Golay Code Transformations for Ensemble Clustering in Application to Medical Diagnostics

Faisal Alsaby
Computer Science Department
The George Washington University
Washington, DC

Kholood Alnowaiser
Computer Science Department
The George Washington University
Washington, DC

Simon Berkovich
Computer Science Department
The George Washington University
Washington, DC

***Abstract*—Clinical Big Data streams have accumulated large-scale multidimensional data about patients' medical conditions and drugs along with their known side effects. The volume and the complexity of this Big Data streams hinder the current computational procedures. Effective tools are required to cluster and systematically analyze this amorphous data to perform data mining methods including discovering knowledge, identifying underlying relationships and predicting patterns. This paper presents a novel computation model for clustering tremendous amount of Big Data streams. The presented approach is utilizing the error-correction Golay Code. This clustering methodology is unique. It outperforms all other conventional techniques because it has linear time complexity and does not impose predefined cluster labels that partition data. Extracting meaningful knowledge from these clusters is an essential task; therefore, a novel mechanism that facilitates the process of predicting patterns and likelihood diseases based on a semi-supervised technique is presented.**

Keywords—medical Big Data; clustering; machine learning; pattern recognition; prediction tool; Big Data classification; Golay Code

I. INTRODUCTION

Medical research is one of the most significant fields of science for people since no one is completely protected from physical ailments and biological degradation. It is not a surprise that health care is expensive. In 2010, the United States alone spent $2.6 trillion in health care expenditures, nearly 17.9 percent of the United States gross domestic product (GDP). The expenses are projected to consume 19.9 percent of GDP by 2022 [1]. According to estimates, 3 million baby boomers will hit retirement age every year for the next 20 years, challenging an already stressed health care system [2]. Chronic diseases form an even bigger challenge, considering that more than 75 percent of health care expenditures are spent on people with chronic conditions [3]. Even though this number is high, it can be dramatically decreased by the power of prevention. Although we are able to generate and store enormous amounts of patients' medical data, physicians nowadays lack techniques that deal with Big Data challenge. More specifically, physicians are not capable of effectively quantify and analyze the relationship between medical data and causes of diseases, and predict the likelihood of diseases based on discovered patterns. However, risk is estimated by considering the patient's family history and the results of necessary laboratory exams. This is highly dependent on the physician's limited experience. Therefore, this model of health care must be replaced with a new one that helps not only to early predict diseases but to prevent them even before patients show any symptoms. This paper presents a mechanism to encode the medical records patterns and generate the codewords that will be clustered by utilizing the perfect Golay code. This novel approach is suitable for processing continuous data streams [4]. With this clustering methodology, sensible information from underlying clusters can be extracted.

A cluster is defined as a data container with homogeneous data points inside of it. On the other hand, the data points from different clusters are non-homogenous. Technically, clusters isolate data points with boundaries such that the data points within the same cluster share common patterns or characteristics [5]. The Golay code clustering technique requires using vectors to represent any type of data, such as person's information, RNA sequencing, DNA sequencing, diseases, drugs and their side effects and so on. Each vector consists of 23-bit, where each bit represents the presence or the absence of a feature. For example, if a patient is tested positive to symptom x, it is represented in the vector as 1. Otherwise, the symptom is represented as 0. However, in some cases the proposed methodology provides the option of using Gray code property where 2 bits might be used to represent a single feature such as blood pressure level. Using Gray code property, this can be represented as 10, 00, or 01, to express high, normal, or low blood pressure level respectively. For the realization of this clustering method, a particular ontology approach must be considered which is called "Meta Knowledge of 23-bit Templates" [6]. These templates are an essential aid that assists in providing highly efficient clustering algorithm. The 23 questions needed to form vectors are included in 23-bit Meta knowledge template, which must not be arbitrary developed. In some circumstances the number of questions might be less than 23. Golay code clustering algorithm is distinctive, because of its linear time complexity and by allowing Fuzzy clustering. Therefore, it outperforms all other conventional clustering methods such as K-means. As a result, this method is an effective tool for handling the convoluted problems arising with the "Big Data" computational model in the medical field. Many medical applications might be considered in this regard. For instance, comparisons of protein and DNA sequences.

