ABSTRACT

In web search, data mining, and machine learning, two popular measures of data similarity are the cosine and the resemblance (the latter is for binary data). In this study, we develop theoretical results for both the cosine and the GMM (generalized min-max) kernel [26], which is a generalization of the resemblance. GMM has direct applications in machine learning as a positive definite kernel [26], which is a generalization of the resemblance. GMM typically holds for data commonly encountered in practice.

Furthermore, we establish the asymptotic normality of GMM. In other words, we have approximately

\[ c_n(x, y) \sim N \left( \rho, \frac{1}{n} (1 - \rho^2)^2 \right) \]

This important theoretical result can be found in [2].

The normality assumption in the above result (2) is actually crucial. When data are not normal, as long as the second moment is bounded, \( c_n(x, y) \) will still converge, although the variance will not exist unless data have bounded fourth moment. If \( c_n(x, y) \) does not converge to a fixed value, it means one can not obtain meaningful empirical results from the use of \( c_n \). If \( c_n(x, y) \) converges but its variance is not bounded (or too large), it also means the results would not be reliable. In this paper, we will provide a precise variance formula for \( c_n(x, y) \) without the normality assumption.

The data encountered in the real world are virtually always heavy-tailed [24, 11, 15]. [32] argued that the many natural datasets follow the power law with exponent (denoted by \( \nu \)) varying largely between 1 and 2, for example, \( \nu = 1.2 \) for the frequency of use of words, \( \nu = 2.04 \) for the number of citations to papers, \( \nu = 1.4 \) for the number of hits on the web sites, etc. Basically, \( \nu > 2 \) means that the data have bounded second moment. As \( n \to \infty \), the cosine similarity (1) will not converge to a fixed limit without the assumption of bounded second moment.

In this study, we analyze the “generalized min-max” (GMM) similarity [26, 27]. First, we define \( x_i = x_{i+} - x_{i-} \), where

\[ x_{i+} = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{otherwise} \end{cases} \]

\[ x_{i-} = \begin{cases} -x_i & \text{if } x_i < 0 \\ 0 & \text{otherwise} \end{cases} \]

Then we compute GMM as follows:

\[ GM(x, y) = \frac{\sum_{i=1}^{n} \left[ \min(x_{i+}, y_{i+}) + \min(x_{i-}, y_{i-}) \right]}{\sum_{i=1}^{n} \left[ \max(x_{i+}, y_{i+}) + \max(x_{i-}, y_{i-}) \right]} \]

The use of GMM for machine learning has been demonstrated in [26, 27]. The initial work [26] focused on comparisons with the radial basis function (RBF) kernel and normalized random Fourier features (NRFF), both empirically and theoretically. The follow-up work [27] illustrated that by introducing tuning parameters to the GMM kernel, the performance could be substantially improved. In fact, the performance of tunable GMM kernels might be even comparable to complex models such as deep nets or boosted trees.
Given the empirical success of GMM, it becomes important (and urgent) to understand what exactly is $g_n(x, y)$ and how it is connected to cosine $c_n(x, y)$. This paper will address these important questions. In particular, we are interested in the limit of $g_n(x, y)$ as $n \to \infty$ and how fast $g_n$ converges to the limit.

We should mention that GMM is related to several similarity measures widely used in data mining and web search. When the data are nonnegative, GMM becomes the “min-max” kernel, which has been studied in the literature [23, 7, 30, 22, 25]. When the data are binary (0/1), GMM becomes the well-known “resemblance” similarity. The minwise hashing algorithm for approximating resemblance has been a highly successful tool in web search for numerous applications [6, 17, 28, 3, 18, 8, 13, 9, 19, 31, 29].

2. KERNELES AND LINALIZATION

It is common in practice to use linear learning algorithms such as logistic regression or linear SVM. It is also known that one can often improve the performance of linear methods by using nonlinear algorithms such as kernel SVMs, if the computational/storage burden can be resolved. A straightforward implementation of a nonlinear kernel, however, can be difficult for large-scale datasets [5]. For example, for a small dataset with merely 60, 000 data points, the 60, 000 x 60, 000 kernel matrix has $3.6 \times 10^9$ entries. In practice, being able to linearize nonlinear kernels becomes highly beneficial. Randomization (hashing) is a popular tool for kernel linearization. After data linearization, we can then apply our favorite linear learning packages such as LIBLINEAR [16] or SGD (stochastic gradient descent) [4].

We focus on two types of nonlinear kernels, the GMM kernel and the RBF kernel, and their linearization methods. For any positive definite kernels, we can always utilize the Nyström method [33] for kernel approximation [39]. For example, [40] applied the Nyström method for approximating the RBF kernel, a procedure we call “RBF-NYS”. Analogously, we propose “GMM-NYS”, which is the use of the Nyström method for approximating the GMM kernel.

