ISSN 2579-2784 (Print)
ISSN 2538-2788 (Online)

MATHEMATICAL PROBLEMS OF COMPUTER SCIENCE

LVII

Yerevan
2022

Հայաստանի Հանրապետության Գիտությունների ազգային ակադեմիայի Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ

Институт проблем информатики и автоматизации Национальной академии наук Республики Армения

Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ

Математические проблемы компьютерных наук

Mathematical Problems of Computer Science

LVII

ՀՐԱՏԱՐԱԿՎԱԾ Է ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏԻ ԿՈՂՄԻՑ
ОПУБЛИКОВАНО ИНСТИТУТОМ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА
PUBLISHED BY INSTITUTE FOR INFORMATICS AND AUTOMATION PROBLEMS OF NAS RA

ԵՐԵՎԱՆ 2022 YEREVAN

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ, LVII

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ պարբերականը հրատարակվում է տարեկան երկու անգամ ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի (ԻԱՊԻ) կողմից: Այն ընդգրկում է տեսական և կիրառական մաթեմատիկայի, ինֆորմատիկայի և հաշվողական տեխնիկայի ժամանակակից ուղղությունները:

Այն ընդգրկված է Բարձրագույն որակավորման հանձնաժողովի ընդունելի ամսագրերի ցանկում:

Տպագրվում է Խմբագրական խորհրդի 2022թ. մայիսի 27-ի N 22-05/1
նիստի որոշման հիման վրա

ISSN 2579-2784 (Print)
ISSN 2738-2788 (Online)

Математические проблемы компьютерных наук, LVII

Журнал **Математические проблемы компьютерных наук** издается два раза в год Институтом проблем информатики и автоматизации НАН РА. Он охватывает современные направления теоретической и прикладной математики, информатики и вычислительной техники.

Он включен в список допустимых журналов Высшей квалификационной комиссии.

Печатается на основании решения N 22-05/1 заседания
Редакционного совета от 27 мая 2022г.

ISSN 2579-2784 (Print)
ISSN 2738-2788 (Online)

Mathematical Problems of Computer Science, LVII

The periodical **Mathematical Problems of Computer Science** is published twice per year by the Institute for Informatics and Automation Problems of NAS RA. It covers modern directions of theoretical and applied mathematics, informatics and computer science.

It is included in the list of acceptable journals of the Higher Qualification Committee.

Printed on the basis of decision N 22-05/1 of session of the Editorial Council dated May 27, 2022.

ISSN 2579-2784 (Print)
ISSN 2738-2788 (Online)

CONTENTS

Mathematical Problems of Computer Science 57, 7–17, 2022.
doi: 10.51408/1963-0082

UDC 004.934

Emotion Classification of Voice Recordings Using Deep Learning

Narek T. Tumanyan

Weizmann Institute of Science, Rehovot, Israel
e-mail: narek.tumanyan@weizmann.ac.il

Abstract

In this work, we present methods for voice emotion classification using deep learning techniques. To processing audio signals, our method leverages spectral features of voice recordings, which are known to serve as powerful representations of temporal signals. To tackling the classification task, we consider two approaches to processing spectral features: as temporal signals and as spatial/2D signals. For each processing method, we use different neural network architectures that fit the approach. Classification results are analyzed and insights are presented.

Keywords: Voice sentiment detection, Mood recognition, Speech emotion recognition, Cepstral features.

Article info: Received 10 February 2022; received in revised form 17 April 2022; accepted 25 April 2022.

1. Introduction

The problem that is addressed in this work is the emotion classification from voice recording. Formally, given some representation X of voice recording data and a set of n emotion labels/classes $\{y_1, y_2, ..., y_n\}$, the aim is to come up with a classifier $F(X) = y_i$ that maps X to a label $y_i \in \{y_1, ..., y_n\}$. Practically, having such a classifier F can have a wide range of applications, such as recommendation systems of movies or music driven by users' mood, systems for tracking the emotional state and satisfaction of clients through time, security systems for preventing harmful actions based on emotion, and so on.

Previous attempts to tackle the voice emotion classification problem include SVM-based algorithms of classifying voice into 5 categories - angry, happy, neutral, sad, or excited [1], which also considers the facial expression of the speaker during speech as an additional signal. Glüge et al. [2] propose a Deep Neural Network Extreme Learning method with efficient performance on small datasets. Eskimez et al. [3] tackle the speech emotion recognition problem through an unsupervised approach, by which they come up with meaningful speech representations by learning the underlying structure of the data, which aids in solving the main task. Bertero et al. [4] introduce a Convolutional Neural Network (CNN)-based approach of 3-label ("angry", "happy", "sad") emotion recognition of speech, where they use the standard pulse-code modulation (PCM) temporal representation of the audio signal as input. Mirsamadi et al. [5] propose a 4-label ("angry", "happy", "sad", "neutral") speech emotion recognition model based on Long Short Term Memory Network (LSTM) architecture and local attention, and base their model on Mel-Frequency Cepstral Coefficients (MFCC), Fast Fourier Transform (FFT), fundamental frequency and zero-crossing rate features of the audio. In our setups, we experiment with both CNN-based and LSTM-based architectures and consider 8 emotional labels for classification, which are described in Section 2.

In this paper, we use cepstral features as representations of voice data, particularly, we utilize Mel-Frequency Cepstral Coefficients (MFCC) for representing the audio signal. We experiment with two views for processing MFCCs: processing them as sequential data in the time domain, and processing them as spatial data. For each of the approaches, we use the appropriate neural network architecture. Specifically, for processing MFCCs as temporal data, we utilize Long Short Term Memory Networks (LSTM), and for processing MFCC as spatial/2D data, we make use of Convolutional Neural Networks (CNN).

2. Datasets

In our setup, we consider 8 emotion labels for classification. The databases used in the paper are as follows: Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [6], Surrey Audio-Visual Expressed Emotion (SAVEE) [7] and Toronto Emotional Speech Set (TESS) [8]. Each item in the datasets is a recording of an actor that pronounces some statement with a certain expressed emotion. Voice recordings in the databases come in a .wav format, which describes the amplitude of air pressure oscillations in the temporal domain. Each voice recording has an emotion label attached to it. The RAVDESS database has 24 actors that pronounce 2 phrases: "Kids are talking by the door" and "Dogs are sitting by the door" with 2 intensities: Normal and High each repeated twice. Neutral emotion has no high intensity so it is only repeated twice. The emotion labels are: "neutral", "calm", "happy", "sad", "angry", "fearful", "disgust", "surprised". TESS dataset has 2 actors, young and old, and both of them are female. There are 2800 voices in total with each phrase being of the form "Say the word x, where x stands for some word. Recordings in the TESS dataset have the same labeled emotions as in RAVDESS, except for the calm label, which is absent in this dataset. SAVEE dataset has 4 English male actors with 480 voice recordings. 7 emotions are present, with the calm emotion missing. In total, there are 4720 samples. The distribution of samples and classes is summarised in Table 1 and in Table 2.

Table 1: Summary of datasets used.

Database	Num of Recordings	Num of Actors	Emotion Labels
RAVDESS	1440	24	8
SAVEE	480	4	7
TESS	2880	2	7

Table 2: Number of voice recordings per emotion label across all databases.

