Correspondence Analysis Using Neural Networks

Hsiang Hsu
Harvard University
hsianghsu@g.harvard.edu

Salman Salamatian
Massachuestts Institute of Technology.
salmansa@mit.edu

Flavio P.Calmon
Harvard University
flavio@seas.harvard.edu

Abstract

Correspondence analysis (CA) is a multivariate statistical tool used to visualize and interpret data dependencies. CA has found applications in fields ranging from epidemiology to social sciences. However, current methods used to perform CA do not scale to large, high-dimensional datasets. By reinterpreting the objective in CA using an information-theoretic tool called the principal inertia components, we demonstrate that performing CA is equivalent to solving a functional optimization problem over the space of finite variance functions of two random variable. We show that this optimization problem, in turn, can be efficiently approximated by neural networks. The resulting formulation, called the correspondence analysis neural network (CA-NN), enables CA to be performed at an unprecedented scale. We validate the CA-NN on synthetic data, and demonstrate how it can be used to perform CA on a variety of datasets, including food recipes, wine compositions, and images. Our results outperform traditional methods used in CA, indicating that CA-NN can serve as a new, scalable tool for interpretability and visualization of complex dependencies between random variables.

1 Introduction

Correspondence Analysis (CA) is an exploratory multivariate statistical technique that converts data into a graphical display with orthogonal factors. CA's history in the applied statistics literature dates back several decades (Benzécri, 1973; Greenacre, 1984; Lebart, 2013; Greenacre, 2017). In a similar vein to Principal Component Analysis (PCA) and its kernel variants (Hoffmann, 2007), CA is a technique that maps the data onto a low-dimensional representation. By construction, this new representation captures possibly non-linear relationships between the underlying variables, and can be used to interpret the dependence between two random variables X and Y from observed samples. CA has the ability to produce interpretable, low-dimensional visualizations (often two-dimensional) that capture complex relationships in data with entangled and intricate dependencies. This has led to its successful deployment in fields ranging from genealogy and epidemiology to social and environmental sciences (Tekaia, 2016; Sourial et al., 2010; Carrington et al., 2005; ter Braak and Schaffers, 2004; Ormoli et al., 2015; Ferrari et al., 2016).

Despite being a versatile statistical technique, CA has been underused on the large datasets currently found in the machine learning landscape. This can potentially be explained by the fact that, traditionally, CA utilizes as its main ingredient a singular value decomposition (SVD) of the normalized contingency table of X and Y (i.e., an empirical approximation of the joint distribution $P_{X,Y}$). This contingency table-based approach for performing CA has three fundamental limitations. First, it is restricted to data drawn from *discrete* distributions with finite support, since contingency tables for continuous variables will be highly dependent on a chosen quantization which, in turn, may jeopardize information in the data. Second, even when the underlying distribution of the data is discrete, reliably estimating the contingency table (i.e., approximating $P_{X,Y}$) may be infeasible due to limited number of samples. This inevitably hinges CA on the more (statistically) challenging problem of estimating $P_{X,Y}$. Third, building contingency tables is not feasible for *high-dimensional* data. For example, if $X \in \{0,1\}^a$ and all outcomes have non-zero probability, then the contingency table has 2^a rows.

We address these limitations by taking a fresh theoretical look at CA and re-interpreting the low-dimensional representations produced by this technique from a functional analysis vantage point. We bring to bear an information-theoretic tool called the *principal inertia components* (PICs) of a joint distribution $P_{X,Y}$ (Calmon et al., 2017). In essence, the PICs provide a fine-grained decomposition of the statistical dependency of X and Y, fully determining an orthornormal

set of finite-variance functions of X that can be reliably estimated from Y (and vice-versa) called the *principal functions* (PFs). The débute of PICs under different guises can be traced back to the works of Hirschfeld (1935), Gebelein (1941), and Rényi (1959). The PICs are at the heart of the Alternating Conditional Expectations (ACE) algorithm (Buja, 1990; Breiman and Friedman, 1985) and have been studied in the information theory and statistics literature (Witsenhausen, 1975; Makur and Zheng, 2015; Huang et al., 2017; Calmon et al., 2017).

We demonstrate that the low-dimensional projections produced by CA are *exactly* the principal functions found in the theory of PICs. The principal functions, in turn, can be determined by solving a quadratic optimization problem over the space of finite variance functions of X and Y. Solving this optimization for arbitrary variables is, at first glance, infeasible. However, by restricting our search to functions representable by multi-layer neural networks, we demonstrate how the principal functions can be efficiently approximated for both discrete and continuous (potentially high-dimensional) random variables. In summary, by first formulating CA in terms of a PIC-based optimization program, and then approximating this program using neural networks, we are able to perform CA at an unprecedented scale.

The contributions of this paper are as follows:

1. We show how the PICs and principal functions can be used for correspondence analysis (Section 2).

2. We introduce the Correspondence Analysis Neural Net (CA-NN) to estimate the PICs and principal functions, thereby making CA scalable to discrete and continuous (high-dimensional) data (Section 3).

3. We use synthetic data to demonstrate that the principal functions found by CA-NN match the functions predicted by theory (Section 4.1).

4. Moreover, we apply the CA-NN on several real-world datasets, including images (MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky and Hinton, 2009)), recipes (Yummly, 2015), and UCI wine quality (Asuncion and Newman, 2007). These examples demonstrate how the interpretable analysis and visualizations found in the CA literature can now be produced at a much larger scale (Section 4.2). All codes and experiments are available in (Hsu, 2019).

1.1 Related Work

Several statistical methods exist for producing correlated low-dimensional representations of two variables X and Y. For example, Canonical Correlation Analysis (CCA) (Hotelling, 1936) seeks to find linear relationships between variables. Kernel Canonical Correlation Analysis (KCCA) (Bach and Jordan, 2002) extends this approach by first projecting the variables onto a reduced kernel Hilbert space. The method closest to the one described here is Deep Canonical Correlation Analysis (DCCA) (Andrew et al., 2013), where non-linear representations of multi-view data is produced using neural nets. The objective of DCCA in (Wang et al., 2015, Eq. 1) is similar to finding the PICs. However, the non-linear projections found by DCCA are not exactly the principal functions, and Wang et al. (2015) do not make a connection to CA, PICs, nor Hilbert spaces. DCCA is also closely related to maximal correlated PCA (Feizi and Tse, 2017).

