Word2Vec is only a special case of Kernel Correspondence Analysis and Kernels for Natural Language Processing

Hirotaka Niitsuma
Okayama University

Word2Vec is a special case of Kernel Correspondence Analysis and Kernels for Natural Language Processing

Hirotaka Niitsuma
Department of Computer Science
Okayama University
Okayama, Japan
niitsuma@cs.okayama-u.ac.jp

Minho Lee
Graduate School of Science and Technology
Kyungpook National University
Daegu, South Korea
mholee@knu.ac.kr

August 16, 2016

Abstract

We show that Correspondence Analysis (CA) is equivalent to defining the Gini-index with appropriately scaled one-hot encoding. Using this relation, we introduce nonlinear kernel extension of CA. The extended CA gives well-known analysis for categorical data (CD) and natural language processing by specializing kernels. For example, our formulation can give G-test, skip-gram with negative-sampling (SGNS), and GloVe as special cases. We introduce two kernels for natural language processing based on our formulation: a stop word (SW) kernel and a word similarity (WS) kernel. The SW kernel is a system introducing appropriate weights for SW. The WS kernel enables the use of WS test data as training data for vector space representations of words. We show that these kernels enhance accuracy when training data are insufficiently numerous.

1 Introduction

Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) are well-known multivariate data reduction techniques. These two methods are analyses of a correlation or variance–covariance matrix of a multivariate quantitative dataset. CA is an analysis of CD.

Also, CA is a statistical visualization method for picturing the associations between the levels of a two-way contingency table. To illustrate CA, one might consider the contingency table presented in Table 1, which shows data well known as Fisher's data[4] for eye and hair color of people in Caithness, Scotland. The CA of these data yield the graphical display presented in Figure 1. Figure 1 shows the correspondence between eye and hair color.

PCA is the eigenvalue decomposition of a covariance matrix. CCA is the singular value decomposition (SVD) based on the matrix representing correlation with modification of the average and the inhomogeneousness. CA can be regarded as SVD of the covariance of CD with modification of the average and inhomogeneousness like that of CCA. And the Gini-index is another way of addressing the variance of CD. Okada [11, 10] reported that defining the Gini-index using one-hot encoding on appropriately rotated space yields reasonable values for the covariance. This research describes the rotated Gini-index with modifying inhomogeneousness is equivalent to CA.

Picca et al. [13] introduced nonlinear kernel extension of CA. Niitsuma et al. [9] introduced nonlinear kernel extension of the rotated Gini-index. Equivalence between CA and the modified Gini-index can define more general nonlinear kernel extension of CA. Calling this nonlinear kernel extension kernel Correspondence Analysis (KCA), we note that gives well-known analysis for CD and natural language processing by specializing kernels. For example, our KCA can give G-test, skip-gram with negative-sampling (SGNS)[8, 6], and GloVe[12] as a special case.

We apply KCA to natural language processing. Especially we specifically examine vector representations of words. Mikolov et al.[8] introduced dense vector representations of words referred to as `word2vec`. The vector representations give meaning subtraction or addition operations of the vectors. For example, the analogy "King is to queen as man is to woman." is encoded in the vector space by the following vector equation: king − queen = man − woman. Sometimes CA also has a similar property. For example, we can see the vector equation: (eye:dark) − (eye:medium) = (hair:dark) − (hair:medium) in Figure 1.

Levy et al. [6] showed that skip-gram with negative-sampling (SGNS), which is one of `word2vec`, can be regarded as analysis of the two-way contingency table in which the row represents a word, and the column represents context. [6] showed that SGNS is the matrix factorization of pointwise mutual information (PMI) of the contingency table. Actually, PMI can be regarded as a log-scale representation of a contingency table. We formulate this mechanism within KCA by providing the appropriate kernels.