Neutral	Calm	Sad	Fear	Anger	Surprises	Happiness	Disgust
616	192	652	652	652	652	652	652

3. Method

3.1 Feature Extraction

To extract audio features from voice recordings, we use librosa library for python [9]. It handles most of the transformations done to voice recordings to get final features used for classification. The first step before extracting features is to resample voice recording files to obtain their time domain and amplitude representation. Voice recordings from our databases have different original sampling rates, which range from 22Khz to 48Khz. However, the content that we are trying to analyze from those recordings are the human voices themselves. Normally, the human voice ranges from low range frequencies 300Hz to higher ranges of 4 - 10Khz. This means that we can use lower sampling rates to resample our voice recording. We chose 22.05Khz sampling rate, which preserves all human voices in original audio recordings and also preserves some possible frequency deviations from the normal range, which can be caused by pronouncing high-frequency tones, e.g. fricatives. The result is a floating-point time series describing the amplitude of air pressure oscillations from a mean frequency of 0 at each time point. Thus, we obtain a time-domain representation of the signal. An example is illustrated below in Fig. 1.

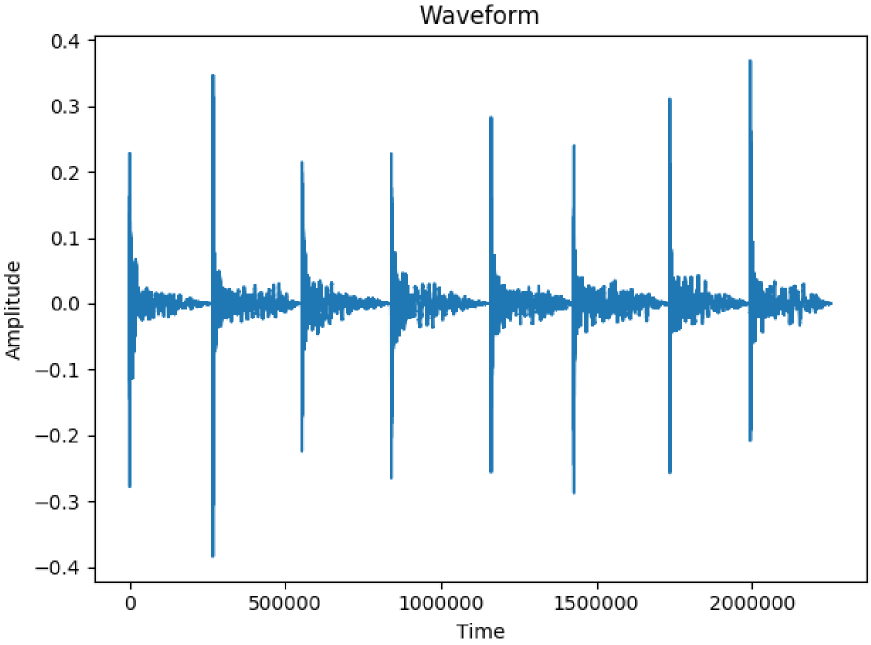


Fig. 1. Sample waveform representation of a voice recording signal.

Having the temporal signal representation of the voice signal, we then process it to obtain its spectral features, which serves as the main data representation for our models.

3.2 Spectral Features Extraction

Conceptually, given a temporal signal $x(t)$, we can represent it as a combination of periodic functions of varying frequencies [10]:

$$x(t) = \int_{-\infty}^{\infty} X(w)e^{jwt}dw,$$

where w is the frequency of the corresponding periodic function. Thus, having the coefficients $X(w)$ is equivalent to having the original signal $x(t)$, and we can use these coefficients as a representation of the temporal signal in the frequency space. To achieving such a representation, the Fourier Transform operation is used [10]. Since we are dealing with discrete data, the equivalent operation used is Discrete Fourier Transform (DFT), which converts discrete temporal signal $x[n]$ of length K to a representation of this signal in frequency space by obtaining the coefficients / intensities $X[k]$ for each frequency k [11]:

$$X[k] = \sum_{n=1}^{K} x[n]e^{-i2\pi kn/N};\ 1 \leq k \leq K.$$

In signal processing, frequency decomposition is often performed by dividing the signals into time intervals of specified window size and performing DFT on each windowed signal, thus coming up with frequency components in multiple time intervals. Such representation of a signal is called the Short-Time Fourier Transform (STFT) of a signal [10].

For audio signals, in some cases, more sophisticated representations of the signal based on STFT are necessary for higher efficiency. Mel-frequency cepstral coefficients, a.k.a. MFCCs, are features, which represent a given signal by cepstral energy coefficients at specific short intervals of time. The advantage of MFCC features is that they represent the signal in a way that is close to the signal perception by the human ear, which, is intuitively achieved by applying smaller window-sized cepstral filters on low frequencies on a signal and increasing the window size of the filters as the considered frequency increases. The reason behind such intuition is that the human ear perceives frequencies in lower ranges much better than in higher ones. Hence, higher resolution at lower ranged frequencies is used while computing MFCCs [12].

In its final form, the MFCC of a signal can be represented simply as a function $P_i(k)$, where the outputted value is the intensity of k-th cepstral coefficient in i-th temporal frame index.

An example of an extracted MFCC feature is demonstrated in Fig. 2

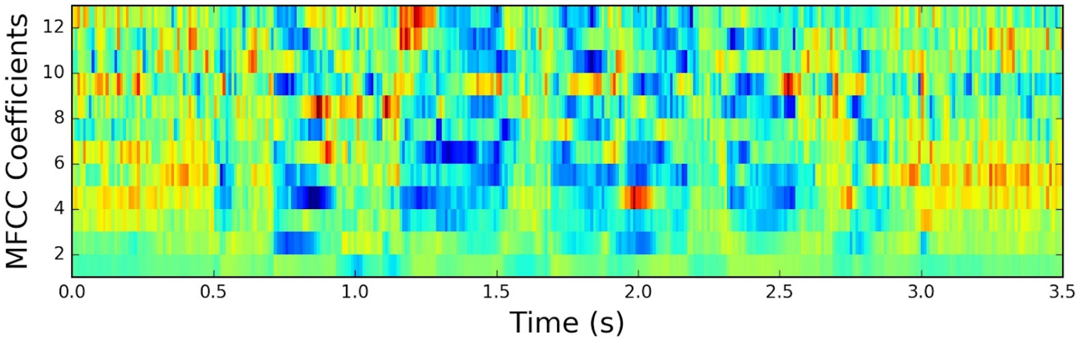


Fig. 2. Sample MFCC representation of a voice recording signal.

3.3 Architectures and Results

3.3.1 Long Short Term Memory Networks (LSTMs)

Considering the temporal nature of the data in hand, i.e., the voice recordings that are represented as magnitudes of air pressure (amplitude) across time, and the computed MFCC's that are a time series of energy coefficient values, it is sensible to use architectures that are by design intended for processing sequential data and have the appropriate inductive bias. One example of such architectures are Long Short Term Memory Networks (LSTM) [13], which are a variant of Recursive Neural Networks (RNN). The main idea behind LSTM is the usage of feedback connections for preventing the vanishing gradient problem. The architecture of LSTM used is summarized in Fig. 3.

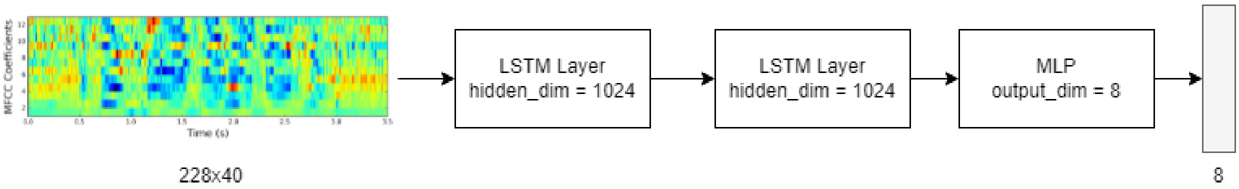


Fig. 3. The architecture of the trained LSTM model.