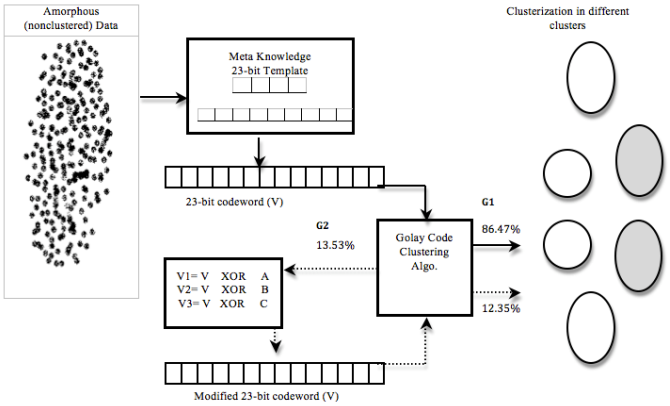

Fig. 1. Overview of Golay Code Clustering Method

This method can also be used to search sequences, find patterns, evaluate similarity and periodic structures based on local sequence similarity [7]. This paper is organized as follows: in section 2, we present some theoretical analysis to demonstrate the applicability of the proposed algorithm. In section 3, we discuss the proposed clustering algorithm. Section 4 presents the pattern recognition method. In section 5, experimental results on synthetic data are presented. Finally, section 6 contains our concluding remarks.

II. GOLAY CODE

The proposed clustering system is based on a reverse of the traditional error-correction scheme using the perfect Golay code (23, 12, 7) as described in [4]. Utilizing this perfect code, the whole set of 23-bit vectors is partitioned into 2^{12} spheres with radius 3. Thus, a transformation that maps the 23-bit string into the 12-centers of these spheres is able to tolerate certain dissimilarity in some bit positions of the 23-bit strings. Luckily, the Golay code is a perfect code that can tolerate up to three error bits [8]. Hence, this property allows adequate codewords to be associated with a single data word i.e. $\binom{23}{3}$ =1771 different codewords. The binary Golay code has a very large data word (2^{12} data words) and a larger codeword space (2^{23} = 8,388,608 codewords). This large space makes Golay Code appropriate for clustering. One interesting property of the Golay code scheme appears when decoding different codewords from the same sphere *m* . The different codewords will all be restored into the same data word. Hence, two random spheres *n1, n2* will have one or more data words (indices) in common if and only if they have common hosting spheres. Therefore, the six data words that are associated with any *n* can be used to create clustering keys for the codeword *n* (Yu, 2011). For example, suppose we have two 23-bit vectors represented by two integers: 1036 ($2^{10} + 2^3 + 2^2$) and 1039 ($2^{10} + 2^3 + 2^2 + 2^1 + 2^0$). The two vectors differ in the two last bit positions. Their six hash indices turn out to be (0, **1054**, 1164, **1293**, 1644, 3084) and (527, **1054**, 1063, 1099, **1293**, 3215) respectively. The hamming distance between the code words 1036 and 1039 is 2, thus, they generate more than one identical index. This property guarantees that the two codewords are placed into a common cluster. As shown in the example, there are two common indices that are generated by both vectors, 1054 and 1293. Intrinsically, concatenating these two data words would provide us with a clustering address where both 1036 and 1039 would be placed in it. Such an approach leads to access the same cluster that contains both of them when searching for either pattern or their neighbors. Based on that, in order to utilize this clustering scheme, *n* must be restored back to six different data words. But the only way that *n* can be decoded into six different data words is when the center of *n* is 3 Hamming distance away from the hosting sphere [9]. In practice, using only one Golay code scheme results in clustering 86.5% of the total vectors (we call them *G1*) while the remaining13.5% does not fit to this scheme (we call them *G2*). In other words, 86.5% of the vectors are able to generate the six data words (indices), which are required for the clustering process, while the remaining 13.5% of the codewords are not able to produce the necessary indices. One possible attempt of clustering *G2* codewords is shown in Fig.1. We investigate the ability of each vector $V \varepsilon G2$ to generate the six indices. A tolerance of 1-bit mismatch can be implemented by probing each hash index corresponding to all 1-bit modification of a given codeword. Therefore, we create three 23-bit codewords *A*, *B*, and *C* where their values are the numbers 1,2 and 4 respectively. After that, by performing bitwise XOR operation between the original codeword and each one of the new codewords *A*, *B* and *C*, new vectors *V1*, *V2* and *V3* are created. As a result of applying Golay code hash transformation to these vectors, two situations are presented. In the first case, 12.35% of the modified *G2* codewords are able to generate the six indices, thus the clustering method proceeds as normal. The remaining 1.17% can only generate one index; hence, in some circumstances, these codewords might be neglected [10]. Another way of clustering *G2* codewords is based on using double Golay codes, which can be generated by the polynomials 2787 and 3189. Based on a previous work [4], this approach, however, is able to cluster 98.2% of the 8,388,608 codewords.