In addition to the Nyström method, we can approximate GMM by the “generalized consistent weighted sampling (GCWS)” [26], which we name as “GMM-GCWS”. For the RBF kernel, a popular scheme is the “random Fourier features (RFF)” [35] (named “RBF-RFF”). Thus, a thorough assessment requires comparing four methods: GMM-NYS, GMM-GCWS, RBF-NYS, and RBF-RFF.

2.1 GCWS: Generalized Consistent Weighted Sampling for Hashing GMM

The definition of GMM in (3) will be mathematically convenient later in our theoretical analysis. To better illustrate the concept, it might be more clear if we re-write (3) as follows [26, 27]:

$$
\begin{align*}
\{ & \tilde{x}_{2i-1} = x_i, \quad \tilde{x}_{2i} = 0 \quad \text{if } x_i > 0 \\
& \tilde{x}_{2i-1} = 0, \quad \tilde{x}_{2i} = -x_i \quad \text{if } x_i \leq 0
\end{align*}
$$

For example, if $x = [2 \quad 0 \quad 1 \quad 3]$, then the transformed data vector becomes $\tilde{x} = [2 \quad 0 \quad 0 \quad 1 \quad 3 \quad 0]$. After the transformation, the GMM similarity between two original vectors $x, y \in \mathbb{R}^d$ is defined as

$$
GMM(x, y) = \sum_{i=1}^{2n} \min(\tilde{x}_i, \tilde{y}_i) / \sum_{i=1}^{2n} \max(\tilde{x}_i, \tilde{y}_i)
$$

(6)

It is clear that, since the data are now nonnegative, one can apply the original “consistent weighted sampling” [30, 22, 25] to generate hashed data. [26, 27] named this procedure “generalized consistent weighted sampling” (GCWS), as summarized in Algorithm 1.

Algorithm 1 Generalized Consistent Weighted Sampling (GCWS)

**Input:** Data vector $x_i$ ($i = 1$ to $n$)

Generate vector $\tilde{x}$ in $2n$-dim by (5)

For $i$ from 1 to $2n$

$r_i \sim$ Gamma(2,1), $c_i \sim$ Gamma(2,1), $\beta_i \sim$ Uniform(0,1) 

$t_i \leftarrow \left\lfloor \frac{c_i}{\tilde{x}_i} + \beta_i \right\rfloor$, $a_i \leftarrow \log(c_i) - r_i(t_i + 1 - \beta_i)$

End For

**Output:** $i^* \leftarrow \arg\min_{i} a_i$, $t^* \leftarrow t_{i^*}$

Given two data vectors $x$ and $y$, one will transform them into nonnegative vectors $\tilde{x}$ and $\tilde{y}$ as in (5) and generate random tuples:

$$
\{(i^*_x, j^*_x), (i^*_y, j^*_y)\}, \quad j = 1, 2, ..., k
$$

where $i^* \in [1, 2n]$ and $t^*$ is unbounded. Following [30, 22], we have the basic probability result.

$$
\Pr\{(i^*_x, j^*_x) = (i^*_y, j^*_y)\} = GMM(x, y)
$$

(7)

Recently, [25] made an interesting observation that for practical data, it is ok to completely discard $t^*$. The following approximation

$$
\Pr\{i^*_x = i^*_y\} \approx GMM(x, y)
$$

(8)

is accurate in practical settings and makes the implementation convenient when using the idea of $b$-bit minwise hashing [28].

As described in [26, 27], for each data vector $x$, we obtain $k$ random samples $i^*_x, j^*_x = 1$ to $k$. We store only the lowest $b$ bits of $i^*$. We need to view those $k$ integers as locations (of the nonzeros) instead of numerical values. For example, when $b = 2$, we should view $i^*$ as a vector of length $2^2 = 4$. If $i^* = 3$, then we code it as $[0 \quad 0 \quad 0 \quad 1]$; if $i^* = 0$, we code it as $[0 \quad 0 \quad 1 \quad 0]$, etc. We concatenate all such vectors into a binary vector of length $2^b \times k$, which contains exactly $k$ 1’s. After we have generated such new data vectors for all data points, we feed them to a linear SVM or logistic regression solver. We can, of course, also use the new data for many other tasks including clustering, regression, and near neighbor search.

Note that, when using linear learning methods (especially online algorithms), both the storage cost and computational cost are largely determined by the number of nonzeros in each data vector, i.e., the $k$ in our case. It is thus crucial not to use a too large $k$.