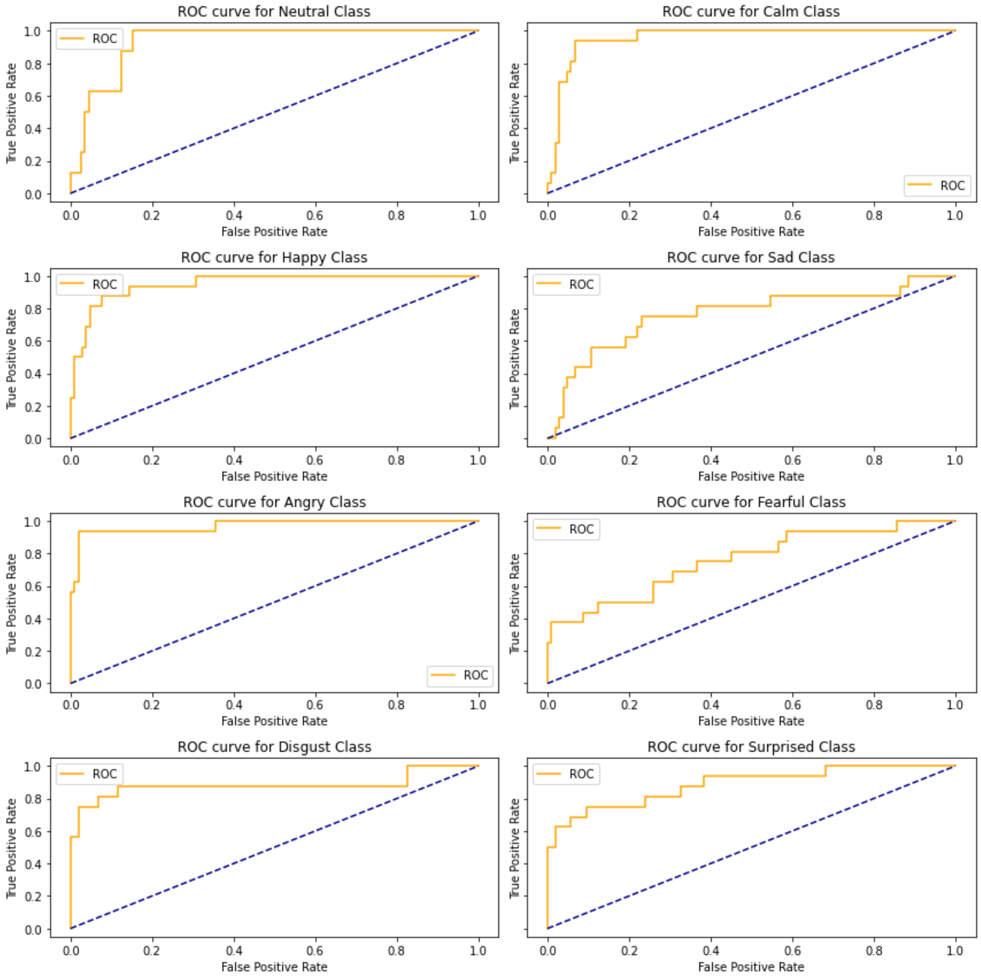


Fig. 4. ROC curves of the trained LSTM model. Each curve corresponds to an emotion label.

MFCC sequences are fed into an LSTM recurrent layer with a hidden dimension of size 1024. There are 2 LSTM layers stacked on top of each other, meaning that the outputs of the first layer are processed by the second one. This increases the perceptiveness of the network towards the features present in the sequence. Due to the dataset being small, we used dropout with high probability (p = 0.5) on the outputs of the first LSTM unit to prevent overfitting. The output of the last LSTM layer is then passed to a Multilayer perceptron (MLP), which outputs an 8-dimensional vector representing the logits of each emotion label.

The network was trained using only the RAVDESS dataset. The recordings of the 1st and 2nd actors (one male and one female) were used as a testing set, the rest of the recordings were used for training the network. Adam optimizer with learning rate of 0.0005 was used and the loss function to minimize was cross entropy loss given by:

$$l\left(\hat{y}_i\right) = \log\left(\frac{\exp\left(\hat{y}_i\right)}{\sum_j \exp\left(\hat{y}_j\right)}\right),$$

$$L\left(\widehat{y}_i\right) = -\sum_i y_i l\left(\widehat{y}_i\right),$$

where $\{\hat{y}_i\}$ are the estimated class labels, and $\{y_i\}$ are the ground-truth labels.

The classification results and comparison to the existing relevant method are demonstrated in Table 3. The Receiver Operating Characteristic curves (ROC curves) of the results are shown in Fig. 4.

3.3.2 Convolutional Neural Networks (CNNs)

As stated in subsection 3.2, the MFCC of a recording can be observed as a 2D feature map of a signal, with one dimension being the temporal dimension and the other being the cepstral coefficient dimension. Thus, a possible approach to working with MFCC's is processing them as spatial signals. Convolutional Neural Networks (CNN) are one of the most prominent architectures used for processing spatial data due to their shift equivariance, their inductive bias in searching for local patterns, and many other inherent benefits.

Thus, we consider solving the voice emotion classification task by training a CNN on extracted MFCC data. For MFCC calculation, the window size of 4096 and the overlap of between subsequent windows were chosen. Decreasing the window size by half degrades the performance of the network. On average, these settings produced better results. 4096 for a window size is good because it allows computing the FFT of length 4096 on that window to capture frequency spectrum of up to 4Khz. This means that the majority of human speech in those recordings is captured in each window. After calculating MFCCs for every recording and padding sequences with less length than the longest sequence, we obtain input matrices to our network of size (40 x 160) where at each sequence point we have 40 MFCCs.

The architecture of the CNN used is depicted in Fig. 5, and the method is summarized as follows:

There are 3 convolutional layers in the network followed by average pooling layers of size (2x2). The last layer is a fully connected layer that maps output of convolutional layers to an 8 length vector. Log softmax activation is applied to use cross entropy loss. Each layer has 32 kernels of parameters. The first layer has kernels of size (10x3), and it is deliberately chosen to be narrow and heighty to capture features from change of MFCCs through the sequence. Between layers, leaky rectified linear unit (ReLU) activation function given as h(x)=max(x, 0)+0.01*min(0, x)is used both to enable fast training and to prevent neurons from dying. Leaky ReLU adds a small slope to non activated neurons thus preventing them from becoming 0 and not contributing to backpropagation in later epochs [14]. Since our dataset is very small, we used dropout with high probability (p = 0.5) as well as L2 regularization to prevent overfitting, which penalizes the sum of squares of the weights of the model.

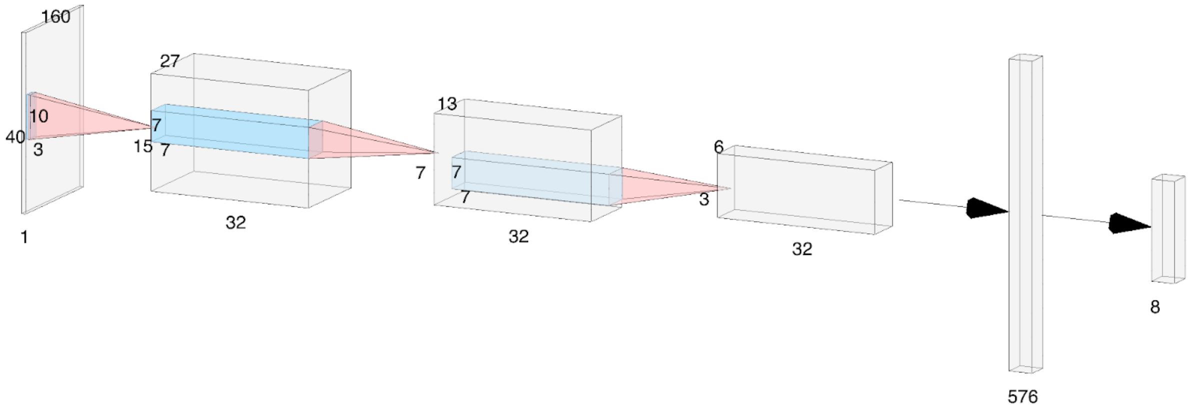


Fig. 5. The architecture of the trained CNN model.

