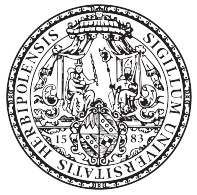

Machine Learning for Complex Networks
SoSe 2024

Prof. Dr. Ingo Scholtes
Chair of Informatics XV
University of Würzburg

Exercise Sheet 09

Published: July 3, 2024
Due: July 10, 2024
Total points: 10

Please upload your solutions to WueCampus as a scanned document (image format or pdf), a typesetted PDF document, and/or as a `jupyter` notebook.

1. Using `word2vec` for Deep Representation Learning on Graphs

The Skip-gram model was introduced in the first `word2vec` paper as a log-linear model. Its maximum log-likelihood objective can be expressed as

$$L = \frac{1}{T}\sum_{t=1}^{T}\sum_{-j_c \leq j \leq j_c, c \neq 0} \log q\left(w_{t+j}|w_t\right) \tag{1}$$

where $q\left(w_{t+j}|w_t\right)$ is the softmax σ of the inner product of $\vec{w}_t$ and $\vec{c}_{t+j}$, the word and context embeddings for word w_t and context w_{t+j}, respectively.

$$q\left(w_{t+j}|w_t\right) = \sigma\left(\vec{w}_i \cdot \vec{c}_{t+j}\right) = \frac{e^{\vec{w}_t \cdot \vec{c}_{t+j}}}{\sum_{k=1}^{K} e^{\vec{w}_t \cdot \vec{c}_k}}$$

This particular form of the softmax function is that of multinomial logistic regression – a log linear model.

(a) One way to fit the logistic / softmax function to data is to use cross-entropy. Cross-entropy measures the log expectation of distribution q with respect to distribution p, and can be formulated as 2P

$$H\left(p, q\right) = -E_p\left[\log q\right]$$

where $E_p[\log q]$ is the expected value $\log q$ with respect to the distribution p, or alternatively as the entropy of p plus the Kullback-Leibler divergence $D_{\mathrm{KL}}(p \parallel q)$ of p from q.

$$H\left(p, q\right) = H\left(p\right) + D_{\mathrm{KL}}\left(p \parallel q\right)$$

Show that these two definitions of the cross-entropy are equivalent (refer to Exercise Sheet 04 for a definition of the KL divergence).

(b) When used as a loss function, the cross-entropy takes the names *cross-entropy loss*, log loss, or *logistic loss*. The cross-entropy loss has a unique relationship to logistic regression. For example, the gradient of the cross-entropy loss for logistic regression is the same as the gradient of the squared error loss for linear regression. We can minimize this loss function with gradient descent in the neural networks discussed in the lectures. 2P

It is often useful to instead frame the problem as maximizing an objective, instead of minimizing a loss. Maximum likelihood estimation aims to maximize the *average log likelihood* of all possible

outcomes. The estimated probability of outcome x is $q(x)$ while its true probability is $p(x)$. For N conditionally independent trials, the likelood of the parameters β of $q(x)$ is given by the probability of x given the estimated $q(x)$

$$\mathcal{L} = \prod_x \left(q(x)\right)^{Np(x)}$$

Show that the average log likelihood is equivalent to the negative cross-entropy loss. What does this mean for the Skip-gram model?

(c) Consider an undirected graph with no self-loops or multi-edges, with adjacency matrix A. Write down the probability $\Pr(i \to j)$ of walking from node i to node j in an infinite random walk. 1P

(d) We can rewrite the objective of the Skip-gram model (Equation 1) to consider the frequency $\Pr(w, c)$ that a word-context pair (w, c) occurs in an infinitely large corpus. 1P

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{-j_c \le j \le j_c, j\neq 0} \log q\left(w_{t+j}|w_t\right) = \sum_{w\in V_w} \sum_{c\in V_c} \Pr(w,c) \log q\left(c|w\right)$$

Consider a simplified version of DeepWalk, where an infinite random walk is used to discover contexts / neighborhoods of only size 1. The maximum likelihood objective function is then

$$L = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \log q\left(w_{t+1}|w_t\right) = \sum_{h,l\in V} \Pr(h \to l) \log \sigma\left(\vec{w}_h \cdot \vec{c}_l\right) = \sum_{h,l\in V} \Pr(h \to l) \log \sigma\left(x_{hl}\right)$$

where $x_{hl} = \vec{w}_h \cdot \vec{c}_l$, and V is the set of vertex indices. Consider its corresponding cross-entropy loss H and argue that

$$\frac{\partial}{\partial x_{ij}} H = -\frac{\partial}{\partial x_{ij}} \sum_{h,l\in V} \delta_{ih} \Pr(h \to l) \log \sigma\left(\vec{w}_h \cdot \vec{c}_l\right) = -\frac{\partial}{\partial x_{ij}} \sum_{l\in V} \Pr(i \to l) \log \sigma\left(\vec{w}_i \cdot \vec{c}_l\right)$$

(e) Evaluate the derivate and substitute in your solution for $\Pr(i \to j)$. Thereby solve for the $\sigma\left(\vec{w}_i \cdot \vec{c}_j\right)$ that minimizes H. Considering your solution, argue the utility of the softmax function σ. Defining $\sigma_{ij} = \sigma\left(\vec{w}_i \cdot \vec{c}_j\right)$, the following derivative may be useful: 2P

$$\frac{\partial}{\partial x_{ij}} \log \sigma_{il} = \delta_{jl} - \sigma_{ij}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.

(f) $x_{ij} = \vec{w}_i \cdot \vec{c}_j$ can be considered as entries in the matrix multiplication $\vec{w}_i \cdot \vec{c}_j = \sum_{k=1}^{d} W_{ik} C_{kj} = M_{ij}$. The matrices W and C are known as the word and context embeddings, respectively, and it is W that is usually used as the resulting word (node) embeddings for the Skip-gram model (DeepWalk / node2vec). Argue what this simplified version of DeepWalk implicitly factorizes, and argue the utility of the Skip-gram model / similar NLP methods for embedding walks on graphs vs [any form of] explicit matrix factorization. 2P