PICs are a generalization of Rényi maximal correlation; in fact, the first PIC is identical to Rényi maximal correlation (Buja, 1990). Maximal correlation can be estimated using the Bivariate ACE algorithm, which determines non-linear projections $f(X)$ and $g(Y)$ that are maximally correlated whilst having zero mean and unit variance (Breiman and Friedman, 1985). The projections are found by iteratively computing $\mathbb{E}\left[g(Y)|X\right]$ and $\mathbb{E}\left[f(X)|Y\right]$, and converge to the first principal functions. However, for large, high-dimensional datasets, iteratively computing conditional expectations is intractable. To overcome this problem, neural-based approaches such as Correlational Neural Nets (Chandar et al., 2016) have been proposed. Unlike previous efforts, the approach that we outline here allows the PICs and principal functions to be *simultaneously* computed, generalizing existing methods in the literature.

Finally, we mention Karhunen-Loève Transform, Principal Component Analysis (PCA), and its kernel version (kPCA) (Hoffmann, 2007). Similar to CA, these methods aim at representing data in terms of orthogonal (uncorrelated) components. As such, they capture the structural relationship within a high dimensional random vector of features X. However, these methods disregard whether these representations are relevant from the point of view of another variable Y. Instead, CA finds orthogonal components of both X and Y *jointly*, with the resulting components being highly correlated. Note that this is different from performing PCA or kPCA on the joint pair (X, Y), as evidenced by the derivations in Section 2. Moreover, unklike kPCA, CA produces non-linear, highly correlated representations without requiring a kernel to be defined *a priori*.

1.2 Notation

Capital letters (e.g. X) are used to denote random variables, and calligraphic letters (e.g. $\mathcal{X}$) denote sets.

We denote the probability measure of $X \times Y$ by $P_{X,Y}$, the conditional probability measure of Y given X by $P_{Y|X}$, and the marginal probability measure of X and Y by P_X and P_Y respectively. We denote the fact that X is distributed according to P_X by $X \sim P_X$. If X and Y have finite support sets $|\mathcal{X}| < \infty$ and $|\mathcal{Y}| < \infty$, then we denote the joint probability mass function (pmf) of X and Y as $p_{X,Y}$, the conditional pmf of Y given X as $p_{Y|X}$, and the marginal distributions of X and Y as p_X and p_Y, respectively. A sample drawn from a probability distribution is denoted by lower-case letters (e.g. x and y). Matrices are denoted in bold capital letters (e.g. $\mathbf{X}$) and vectors in bold lower-case letters (e.g. $\mathbf{x}$). The (i,j)-th entry of a matrix $\mathbf{X}$ is given by $[\mathbf{X}]_{i,j}$. We denote the identity matrix of dimension d by $\mathbf{I}_d$, and the all-one vector of dimension d by $\mathbf{1}_d$. Given $\mathbf{v} \in \mathbb{R}^d$, we denote the matrix with diagonal entries equal to $\mathbf{v}$ by $\mathsf{diag}(\mathbf{v})$.

2 Correspondence Analysis and the Principal Inertia Components

In this section, we formally introduce CA, the PICs and the principal functions, as well as the connection between the PICs and CA.

2.1 Correspondence Analysis

Correspondence analysis considers two random variables X and Y with $|\mathcal{X}| < \infty$, $|\mathcal{Y}| < \infty$, and pmf $p_{X,Y}$ (cf. Greenacre (1984) for a detailed overview). Given samples $\{x_k, y_k\}_{k=1}^n$ drawn independently from $p_{X,Y}$, a two-way contingency table $\mathbf{P}_{X,Y}$ is defined as a matrix with $|\mathcal{X}|$ rows and $|\mathcal{Y}|$ columns of normalized co-occurrence counts, i.e. $[\mathbf{P}_{X,Y}]_{i,j} = (\# \text{ of observations } (x_i, y_i) = (i,j))/n$. Moreover, the marginals are defined as $\mathbf{p}_X \triangleq \mathbf{P}_{X,Y}\mathbf{1}_{|\mathcal{Y}|}$ and $\mathbf{p}_Y \triangleq \mathbf{P}_{X,Y}^T\mathbf{1}_{|\mathcal{X}|}$. Consider a matrix

$$\mathbf{Q} \triangleq \mathbf{D}_X^{-1/2}(\mathbf{P}_{X,Y} - \mathbf{p}_X\mathbf{p}_Y^T)\mathbf{D}_Y^{-1/2}, \qquad (1)$$

where $\mathbf{D}_X \triangleq \mathsf{diag}(\mathbf{p}_X)$ and $\mathbf{D}_Y \triangleq \mathsf{diag}(\mathbf{p}_Y)$, and let the SVD of $\mathbf{Q}$ be $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$. Let $d = \min\{|\mathcal{X}|, |\mathcal{Y}|\} - 1$, and $\{\sigma_i\}_{i=1}^d$ be the singular values, then we have the following definitions (Greenacre, 1984):

- The orthogonal factors of X are $\mathbf{L} \triangleq \mathbf{D}_X^{-1/2}\mathbf{U}$.
- The orthogonal factors of Y are $\mathbf{R} \triangleq \mathbf{D}_Y^{-1/2}\mathbf{V}$.
- The factor scores are $\lambda_i = \sigma_i^2, 1 \le i \le d$.
- The factor score ratios are $\frac{\lambda_i}{\sum_{i=1}^d \lambda_i}, 1 \le i \le d$.

CA makes use of the orthogonal factors $\mathbf{L}$ and $\mathbf{R}$ to visualize the correspondence (i.e., dependencies), between X and Y. In particular, the first and second columns of $\mathbf{L}$ and $\mathbf{R}$ can be plotted on a two-dimensional plane (with each row corresponding to a point) producing the so-called *factoring plane*. The remaining planes can be produced by plotting the other

columns of $\mathbf{L}$ and $\mathbf{R}$. The factor score ratio quantifies the variance ("correspondence") captured by each orthogonal factor, and is often shown along the axes in factoring planes.

We provide next the definition of the PICs, which will enable the CA decomposition in (1) to be performed for arbitrary random variables under appropriate compactness assumptions.