All recordings of the 1st and 2nd actors, one male and one female from the RAVDESS database were used for testing, which the neural network was not trained on. All remaining recordings were used for training. We used Adam optimizer with a learning rate of 0.00005 and L2 regularization with decay of 10^{-4}. The final loss function becomes:

$$l(\widehat{y}_i) = \log\left(\frac{\exp(\hat{y}_i)}{\sum_j \exp(\hat{y_j})}\right),$$

$$L(\widehat{y}_i) = -\sum_i y_i l(\widehat{y}_i) + \lambda \sum_{w \in W} w^2,$$

where W is the set of all trainable weights of the CNN.

The classification results and comparison to the existing related method are summarized in Table 3. Average ROC Area Under Curve (AUC) for all classes was 0.927. ROC curves for all classes are demonstrated in Fig. 6.

Table 3: Classification results.

Architecture	Train Accuracy	Test Accuracy	Mirsamadi et al. [5] Test Accuracy	Bertero et al. [4] Test Accuracy
LSTM	93.58%	65%	63.5%	-
CNN	96%	67.5%	-	66.1%

As it can be observed, the network captures some emotions more easily than others. For instance, Neutral, Calm, Angry and Surprise were captured better than the rest. ROC-AUC metric also suggests that the model learned meaningful representations for the task.

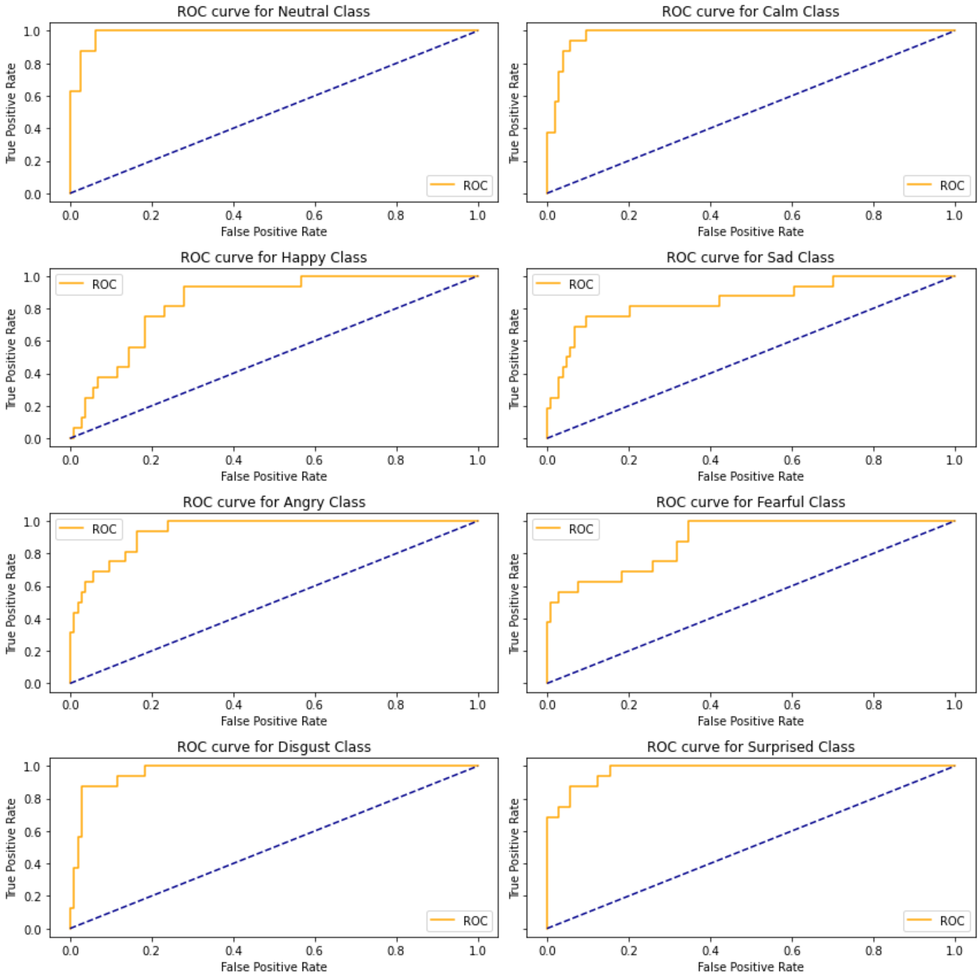


Fig. 6. ROC curves of the trained CNN model. Each curve corresponds to an emotion label.

Overall, the results show that the models managed to learn meaningful representations from the training procedure. In Table 3, we compare our results to the LSTM-based method of Mirsamadi et al. [5], which was trained and tested on the IEMOCAP benchmark [16] with a 4-label ("angry", "happy", "sad", "neutral") classification setting, as well as to the CNN-based method of Bertero et al. [4], which was trained and tested on the TED-LIUM benchmark [15] with a 3-label ("angry", "happy", "sad") classification setting. As it can be observed, our method gains superior results on our 8-label classification setting. In contrast to the 2 methods, we leverage only the MFCC representation of the signal, which highlights the efficiency of the MFCC representation and its usage with deep learning methods for the task.

4. Discussion and Conclusion

This paper proposes deep learning approaches for the voice emotion classification problem. Particularly, CNN and LSTM architectures were trained on MFCC features of voice recordings, depending on processing MFCCs either as a spatial signal or as a sequential signal. The results indicate that the networks have learned meaningful representations from the training data. A possible future direction for improving the classification performance of the proposed models could be adding augmentations to the audio data. The recent advancements in using transformers [17] for multi-modal representation learning [18] and the expressiveness of the resulting feature space can also be a promising direction for solving the speech emotion recognition task.

References

[1] E. Mower, M. J. Mataric and S. Narayanan,“A framework for automatic human emotion classification using emotion profiles”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1057–1070, 2010.

[2] S. Glüge, R. Böck and T. Ott, “Emotion recognition from speech using representation learning in extreme learning machines”, *Proceedings of the 9th International Joint Conference on Computational Intelligence*, Funchal, Portugal, pp. 179–185, 2017.

[3] S.E. Eskimez, Z. Duan and W. Heinzelman, “Unsupervised learning approach to feature analysis for automatic speech emotion recognition”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada., pp. 5099–5103, 2018.

[4] D. Bertero and P. Fung, “A first look into a convolutional neural network for speech emotion detection”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA., pp. 5115–5119, 2017.

[5] S.Mirsamadi, E. Barsoum and C. Zhang, “Automatic speech emotion recognition using recurrent neural networks with local attention”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA., pp. 2227–2231, 2017.

[6] S. R. Livingstone and F. A. Russo, “The ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English”, *PLoS ONE* , vol. 13, no. 5, 2018.

[7] P. Jackson and S. Haq, “Surrey audio-visual expressed emotion (savee) database”, University of Surrey: Guildford, UK. 2014.

[8] M. K. Pichora-Fuller and K. Dupuis, “Toronto emotional speech set (TESS)”, Scholars Portal Dataverse, 2020.