2.2 The RBF Kernel and (Normalized) Random Fourier Features (RFF)

The natural competitor of the GMM kernel is the RBF (radial basis function) kernel, whose definition involves a crucial tuning parameter $\gamma > 0$. In this study, for convenience (e.g., parameter tuning), we use the following version of the RBF kernel:

$$
RBF(x, y; \gamma) = e^{-\gamma(1-\rho)},
$$

$$
\rho = \rho(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}
$$

(9)

Based on Bochner’s Theorem [37], it is known [35] that, if we sample $w \sim \text{uniform}(0, 2\pi)$, $r_i \sim \text{N}(0, 1)$ i.i.d., and let $u = \sum_{i=1}^{n} x_i r_i$, $v = \sum_{i=1}^{n} y_i r_i$, where $\|u\|_2 = \|v\|_2 = 1$, then we have

$$
E \left( \sqrt{2} \cos(\sqrt{t}u + w) \sqrt{2} \cos(\sqrt{t}v + w) \right) = \gamma^{-\gamma(1-\rho)}
$$

(10)

This provides an elegant mechanism for linearizing the RBF kernel. The so-called RFF (random Fourier features) method has become a popular building block in numerous machine learning tasks, e.g., [34, 40, 1, 20, 12, 41, 21, 38, 10, 36].
A major issue with RFF is the high variance. Typically a large number of samples (i.e., large $k$) would be needed in order to reach a satisfactory accuracy, as validated in [26]. Usually, “GMM-GCWS” (i.e., the GCWS algorithm for approximating GMM) requires substantially fewer samples than “RBF-RFF” (i.e., the RFF method for approximating the RBF kernel).

Note that there is an additional normalization step when we feed the RFF hashed data to a linear classifier such as linear SVM. When the hashed data are normalized to unit $l_2$ norm, the variance will be reduced in high similarity region ($\rho \rightarrow 1$). This phenomenon is probably not surprising after we have seen an analogous result (2). In the experiments of [26], the input data were always normalized.

2.3 The Nyström Method

The Nyström method [33] is a sampling scheme for kernel approximation [39]. For example, [40] applied the Nyström method for approximating the RBF kernel, which we call “RBF-NYS”. Analogously, we propose “GMM-NYS”, which is the use of the Nyström method for approximating the GMM kernel.

The concept of the Nyström method is simple. Given a training dataset $A$, we first sample $k$ data points (a smaller matrix denoted by $A_k$) and compute a small kernel matrix denoted by $K_k \in \mathbb{R}^{k \times k}$. Given another dataset $B$ (where $B$ and $A$ could be the same) with $m$ examples, we compute a kernel matrix $K \in \mathbb{R}^{m \times k}$, between every data point in $B$ and every data point in $A_k$. Then we compute a hashed data matrix of $B$, denoted by $H \in \mathbb{R}^{m \times k}$ as follows:

$$K_k = V D V^T, \quad H = K V D^{-1/2}$$

where $D$ is a diagonal (eigenvalue) matrix and $V$ is the eigenvector matrix of $K_k$. Here, we need to interpret $D^{-1/2}$ as a new diagonal matrix whose diagonal elements are the reciprocals of the square roots of the diagonal elements of $D$.

After this procedure, the inner product between any two rows of $H$ approximates the original kernel value between these two data points. Therefore, the new data dimensionality becomes $k$ and we can directly apply linear algorithms on the new data.

The above generic description of the Nyström method is suitable for any positive definite kernels, including GMM and RBF.

2.4 Experiments on Linearized Kernels

We provide an empirical study to compare four algorithms: GMM-GCWS, GMM-NYS, RBF-NYS, and RBF-RFF. The results for GMM-GCWS and RBF-RFF are directly taken from the prior work [26]. Note that “RBF-RFF” really means “RBF-NRFF”, i.e., we always use “normalized random Fourier features” instead of the original version. Once the hashed data are generated, we use the popular LIBLINEAR package on the hashed data produced by these four algorithms, in particular, at the same sample size $k$.

Table 1: Datasets downloaded from the UCI repository or LIBSVM website. In the last column, we report the test classification accuracies for the linear kernel, at the best SVM $l_2$-regularization $C$ value.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># train</th>
<th># test</th>
<th># dim</th>
<th>linear (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter</td>
<td>15,000</td>
<td>5,000</td>
<td>16</td>
<td>61.66</td>
</tr>
<tr>
<td>Webspam</td>
<td>175,000</td>
<td>175,000</td>
<td>254</td>
<td>93.31</td>
</tr>
<tr>
<td>PAMAP105</td>
<td>185,548</td>
<td>185,548</td>
<td>51</td>
<td>83.35</td>
</tr>
<tr>
<td>RCV1</td>
<td>338,699</td>
<td>338,700</td>
<td>47,236</td>
<td>97.66</td>
</tr>
</tbody>
</table>

Table 1 summarizes the datasets for our experimental study. The last column also reports the test classification accuracies for the linear kernel at the best SVM $l_2$-regularization $C$ value. Because the RBF kernel also needs a crucial tuning parameter $\gamma$, we tune $\gamma$ from a wide range $\gamma \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ either on the original dataset or a subset of the data. (For example, 3:1:20 means the number ranges from 3 to 20 spaced at 1.) In short, we have tried our best to optimize the performance of RBF and its linearization.