2.2 Functional Spaces and the Principal Inertia Components

For a random variable X over the alphabet $\mathcal{X}$, we let $\mathcal{L}_2(P_X)$ be the Hilbert Space of all functions from $\mathcal{X} \to \mathbb{R}$ with finite variance with respect to P_X, i.e., $\mathcal{L}_2(P_X) \triangleq \{f : \mathcal{X} \to \mathbb{R} \mid \mathbb{E}[\|f(X)\|_2] < \infty\}$. For $f_1, f_2 \in \mathcal{L}_2(P_X)$, this Hilbert space has an associated inner product given by $\langle f_1, f_2 \rangle = \mathbb{E}[f_1(X)f_2(X)]$. As customary, this inner product induces a distance between two functions $f_1, f_2 \in \mathcal{L}_2(P_X)$, namely the Mean-Square-Error (MSE) distance given by $d(f_1, f_2) = \mathbb{E}[(f_1(X) - f_2(X))^2]$. One can construct the projection operator from $\mathcal{L}_2(P_X)$ to $\mathcal{L}_2(P_Y)$ by

$$\begin{aligned} \Pi_{Y=y}[f] &\triangleq \underset{g \in \mathcal{L}_2(P_Y)}{\arg\min} \mathbb{E}_{X,Y}\left[(f(X) - g(Y))^2 | Y = y\right] \\ &= \mathbb{E}[f(X)|Y = y], \qquad (2) \end{aligned}$$

with adjoint operator $\Pi_{X=x}[g] = \mathbb{E}[g(Y)|X = x]$ defined for $g \in \mathcal{L}_2(P_Y)$. The projection operator describes the closest function, in terms of mean-square-error, to a given function f of the inputs. Since $\mathcal{L}_2(P_X)$ is a Hilbert space, there exists a basis (in fact infinitely many) through which any function $f \in \mathcal{L}_2(P_X)$ can be equivalently represented. However, at a high level, it is of interest to find a basis for $\mathcal{L}_2(P_X)$, which *diagonalizes* the projection operator Π_Y.

This naturally leads to the following proposition.

Proposition 1 (Witsenhausen (1975)). *Without loss of generality, let $|\mathcal{Y}| \le |\mathcal{X}|$ and let $d \triangleq |\mathcal{Y}| - 1$, or be infinity if both sets $\mathcal{X}$ and $\mathcal{Y}$ are infinite. There exists two sets of functions of $\mathcal{F} = \{f_0, f_1, \ldots, f_d\} \subseteq \mathcal{L}_2(P_X)$ and $\mathcal{G} = \{g_0, g_1, \ldots g_d\} \subseteq \mathcal{L}_2(P_Y)$, and a set $\mathcal{S} = \{1, \lambda_1, \ldots, \lambda_d\}$ such that:*

- *$f_0(X)$ and $g_0(Y)$ are constant function almost surely, $\mathbb{E}[f_i(X)f_j(X)] = \delta_{i,j}$ and $\mathbb{E}[g_i(Y)g_j(Y)] = \delta_{i,j}$ (orthornormality).*
- *$\mathbb{E}[f_i(X)|Y = y] = \sqrt{\lambda_i}g_i(y)$, and $\mathbb{E}[g_i(Y)|X = x] = \sqrt{\lambda_i}f_i(x)$ for all $i = 1, \ldots, d$ (diagonalization).*
- *Any function $g \in \mathcal{L}_2(P_Y)$ can be represented as a linear combination $g(y) = \sum_{i=0}^d \beta_i g_i(y)$. Similarly, any function $f \in \mathcal{L}_2(P_X)$ can be represented as a linear combination $f(x) = f^\perp(x) + \sum_{i=0}^d \alpha_i f_i(x)$, where $f^\perp$ is orthogonal to all f_i for all $i = 0, 1, \ldots, d$ (basis).*

The functions within the sets $\mathcal{F}$ and $\mathcal{G}$ are defined here as the *principal functions* of $P_{X,Y}$, and the elements of $\mathcal{S}$ as the *principal inertia components* of $P_{X,Y}$. We call $f_i(x)$ and $g_i(y)$ the ith PFs of X and Y, and $0 \leq \lambda_i \leq 1$ the ith PIC. Without loss of generality, we let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. Moreover, $\sqrt{\lambda_1}$ is also known as Rényi maximal correlation. A more thorough introduction to PICs can be found in (Witsenhausen, 1975; Buja, 1990; Calmon et al., 2017).

2.3 The Reconstitution Formula and Correspondence Analysis

As illustrated in (2), the principal functions precisely characterize the MSE-performance of estimating a function of X from an observation Y (and vice-versa). In fact, the PICs and principal functions can be used to reconstitute the joint distribution entirely (Buja, 1990, Sec. 3), i.e.

$$\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} = 1 + \sum_{i=1}^{d} \sqrt{\lambda_i} f_i(x) g_i(y). \quad (3)$$

This decomposition has also appeared in the CA literature (Greenacre, 1984, Chap. 4). This reconstitution formula is key for bridging the PICs and CA, and enables us to generalize CA to continuous variables. We make this connections precise in the following proposition, which demonstrates that the orthogonal factors found in CA are *exactly* the principal functions.

Proposition 2. *If $|\mathcal{X}|$ and $|\mathcal{Y}|$ are finite, we set $[\mathbf{F}]_{i,j} = f_j(i)$, $[\mathbf{G}]_{k,j} = g_j(k)$ for $1 \leq i \leq |\mathcal{X}|$, $1 \leq j \leq d$ and $1 \leq k \leq |\mathcal{Y}|$, and let $\mathbf{\Lambda} = \mathsf{diag}(\lambda_1, \cdots, \lambda_d)$. Moreover, let $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{\Sigma}$ follow from Section 2.1 and assume the diagonal entries of $\mathbf{\Sigma}$ are in descending order. Then, the PFs $\mathbf{F}$ and $\mathbf{G}$ are equivalent to the orthogonal factors $\mathbf{L}$ and $\mathbf{R}$ in the CA, and the factoring scores $\mathbf{\Sigma}$ are the same as the PICs $\mathbf{\Lambda}$.*

3 The Correspondence Analysis Neural Net (CA-NN)

In the previous section, we demonstrated that the orthogonal factors found via CA are equivalent to the principal functions given by the PIC decomposition of $P_{X,Y}$ (Prop. 2). Thus, we can (at least in theory) perform CA by computing principal functions directly, without having to build a contingency table first. Principal functions, in turn, are well-defined for both discrete and continuous (or mixed) X and Y, enabling CA to be extended to a broader range of data types. For the rest of the paper, we use the term principal functions and PICs to indicate the orthogonal factors and factor scores, respectively.