[9] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thom, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg and S. Seyfarth, librosa/librosa: 0.9.0 (0.9.0). Zenodo, 2022, https://doi.org/10.5281/zenodo.5996429

[10] K. Gröchenig, *Foundations of Time-Frequency Analysis*, First Edition. Birkhuser, Boston, MA, 2001.

[11] A. Kulkarni, M. F. Qureshi, and M. Jha, “Discrete fourier transform: approach to signal processing”, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 03, pp. 12341–12348, 2014.

[12] M. Sahidullah and G. Saha, “Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition”, *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.

[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural Computation,* vol. 9, no. 8, pp. 1735–1780, 1997.

[14] B. Xu, N. Wang, T. Chen and M. Li, “Empirical evaluation of rectified activations in convolutional network”, *CoRR* , vol. abs/1505.00853, 2015.

[15] A. Rousseau and P. Deleglise, “Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks”, *International Conference on Language Resources and Evaluation*, Reykjavik, Iceland, pp. 3935-3939, 2014.

[16] C. Busso, M. Bulut, Chi-Chun Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee and S. S. Narayanan, “Iemocap: Interactive emotional dyadic motion capture database”, *Language Resources and Evaluation* , vol. 42, no. 4, pp. 335–359, 2008.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, “Attention is all you need”, *CoRR* , vol. abs/1505.00853, 2017.

[18] H. Akbari, L. Yuan, R. Qian, W.H. Chuang, S.-Fu Chang, Y. Cui and B. Gong, “VATT: Transformers for multimodal self-supervised learning from raw video, audio and text,” *Advances in Neural Information Processing Systems*, 2021.

Խորը ուսուցման վրա հիմնված ձայնագրությունների էմոցիաների դասակարգման մեթոդներ

Նարեկ Տ. Թումանյան

Վեյցմանի գիտությունների համալսարան, Ռեխովոտ, Իսրայել
e-mail: narek.tumanyan@weizmann.ac.il

Ամփոփում

Տվյալ հոդվածում ներկայացվում են խորը ուսուցման վրա հիմնված ձայնագրությունների դասակարգման մեթոդներ։ Աուդիո ազդանշանները մշակելու համար օգտագործվում են ձայնագրությունների հաճախական տվյալներ, որոնք հայտնի են ժամանակային ազդանշանների արդյունավետ ներկայացմամբ։ Դասակարգման խնդիրը լուծելու համար հոդվածում հաշվի են առնվում հաճախական հատկանիշների մշակման երկու մոտեցում՝ որպես ժամանակային ազդանշանների մշակման մոտեցում և որպես տարածական ազդանշանների մշակման մոտեցում։ Յուրաքանչյուր մոտեցման համար կիրառվում են համապատասխան արհեստական ցանցերի մոդելներ։ Ներկայացվում է դասակարգման արդյունքների վերլուծություն, կատարվում են եզրակացություններ։

Բանալի բառեր՝ ձայնի տրամադրության ճանաչում, խոսքի էմոցիայի դասակարգում, հաճախական հատկանիշներ։

Классификация эмоций в голосе с использованием глубокого обучения

Нарек Т. Туманян

Институт Вейцмана, Реховот, Израиль
e-mail: narek.tumanyan@weizmann.ac.il

Аннотация

В этой статье мы представляем методы классификации эмоций в голосе с использованием методов глубокого обучения. Для обработки аудиосигналов, данный метод использует частотные признаки извлеченные из голосовых записей, которые, как известно, служат мощным представлением временных сигналов. Для решения задачи классификации, в данной работе рассматриваются два подхода обработки частотных признаков: как временные сигналы и как пространственные/2D-сигналы. Для каждого из подходов мы используем подходящие архитектуры нейронных сетей. Были проанализированы результаты классификации и представлены выводы.

Ключевые слова: определение настроения по голосу, распознавание настроения, классификации эмоций в голосе, частотные признаки.

Mathematical Problems of Computer Science 57, 18–29, 2022.
doi: 10.51408/1963-0083

UDC 004.932

A New Image Decolorization Evaluation Quality Metric

Hrach Y. Ayunts[1] and Sos S. Agaian[2]

[1]Yerevan State University, Yerevan, Armenia
[2]College of Staten Island (CSI), City University of New York, New York, USA
e-mail: hrach.ayunc@gmail.com, sos.agaian@csi.cuny.edu

Abstract

Image decolorization, the process of color-to-gray conversion, plays a crucial role in single-channel processing, computer vision, digital printing, and monochrome visualization. This process induces new artifacts, the impact of which on visual quality has to be identified. While visual quality assessment has been the subject of many studies, there are still some open questions regarding new color-to-gray conversion quality metrics. For example, computer simulations show that the commonly used grayscale conversion quality metrics such as CCPR, CCFR, and E-score depend on parameters and may pick different best decolorization methods by changing the parameters.

This paper proposes a new quality metric to evaluate image decolorization methods. It uses the human visual properties information and regression method. Experimental results also show (i) strong correlations between the presented image decolorization quality metric and the Mean Opinion Score (MOS), (ii) more robust than the existing quality metrics, and (iii) help to choose the best state-of-the-art decolorization methods using the presented metric and existing quality metrics.

Keywords: Color-to-gray conversion, Decolorization, Grayscale, Regression, Quality metric.

Article info: Received 18 April 2022; accepted 20 May 2022.

Acknowledgement: This work was supported by the professors of the Department of Informatics and Applied Mathematics of Yerevan State University (S. Sargsyan, E. Danoyan, Yu. Hakopian) and the Armenian Engineers and Scientists of America.

The authors would like to thank the Picsart employees for participating in the user study. The authors would also like to thank Sowmya et al. [13] and Lu et al. [4] for sharing source codes and experiment materials.

1. Introduction

Image decolorization aims to convert a color image into a grayscale image to improve the image's visual appearance or provide a "better" gray-level representation for the future automated image. The overall purpose of image decolorization is to preserve the visible color contrast, which usually suffers from information loss. It plays an essential role in single-channel image processing (analysis, detection, segmentation, and recognition), computer vision, monochrome printing, e-ink display, etc. [1]. The analysis of the existing image decolorization techniques shows the common problems that need to be solved because such methods introduce certain artifacts. It isn't easy to evaluate decolorization methods and select their optimal parameters. There are various quality metrics for color images [2, 3]. Thus, these types of metrics are not suitable for the evaluation of color-to-gray conversion. There is also no efficient measure that can be served as a building criterion for image decolorization.

Practically, all quality metrics for image decolorization are based on the fact that the human visual system cannot perceive color differences smaller than a certain threshold [4, 5, 6]. Extensive computer simulations show that (i) commonly used grayscale conversion quality metrics such as CCPR, CCFR and E-score depend on the color difference parameter, and (ii) by changing the parameter, we pick a different decolorization method. Thus, one needs to develop a new robust threshold-independent quality metric that does not require a reference image.

This paper makes several key contributions:

1. Propose a non-parametric, robust, monotonic, and non-reference quality metric for image decolorization.
2. Present extensive computer simulation results.
3. Present qualitative and perceptual evaluation of state-of-the-art decolorization methods.

The structure of this paper is organized as follows. Section 2. discusses the existing image decolorization methods and quality metrics. Section 3. presents a new non-parametric quality metric. Section 4. provides the results of extensive computer simulation. Section 5. validates the new metric using preference scores from the user study. Finally, Section 6. concludes the work.