III. CLUSTERING COMPONENTS

A. Meta Knowledge Template

To facilitate the using of our clustering algorithm, a template of yes/no questions for each data item is necessary. A group of "23-bit Metadata Template" that is suitable for the medical case is designed. Questions should be based on acute physiological measurements. Each of these questions investigates the presence or absence of a property, a symptom, or a feature as shown in Fig.2. Moreover, this 23-bit Metadata Template can be utilized in a way such that complex values like DNA and RNA sequencing can be represented in the 23-bit codeword. One possible approach of designing such a template is to investigate DNA sequencings and find patterns and examine the correlation between diseases, mutations, Single Nucleotide Polymorphism (SNP), as well as the surrounding environment. Answering the questions results in a unique 23-bit vector *V* as in Fig. 1. *V* is then computed by the Golay code clustering algorithm where the output of this process is six different indices. A pairwiseing process for these six indices is applied to compose 15 cluster addresses. Subsequently, *V* is stored in each of the corresponding 15 clusters. This technique guarantees storing data items in one cluster if the difference between each two of them does not

exceed a certain number of bit-position mismatches. In other words, this clustering technique assures that the distance between any two vectors included in one cluster does not exceed a certain Hamming distance, as with Fig.2. It is important to recall that when mapping each codeword, we employ the binary Golay code, which guarantees that close decimal numbers have low Hamming distance in their binary representation. When applying our proposed clustering algorithm, all 23-bit codewords are classified into a number of clusters. The maximum Hamming distance within each cluster is either 7 or 8. The total number of bit positions that have common bit values within each cluster is either 15 or 16. This is specifically significant since it represents the total number of common attributes between codewords within a certain cluster. Put in mind that as bit positions may have different physical meanings, a low Hamming distance alone does not mean that two codewords are similar.

23-bit Metadata Template	
Q1	Is the patient a female?
Q2	Is the patient obese?
Q3	Does the patient have a family history of breast cancer?
Q4	Does the patient have BRCA1 or BRCA2 gene mutations?
…..	…..
Q23	Did the patient receive any radiation treatments?

Fig. 2. 23-bit Metadata Template

As discussed above, codewords are created through answering the questions in the 23-bit Meta Knowledge Template. Each codeword consists of a 23-bit; each bit represents the presence or absence of a feature. Consequently, when two codewords have similar answers for the same questions within the template, these two codewords have similar features. Hamming distance is used to measure the similarity between codewords. Hamming distance between two codewords is the number of bits we must change to convert one codeword into the other. For example: the Hamming distance between the vectors **0**1**10**101**0** and 11011011 is 4. This methodology is considered one of the most simple, efficient, and accurate distance measures [11].

B. Composing Clustering Addresses

The overall clustering algorithm structure is shown in Fig.1. To illustrate the proposed methodology that uses the Golay code hash transformation, let *V* be a 23-bit codeword that is created by answering the question within the 23-bit Meta Knowledge Template. By using only one Golay code scheme and utilizing the Gray code property; the six 12- bit data words are generated for *V*. Clustering keys will be influenced by these six 12-bit data words. We start by choosing two arbitrary 12-bit data words of the 6 generated indices. Then, we order the selected two data words (such as, *w1<w2*). After that, we remove the least significant bit (LSB) of the smallest pair *w1* and concatenate the result with the second data word to form a 24-bit *A*. After that, we shift *A* one bit to the right to get another 24-bit *B*. We then perform bitwise XOR operation between *A and B* to get a 24-bit, *C*. The last 23-bit of *C* is the clustering key. The following algorithm shows how clustering keys are generated:

At least two common indices are generated by two 23-bit vectors at Hamming distance 2, as with the example aforementioned where the codewords were 1036 and 1039. Thus, when we pairwise (concatenate) these two common indices to generate the 23-bit clustering key, it is possible to place these vectors into the same cluster.