Table 1: Fisher's data.

		x^{hair}				
		fair	red	medium	dark	black
x^{eye}	blue	326	38	241	110	3
	light	688	116	584	188	4
	medium	343	84	909	412	26
	dark	98	48	403	681	85

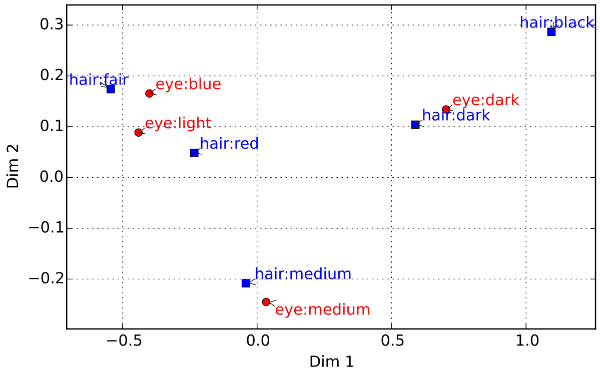

Figure 1: Visualizing Fisher's data.

2 Correspondence Analysis

Actually, CA is a generalized singular value decomposition (GSVD) [1] based on a contingency table for two categorical variables such that

$$\breve{U}\hat{S}\breve{V}^t = \Xi \tag{1}$$

$$\Xi = \mathbf{N}/n - \mathbf{r}\mathbf{c}^t/n^2 \tag{2}$$

$$\breve{U}^t D(\mathbf{r})^{-1}\breve{U} = \breve{V}^t D(\mathbf{c})^{-1}\breve{V} = \mathbf{E} \tag{3}$$

where $\mathbf{N}$ is an $n^r \times n^c$ contingency table, with entries n_{ij} giving the frequency with which row categorical variables $x^r = i$ occur together with column categorical variable $x^c = j$. In addition, $x^r \in \{x^r_1, x^r_2, ..., x^r_{n^r}\}$ is the categorical variable representing the row side of the contingency table $\mathbf{N}$. $x^c \in \{x^c_1, x^c_2, ..., x^c_{n^c}\}$ represents the categorical variable representing the column side. $\mathbf{r} = \mathbf{N1}$ denotes the vector of row marginals, $\mathbf{c} = \mathbf{N}^t\mathbf{1}$ is the vector of column marginals and $n = \mathbf{1}^t\mathbf{c} = \mathbf{1}^t r$ stands for the total number of observations. where $\mathbf{1} = (1, 1, 1...)^t$. $\mathbf{E}$ denotes an identity matrix. $D(\mathbf{v})$ stands for the diagonal matrix for which diagonal entries are the components of vector $\mathbf{v}$. $\breve{U}\hat{S}\breve{V}^t = \Xi$ is GSVD of Ξ so

[1] http://forrest.psych.unc.edu/research/vista-frames/pdf/chap11.pdf

that $\breve{U}^t D(\mathbf{r})^{-1}\breve{U} = \breve{V}^t D(\mathbf{c})^{-1}\breve{V} = \mathbf{E}$. Usually CA use the following matrix for the visualize instead of using $\breve{U}$ and $\breve{V}$.

$$F = D(\mathbf{r})^{-1}\breve{U}\hat{S} \tag{4}$$

$$G = D(\mathbf{c})^{-1}\breve{V}\hat{S} \tag{5}$$

For example, let us consider the contingency table presented in Table 1 . The table presents the joint population distribution of the categorical variable for eye color $x^{eye} \in \{$ fair red medium dark black$\}$ and the categorical variable for hair color $x^{hair} \in \{$ blue light medium dark$\}$. For these data, $\mathbf{r} = (718, 1580, 1774, 1315)^t$, $\mathbf{c} = (1455, 286, 2137, 1391, 118)^t$, and $n = 5387$. CA using Fisher data is associated with the following GSVD.

$$\frac{1}{n}\mathbf{N} - \frac{1}{n^2}\mathbf{rc}^t = \frac{1}{5387}\begin{bmatrix} 326 & 38 & 241 & 110 & 3 \\ 688 & 116 & 584 & 188 & 4 \\ 343 & 84 & 909 & 412 & 26 \\ 98 & 48 & 403 & 681 & 85 \end{bmatrix} + \frac{1}{5387^2}\begin{pmatrix} 718 \\ 1580 \\ 1774 \\ 1315 \end{pmatrix}\begin{pmatrix} 1455 \\ 286 \\ 2137 \\ 1391 \\ 118 \end{pmatrix}^t . \tag{6}$$