2. Background

This section presents the existing color-to-gray conversion methods and quality metrics. Traditional color-to-gray conversion methods usually use a linear combination of R (red), G(green), B (blue) channels of a color image. It is based on the theory of T. Young (1802), which states that any color can be created by combining three primary colors: R, G, and B. $Gray = aR + bG + cB$ [7], were a, b, c coefficients are calculated as

(i) Lightness method: $Gray = \frac{max(R,G,B)+min(R,G,B)}{2}$

(ii) Average method: $a = b = c = 1/3$, or $Gray = \frac{R+G+B}{3}$

(iii) Luminosity method: $a = 0.21, b = 0.72, c = 0.07$, or $Gray = 0.21R + 0.72G + 0.07B$.

These are the most popular and straightforward conversions used in electronic displays, printers, computer vision, image processing, and many other algorithms as a preprocessing step.

However, Fig. 1 shows that the grayscale conversion suffers from information loss (many details didn't preserve, and the color contrast was lost in the grayscale images). It is natural to ask.

(a) Can we have a better decolorization algorithm?

(b) How to quantitatively evaluate the performance of different methods or choose the parameters such as a, b, and c?

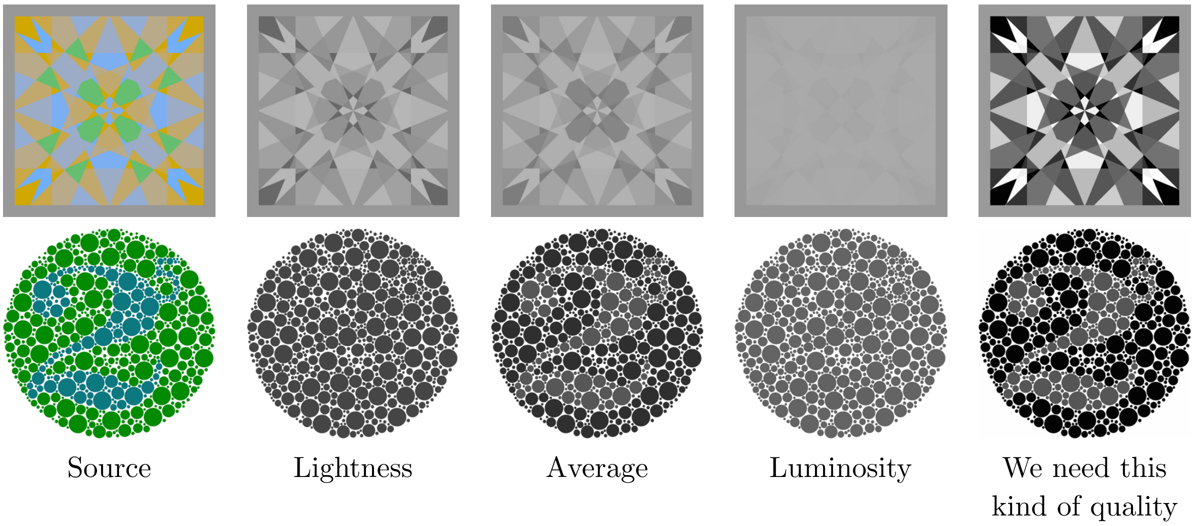


Fig. 1. Comparison of traditional grayscale conversion methods. Decolorized images can lose can lose the contrast and become hardly visible.

(c) How do you improve the quality of a color image using decolorized images?

More advanced decolorization methods use the values of other pixels to specify color orders to preserve the color contrast. Local methods rely on local chrominance edges to enhance the contrast [8, 9]. Most recent notable decolorization methods are based on the parametric decolorization model and its modification [5, 4, 10].

Parametric Decolorization Model(PDM). The basic idea here is to convert a color image into gray using a combination of a polynomial of R, G, and B components: $\{R, G, B, RG, RB, GB, R^2, G^2, B^2\}$. It generalizes commonly used linear and nonlinear color-to-gray conversion/mapping systems. More details on this method one can find in [5].

There are also neural network solutions to this problem [11].

Decolorization needs quantitative evaluation to understand the performance of different methods.

Exiting decolorization quality metrics. The most commonly used decolorization quality metrics are based on the fact that the human visual system cannot perceive color difference δ smaller than a certain threshold. For example, the Color Contrast Preserving Ratio (CCPR) (suggested by Lu et al. [4]), defined as

$$\text{CCPR} = \frac{\#\{(x,y)|(x,y) \in \Omega, |g_x - g_y| \geq \tau\}}{||\Omega||}, \tag{1}$$

where Ω is the set of all pixel pairs with $\delta_{x,y} \geq \tau$, and g_x, is the value of the x pixel after decolorization.

CCPR shows the percentage of distinctive pixel pairs after the conversion, but it does not necessarily indicate if the grayscale image was “distorted” after conversion. To complement CCPR, Lu et al. [4] suggested Color Content Fidelity Ratio (CCFR). It is defined as

$$\text{CCFR} = 1 - \frac{\#\{(x,y)|(x,y) \in \Theta, \delta_{x,y} \leq \tau\}}{||\Theta||}, \tag{2}$$

where Θ is the set of all pixel pairs with $|g_x - g_y| > \tau$. This metric shows how much the converted image has changed in terms of structure.

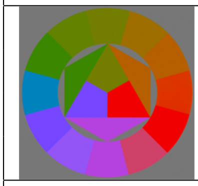
Finally, the combination of CCPR and CCFR, E-score [4], is defined as

$$\text{E-score} = \frac{2 \cdot \text{CCPR} \cdot \text{CCFR}}{\text{CCPR} + \text{CCFR}}. \tag{3}$$

3. Proposed Quality Metric

This section shows the shortcomings of the existing decolorization metrics and suggests a better quality metric for quantitative evaluations.

Table 1: E-score metric for some threshold values for different decolorization methods

Image	Method	$\tau = 3$	$\tau = 5$	$\tau = 7$	$\tau = 9$	$\tau = 15$	$\tau = 25$
	PDM	**0.9934**	0.9776	0.9759	0.9737	**0.9644**	**0.9551**
	LUM	0.9613	0.9174	0.8526	0.7990	0.5956	0.3997
	SPD	0.9862	0.9769	0.9751	0.9713	0.9475	0.9192
	SVD	0.9896	**0.9821**	**0.9765**	**0.9744**	0.9279	0.8514
	PDM	0.9726	0.9502	0.9272	**0.9035**	**0.8298**	**0.6647**
	LUM	0.9646	0.9356	0.9046	0.8704	0.7447	0.4993
	SPD	**0.9777**	**0.9550**	0.9275	0.8956	0.7823	0.5662
	SVD	0.9745	0.9525	**0.9279**	0.9003	0.7987	0.5965

The commonly used grayscale conversion quality metrics such as CCPR, CCFR, and E-score depend on the color difference parameter τ. Computer simulations show that by changing the parameter τ, we pick a different decolorization method.

To verify this statement, we compare different decolorization methods on a couple of images from Ĉadíks dataset [12].

We calculate the E-score quality metric for different values of threshold. We use three state-of-the-art methods (Lu et al. [5], Sowmya et al. [13], Liu et al. [10]) and the Luminosity method for comparison. The results are listed in Table 1. Obviously, the best method differs depending on the threshold value. For example, we can pick three different best methods by changing parameter τ in the case of the second image. The visual results of decolorization on these images are shown in Fig. 4..

In the previous work, the quantitative evaluation of color-to-gray conversion was performed using E-score for fixed values of threshold [4, 5]) or the average of several threshold values [10]. Therefore, there is a need for more independent metrics to investigate the conversion process for each image.

We introduce a new quality metric called Threshold-Independent Slope (TIS), which shows the decreasing speed of the E-score as the threshold value grows. We calculate the E-score metric for different τ values ($\tau = 1, 2, ..15$) and choose the slope of the linear regression of this data as a new metric. The main advantage of the new metric is that it is not dependent on the τ parameter.