Equations (2), (3), and Prop. 1 suggest that the principal functions can be computed for arbitrary variables

by finding maximally correlated functions in $\mathcal{L}_2(P_X)$ and $\mathcal{L}_2(P_Y)$. Finding such functions, however, require a search over the space of all finite-variance functions of X or Y, which is not feasible for high dimensional data. Thus, in order to approximate the principal functions and scale up CA, we restrict our search to *functions representable by neural nets*. Note that the output of any neuron of a feed-forward neural net that receives X as an input can be viewed[1] as a point in $\mathcal{L}_2(P_X)$ (and equivalently when receiving Y as input).

In this section, we introduce the Correspondence Analysis Neural Net (CA-NN). The CA-NN estimates the PICs and the principal functions of $P_{X,Y}$ by minimizing an appropriately defined loss function (described next) using gradient descent and backpropagation. We will use the CA-NN to perform CA at scale.

3.1 Method

For two random variables (X, Y), we denote the d principal functions of X and Y, respectively, as

$$\widetilde{\mathbf{f}}(X) \triangleq [\widetilde{f}_1(X), \cdots, \widetilde{f}_d(X)]^{\mathsf{T}} \in \mathbb{R}^{d \times 1}, \quad (4)$$

$$\widetilde{\mathbf{g}}(Y) \triangleq [\widetilde{g}_1(Y), \cdots, \widetilde{g}_d(Y)]^{\mathsf{T}} \in \mathbb{R}^{d \times 1}. \quad (5)$$

Under these assumptions, the solution of the optimization problem

$$\min_{\mathbf{A} \in \mathbb{R}^{d \times d}, \widetilde{\mathbf{f}}, \widetilde{\mathbf{g}}} \mathbb{E} \left[\|\mathbf{A}\widetilde{\mathbf{f}}(X) - \widetilde{\mathbf{g}}(Y)\|_2^2 \right]$$
$$\text{subject to } \mathbb{E} \left[\mathbf{A}\widetilde{\mathbf{f}}(X)(\mathbf{A}\widetilde{\mathbf{f}}(X))^{\mathsf{T}} \right] = \mathbf{I}_d \quad (6)$$

recovers the d largest PICs. To see why this is the case, let

$$\mathbf{f}(X) = \mathbf{A}\widetilde{\mathbf{f}}(X) = [\mathbf{f}_1(X), \cdots, \mathbf{f}_d(X)]^{\mathsf{T}}, \quad (7)$$

and suppose that $\mathbf{f}, \widetilde{\mathbf{g}}$ and $\mathbf{A}$ achieve optimality in (6). Optimality under quadratic loss implies that $\widetilde{g}_i(y) = \mathbb{E}\left[f_i(X) \mid Y = y \right]$ for $i \in \{1, \ldots, d\}$. Moreover, the orthogonality constraint assures that the entries of $\mathbf{f}(X)$ satisfy $\mathbb{E}\left[f_i(X)f_j(X) \right] = \delta_{i,j}$, and thus form a basis for a d-dimensional subspace of $\mathcal{L}_2(P_X)$. As discussed in Section 2, conditional expectation on Y is a compact operator from $\mathcal{L}_2(P_X) \to \mathcal{L}_2(P_Y)$, and from orthogonality of $\mathbf{f}(X)$, it follows directly from the Hilbert-Schmidt Theorem (Reed and Simon, 1980) that the optimal value of (6) is $\sum_{i=0}^{d-1} \lambda_i$, with $\mathbf{f}$ and $\widetilde{\mathbf{g}}$ corresponding to the d largest principal functions.

We can further simplify the objective function in (6) using the following proposition.

---

[1] We assume that the outputs of a neural network have finite variance — a reasonable assumption since several gates used in practice have bounded value (e.g., sigmoid, tanh) and, at the very least, the output is limited by the number of bits used in floating point representations.
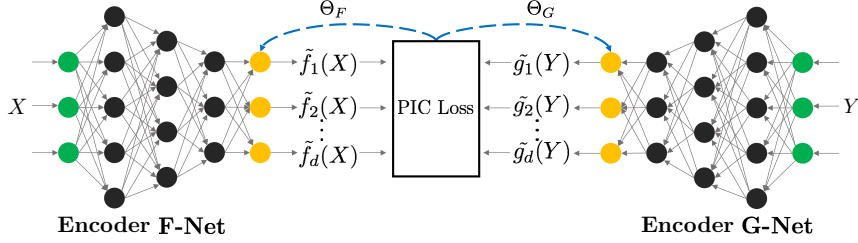
Hsiang Hsu, Salman Salamatian, Flavio P.Calmon

Figure 1: The architecture of the CA-NN, consisting of two encoders F-Net and G-Net for X and Y respectively to estimate the principal functions. The PIC loss is given by (8).

Proposition 3. *The minimization in (6) is equivalent to the following unconstrained optimization problem.*

$$\min_{\widetilde{\mathbf{f}},\widetilde{\mathbf{g}}} \quad -2\|\mathbf{C}_f^{-\frac{1}{2}}\mathbf{C}_{fg}\|_d + \mathbb{E}[\|\widetilde{\mathbf{g}}(Y)\|_2^2], \qquad (8)$$

where $\mathbf{C}_f = \mathbb{E}[\widetilde{\mathbf{f}}(X)\widetilde{\mathbf{f}}(X)^{\mathsf{T}}]$, $\mathbf{C}_{fg} = \mathbb{E}[\widetilde{\mathbf{f}}(X)\widetilde{\mathbf{g}}(Y)^{\mathsf{T}}]$, *and* $\|\mathbf{Z}\|_d$ *is the d-th Ky-Fan norm, defined as the sum of the singular values of* $\mathbf{Z}$ *(Horn et al., 1990, Eq. (7.4.8.1)). Denoting by* $\mathbf{A}$ *and* $\mathbf{B}$ *the whitening matrices[2] for* $\widetilde{\mathbf{f}}(\mathbf{X})$ *and* $\widetilde{\mathbf{g}}(\mathbf{Y})$, *the principal functions are given by* $\mathbf{f}(X) = [f_0(X), \cdots, f_d(X)]^{\mathsf{T}} = \mathbf{A}\widetilde{\mathbf{f}}(X)$ *and* $\mathbf{g}(Y) = [g_0(Y), \cdots, g_d(Y)]^{\mathsf{T}} = \mathbf{B}\widetilde{\mathbf{g}}(Y)$.