3 Gini's definition of Variance and its Extension

A contingency table can be constructed using the matrix product of two indicator matrices of

$$\mathbf{N} = H^{r\,t} H^c \tag{7}$$

where

$$\begin{aligned} H^r &= [\mathbf{e}^r(x^r(a_1)), \mathbf{e}^r(x^r(a_2)), ..., \mathbf{e}^r(x^r(a_n))]^t \\ H^c &= [\mathbf{e}^c(x^c(a_1)), \mathbf{e}^c(x^c(a_2)), ..., \mathbf{e}^c(x^c(a_n))]^t. \end{aligned} \tag{8}$$

H^r is the $n \times n^r$ indicator matrix of x^r. $\mathbf{e}^r(x^r)$ is one-hot encoding of x^r. $a_1 ..., a_n$ represent the observations. $x^r(a_b)$ is the value of x^r of b-th observation. H^c is an $n \times n^c$ indicator matrix of x^c. $\mathbf{e}^c(x^c)$ is one-hot encoding of x^c. $x^c(a_b)$ is the value of x^c of b-th observation.

For example, the one-hot encodings and indicator matrix of x^{eys} in Fisher's data are

$$\begin{aligned} \mathbf{e}^{\text{eye}}(\text{blue}) &= (1, 0, 0, 0)^t \\ \mathbf{e}^{\text{eye}}(\text{light}) &= (0, 1, 0, 0)^t \\ \mathbf{e}^{\text{eye}}(\text{medium}) &= (0, 0, 1, 0)^t \\ \mathbf{e}^{\text{eye}}(\text{dark}) &= (0, 0, 0, 1)^t \end{aligned} \quad , \quad H^{eye} = \begin{bmatrix} \mathbf{e}^{\text{eye}}(\text{eye color of 1st person}))^t \\ \cdots \\ 0, 0, 1, 0 \\ \cdots \\ \mathbf{e}^{\text{eye}}(\text{eye color of } n \text{ th person}))^t \end{bmatrix} \tag{9}$$

H^{hair} can be constructed similarly. The contingency table of Fisher's data is

$$\mathbf{N} = H^{\text{eye}t} H^{\text{hair}}. \tag{10}$$

Based on each observation, we can define the Gini-index as

$$\sigma^2(x) = \frac{1}{2n^2}\sum_{a=1}^{n}\sum_{b=1}^{n}(x(a)-x(b))^2 \tag{11}$$

where

$$(x(a)-x(b))^2 = \frac{|\mathbf{e}^x(x(a)) - \mathbf{e}^x(x(b))|^2}{2} = \begin{cases} 0 & x(a) = x(b) \\ 1 & x(a) \neq x(b). \end{cases} \tag{12}$$

Gini-index is the variance of a categorical variable[5]. For example, $\sigma^2(x^{eye})$ using Fisher's data is

$$\sigma^2(x^{\mathrm{eye}}) = \frac{1}{2 \cdot 5387^2}\sum_{a=1}^{5387}\sum_{b=1}^{5387}((\text{eye color of } a \text{ th person}) - (\text{eye color of } b \text{ th person}))^2.$$

A simple extension of this definition to the covariance $\sigma^2(x^r, x^c)$ by replacing $(x(a)-x(b))^2$ to $(x^r(a)-x^r(b))(x^c(a)-x^c(b))$ does not give reasonable values for the covariance $\sigma^2(x^r, x^c)$ [11]. Okada[11, 10] shown using rotated one-hot encoding provides reasonable values for the covariance of CD:

$$\begin{aligned} \sigma^2(x^r, x^c) = &\underset{R}{\text{maximize}} \quad \frac{1}{2n^2}\sum_{a=1}^{n}\sum_{b=1}^{n}\frac{1}{2}(\mathbf{e}^r(x^r(a)) - \mathbf{e}^r(x^r(a)))^t R(\mathbf{e}^c(x^c(a)) - \mathbf{e}^c(x^c(b)) \\ &\text{subject to} \quad R^tR = \mathbf{E} \end{aligned} \tag{13}$$

where R is a rotation matrix to maximize the covariance. We can rewrite Eq. (13) using Ξ, which is used in CA.

$$\begin{aligned} \sigma^2(x^r, x^c) = &\underset{R}{\text{maximize}} \quad && \frac{1}{2}\mathrm{tr}(R^t\Xi) \\ &\text{subject to} && R^tR = \mathbf{E}. \end{aligned} \tag{14}$$