C. The Structure of Clusters

Clusters are essential components in our classification and prediction methodology due to its ability to discover the connected components of patients [12]. Because fuzziness is one of the most salient features of the "Big Data" concept, underlying relationships can be detected by using Golay code clustering technique. Furthermore, clusters assist in reducing the influence of patients who have little or no similarity i.e. common symptoms. When applying the Golay Code clustering algorithm to the possible 23-bit vectors (8,388,608 vectors), a total of 1,267,712 non-empty clusters were created. Each one of the generated clusters contains (139) or (70) codewords. For simplicity, we call them larger cluster (LC) and smaller cluster (SC) respectively. The maximum Hamming distance within each cluster is either 7 or 8. More importantly, the minimum total number of bit positions that have common bit values within each cluster is either 15 or 16. This is specifically a significant feature since it represents the total number of common attributes between codewords within a certain cluster.

Algorithm 1: Composing Clustering Addresses

1. *generate the 6 data words*
2. *loop i=1 to 15*
3. *pick 2 random data words: w1,w2*
4. *order them such as w1<w2*
5. *right shift the smallest data word such as w1 >>1*
6. *A= w1 w2*
7. *B= A >> 1*
8. *Ci= B XOR A*
9. *clustering_keys[i]= Ci (only last 23 bit of C is used)*
10. *end loop*
11. *Return clustering_keys*

For example, in Fig.2, the first two codewords have 19 features in common. More importantly, within each one of the SLs, 98.55% of the codewords have at least 17 common features, while the remaining codewords have either 16 or 15. On the other hand, 86.25% of the codewords in LCs have at least 17 common features, while 13.75% share either 16 or 15.

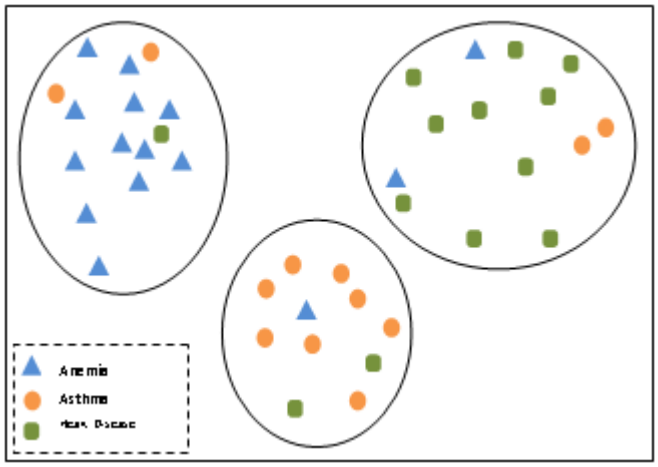

Fig. 3. Labeled Clusters based on the Majority Vote

IV. Data Items and Clusters Labeling Method

A. Training Method

Unlabeled data forms a major challenge that machine learning and data mining systems are facing [13][14][15]. Far better results can be obtained by adopting a machine learning approach in which a large set of N vectors {x1,..., xN} called a training set is used to tune the parameters of an adaptive model[book]. Our pattern recognition procedure starts by training the system with a fully labeled training dataset (we call them centers). Specifically, the dataset is a collection of vectors that represent the identity of corresponding medical conditions or diseases, for instance, heart disease, Asthma, Breast cancer etc. These vectors will be employed to label objects that already were clustered. We sequence through clusters and find the nearest center to each clustered codeword in terms of Hamming distance. The label of the codeword is basically the exact label of the nearest center. When all codewords labeling process is fulfilled, labeling clusters becomes trivial. For example, assume that *V1* represents Asthma, *V2* represents Anemia, and *V3* depicts Heart diseases. Subsequently, we find the minimum Hamming distance between each vector in the system and *V1*, *V2*, *V3*. If the Hamming distance does not exceed a certain number of distortions, the vector's label is the same as the label of the nearest center. After labeling codewords, we rank objects among every cluster according to their frequency, regardless of whether they occur in other clusters within the system [14][15]. The label of the cluster depends solely on the majority weight within this cluster, i.e. prevalent element. Some clusters have different types where one type dominates that cluster or weighs more. Thus, the weight *Wn* of each object within a cluster is simply its frequency in that cluster.