The proof of this propositon is related to the orthogonal Procrustes problem (Gower and Dijksterhuis, 2004), which convergence properties have been studied in (Nie et al., 2017).

3.2 Implementation

Observe that (8) is an unconstrained optimization problem over the space of all finite variance functions of X and Y. As discussed in the introduction of this section, we restrict our search to functions given by outputs of neural nets, parameterizing $\widetilde{\mathbf{f}}(X)$ and $\widetilde{\mathbf{g}}(Y)$ by θ_F and θ_G, respectively. Here, θ_F and θ_G denote weights of two neural nets, called the F-net and the G-net (Fig. 1). The F-Net and the G-Net encode X and Y to $\mathbb{R}^d$, respectively. The parameters of each network can be fit using gradient-based back-propagation of the objective (8), as described next.

Given n samples $\{x_k, y_k\}_{k=1}^n$ from $P_{X,Y}$, we denote $\mathbf{x}_n \triangleq [x_1, \cdots, x_n]$, $\mathbf{y}_n \triangleq [y_1, \cdots, y_n]$, $\widetilde{\mathbf{F}}_n(\mathbf{x}_n) = [\widetilde{\mathbf{f}}(x_1, \theta_F), \cdots, \widetilde{\mathbf{f}}(x_n, \theta_F)]^{\mathsf{T}} \in \mathbb{R}^{d \times n}$ and $\widetilde{\mathbf{G}}_n(\mathbf{y}_n) = [\widetilde{\mathbf{g}}(y_1, \theta_G), \cdots, \widetilde{\mathbf{g}}(y_n, \theta_G)] \in \mathbb{R}^{d \times n}$. The empirical eval-

uations of the terms in (8) are

$$\mathbf{C}_f \approx \frac{1}{n}\widetilde{\mathbf{F}}_n(\mathbf{x}_n, \theta_F)\widetilde{\mathbf{F}}_n(\mathbf{x}_n, \theta_F)^{\mathsf{T}}, \quad (9a)$$

$$\mathbf{C}_{fg} \approx \frac{1}{n}\widetilde{\mathbf{F}}_n(\mathbf{x}_n, \theta_F)\widetilde{\mathbf{G}}_n(\mathbf{y}_n, \theta_G)^{\mathsf{T}}, \quad (9b)$$

$$\mathbb{E}[\|\widetilde{\mathbf{g}}(Y)\|_2^2] \approx \frac{1}{n}\sum_{i=1}^n \sum_{j=1}^d \widetilde{\mathbf{g}}_j(y_i, \theta_G)^2. \quad (9c)$$

After extracting $\widetilde{\mathbf{F}}_n(\mathbf{x}_n)$ and $\widetilde{\mathbf{G}}_n(\mathbf{y}_n)$ from the F-Net and G-Net, respectively, Proposition 3 suggests that the principal functions can be recovered by producing whitening matrices $\mathbf{A}$ for $\widetilde{\mathbf{f}}$ and $\mathbf{B}$ for $\widetilde{\mathbf{g}}$. Without loss of generality, we will assume that $\widetilde{\mathbf{F}}_n(\mathbf{x}_n)$ and $\widetilde{\mathbf{G}}_n(\mathbf{y}_n)$ have zero-mean columns, which can always be done by subtracting the column-mean element-wise. Then $\mathbf{A}$ is given by $\mathbf{A} = \mathbf{U}^{\mathsf{T}}\mathbf{C}_f^{-1/2}$, with $\mathbf{U}$ the left singular vectors of the matrix

$$\mathbf{L} = \frac{1}{n}(\mathbf{C}_f^{-1/2}\widetilde{\mathbf{F}}_n(\mathbf{x}_n))(\mathbf{C}_g^{-1/2}\widetilde{\mathbf{G}}_n(\mathbf{y}_n))^{\mathsf{T}}. \quad (10)$$

The matrix $\mathbf{C}_f^{-1/2}$ guarantees that $\widetilde{\mathbf{F}}_n(\mathbf{x}_n)$ has orthonormal columns, while $\mathbf{U}$ rotates the set of vectors to align with $\widetilde{\mathbf{G}}_n(\mathbf{y}_n)$. By symmetry, $\mathbf{B} = \mathbf{V}^{\mathsf{T}}\mathbf{C}_g^{-1/2}$, where $\mathbf{V}$ are the right singular vectors of $\mathbf{L}$. The produced matrices $\mathbf{F}_n(\mathbf{x}_n) = \mathbf{A}\widetilde{\mathbf{F}}_n(\mathbf{x}_n)$ and $\mathbf{G}_n(\mathbf{y}_n) = \mathbf{B}\widetilde{\mathbf{G}}_n(\mathbf{y}_n)$ satisfy

$$\frac{1}{n}\mathbf{F}_n(\mathbf{x}_n)^{\mathsf{T}}\mathbf{F}_n(\mathbf{x}_n) = \frac{1}{n}\mathbf{G}_n(\mathbf{y}_n)^{\mathsf{T}}\mathbf{G}_n(\mathbf{y}_n) = \mathbf{I}_d, \quad (11)$$

and $\frac{1}{n}\mathbf{F}_n(\mathbf{x}_n)^{\mathsf{T}}\mathbf{G}_n(\mathbf{y}_n) = \mathbf{\Lambda}$ being the diagonal matrix with the estimated PICs. It should be emphasized that in the implementation and subsequent experiments, we estimate the whitening matrices $\mathbf{A}$ and $\mathbf{B}$ on the training set alone prior to evaluation on the test set. For clarity, we summarize this whitening process by Algorithm 1 in the Supplementary Material.

4 Experiments

The experiments contain two parts. First, we apply the CA-NN to synthetic data where the PICs and principal functions can be computed analytically, and

---

[2]We call $\mathbf{A}$ and $\mathbf{B}$ the whitening matrices since in Proposition 1, it is cleat that the covariance matrices of $\mathbf{f}(X)$ and $\mathbf{g}(Y)$ should both be identity matrices.
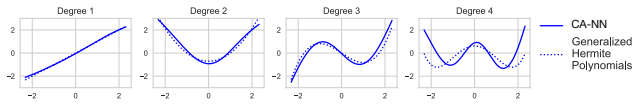
Figure 2: CA-NN recovers the Hermite polynomials, the principal functions in the Gaussian example.

Table 1: CA-NN reliably approximates the top four PICs in discrete and Gaussian cases.