Here we use the following relation

$$\begin{aligned} &\frac{1}{2n^2}\sum_{a=1}^{n}\sum_{b=1}^{n}\frac{1}{2}(\mathbf{e}^r(x^r(a)) - \mathbf{e}^r(x^r(b)))^t R(\mathbf{e}^c(x^c(a)) - \mathbf{e}^c(x^c(b))) \\ &= \frac{1}{4n^2}\mathrm{tr}(R^t\sum_{a=1}^{n}\sum_{b=1}^{n}(\mathbf{e}^r(x^r(a)) - \mathbf{e}^r(x^r(b)))(\mathbf{e}^c(x^c(a)) - \mathbf{e}^c(x^c(b)))^t) \\ &= \mathrm{tr}(R^t(\frac{1}{2n}\sum_{a=1}^{n}\mathbf{e}^r(x^r(a))\mathbf{e}^c(x^c(a))^t - \frac{1}{2n^2}\sum_{a=1}^{n}\mathbf{e}^r(x^r(a))\sum_{b=1}^{n}\mathbf{e}^c(x^c(b))^t)) \\ &= \frac{1}{2}\mathrm{tr}(R^t(\frac{H^{r\,t}H^c}{n} - \frac{\mathbf{r}\mathbf{c}^t}{n^2})) = \frac{1}{2}\mathrm{tr}(R^t(\frac{\mathbf{N}}{n} - \frac{\mathbf{r}\mathbf{c}^t}{n^2})) = \frac{1}{2}\mathrm{tr}(R^t\Xi) \end{aligned} \tag{15}$$

One can solve the maximization problem by differentiating the following Lagrangian.

$$\mathcal{L} = \mathrm{tr}(R^t\Xi) - \mathrm{tr}(\Lambda^t(R^tR - \mathbf{E})) \tag{16}$$

Therein, Λ is a Lagrange multiplier. The conditions for this Lagrangian $\mathcal{L}$ to be stationary with respect to R are $R^t\Xi = (\Lambda + \Lambda^t)$. This result demonstrates that $(R^t\Xi)$ must be a symmetric matrix. We can give a solution to this maximization problem using the following SVD.

$$USV^t = \Xi \tag{17}$$

It is readily apparent that $R = UV^t$ satisfies all necessary conditions. One can conclude that $R = UV^t$ is the solution of the problem (13).

Recalling that CA is the GSVD of Ξ, then the GSVD $\breve{U}\hat{S}\breve{V}^t = \Xi$ can be computed using SVD of

$$\hat{U}\hat{S}\hat{V}^t = \hat{\Xi} \tag{18}$$

$$\hat{\Xi} = D(\mathbf{r})^{-1/2}\Xi D(\mathbf{c})^{-1/2} \tag{19}$$

as shown below.

$$\breve{U} = D(\mathbf{r})^{1/2}\hat{U} \quad , \quad \breve{V} = D(\mathbf{c})^{1/2}\hat{V} \tag{20}$$

Substituting this relation to Eq. (13) produces the following.

$$\begin{aligned} &\underset{\breve{R}}{\text{maximize}} && \frac{1}{2}\mathrm{tr}(\breve{R}^t D(\mathbf{r})^{-1}\Xi D(\mathbf{c})^{-1}) \\ &\text{subject to} && \breve{R}^t D(\mathbf{r})^{-1}\breve{R}D(\mathbf{c})^{-1} = \mathbf{E} \end{aligned} \tag{21}$$

The following relations show that the solution of this optimization problem is $\breve{R} = \breve{U}\breve{V}^t$. And we can say CA is equivalent to the problem (21).