Linear Regression can be solved using several linear models. A simple linear model function is defined as

$$y = \alpha + \beta x, \tag{4}$$

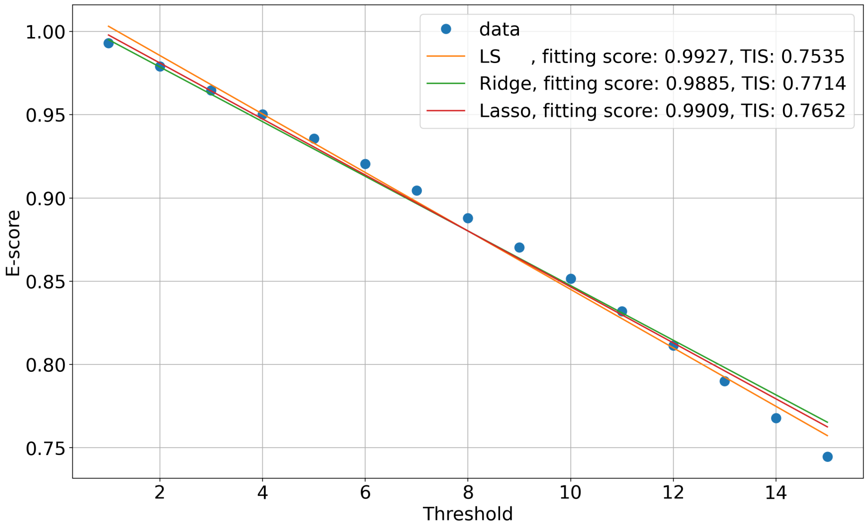


Fig. 2. Simple linear function estimation using the Least Squares method, Ridge regression, and Lasso regression.

which describes a line with a slope β and y-intercept α. One of the easiest ways to estimate the slope is to use the **Least Squares** method:

$$\hat{\beta}_{ls} = \arg\min_{\beta} ||y - \beta x||_2^2. \tag{5}$$

Another method for coefficient estimation of (4) is **Ridge regression** [14]. It is most suitable when data contains a higher number of predictor variables than the number of observations. The ridge regression estimator solves the regression problem using l_2 penalized least squares:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} ||y - \beta x||_2^2 + \lambda||\beta||_2^2, \tag{6}$$

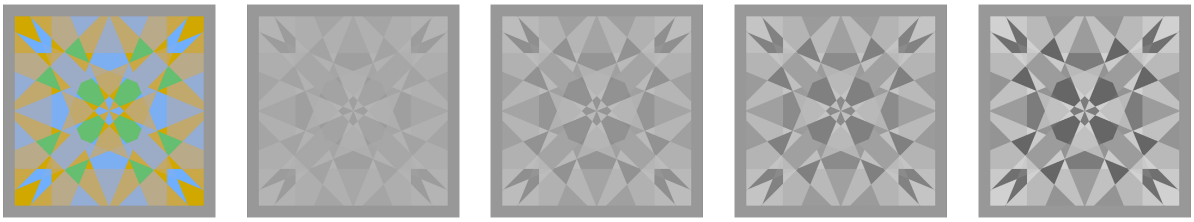
where $\lambda > 0$ is a tuning parameter that controls the strength of the penalty term. Similar to ridge regression, **Lasso regression** can be used for slope estimation [15]. The lasso estimator uses l_1 penalized least squares for solving the following optimization problem with λ tuning parameter:

$$\hat{\beta}_{lasso} = \arg\min_{\beta} ||y - \beta x||_2^2 + \lambda||\beta||_1. \tag{7}$$

Fig. 2 compares these three regression models on a sample image from Ĉadíks dataset [12]. Each of these models is used to calculate the TIS metric. To find the best model for our case, we calculate the fitting scores of each model on every image from the dataset. The Least Squares method has the best average fitting score: thus, we use it for further evaluations. Therefore, our TIS metric is defined as

$$\mathrm{TIS} = \max(1 - |\alpha\beta|, 0), \tag{8}$$

where α and β are coefficients of a simple linear function (4) estimated with the Least Squares method (5). TIS ranges in $[0, 1]$, and higher values mean a lower decreasing speed of the E-score metric when the threshold is increased.



TIS	0.53565	0.72862	0.82799	0.87882

Fig. 3. The TIS metric grows with a contrast and visibility increase.

Fig. 3. shows the decolorization result on a sample image with four different levels of visibility. The value of our TIS metric grows with better visibility and contrast in the result. Therefore, the TIS is also a monotonic metric.

4. Computer Simulation

This section evaluates four decolorization methods using our TIS metric and the existing quality metrics. We also show the usefulness of our metric in picking the best parameters for grayscale conversion.

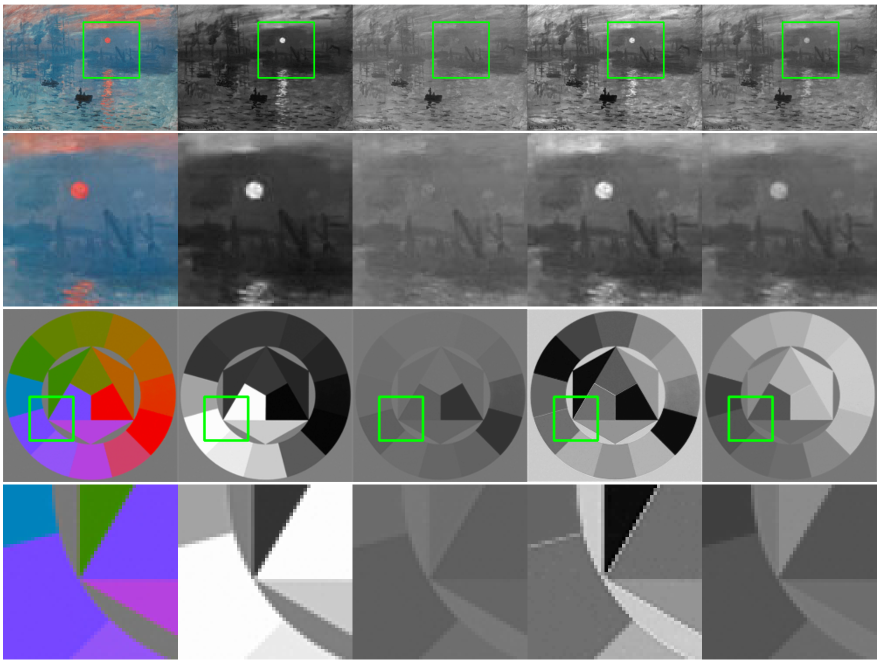


Fig. 4. Visual results of different decolorization algorithms (from left to right: source image, PDM, LUM, SPD, SVD)

Evaluation of decolorization methods. We chose one traditional conversion method: the Luminosity method (denoted as LUM in tables) is the most popular conversion used in many image processing algorithms and electronic devices. In many cases, it fails to preserve the contrast because the conversion considers only current pixel information. We also chose three state-of-the-art contrast preserving decolorization methods for evaluation. These methods are suggested by Lu et al. [5], Liu et al. [10], and Sowmya et al. [13] (we use PDM, SPD, and SVD acronyms in the tables, respectively). These methods consider global pixel information and color differences in the image for better conversion.