	Discrete PICs			
CA-NN	0.8011	0.7942	0.7918	0.7883
Analytic value	0.8000	0.8000	0.8000	0.8000
	Gaussian PICs			
CA-NN	0.7007	0.4938	0.3376	0.2037
Analytic value	0.6977	0.4675	0.2979	0.2113

demonstrate that the proposed method recovers the values predicted by theory. Second, we use the CA-NN to perform CA on two real-world datasets where contingency table-based CA fails: the Kaggle What's Cooking Recipes (Yummly, 2015) and UCI Wine Quality data (Asuncion and Newman, 2007). In particular, the UCI Wine Quality dataset includes a mixture of discrete and continuous variables, demonstrating the versatility of the proposed methods. We select these datasets since their features naturally lend themselves to interpretable visualizations of the underlying dependencies in the data by factoring planes. In order to demonstrate that the CA-NN can be used to perform CA at an unprecedented scale, we also apply this method to image datasets (MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky and Hinton, 2009)), albeit these experiments do not lend themselves to the same kind of interpretable analysis found in the food related datasets. Detailed experimental setups (e.g., architecture of the CA-NN, training details, depths of encoders, etc.) are provided in the Supplementary Material. All the 95%-confidence intervals of the estimation of the PICs in Tables 1 and 2 are less than 1% for CA-NN.

4.1 Synthetic Data

We demonstrate next through two examples — one on discrete data and one on continuous data — that the CA-NN is able to reliably recover the PICs and the principal functions predicted by theory.

4.1.1 Discrete Synthetic Data

We consider $X \sim Bernoulli(p)$, $Z \sim Bernoulli(\delta)$, and $Y = X \oplus Z$, where $\oplus$ is the exclusive-or operator and δ is the crossover probability. Note that Y can be viewed as the output of a discrete memoryless bi-

nary symmetric channel (BSC) (Cover and Thomas, 2012) with input X. For any additive noise binary channel, the PICs can be mathematically determined (O'Donnell, 2014; Calmon et al., 2017). We set X to be a binary string of length 5, $\delta = 0.1$, and $p = 0.1$. The results in Table 1 show that the CA-NN reliably approximates the PICs, and we observed this consistent behaviour over multipe runs of the experiment. Details (including analytical expressions for the PICs and principal functions) are given in the Supplementary Material.

4.1.2 Gaussian Synthetic Data

When $X \sim \mathcal{N}(0, \sigma_1^2 \boldsymbol{I}_n)$, $Y|X \sim \mathcal{N}(X, \sigma_2^2 \boldsymbol{I}_n)$, the set of principal functions are the Hermite polynomials (Abbe and Zheng, 2012). More precisely, if the ith degree Hermite polynomial is given by

$$H_i^{(r)}(x) \triangleq \frac{(-1)^i}{\sqrt{i!}} e^{\frac{x^2}{2r}} \frac{d^i}{dx^i} e^{-\frac{x^2}{2r}}, r \in (0, \infty), \quad (12)$$

then the ith principal functions f_i and g_i are $H_i^{(\sigma_1)}$ and $H_i^{(\sigma_1+\sigma_2)}$ respectively, and the ith PIC can then be given by their inner product. We pick $\sigma_1 = \sigma_2 = 1$, and show estimation of PICs and principal functions in Table 1 and Fig. 2. Observe that the CA-NN closely approximates the first Hermite polynomials.

4.2 Real-World Data

We first investigate two datasets, Kaggle What's Cooking Recipes(Yummly, 2015) and UCI Wine Quality (Asuncion and Newman, 2007). These dataset contain highly non-linear dependencies which are interpretable via CA. In order to illustrate that we can perform CA on high-dimensional, continuous data, we conclude this section by applying the CA-NN to two image datasets.

4.2.1 Kaggle What's Cooking Recipe Data

The Kaggle What's cooking dataset (Yummly, 2015) contains 39774 recipes as X, composed of 6714 ingredients (e.g. peanuts, sesame, beef, etc.), from 20 types of cuisines as Y (e.g. Japanese, Greek, Southern US, etc.). The recipes are given in text form, so we preprocess the data in order to combine variations of the same ingredient and, for the sake of example and interpretable visualizations, keep only the 146 most common ingredients. In Table 2, we show that CA-NN outperforms contingency table-based CA using SVD[3], with the resulting PICs being more correlated (i.e., achieving a higher value of Eq. (8)) than its contingency table-based counterpart.

In Table 2, we compare the CA-NN with baseline techniques such as CCA and KCCA (with radial basis func-

---

[3]We only consider combinations of ingredients observed in the dataset as possible outcomes of X.

Table 2: CA-NN outperforms contingency table-based CA (SVD) and CCA/ KCCA (which also produces transformation of X and Y, cf. Section 1.1) on Kaggle What's Cooking dataset to explore dependencies in samples.