$$w_n = Frequency[n]$$

Wn is the weight of the object *n*.

As a result, the vote of the majority within a cluster influences the label of the cluster. Noise is a factor that might reduce the accuracy of labeling process. Therefore, a threshold is recommended to insure high accuracy and efficiency. Cluster is granted the right to vote when it contains at least 10 codewords. Table (1) presents an example of the labeling process. Eventually, when the majority of clusters are labeled, the process of assigning a label to a new vector becomes a trivial. The label of a new vector is determined during the clusterization method. In particular, after attaching the new vector to the appropriate 15 clusters, its label will be assigned instantly. The assignment works by receiving a vote from each one of the 15 clusters i.e. the vote of a cluster is basically its label. Hence, the label of the new vector is the majority vote among its 15 clusters [16]. For example, if 10 out of 15 clusters are labeled with Asthma and 5 are labeled with Heart disease, the new vector is labeled with Asthma. Prior work indicates that the accuracy of the assignment is 92.7% [4].

TABLE I. Identyfying the Pattern of a New Data Item

Cluster #	Object Frequency		Cluster size	Label
	Asthma	*Anemia*		
1	20	3	23	Asthma
2	12	2	14	Asthma
3	104	10	114	Asthma
4	14	1	15	Asthma
5	1	2	3< threshold	*Ineligible for voting*
6	65	6	71	Asthma
7	2	1	3< threshold	*Ineligible for voting*
8	15	7	22	Asthma
---	----	----	--	----
14	78	3	81	Asthma
15	30	2	32	Asthma
The new pattern is:				Asthma

V. Classification and Prediction Methodology

This proposed approach is suitable for Big Data problems, because it requires less complex mathematical calculations. Not like other conventional methods that depend on performing complex probabilistic operations, which are time consuming and requiring large-scale computational capabilities. The approach is an efficient technique in a sense that smarter decisions can be made much faster for quick responses. To simply describe the prediction methods, assume that a codeword *C* is generated based on diagnosing a patient *P* and answering the 23-bit questions of the Meta knowledge template. Thus, *C* represents the symptoms *S* that *P* has or has not. Our prediction approach works as follows: *C* goes in a process of generating and composing the clustering keys which was described above. Then, a pointer to *C* is placed in each one of the 15 clusters. Two different ways of prediction and classification are presented. First prediction approach works by identifying the type of the disease that *P* might develop based on the majority vote among the 15 clusters. For instance, if 10 out of the 15 clusters were labeled with "Asthma ", then *P* is most likely to develop asthma based on the current symptoms.

The second approach works by discovering relationships between symptoms based on other patients' metadata analysis. This relationship yields a prediction on the type of the *S* that *P* might develop in the future. For example, let *A* be the group of neighbor vectors. Vectors in *A* are placed with *C* in the same

cluster(s) and have no more than a certain Hamming distance, let's say 1. Then, we follow Fuzzy search method to retrieve codewords in *A*. After that, we sequence in *A* to find all the bit positions that mismatch with *C*, and place these mismatches in a group named *L*. As we described earlier, each bit represents the presence or absence of a property, which is in our example a symptom. Subsequently, we rank these symptoms in *L* based on their frequency. Therefore, our system can predict the *S* and their likelihood for a specific *P* based on the frequency of *S*. As a result, a symptom *S* with high frequency has high chance of occurrence in *P* and vice versa.