$$\hat{R} = \hat{U}\hat{V}^t \tag{22}$$

$$\breve{R} = D(\mathbf{r})^{1/2}\hat{R}D(\mathbf{c})^{1/2} \tag{23}$$

$$\hat{R} = D(\mathbf{r})^{-1/2}\breve{R}D(\mathbf{c})^{-1/2} \tag{24}$$

$$\breve{R}^t D(\mathbf{r})^{-1}\breve{R}D(\mathbf{c})^{-1} = D(\mathbf{c})^{1/2}\hat{R}^t\hat{R}D(\mathbf{c})^{-1/2} = \mathbf{E} \Leftrightarrow \hat{R}^t\hat{R} = \mathbf{E} \tag{25}$$

$$\mathrm{tr}(\hat{R}^t\hat{\Xi}) = \mathrm{tr}(D(\mathbf{c})^{-1/2}\breve{R}^t D(\mathbf{r})^{-1/2}\hat{\Xi}) = \mathrm{tr}(\breve{R}^t D(\mathbf{r})^{-1}\Xi D(\mathbf{c})^{-1}) \tag{26}$$

Based on these relation, we introduce scaled one-hot encoding as the following.

$$\hat{\mathbf{e}}^r(x^r) = D(\mathbf{r})^{-1}\mathbf{e}^r(x^r) \quad , \quad \hat{\mathbf{e}}^c(x^c) = D(\mathbf{c})^{-1}\mathbf{e}^c(x^c) \tag{27}$$

It is noteworthy that

$$\begin{gathered} [\hat{\mathbf{e}}^r(x_1^r), \hat{\mathbf{e}}^r(x_2^r), ..., \hat{\mathbf{e}}^r(x_{n^r}^r)] = D(\mathbf{r})^{-1}[\mathbf{e}^r(x_1^r), \mathbf{e}^r(x_2^r), ..., \mathbf{e}^r(x_{n^r}^r)] = D(\mathbf{r})^{-1}\mathbf{E} = D(\mathbf{r})^{-1} \\ [\hat{\mathbf{e}}^c(x_1^c), \hat{\mathbf{e}}^c(x_2^c), ..., \hat{\mathbf{e}}^c(x_{n^c}^c)] = D(\mathbf{c})^{-1}[\mathbf{e}^c(x_1^c), \mathbf{e}^c(x_2^c), ..., \mathbf{e}^c(x_{n^c}^c)] = D(\mathbf{c})^{-1}\mathbf{E} = D(\mathbf{c})^{-1} \end{gathered} \tag{28}$$

Substituting this relation into problem (21) yields the following.

$$\begin{aligned} &\underset{\breve{R}}{\text{maximize}} && \frac{1}{4n^2}\sum_{a,b}(\hat{\mathbf{e}}^r(x^r(a)) - \hat{\mathbf{e}}^r(x^r(b)))^t\breve{R}(\hat{\mathbf{e}}^c(x^c(a)) - \hat{\mathbf{e}}^c(x^c(b))) \\ &\text{subject to} && \breve{R}^t[\hat{\mathbf{e}}^r(x_1^r), \hat{\mathbf{e}}^r(x_2^r), ...]^t\breve{R}[\hat{\mathbf{e}}^c(x_1^c), \hat{\mathbf{e}}^c(x_2^c), ...] = \mathbf{E} \end{aligned} \tag{29}$$

It is readily apparent that this problem defines the rotated Gini-index using scaled one-hot encoding. This generalized Gini-index is equivalent to CA.

4 Non-linear extension

This section presents an attempt to generalize CA as possible to generalize while keeping the relation between Gini-index and CA. Let us introduce non-linear mapping Φ^r, Φ^c and operators on nonlinear mapped spaces: $\ominus^r, \ominus^c, \ominus, \oplus$. Rewriting the problem (29) using this nonlinear extension yields the following.

$$\begin{aligned}
&\underset{R}{\text{maximize}} && \frac{1}{4n^2}\sum_{a,b}(\Phi^r(\hat{\mathbf{e}}^r(x^r(a))) \ominus^r \Phi^r(\hat{\mathbf{e}}^r(x^r(b))))^t R(\Phi^c(\hat{\mathbf{e}}^c(x^c(a))) \ominus^c \Phi^c(\hat{\mathbf{e}}^c(x^c(b)))) \\
&\text{subject to} && R^t[\Phi^r(\hat{\mathbf{e}}^r(x^r_1)), \Phi^r(\hat{\mathbf{e}}^r(x^r_2)), ...]^t R[\Phi^c(\hat{\mathbf{e}}^c(x^c_1)), \Phi^c(\hat{\mathbf{e}}^c(x^c_2)), ...] = \mathbf{E}
\end{aligned} \tag{30}$$