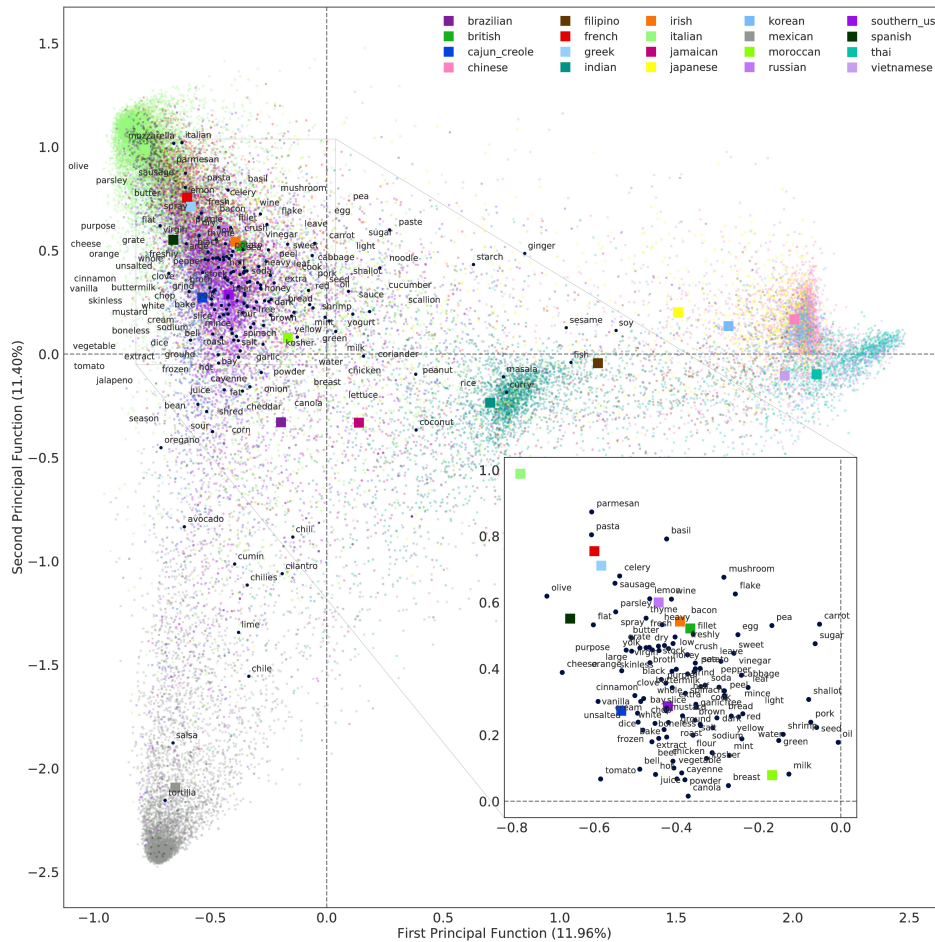|	Top ten principal inertia components									
CA-NN	**0.9092**	**0.8667**	**0.8412**	**0.7932**	**0.7391**	**0.6413**	**0.6018**	**0.4792**	**0.4508**	**0.2821**
SVD	0.4504	0.3894	0.3149	0.2943	0.2413	0.1958	0.1547	0.1191	0.1146	0.1035
	Correlations between transformed samples									
CCA	0.1915	0.1751	0.1342	0.1083	0.1050	0.0823	0.0623	0.0488	0.0485	0.0431
KCCA	0.6585	0.1223	0.0860	0.0636	0.0320	0.0131	0.0090	0.0089	0.0051	0.0011

Figure 3: The first factoring plane of CA on Kaggle What's cooking dataset (Colored dots: recipe, dark blue: ingredient).

tion kernels). The resulting low-dimensional representation produced by CA-NN captures a higher correlation/variance than these other embeddings. We recognize that these results may vary if other kernels were selected, but note that the CA-NN *does not require any form of kernel selection by a human prior to application*. In Fig. 3 we display a traditional CA-style plot produced using CA-NN, showing the first factoring plane of the CA (i.e., the first and second principal

functions for X and Y). The intersection of two dashed lines ($x = 0$ and $y = 0$) indicates the space where X and Y have insignificant correlation.

There are three key observations which can be extracted from Fig. 3. First, we observe clear clusters under the representation learned by CA-NN which can be easily interpreted. The cluster on the right-hand side represents East-Asian cuisines (e.g. Chinese, Korean), the one on the left-hand side represents West-
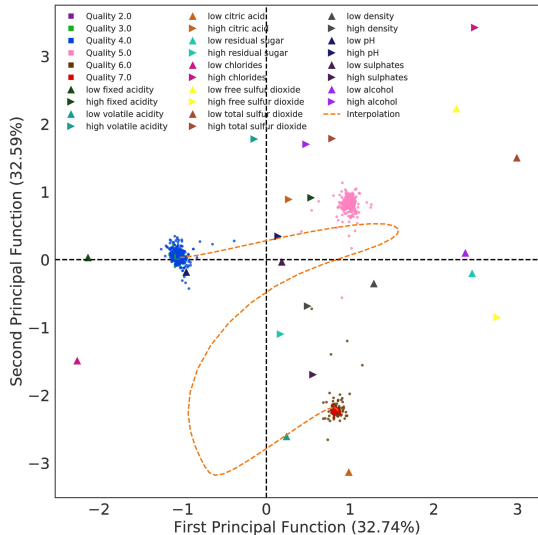
Figure 4: The first factoring plane of CA on UCI wine quality dataset.

ern cuisine (e.g. French, Italian) and in between sits Indian cuisine. Second, we observe that the first principal function learns to distinguish Asian cuisine (e.g. Chinese, Korean) from Western cuisine, naturally separating these contrasting culinary cultures. Interestingly, Indian cuisine sits in between Asian and Western cuisine, and Filipino cuisine is represented between Indian and Asian cuisine over this axis. The second principal function further indicates finer differences between Western cuisines, and singles out Mexican cuisine. Third, by plotting the ingredients on this plane (i.e., recipes containing only one ingredient), we can determine *signature* ingredients for different kinds of cuisines. For example, despite the fact that ginger is in both Western and Asian dishes, it is closer in the factor plane to Asian cuisine, revealing that it plays a more prominent role in this cuisine. Some ingredients share much stronger correlation with the cuisine type, e.g. curry in Indian dishes and tortilla in Mexican ones. See the supplementary material for additional factorial planes.

4.2.2 UCI Wine Quality Data

The UCI wine quality dataset contains 4898 red wines with 11 physico-chemical attributes (e.g. pH value, acid, alcohol) and 6 levels of qualities (from 2 to 7). We set X to be the 11 attributes and Y be the quality, and report the results of CA in Fig. 4. Note that since the attributes are continuous, performing contingency table-based CA is not well-defined for this case. In Fig. 4, we can see that despite the existence of 6 classes of qualities, the principal functions discover three sub-clusters, namely poor quality (less or equal to 4), medium quality (equal to 5), and high quality
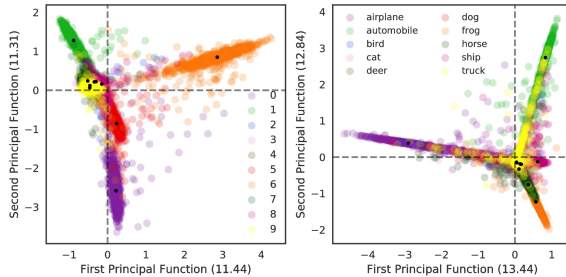

Figure 5: The first factoring planes of noisy MNIST (left) and CIFAR-10 (right).

(6 and above) Moreover, Fig. 4 also shows how the attributes affects the quality of a wine. For example, high quality wines (quality 6 and 7) tends to have low citric and volatile acidity, but with high sulphates. Finally, we randomly sample a low quality and a high quality wine, and take the linear interpolation of their features. We represent the path that this linear interpolation draws in the factorial plane by the orange line in Fig. 4. This sheds light on how a "bad" wine can be transformed into a "good" wine.

