Appendix: Distributional Semantics meets Multi-Label Learning

SGNS as Implicit SPPMI factorization

The SGNS (Mikolov et al. 2013) objective is as follows:

$$\mathbb{O}_i = \sum_{j \in S_i} \log(\sigma(K_{ij})) + \sum_{k \sim P_D}^{M} \mathbb{E}_{k \sim P_D}[\log(\sigma(-K_{ik}))] \tag{13}$$

where, $P_D = \frac{(\#k)^{0.75}}{\#D}$, D is collection of all word-context pairs and K_{ij} represent dot-product similarity between the embeddings of a given word (i) and context (j).

Here, $\#k$ represent total number of word-context pairs with context (k).

$$\mathbb{O}_{\{i,j\}} = \log(\sigma(K_{ij})) + \sum_{k \sim P_D}^{\frac{M}{|S|}} \mathbb{E}_{k \sim P_D}[\log(\sigma(-K_{ik}))] \tag{14}$$

$$\mathbb{E}_{k \sim P_D}[\log(\sigma(-K_{ik}))] = \sum_{k \sim P_D} \frac{(\#k)^{0.75}}{\#D} \log(\sigma(-K_{ik})) \tag{15}$$

$$\begin{aligned} \mathbb{E}_{k \sim P_D}[\log(\sigma(-K_{ik}))] = & \frac{(\#j)^{0.75}}{\#D} \log(\sigma(-K_{ij})) \\ & + \sum_{k \sim P_D \& k \neq j} \frac{(\#k)^{0.75}}{\#D} \log(\sigma(-K_{ik})) \end{aligned} \tag{16}$$

Therefore,

$$\mathbb{E}_{j \sim P_D}[\log(\sigma(-K_{ij}))] = \frac{(\#j)^{0.75}}{\#D} \log(\sigma(-K_{ij})) \tag{17}$$

$$\mathbb{O}_{\{i,j\}} = \log(\sigma(K_{ij})) + \frac{M}{|S|} \frac{(\#j)^{0.75}}{\#D} \log(\sigma(-K_{ij})) \tag{18}$$

Let γK_{ij} = x, then

$$\nabla_x \mathbb{O}_{\{i,j\}} = \sigma(-x) - \frac{M}{|S|} \frac{(\#j)^{0.75}}{\#D} \sigma(x) \tag{19}$$

equating $\nabla_x \mathbb{J}_{\{i,j\}}$ to 0, we get :

$$e^{2x} - \left(\frac{1}{\frac{M}{|S|} \frac{(\#j)^{0.75}}{\#D}} - 1 \right) e^x - \left(\frac{1}{\frac{M}{|S|} \frac{(\#j)^{0.75}}{\#D}} \right) = 0 \tag{20}$$

If we define $y = e^x$, this equation becomes a quadratic equation of y, which has two solutions, y =- 1 (which is invalid given the definition of y) and

$$y = \frac{1}{\frac{M}{|S|} \frac{(\#j)^{0.75}}{\#D}} = \frac{\#D * |S|}{M * (\#j)^{0.75}} \tag{21}$$

Substituting y with e^x and x with K_{ij} reveals :

$$K_{ij} = \log\left(\frac{\#D * |S|}{M * (\#j)^{0.75}} \right) \tag{22}$$

Here $|S| = \#(i,j)$ and $M = \mu\#(i)$ i.e. μ proportion of total number of times label vector (i) appear with others.

$$K_{ij} = \log\left(\frac{\#(i,j)(\#D)}{\#(i)(\#j)^{0.75}} \right) - log(\mu) \tag{23}$$

$$K_{ij} = \log\left(\frac{P(i,j)}{P(i)P(j)} \right) - log(\mu) \tag{24}$$

Here P(i,j),P(i) and P(j) represent probability of co-occurrences of $\{i,j\}$, occurrence of i and occurrence of j respectively,
Therefore,

$$K_{ij} = \mathsf{PMI}_{ij} - \log(\mu) = \log(P(i|j)) - log(\mu) \tag{25}$$

Note that PMI^+ is inconsistent, therefore we used the sparse and consistent positive PMI (PPMI) metric, in which all negative values and nan are replaced by 0:

$$\mathsf{PPMI}_{ij} = \max(\mathsf{PMI}_{ij}, 0) \tag{26}$$

Here, PMI is point wise mutual information and PPMI is positive point wise mutual information. Similarity of two $\{i,j\}$ is more influenced by the positive neighbor they share than by the negative neighbor they share as *uninformative* i.e. 0 value. Hence, SGNS objective can be cast into a weighted matrix factorization problem, seeking the optimal lower d-dimensional factorization of the matrix SPPMI under a metric which pays more for deviations on frequent $\#(i,j)$ pairs than deviations on infrequent ones.