We report the test classification accuracies for four algorithms and $k \in \{32, 64, 128, 256, 512, 1024\}$. For RCV1, we also report results for $k \in \{16, 2048, 4096\}$. In practice, we expect that even $k = 1024$ might be too large. To ensure repeatability, we always report the results for a wide range of $l_2$-regularization $C$ values.

Figure 1: Letter: Test classification accuracies for 6 different $k$ values and 4 linearization: 1) GMM-GCWS, 2) GMM-NYS, 3) RBF-NYS, 4) RBF-RFF. We use LIBLINEAR package [16] for training linear SVMs for a wide range of $l_2$-regularization $C$ values. Results are averaged over 10 repetitions.

Figure 1 reports the test classification accuracies on Letter, for 6 different samples sizes ($k$) and 4 different algorithms: 1) GMM-GCWS, 2) GMM-NYS, 3) RBF-NYS, 4) RBF-RFF. Again, we should emphasize that the storage and computational cost are largely determined by the sample size $k$, which is also the number of nonzero entries per data vector in the transformed space. Thus, it might be more practically useful to examine the classification results at smaller $k$ values. Since the original dimensionality of Letter is merely 16, in this case even $k = 32$ might be interesting.

Note that, for Letter, the original classification accuracy using linear SVM is very low (61.66%, see Table 1). We can see from Figure 1, that with merely $k = 32$, both GMM-GCWS and GMM-NYS already produce much more accurate results linear. With larger $k$, the results become even much better.
Figures 2 and 3 report the test classification accuracies for Webspam and PAMAP105, respectively. Again, these figures confirm that GMM-GCWS and GMM-NYS produce good results.

Figures 4, 5, and 6 report the test classification results on the RCV1 dataset. Because the performance of RBF-RFF is so much worse than other methods, we report in Figure 4 only the results for GMM-GCWS, GMM-NYS, and RBF-NYS, for better clarity. In addition, we report the results for \( k \in \{4096, 2048, 16\} \) in Figure 6, to provide a more comprehensive comparison study.

The empirical results support the claim that GMM and its linearization methods can be effective in machine learning tasks. Thus, it becomes interesting to understand the behavior of GMM \( g_n(x,y) \). For example, what is the limit of \( g_n(x,y) \) and how fast does it converge to the limit? These questions can not be precisely answered without making assumptions on the data. While normality is a typical assumption, it is too restrictive and in fact the results based on normal assumptions can be misleading. The \( t \) distribution is much more general with the capability of modeling heavy-tailed data. In this paper, we will go much further by considering the general family of elliptic distributions.

We should also mention that, owing to the discrete nature, the hashed data produced by GMM-GCWS can be directly used for building hash tables for efficient (sublinear time) near neighbor search, while the hashed data produced by other 3 algorithms can not. This is an additional advantage of GMM-GCWS.

Compared with the original empirical study on GMM in the initial work [26], the new experiments in this paper do not seem to provide a surprising conclusion. On these datasets in Table 1, GMM-NYS significantly outperforms RBF-RFF. Nevertheless, GMM-GCWS still performs the best among the four methods.
Figure 5: RCV1: Test classification accuracies for 6 different $k$ values and 4 different algorithms.

Figure 6: RCV1: Test classification accuracies for $k \in \{16, 2048, 4096\}$ and 4 different algorithms.

3. BEYOND NORMAL ASSUMPTION

The cosine similarity $c_n(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$ is popular largely due to the normal assumption. That is, if each coordinate $(x_i, y_i)$ is an iid copy of $(X, Y)$, where

$$(X, Y)^T \sim N \left( \mathbf{0}, \Sigma = \begin{pmatrix} \sigma_X^2 & \rho \sigma_X \sigma_Y \\ \rho \sigma_X \sigma_Y & \sigma_Y^2 \end{pmatrix} \right)$$

then as $n \to \infty$, $c_n(x, y)$ converges in distribution to a normal [2]

$$n^{1/2} (c_n - \rho) \overset{D}{\to} N \left( 0, (1 - \rho^2)^2 \right)$$

Even though it is known that practical data are typically not normal, practitioners tend to believe that the normality assumption is not essential in that as long as the data have bounded second moment, cosine will still converge to the right quantity. While this is true, we need to pay close attention to the variance. If the variance does not exist or is too large, then cosine is not reliable.

Beyond normality, the obvious choice of distribution is the $t$-distribution with $\nu$ degrees of freedom, which converges to normal when $\nu \to \infty$. Denote by $t_{\nu, \Sigma}$ the bivariate $t$-distribution with covariance matrix $\Sigma$. Basically, if two independent variables $Z \sim N(0, \Sigma)$ and $u \sim \chi^2_{\nu}$, then we have $Z / \sqrt{\nu / u} \sim t_{\Sigma, \nu}$.