VI. Conclusion Remarks

Formulating meaningful groups of scattered data is beginning to gain popularity in many fields, including the medical field. In fact, it is one of the most demanding fields due to the enormous amounts of data generated on a daily basis. In this paper, we presented an efficient medical Big Data processing model based on Golay Code clustering algorithm. Our Big Data methodology works by clustering diverse information items in a data stream mode. The result is a group of clusters where the data items in each cluster are homogeneous. In contrast, the data points from different clusters are non-homogenous. This technique improves our ability to extract knowledge and insights from large and complex collections of medical data. Granting all the clustering methods that have been published before, the proposed method surpasses others as it improves the time complexity to O(n). We recommend the presented algorithm to be used as a tool in the medical field due to its competence in classification and prediction of risks, symptoms, and diseases.

References

[1] Centers for Medicare and Medicaid Services, Office of the Actuary (2012), National Health Expenditure Projections, 2012–2022. Washington, DC: CMS.

[2] *Barr, P. (2014, January). The Baby Boomer Challenge. Hospitals & Health Networks. Retrieved from* http://www.hhnmag.com/display/HHN-news-article.dhtml?dcrPath=/templatedata/HF_Common/NewsArticle/data/HHN/Magazine/2014/Jan/cover-story-baby-boomers

[3] Chronic Conditions: Making the Case for Ongoing Care. 2002. Johns Hopkins University. Baltimore.

[4] D. Greene, A. Tsymbal, N. Bolshakova and P. Cunningham, "Ensemble Clustering in Medical Diagnostics," Proc. 17th IEEE Symp. Computer-Based Medical Systems (CBMS ',04), pp. 576-581, 2004.

[5] F. Alsaby and S. Berkovich. Realization of clustering with Golay code transformations. Global Science and Technology Forum, J. on Computing, 2014.

[6] D. Liao, and S. Berkovich. "On clusterization of Big Data streams," Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications, article no.26. ACM press, New York 2012

[7] H. Yu. Golay Code Clustering Using Double Golay Encoding Technique, Doctoral Dissertation, GWU, October 2011.

[8] D. Davis, N. Chawla, N. Blumm,, N. Christakis, and N. Barabasi. Predicting individual disease risk based on medical history, Proceedings of the 17th ACM conference on Information and knowledge management, October 26-30, 2008, Napa Valley, California, USA [doi>10.1145/1458082.1458185]

[9] H. Yu, T. Jing, and S. Berkovich. Golay Code Clustering for Mobility Behavior Similarity Classification in Pocket Switched Networks, J. of Communication and Computer, USA, 2012.

[10] W. Pearson, and D. Lipman, Improved tools for biological sequence comparison. Proc Natl Acad Sci USA, 1988. 85: p. 2444 - 2448.

[11] S. Berkovich and E. El-Qawasmeh. "Reversing the Error-Correction Scheme for a Fault-Tolerant Indexing", The Computer Journal, Vol. 43, No. 1, pp. 54 – 64

[12] M. Yammahi, K. kowsari, C. Shen, and Simon Berkovich. An efficient technique for searching very large files with fuzzy criteria using the Pigeonhole Principle.

[13] A. Gruber1, S. Bernhart, I. Hofacker, and S. Washietl. Strategies for measuring evolutionary conservation of RNA secondary structures, BMC Bioinformatics, Volume 9, 2008

[14] A. Blum, and S. Chawla. Learning from Labeled and Unlabeled Data using Graph Mincuts, Proceedings of the Eighteenth International Conference on Machine Learning, p.19-26, June 28-July 01, 2001

[15] M. Charkhabi, T. Dhot, and S.Mojarad. Cluster Ensembles, Majority Vote, Voter Eligibility and Privileged Voters International Journal of Machine Learning and Computing, Vol. 4, No. 3, June 2014

[16] T. Yokoi, T, T. Yoshikawa and T. Furuhashi. Incremental learning to reduce the burden of machine learning for P300 speller, Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on , vol., no., pp.167,170, 20-24 Nov. 2012

[17] G. Kowalski, Document and Term Clustering, In Information retrieval architecture and algorithms, p. 173, New York: Springer,2011

[18] E. Berkovich. Method of and system for searching a data dictionary with fault tolerant indexing. US patent No.: US 7,168,025 B1 (2007)

[19] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[20] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[21] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[22] K. Elissa, "Title of paper if known," unpublished.

[23] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[24] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[25] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.