4.2.3 MNIST and CIFAR-10

In Fig. 5, we show that CA-NN enables CA on image datasets such as noisy MNIST (Wang et al., 2015) and CIFAR-10 (Krizhevsky and Hinton, 2009) datasets. Noisy MNIST is a more challenging version of MNIST (LeCun et al., 1998), containing 60000 gray-scale images for training and 10000 for test, where each image is a 28×28 pixels handwritten digit with random rotation and noise; CIFAR-10 contains 50000 colored images for training and 10000 for test in 10 classes, where each images has 32×32 pixels. Since the features (pixels) of these images are not very informative, we do not show them in Fig. 5.

5 Conclusion

We proposed a neural-based estimator for the principal inertia components and the principal functions, called the Correspondence Analysis Neural Network (CA-NN). By proving that the principal functions are equivalent to orthogonal factors in CA, we are able to use the CA-NN to scale up CA to large, high-dimensional datasets with continuous features. We validated the CA-NN on synthetic data, and showed how it enables CA to be performed on large real-world datasets. These experiments indicate that CA-NN significantly outperforms other approaches of CA. Future research directions include characterizing generalization properties of CA-NN in terms of the number of training samples, as well as the impact of network architecture on the resulting principal functions. We hope that the CA-NN can allow CA to be more widely applied to the large datasets found in the current machine learning landscape.

References

Abbe, E. and Zheng, L. (2012). A coordinate system for gaussian networks. *IEEE Transactions on Information Theory*, 58(2):721–733.

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In *International Conference on Machine Learning*, pages 1247–1255.

Asuncion, A. and Newman, D. (2007). UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48.

Benzécri, J.-P. (1973). *L'Analyse des Données. Volume II. L'Analyse des Correspondances*. Paris, France: Dunod.

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598.

Buja, A. (1990). Remarks on functional canonical variates, alternating least squares methods and ace. *The Annals of Statistics*, pages 1032–1069.

Calmon, F. P., Makhdoumi, A., Médard, M., Varia, M., Christiansen, M., and Duffy, K. R. (2017). Principal inertia components and applications. *IEEE Transactions on Information Theory*, 63(8):5011–5038.

Carrington, P. J., Scott, J., and Wasserman, S. (2005). *Models and methods in social network analysis*, volume 28. Cambridge university press.

Chandar, S., Khapra, M. M., Larochelle, H., and Ravindran, B. (2016). Correlational neural networks. *Neural computation*, 28(2):257–285.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Feizi, S. and Tse, D. (2017). Maximally correlated principle component analysis. *arXiv preprint arXiv:1702.05471*.

Ferrari, A., Vincent-Salomon, A., Pivot, X., Sertier, A.-S., Thomas, E., Tonon, L., Boyault, S., Mulugeta, E., Treilleux, I., Macgrogan, G., et al. (2016). A whole-genome sequence and transcriptome perspective on her2-positive breast cancers. *Nature communications*, 7:12222.

Gebelein, H. (1941). Das statistische problem der korrelation als variations-und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 21(6):364–379.

Gower, J. C. and Dijksterhuis, G. B. (2004). *Procrustes problems*, volume 30. Oxford University Press on Demand.

Greenacre, M. (2017). *Correspondence analysis in practice*. Chapman and Hall/CRC.

Greenacre, M. J. (1984). *Theory and applications of correspondence analysis*. London (UK) Academic Press.

Hirschfeld, H. O. (1935). A connection between correlation and contingency. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 520–524. Cambridge University Press.

Hoffmann, H. (2007). Kernel pca for novelty detection. *Pattern recognition*, 40(3):863–874.

Horn, R. A., Horn, R. A., and Johnson, C. R. (1990). *Matrix analysis*. Cambridge university press.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Hsu, H. (2019). Correspondence analysis using neural networks. https://github.com/HsiangHsu/2019-AISTATS-CA.

Huang, S.-L., Makur, A., Zheng, L., and Wornell, G. W. (2017). An information-theoretic approach to universal feature selection in high-dimensional inference. In *Information Theory (ISIT), 2017 IEEE International Symposium on*, pages 1336–1340. IEEE.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. *Technical report, University of Toronto*.

Lebart, L. (2013). Correspondence analysis. In *Data Science, Classification, and Related Methods: Proceedings of the Fifth Conference of the International Federation of Classification Societies (IFCS-96), Kobe, Japan, March 27–30, 1996*, page 423. Springer Science & Business Media.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Makur, A. and Zheng, L. (2015). Bounds between contraction coefficients. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 1422–1429. IEEE.

Nie, F., Zhang, R., and Li, X. (2017). A generalized power iteration method for solving quadratic problem on the stiefel manifold. *Science China Information Sciences*, 60(11):112101.

O'Donnell, R. (2014). *Analysis of boolean functions*. Cambridge University Press.

Ormoli, L., Costa, C., Negri, S., Perenzin, M., and Vaccino, P. (2015). Diversity trends in bread wheat

in italy during the 20th century assessed by traditional and multivariate approaches. *Scientific reports*, 5:8574.

Reed, M. and Simon, B. (1980). Functional analysis.

Rényi, A. (1959). On measures of dependence. *Acta mathematica hungarica*, 10(3-4):441–451.

Sourial, N., Wolfson, C., Zhu, B., Quail, J., Fletcher, J., Karunananthan, S., Bandeen-Roche, K., Béland, F., and Bergman, H. (2010). Correspondence analysis is a useful tool to uncover the relationships among categorical variables. *Journal of clinical epidemiology*, 63(6):638–646.

Tekaia, F. (2016). Genome data exploration using correspondence analysis. *Bioinformatics and Biology insights*, 10:BBI–S39614.

ter Braak, C. J. and Schaffers, A. P. (2004). Co-correspondence analysis: a new ordination method to relate two community compositions. *Ecology*, 85(3):834–846.

Wang, W., Arora, R., Livescu, K., and Bilmes, J. (2015). On deep multi-view representation learning. In *International Conference on Machine Learning*, pages 1083–1092.

Witsenhausen, H. S. (1975). On sequences of pairs of dependent random variables. *SIAM Journal on Applied Mathematics*, 28(1):100–113.

Yummly (2015). Kaggle what's cooking? https://www.kaggle.com/c/whats-cooking/data. Accessed: 2018-05-13.