The following Corollary 1 can be inferred from Theorem 3.

**Corollary 1.** Consider $n$ iid samples $(x_i, y_i) \sim t_{\Sigma, \nu}$, where $\nu > 4$ and $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$, $-1 \leq \rho \leq 1$. As $n \to \infty$, $c_n(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$ converges in distribution to a normal:

$$n^{1/2} (c_n - \rho) \overset{D}{\to} N \left( 0, \frac{\nu - 2}{\nu - 4} (1 - \rho^2)^2 \right)$$

Corollary 1 basically says that, if the data have bounded fourth moment ($\nu > 4$), then it might be ok to use the cosine similarity. However, to be more safe, it is better that the data have finite higher-order moments. This means there are actually quite stringent conditions we must check if we hope to use cosine safely.

Next, we study the GMM kernel $g_n(x, y)$ defined in (4). The following Corollary 2 can be inferred from Theorem 2.

**Corollary 2.** Consider $n$ iid samples $(x_i, y_i) \sim t_{\Sigma, \nu}$, where $\nu > 2$ and $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$, $-1 \leq \rho \leq 1$. As $n \to \infty$, $g_n(x, y)$ defined in (4) converges in distribution to a normal:

$$n^{1/2} \left( g_n - \frac{1 - \sqrt{(1 - \rho)/2}}{1 + \sqrt{(1 - \rho)/2}} \right) \overset{D}{\to} N \left( 0, \frac{V}{H^2 \pi} \frac{1}{\nu - 2} \Gamma^2(\nu/2) \right)$$

where $\Gamma(.)$ is gamma function, $H = \frac{\pi^2}{2} (1 + \sin \alpha)$, $\alpha = \sin^{-1} \left( \sqrt{1/2 - \rho/2} \right)$, and $V = \frac{4}{\pi} \sin^2 \alpha \times (3\pi - 8\cos \alpha + 2\sin 2\alpha + \pi \cos 2\alpha - 8\alpha \sin \alpha - 4\alpha \cos 2\alpha)$.

We can see that the results for GMM are sophisticated even for this “simple” scenario. To help readers understand the behavior of GMM and verify the theoretical results, in this section we provide a simulation study for $g_n(x, y)$. We consider $n$ iid samples $(x_i, y_i) \sim t_{\Sigma, \nu}$ and compute $g_n(x, y)$ according to (4), for $n \in \{1, 10, 100, 1000, 10000\}$ and $\nu \in \{3, 2, 1, 0.5\}$. 

1057
As shown by [32], most natural datasets have the equivalent zoomed-in version of the left panels. Because and the convergence still holds at the boundary case (i.e., \( \nu = 1 \)).

In each panel, the top (dashed and green if color is available) curve represents \( f_1 \), and the bottom (dashed and red) curve represents \( f_\infty \). For \( \nu = 1 \), \( g_n \) still converges to \( f_\infty \) but much slower. With \( \nu = 0.5 \), \( g_n \) does not converge to \( f_\infty \). Basically, the simulations suggest that \( g_n \to f_\infty \) as long as data have bounded first moment (i.e., \( \nu > 1 \)) and the convergence still holds at the boundary case (i.e., \( \nu = 1 \)). Because \( \rho \) measures similarity, the fact that \( g_n \to f_\infty \) as long as \( \nu \geq 1 \) is important because it means we have a robust measure of \( \rho \).

As shown by [32], most natural datasets have the equivalent \( \nu > 1 \). While the \( t \)-distribution is a popular choice for modeling data beyond normality, practical data can be much more complex. We will provide general results for elliptical distributions.

4. THEORETICAL ANALYSIS OF GMM UNDER ELLIPTICAL DISTRIBUTIONS

We consider \( (x_i, y_i) \), \( i = 1 \) to \( n \), are iid copy of \( (X, Y) \). Our goal is to analyze the statistical behavior of GMM, as \( n \to \infty \),

\[
g_n(x, y) = \frac{\sum_{i=1}^{n} \min(x_i + y_i, y_i) + \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i) + \max(x_i, y_i)}
\]

To proceed with the theoretical analysis, we make a very general distributional assumption on the data. We say the vector \((X, Y)\) has an elliptical distribution if

\[
(X, Y)^T = AUT = \left( a_1^T UT \right) \left( a_2^T UT \right)
\]

where \( A = (a_1, a_2)^T \) is a deterministic \( 2 \times 2 \) matrix, \( U \) is a vector uniformly distributed in the unit circle and \( T \) is a positive random variable independent of \( U \). See [2] for an introduction.

In this family, there are two important special cases:

1. **Gaussian distribution**: In this case, we have \( T^2 \sim \chi^2_2 \)

\[
(X, Y)^T \sim N(0, \Sigma) \sim AU \sqrt{\chi^2_2}.
\]

where \( \Sigma = AA^T = \left( \sigma^2 \sigma^2 \right) \).

