместное обучение кодировщика, декодировщика и предсказательной модели (классификации, регрессии или др.):

$$\sum_{i=1}^{\ell} \mathscr{L}(g(f(x_i,\alpha),\beta),x_i) + \lambda \sum_{i=\ell+1}^{\ell+k} \tilde{\mathscr{L}}(\hat{y}(f(x_i,\alpha),\gamma),y_i) \to \min_{\alpha,\beta,\gamma}$$

$z_i = f(x_i, \alpha)$ — кодировщик
$\hat{x}_i = g(z_i, \beta)$ — декодировщик
$\hat{y}_i = \hat{y}(z_i, \gamma)$ — классификатор



Функции потерь:
$\mathscr{L}(\hat{x}_i, x_i)$ — реконструкция
$\tilde{\mathscr{L}}(\hat{y}_i, y_i)$ — предсказание

Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders. 2020

Векторные представления графов как автокодировщики

Все рассмотренные выше методы векторных представлений графов суть автокодировщики данных о рёбрах:

- многомерное шкалирование: $R_{ij} \to \|z_i - z_j\|$
- SNE и t-SNE: $p(i,j) \to q(i,j)$
- матричные разложения: $S_{ij} \to \langle \varphi_i, \theta_j \rangle$

Вход кодировщика:

- W_{ij} — данные о ребре графа (i,j)

Выход кодировщика:

- векторные представления вершин z_i

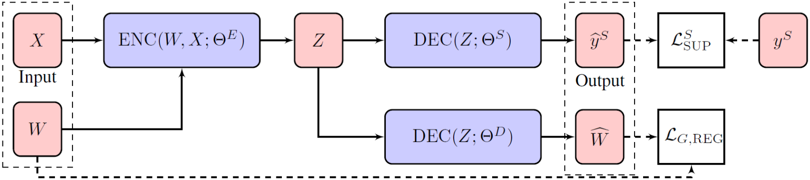
Выход декодировщика:

- аппроксимация $\hat{W}_{ij}$, вычисляемая по (z_i, z_j)

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

GraphEDM: обобщённый автокодировщик на графах

Graph Encoder Decoder Model — обобщает более 30 моделей:



$W \in \mathbb{R}^{V \times V}$ — входные данные о рёбрах
$X \in \mathbb{R}^{V \times n}$ — входные данные о вершинах, признаковые описания
$Z \in \mathbb{R}^{V \times d}$ — векторные представления вершин графа
$\mathrm{DEC}(Z; \Theta^D)$ — декодер, реконструирующий данные о рёбрах
$\mathrm{DEC}(Z; \Theta^S)$ — декодер, решающий supervised-задачу
y^S — (semi-)supervised данные о вершинах или рёбрах
$\mathcal{L}$ — функции потерь

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

Графовые нейронные сети (Graph Neural Network, GNN)

Один из вариантов кодировщика в GraphEDM — рекуррентная сеть с передачей сообщений по графу (Message Passing):

$$z_i^t = \sum_{j \in N(i)} f(x_i, x_j, z_j^{t-1}; \alpha)$$

$$\hat{y}_i^t = \hat{y}(x_i, z_i^t; \gamma)$$

t — номер итерации
x_i — вектор признакового описания вершины i
$N(i)$ — множество соседних вершин вершины i
f — многослойная нейронная сеть (используются различные архитектуры, в частности, свёрточные и рекуррентные)

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020
Ziwei Zhang, Peng Cui and Wenwu Zhu. Deep learning on graphs: A survey. 2020
Zonghan Wu et al. A comprehensive survey on graph neural networks. 2019
Jie Zhou et al. Graph neural networks: A review of methods and applications. 2019

Резюме

- Синтез векторных представлений (эмбедингов) — это
  — *обучение представлений* (Representation Learning)
  — *генерация признаков* (Feature Generation)
  — векторизация сложно структурированных данных
- Многокритериальная информативность эмбедингов:
  — качество реконструкции объекта по эмбедингу
  — качество предсказательной модели
- Эмбединги графов обобщают многие задачи векторизации текстов, дискретных сигналов, изображений и др.