Consider expansion of this expression using below rules:

$$X(Y_1 \ominus^c Y_2)^t = XY_1^t \ominus XY_2^t \tag{31}$$
$$(X_1 \ominus^r X_1)Y^t = X_1Y^t \ominus X_2Y^t \tag{32}$$
$$X \ominus (Y \ominus Z) = (X \oplus Z) \ominus Y \tag{33}$$
$$(X \ominus Y) \oplus Z = (X \oplus Z) \ominus Y \tag{34}$$
$$(X \ominus Y) \ominus Z = X \ominus (Z \oplus Y) \tag{35}$$
$$X \oplus (Y \ominus Z) = (X \oplus Y) \ominus Z \tag{36}$$
$$\theta_{ij} = \Phi^r(\hat{\mathbf{e}}^r(x^r_i))R\Phi^c(\hat{\mathbf{e}}^c(x^c_j)) \tag{37}$$
$$\Theta = [\theta_{ij}] \tag{38}$$

We consider the case of how to evaluate an expression including the operators $\ominus$, $\ominus^r$, and $\ominus^c$, which are not defined. Subsequently, we move these operators outside parentheses. To move the operators outside parentheses, some rules (31)–(36) are useful. In these rules, rewriting the left-hand side to the right-hand side moves the operators outside parentheses. Applying these rewriting rules to the problem (30) of yields the following.

$$\begin{aligned}
&\underset{R}{\text{maximize}} && \frac{1}{4n^2}\text{tr}(R^t(2n\mathbf{N}^{\oplus} \ominus 2\Upsilon^{\oplus}) \\
&\text{subject to} && R^t\Theta = \mathbf{E}
\end{aligned} \tag{39}$$

where

$$\begin{aligned}
\mathbf{N}^{\oplus} &= \sum_{a=1}^{n} \Phi^r(\hat{\mathbf{e}}^r(x^r(a)))\Phi^c(\hat{\mathbf{e}}^c(x^c(a)))^t \\
2\Upsilon^{\oplus} &= \sum_{a=1}^{n}\sum_{b=1}^{n} \Phi^r(\hat{\mathbf{e}}^r(x^r(a)))\Phi^c(\hat{\mathbf{e}}^c(x^c(b)))^t
\end{aligned} \tag{40}$$

When $\oplus = +$, and introducing kernel matrices K^r and K^c to be

$$\Theta = [\theta_{ij}] = K^rRK^c \tag{41}$$

Table 2: Relations between known methods and kernel specializations.

Name	K^r	K^c	$X \ominus Y$	$X \oplus Y$
Linear CA	$D(\mathbf{r})^{-1}$	$D(\mathbf{c})^{-1}$	$X - Y$	$X + Y$
Gini-index [11, 10]	$\mathbf{E}$	$\mathbf{E}$	$X - Y$	$X + Y$
G-test	$\mathbf{E}$	$\mathbf{E}$	$X(\log X - \log Y)$	$X + Y$
SGNS[8, 6]	$\mathbf{E}$	$\mathbf{E}$	$(\log X - \log Y - \log k)$	$X + Y$
GloVe[12]	$\mathbf{E}$	$\mathbf{E}$	$(\log X - \log Y + b^w + b^c)$	$X + Y$
kernel PCA for CD[9]	$e^{\alpha\|\mathbf{e}^r(x_i^r) - \mathbf{e}^r(x_j^r)\|^2}$	$\mathbf{E}$	$X - Y$	$X + Y$

gives

$$\underset{R}{\text{maximize}} \qquad \frac{1}{2}\mathrm{tr}(R^t K^r (\frac{1}{n}\mathbf{N} \ominus \frac{1}{n^2}\mathbf{r}\mathbf{c}^t)K^c) \tag{42}$$

$$\text{subject to} \qquad R^t K^r R K^c = \mathbf{E}. \tag{43}$$

Specifying the operators and kernel matrices provides various well known analyses for CD and natural language processing. Table 2 presents the relation between the specification and known methods. Linear CA is a simple CA without the nonlinear extension. Levy et al. [7] showed GloVe[12] is matrix factorization of sifted PMI. Based on that discussion, one can regard GloVe[12] as that of the KCA, as shown in Table 2. When some element of a contingency table is zero: $n_{ij} = 0$ then PMI $= -\infty$. The G-test for contingency tables can avoid this problem. Let us call SVD of Ξ when $X \ominus Y = X(\log X - \log Y)$ the G-test in this research.