Note that for analyzing \( g_n \), it suffices to set \( \text{Var}(X) = 1 \), due to cancelation effect in the definition of GMM and cosine.

2. **\( t \)-distribution**: In this case, we have \( T \sim \sqrt{\chi^2_2 / \chi^2_2} \)

\[
(X, Y)^T \sim N(0, \Sigma) \sim \sqrt{\chi^2_2 / \chi^2_2}.
\]

Note that in \( \Sigma \) we consider \( \sigma \neq 1 \) to allow the situation where two vectors have different scales. For the convenience of presenting our theoretical results, we summarize the notations:

- \( \Sigma = \left( \begin{array}{cc} \sigma^2 & \rho \\ \rho & \sigma^2 \end{array} \right) \) \( \), where \( \rho \in [-1, 1] \) and \( \sigma > 0 \).

- \( \alpha = \sin^{-1}(\sqrt{1 - \rho^2}) \) \( \), \( \in [0, \pi/2] \).

- \( \tau \in [-\pi/2 + 2\alpha, \pi/2] \) is the solution of \( \cos(\tau - 2\alpha)/\cos(\tau) = \sigma \), i.e., \( \tau = \arctan(\sigma / \sin(2\alpha) - \cot(2\alpha)) \).

Note that \( \tau = \alpha \) if \( \sigma = 1 \).

In addition, we need the following definitions of \( f_1(\rho, \sigma) \) and \( f_\infty(\rho, \sigma) \), for general \( \sigma \) as well as \( \sigma = 1 \):

\[
f_1(\rho, \sigma) = \frac{1}{\sigma^2} \left[ (\tau + \pi/2 - 2\alpha) \cos(2\alpha) + \sin(2\alpha) \log \frac{\cos(2\alpha - \pi/2)}{\cos(\tau)} \right] + \frac{\sigma}{\pi} \left( (\pi/2 - \tau) \cos(2\alpha) + \sin(2\alpha) \log \frac{\cos(2\alpha - \pi/2)}{\cos(2\alpha - \tau)} \right)
\]

\[
f_\infty(\rho, \sigma) = \frac{1}{\sigma^2} \left( \sqrt{1 - \rho^2} \log(2 - 2\rho) - 2\rho \sin^{-1}(\sqrt{(1 - \rho)/2}) \right)
\]

In each panel, the top (dashed and green if color is available) curve represent \( f_1 \) and the bottom (dashed and red) curve represent \( f_\infty \). We can see that for \( \nu = 3 \) and \( \nu = 2 \), \( g_n \) converges to \( f_\infty \) fast. For \( \nu = 1 \), \( g_n \) still converges to \( f_\infty \) but much slower. With \( \nu = 0.5 \), \( g_n \) does not converge to \( f_\infty \). Basically, the simulations suggest that \( g_n \to f_\infty \) as long as data have bounded first moment (i.e., \( \nu > 1 \)) and the convergence still holds at the boundary case (i.e., \( \nu = 1 \)).
Theorem 1 presents the results for consistency, i.e., \( g_n \to f_\infty \). Theorem 2 presents the results for asymptotic normality, i.e., the speed of the convergence. The proofs are technical and tedious and we will provide simulations to verify the results.

**Theorem 1. (Consistency)** Assume \((X, Y)\) has an elliptical distribution with \((X, Y)^T = \mathbf{AUT} \) and 
\[
\Sigma = AA^T = \begin{pmatrix} 1 & \sigma_p \\ \sigma_p & \sigma^2 \end{pmatrix}.
\]
Let \( (x_i, y_i) \), \( i = 1 \) to \( n \), be iid copies of \((X, Y)\), and \( g_n(x, y) \) as defined in \((4)\). Then the following statements hold:

(i) \( g_1 = f_1(\rho, \sigma) \)

(ii) If \( \mathbb{E}T < \infty \), then \( g_n \to f_\infty(\rho, \sigma) \), almost surely.

(iii) If we have 
\[
\lim_{t \to \infty} t \frac{\mathbb{P}(T > t)}{\mathbb{E} \min(T, t)} = 0,
\]
then \( g_n \to f_\infty(\rho, \sigma) \), in probability.

(iv) If \((X, Y)\) has a t-distribution with \( \nu \) degrees of freedom, then \( g_n \to f_\infty(\rho, \sigma) \) almost surely if \( \nu > 1 \) and \( g_n \to f_\infty(\rho, \sigma) \) in probability if \( \nu = 1 \).