Using a similar derivation, it can be shown that noise-contrastive estimation (NCE) which is alternative to (SGNS) can be cast as factorization of (shifted) log-conditional-probability matrix

$$K_{ij} = \log\left(\frac{\#(i,j)}{(\#j)} \right) - \log(\mu) \tag{27}$$

Gradient Computation

Gradient of objective 4 w.r.t to V i.e. $\nabla_V \mathbb{O}_i$ is :

$$\begin{aligned} \nabla_V \mathbb{O}_i = & \sum_{j : \mathcal{N}_k(\mathbf{y}_i)} \sigma(-K_{ij}) \nabla_V K_{ij} \\ & - \frac{n_-}{n} \sum_{j'} \sigma(K_{ij'}) \nabla_V K_{ij'} \end{aligned} \tag{28}$$

Table 5: Dataset Statistics

Dataset	Feature	Label	Train	Test
Bibtex (Katakis, Tsoumakas, and Vlahavas 2008; Prabhu and Varma 2014)	1836	159	4880	2515
Delicious (Tsoumakas, Katakis, and Vlahavas 2008; Prabhu and Varma 2014)	500	983	12920	3185
EURLex-4K (Loza Mencía and Fürnkranz 2008; Prabhu and Varma 2014)	5000	3993	15539	3809
rcv1v2 (Lewis et al. 2004; Prabhu and Varma 2014)	47236	101	3000	3000
Delicious-200K (Tsoumakas, Katakis, and Vlahavas 2008; Bhatia et al. 2015)	782585	205443	196606	100095
MediaMill (Snoek et al. 2006; Bhatia et al. 2015)	120	101	30993	12914
Wiki10-31K (Bhatia et al. 2015; Zubiaga 2012)	101938	30938	14146	6616
AmazonCat-13K (McAuley and Leskovec 2013)	203882	13330	1186239	306782
WikiLSHTC-325 (Prabhu and Varma 2014; Bhatia et al. 2015)	1617899	325056	1778351	587084
Wikipedia-500K	2381304	501070	1813391	783743
Amazon-670K (McAuley and Leskovec 2013; Bhatia et al. 2015)	135909	670091	490449	153025

$$\begin{aligned}\nabla_V \langle \mathbf{z}_i \mathbf{z}_j \rangle &= \nabla_V (V\mathbf{x}_i)\mathbf{z}_j + \nabla_V (V\mathbf{x}_j)\mathbf{z}_i \\ &= \langle \mathbf{z}_i \mathbf{x}_j^T \rangle + \langle \mathbf{z}_j \mathbf{x}_i^T \rangle = V(\langle \mathbf{x}_i \mathbf{x}_j^T + \mathbf{x}_j \mathbf{x}_i^T \rangle)\end{aligned} \tag{29}$$

$$\begin{aligned}\nabla_V \frac{1}{\|\mathbf{z}_i\|} = \nabla_V \mathbf{z}_i \mathbf{z}_i^T{}^{\frac{-1}{2}} &= \frac{-1}{2}\mathbf{z}_i \mathbf{z}_i^T{}^{\frac{-3}{2}} \nabla_V \mathbf{z}_i \mathbf{z}_i^T \\ &= \frac{-1}{2}\mathbf{z}_i \mathbf{z}_i^T{}^{\frac{-3}{2}} \mathbf{x}_i \mathbf{z}_i^T\end{aligned} \tag{30}$$

$$\begin{aligned}\nabla_V \frac{1}{\|\mathbf{z}_j\|} = \nabla_V \mathbf{z}_j \mathbf{z}_j^T{}^{\frac{-1}{2}} &= \frac{-1}{2}\mathbf{z}_j \mathbf{z}_j^T{}^{\frac{-3}{2}} \nabla_V \mathbf{z}_j \mathbf{z}_j^T \\ &= \frac{-1}{2}\mathbf{z}_j \mathbf{z}_j^T{}^{\frac{-3}{2}} \mathbf{x}_j \mathbf{z}_j^T\end{aligned} \tag{31}$$

Let,

$$a = \mathbf{z}_i^T \mathbf{z}_j, b = \frac{1}{\|\mathbf{z}_i\|}, c = \frac{1}{\|\mathbf{z}_j\|} \tag{32}$$

Thus, we have,

$$\begin{aligned}\nabla_V K_{ij} = -ab^3 c\mathbf{z}_i(\mathbf{x}_i)^T - abc^3 \mathbf{z}_j(\mathbf{x}_j)^T \\ + bc(\mathbf{z}_i \mathbf{x}_j^T + \mathbf{z}_j \mathbf{x}_i^T)\end{aligned} \tag{33}$$

Similarity to graph embedding:

Graph embedding algorithm Grarep(Cao, Lu, and Xu 2015) and DNGR(Cao, Lu, and Xu 2016), which aim to learn embeddings for each node of graph are based on similar idea of weighted matrix factorization of shifted PPMImatrix of input adjacency matrix. These node embedding outperforms has outperform previous state-of-the-art in the task of community classification/detection.

Dataset Statistics:

We have provided the details datasets statistics in Table 5.