5 Kernels for Natural language processing

In this section, we specifically examine vector representations of words. We consider the following contingency table

$$\mathbf{N} = [n_{ij}] = [\#(w_i, c_j)] \tag{44}$$

where $\#(w, c)$ is the number of times that word w appears in context c. Using this contingency table, we can formulate SGNS as one specialized KCA, as shown in Table 2. Linear CA and G-test also provide vector representations of words using the contingency table.

KCA can introduce tunable weights for stop words (SW) using the following kernel as

$$K^r = D(\text{StopWordVector}(\alpha^w))D(\mathbf{r})^{-1} \qquad K^c = D(\text{StopWordVector}(\alpha^c))D(\mathbf{c})^{-1} \tag{45}$$

where

$$\text{StopWordVector}(\alpha) = (..., 1 + \alpha\delta_{i \in \text{Stop Words}}, ...) \tag{46}$$

is the vector for which the weight α is added when i-th world is SW. We designate this mechanism as a stop word kernel.

When we have datasets about similarity scores between two words, the scores can be encoded into the vector representations of words using our kernel mechanism. For example[3],

$$\begin{aligned}&\mathrm{score}(\mathrm{movie}, \mathrm{theater}) = 7.92\\&\mathrm{score}(\mathrm{movie}, \mathrm{star}) = 7.38\\&\mathrm{score}(\mathrm{movie}, \mathrm{popcorn}) = 6.19\end{aligned}$$

We can encode such scores using defined operators $\ominus^r$ and $\ominus^c$ in Eq. (30) as presented below.

$$\begin{aligned}\Phi^r(\hat{\mathbf{e}}^r(x^r_a)) \ominus^r \Phi^r(\hat{\mathbf{e}}^r(x^r_b)) &= (\hat{\mathbf{e}}^r(x^r_a) - \hat{\mathbf{e}}^r(x^r_b))\gamma^r(x_a, x_b)\\ \Phi^c(\hat{\mathbf{e}}^c(x^c_a)) \ominus^c \Phi^c(\hat{\mathbf{e}}^c(x^c_b)) &= (\hat{\mathbf{e}}^c(x^c_a) - \hat{\mathbf{e}}^c(x^c_b))\gamma^c(x^c_a, x^c_b) \qquad (47)\end{aligned}$$

where

$$\gamma^r_{ij} = \alpha^r \mathrm{score}(i, j) + \beta^r \quad \gamma^c_{ij} = \alpha^c \mathrm{score}(i, j) + \beta^c \qquad (48)$$

$\alpha^r, \beta^r, \alpha^c, \beta^c$ are tunable parameters. When $\ominus = -$ and $\oplus = +$, the problem (30) is

$$\begin{aligned}\underset{R}{\text{maximize}} \quad & \frac{1}{2n^2}\mathrm{tr}(R^t D(\mathbf{r})^{-1}(\mathbf{N} \circ (\Gamma^r \mathbf{N} \Gamma^c) - (\Gamma^r \mathbf{N}) \circ (\mathbf{N}\Gamma^c))D(\mathbf{c})^{-1}\\ \text{subject to} \quad & R^t D(\mathbf{r})^{-1} R D(\mathbf{c})^{-1} = \mathbf{E}\\ & r = (\Gamma^r \mathbf{N}) \circ (\mathbf{N}\Gamma^c)\mathbf{1}\\ & c = ((\Gamma^r \mathbf{N}) \circ (\mathbf{N}\Gamma^c))^t \mathbf{1}\\ & \Gamma^r = [\gamma^r_{ij}]\\ & \Gamma^c = [\gamma^c_{ij}] \qquad (49)\end{aligned}$$

where $\circ$ is the Hadamard product. We designate this mechanism as the word similarity kernel.