**Theorem 2. (Asymptotic Normality)** With the same notation and definitions as in Theorem 1, the following statements hold:

(i) If \( \mathbb{E}T^2 < \infty \), then 
\[
\frac{n^{1/2}}{} (g_n(x, y) - f_\infty(\rho, \sigma)) \overset{D}{\to} N \left( 0, \frac{V}{H^4} \frac{\mathbb{E}T^2}{\mathbb{E}T} \right)
\]

where 
\[
V = \frac{1}{4\pi^3} \left\{ 2 \tau + \pi - 4 \alpha + \sin(2 \tau - 4 \alpha) + \sigma^2 (\pi - 2 \tau - \sin(2 \tau)) \right\} 
\times \left\{ \sigma(1 + \sin(2 \tau)) + 1 + \sin(2 \tau - \sin(2 \tau)) \right\} 
+ \frac{1}{4\pi^3} \left\{ \sigma^2 (2 \tau + \sin(2 \tau) + \pi) + 4 \sigma (\sin 2 \tau - 2 \cos 2 \alpha) \sin 2 \alpha \right\} 
\times \left\{ (\pi - 2 \alpha - 2 \sin 2 \alpha - 2 \sin 2 \alpha - 2 \cos 2 \alpha) \right\} 
\times \left\{ (\pi - 2 \alpha) - 2 \cos 2 \alpha + 2 \sin 2 \alpha + 2 \sin 2 \alpha - 2 \sin 2 \alpha \right\}
\]

and 
\[
H = \frac{1}{\pi} \left\{ \sigma(1 + \sin(2 \tau)) + 1 + \sin(2 \tau - \sin(2 \tau)) \right\} \sigma^2 \frac{2}{\pi} \frac{1 + \sin(2 \tau - \sin(2 \tau))}{\pi} \]

(ii) If \((X, Y)\) has a t-distribution with \( \nu \) degrees of freedom and \( \nu > 2 \), then 
\[
\frac{n^{1/2}}{} (g_n(x, y) - f_\infty(\rho, \sigma)) \overset{D}{\to} N \left( 0, \frac{V}{H^4} \frac{\mathbb{E}T^2}{\mathbb{E}T} \right),
\]

where \( \mathbb{E}T^2 = \frac{2 \nu}{\nu - 2} \) and \( \mathbb{E}T = \frac{\sqrt{\Gamma(\nu/2 - 1/2)} \Gamma(\nu/2)}{2 \Gamma(\nu/2)} \).

(iii) If \((X, Y)\) has a t-distribution with \( \nu = 2 \) degrees of freedom, then 
\[
\left( \frac{n}{\log n} \right)^{1/2} (g_n(x, y) - f_\infty(\rho, \sigma)) \overset{D}{\to} N \left( 0, \frac{V}{H^4} \frac{4}{\pi^2} \right).
\]

Figure 8 presents a simulation study to verify the asymptotic normality, in particular, the asymptotic variance formula
\[
\text{Var}(g_n) = \frac{V}{H^4} \frac{\mathbb{E}T^2}{\mathbb{E}T} + O \left( \frac{1}{n^2} \right) \quad (14)
\]

by considering data from a t-distribution with \( \nu \) degrees of freedom and \( \nu \in \{2.5, 3, 4, 5\} \). Each case, we repeat the simulation 10000 times. We report the empirical \( \text{Var}(g_n) \times n \) with the theoretical asymptotic value
\[
\frac{V}{H^4} \frac{\mathbb{E}T^2}{\mathbb{E}T}
\]
plotted as dashed curves. For \( n \) large enough, the asymptotic variance formula \((14)\) becomes accurate. For small \( n \) values, however, the formula can be quite conservative.

5. **ESTIMATION OF SIMILARITY \( \rho \)**

The fact that \( g_n(x, y) \to f_\infty(\rho, \sigma) \) also provides a robust and convenient way to estimate the similarity between data vectors. Here, for convenience we consider \( \sigma = 1 \). For this case, we have 
\[
f_\infty = \frac{1 - \sqrt{1 - \rho^2}}{1 + \sqrt{1 - \rho^2}}.
\]
This suggests an estimator of \( \rho \):
\[
\hat{\rho}_g = 1 - 2 \left( \frac{1 - g_n}{1 + g_n} \right)^2
\]

As \( n \to \infty \), \( g_n \to f_\infty \) and \( \hat{\rho}_g \to \rho \). In other words, the estimator \( \hat{\rho}_g \) is asymptotically unbiased. The asymptotic variance of \( \hat{\rho}_g \) can be computed using “delta method”: (Note that \( \rho_g = 8 \frac{\frac{1}{n} g_n}{(1 + g_n)^2} \))
\[
\text{Var}(\hat{\rho}_g) = \left[ \frac{8}{(1 + f_\infty)^4} \text{Var}(g_n) + O \left( \frac{1}{n^2} \right) \right]
\]
\[
= \frac{1}{n} \left( 1 + \sqrt{1 - \rho^2} \right)^4 \frac{V}{H^4} \frac{\mathbb{E}T^2}{\mathbb{E}T} + O \left( \frac{1}{n^2} \right)
\]

See Theorem 2 for \( V \) and \( H \). This estimator \( \hat{\rho}_g \) is meaningful as long as \( \mathbb{E}T < \infty \) and \( \text{Var}(\hat{\rho}_g) \) is less than as long as \( \mathbb{E}T^2 < \infty \).

It is interesting to compare \( \rho_g \) with the commonly used estimator based on the cosine similarity: 
\[
c_n(x, y) = \frac{1}{\sqrt{\sum_{i=1}^{n} x_i y_i}}
\]
When the data are bivariate normal, it is a known result [2] that 

\[ c_n(x, y) \sim N(0, (1 - \rho^2)^2) \]

Here, we provide a general result for the elliptical distribution.

**Theorem 3.** If \( ET^4 < \infty \), then

\[ n^{1/2} (c_n - \rho) \sim N \left( 0, \frac{ET^4}{2E T^2} (1 - \rho^2)^2 \right) \]

Based on Theorem 3, a natural estimator of \( \rho \) and its asymptotic variance would be

\[ \hat{\rho}_c = c_n, \quad Var(\hat{\rho}_c) = \frac{1}{n} \frac{ET^4}{2E T^2} (1 - \rho^2)^2 + O \left( \frac{1}{n^2} \right) \]

When data follow a \( t \)-distribution with \( \nu \) degrees of freedom, we have \( ET^2 = \frac{\nu - 2}{\nu - 2}, \quad ET^4 = \frac{\nu n^2 (\nu - 4)}{(\nu - 2)(\nu - 4)} \).

Figure 9 presents a simulation study for comparing the two estimators: \( \hat{\rho}_Y \) and \( \hat{\rho}_c \). We assume \( t \)-distribution with \( \nu \) degrees of freedom, where \( \nu \in \{2.5, 3, 4, 5, 6, 8, 10\} \) as well as \( \nu = \infty \) (i.e., normal distribution). In each panel, we plot the empirical mean square errors (MSEs): \( MSE(\hat{\rho}_Y) \) and \( MSE(\hat{\rho}_c) \) (computed from 10000 repetitions), along with the (asymptotic) theoretical variance of \( \hat{\rho}_Y: \frac{\nu}{n} \left( 1 - \rho \right) \left( 1 + \frac{\nu}{n} \right)^2 \left( \frac{1}{E T^2} \right)^2 \). For clarity, we did not plot the theoretical variance of \( \hat{\rho}_C \), which is fairly simple and more straightforward to be verified.

Figure 9 confirms that \( \hat{\rho}_Y \), the estimator based on GMM, can be substantially more accurate than \( \hat{\rho}_C \), the estimator based on cosine. Roughly speaking, when \( \nu < 8 \), \( \hat{\rho}_Y \) is more preferable. Even when the data are perfectly Gaussian (the bottom row in Figure 9), using \( \hat{\rho}_Y \) does not result in much loss of accuracy compared to \( \hat{\rho}_C \).

**6. CONCLUDING REMARKS**

The cosine similarity is still widely used in practice. Whenever we use linear models and normalize input data, we effectively adopt the cosine similarity. Some more sophisticated measures such as the RBF kernel are also based on cosine. It has been realized that real-world data in networking, web search and data mining are much more complex [24, 11, 15, 32] and a robust (and still simple) similarity measure would be desirable. There are already attempts in information retrieval for finding robust data models, e.g., [14]. The contributions of this paper can be summarized as follows:

1. We prove the asymptotic normality of cosine by assuming a general family of elliptical distribution. Our results real that cosine is not robust even when data are not so heavy-tailed.

2. We prove the consistency and asymptotic normality of the GMM kernel, also by assuming elliptical distributions. The proofs are technical and do not fit in the page limit, but we provide the details in the arXiv. The proofs can also be validated by simulations. The consistency and normality results help explain the recent success of GMM (and variants) [26, 27] in machine learning applications, and may lay foundation for future research on this direction.

3. We add an empirical evaluation by linearizing the GMM kernel using the Nyström method, which was not included in the initial work on GMM [26]. Basically, the experimental results demonstrate that GMM linearized via Nyström can also be substantially more accurate than RBF linearized by...
(normalized) random Fourier features. Nevertheless, on the datasets considered in this paper (i.e., Table 1), GCWS hashing as presented in [26] still provides a more accurate approximation to the GMM kernel than the Nystrom method.

We are enthusiastic that the GMM kernel and its variants (e.g., tunable GMMs) are promising to become standard tools in machine learning. Clearly, the line of work on GMM is largely built on the efforts from the web search community, in particular, the prior works on resemblance, minwise hashing, min-max kernel, etc.

**Acknowledgment**

The work is partially supported by NSF-Bigdata-1419210, NSF-III-1360971, and NSF-DMS-1513378.

7. REFERENCES