6 Experiments

This section explains experimental evaluations of vector representations of words based on our proposed system. To calculate the contingency table $\mathbf{N} = [n_{ij}] = [\#(w_i, c_j)]$, we use the program code[2] used in [7]. For the SW kernel, we use stop words set in NLTK[1]. For WS, we use the score of the MEN dataset[2] reported by Bruni et al. First, 20% data of Text8 corpus[3] are used as training text data. The stop words set and word similarity data are too small compared to the entire Text8 corpus. Consequently, the stop word kernel and word similarity kernel have a small effect when using the whole Text8 corpus.

For evaluation, we use WordSim3533 dataset[3], the MEN dataset[2] of Bruni et al. and the Mechanical Turk dataset[14] reported by Radinsky et al.

[2] `https://bitbucket.org/omerlevy/hyperwords`

[3] `http://mattmahoney.net/dc/text8.zip`

Table 3: Comparing word similarity.

Name	WordSim3533[3]	Bruni[2]	Randinsky[14]
SGNS[7]	0.192	0.061	0.314
Linear CA F	0.189	0.081	0.201
Linear CA G	0.213	0.072	0.407
Linear CA with SW F	0.198	**0.088**	0.212
Linear CA with SW G	**0.222**	0.087	**0.421**
Linear CA with WS F	0.190	0.084	0.201
Linear CA with WS G	0.213	0.074	0.407
G-test F	0.139	0.074	0.156
G-test G	0.169	0.051	0.365
G-test with SW F	0.178	0.078	0.179
G-test with SW G	0.186	0.065	0.369
G-test with WS F	0.139	0.074	0.171
G-test with WS G	0.170	0.055	0.365

The 100 dimensional word vectors presentations are evaluated by comparing ranking of their cosine similarities and the ranking in the test dataset. The rankings are compared using Spearman's σ. We also show the SGNS result for comparison. Table 3 presents evaluation results. Linear CA provides vector representations of two types: F and G . We present the results of these two representations. We also show G-test results. The results for Linear CA and G-test with SW kernel and WS kernel are shown as "with SW" or "with WS" in the table.

Although Linear CA and G-test have no tunable parameters, these results are comparable to SGNS. In most cases, the SW kernel enhances accuracy. The WS kernel enhances accuracy slightly in some cases.

7 Conclusion

We demonstrate that Linear Correspondence Analysis (CA) is equivalent to defining the Gini-index with rotated and scaled one-hot encoding. Moreover, we attempted to generalize CA as possible to be able to generalize while maintaining the relation between the Gini-index and CA. Kernel Correspondence Analysis (KCA) is introduced based on the nonlinear generalization of CA. KCA gives various known analyses for categorical data and natural language processing by specializing kernels. For example, KCA can give G-test, skip-gram with negative-sampling (SGNS), and GloVe as a special case. We introduce two kernels for natural language processing based on KCA. The proposed mechanism is evaluated by application of the problem of vector representations of words. Although Linear CA and G-test have no tunable parameter, these results are comparable to those obtained for SGNS. Additionally, we show that kernels with tunable parameters can enhance accuracy.

References

[1] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[2] E. Bruni, G. Boleda, M. Baroni, and N. K. Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[3] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131, Jan. 2002.

[4] R. A. Fisher. The precision of discriminant functions. *Annals of Eugenics (London)*, 10:422–429, 1940.

[5] C. Gini. Variability and mutability, contribution to the study of statistical distributions and relations. studi economico-giuridici della r. universita de cagliari (1912). reviewed in: Light, r.j., margolin, b.h.: An analysis of variance for categorical data. *J. American Statistical Association*, 66:534–544, 1971.

[6] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185, 2014.

[7] O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225, 2015.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[9] H. Niitsuma and T. Okada. Kernel PCA for categorical data. *IEICE technical report. Artificial intelligence and knowledge-based processing*, 103(305):13–17, sep 2003.

[10] H. Niitsuma and T. Okada. Covariance and PCA for categorical variables. In *Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005, Proceedings*, pages 523–528, 2005.

[11] T. Okada. A note on covariances for categorical data. In K. Leung, L. Chan, and H. Meng, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2000*, 2000.

[12] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[13] D. Picca, B. Curdy, and F. Bavaud. Non-linear correspondence analysis in text retrieval: A kernel view. In *Proceedings of JADT*, 2006.

[14] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch. A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 337–346, New York, NY, USA, 2011. ACM.