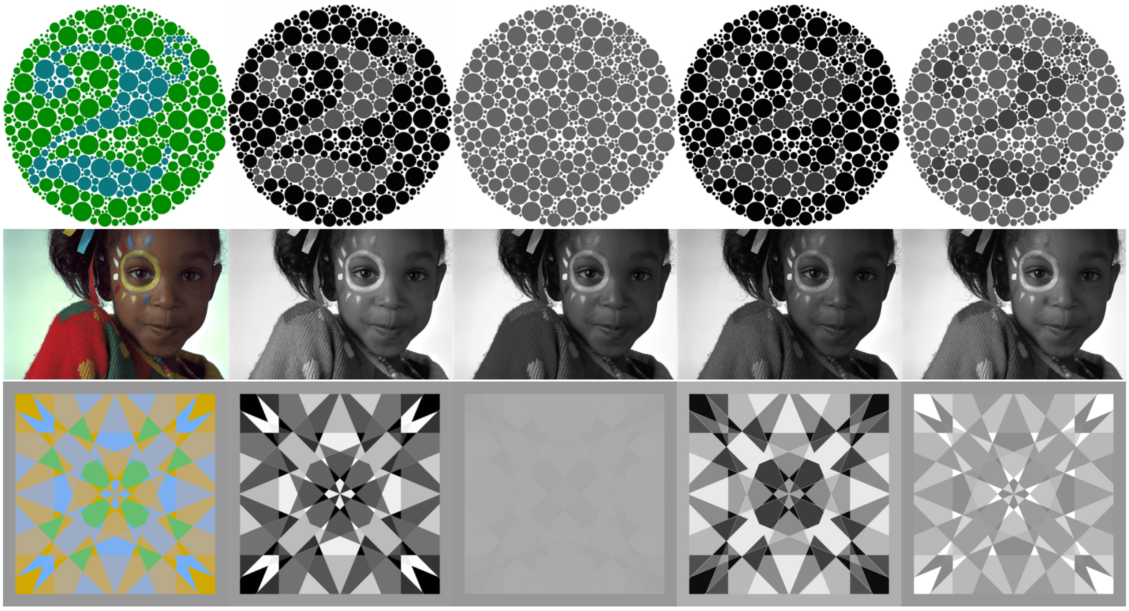


Fig. 5. Visual results of different decolorization algorithms
(from left to right: source image, PDM, LUM, SPD, SVD)

Figs. 4 and 5 show the visual results of four decolorization methods on several images. The simple Luminosity method usually fails to preserve the color contrast, while the other three methods produce better visual outputs.

We use Ĉadík's dataset [12] for performance evaluation. It contains 24 PNG images and mainly consists of synthetically generated images and some colorful real-life photos. Most of these images are challenging for traditional color-to-gray conversion methods. That's why Ĉadík's dataset is the most popular in this field and can be beneficial for the evaluation of decolorization methods.

Table 2: Average TIS and E-score for different thresholds on Ĉadík's dataset.

Method	$\tau = 3$	$\tau = 5$	$\tau = 7$	$\tau = 9$	$\tau = 15$	$\tau = 25$	TIS
PDM	0.98222	0.97009	0.95866	0.94697	**0.90971**	**0.84409**	**0.91635**
LUM	0.96340	0.93755	0.91399	0.89556	0.83167	0.71761	0.84992
SPD	**0.98241**	**0.97060**	**0.95922**	**0.94810**	0.90966	0.83835	0.91560
SVD	0.98045	0.96651	0.95334	0.94040	0.89324	0.81638	0.90121

The quantitative evaluation of four decolorization methods using the E-score metric for different thresholds and our new TIS metric on Ĉadík's dataset are presented in Table 2. It presents the performance of each metric (average value) of all images from Ĉadík's dataset. It also shows that the presented TIS metric is more stable and picks only the best method for this dataset. So it can be helpful in both individual and large-scale evaluations of the grayscale conversion methods.

Picking the best parameter for the simple grayscale conversion. Image decolorization quality metrics can not only be useful in method evaluation, but they can also help pick the best parameters for an algorithm. For example, in simple grayscale conversion, coefficients can be changed to get a "better" conversion.

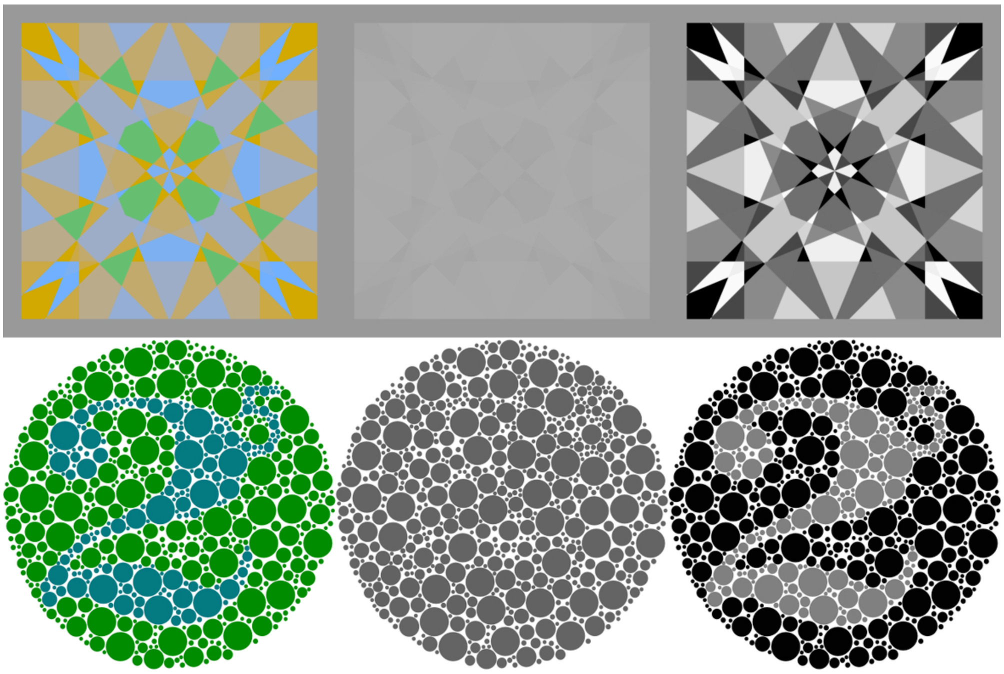


Fig. 6. Comparison of the "best" linear conversion with the luminosity method (from left to right: source image, luminosity, the best conversion).

We pick the best parameters of the linear grayscale conversion by maximizing the value of the quality metric for each image individually. Fig. 6 shows the results corresponding to the highest values of the TIS for two images ($a = 0.02, b = 0, c = 0.98$ for the first image, and $a = 0, b = 0.06, c = 0.94$ for the second one).

5. Perceptual Validation

This section validates our TIS metric using the preference scores.

We invited 20 users to participate in a survey to show the effectiveness and importance of our metric. After a small introduction to decolorization, they were asked to rate the color-to-gray conversion for ten random images from the Ĉadík's dataset on a scale of one to three. To facilitate the scoring process, we use the three-scale modification of the Mean Opinion Score (MOS). One means the conversion is bad, and it failed to preserve the contrast.

Score two corresponds to mediocre conversion. Finally, three is for the best conversion with contrast preservation and the most visually pleasing result.

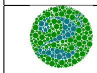
To validate our metric, we use the Kendall rank correlation coefficient [16]. It is defined as

$$R = \frac{\#\{\text{concordant pair}\} - \#\{\text{disconcordant pair}\}}{\frac{1}{2}n(n-1)}, \tag{9}$$

where $n = 4$ denotes the number of methods. Let s_i be the score for the result produced by the ith method, and p_i be the preference score for the same result. If two pairs (s_i, s_j) and (p_i, p_j) are with the same order (i.e., $(s_i - s_j)(p_i - p_j) > 0$), the pair (i, j) is concordant. Otherwise, it is discordant. R ranges in $[-1, 1]$. We get $R > 0$ if the two rankings agree with each other and $R < 0$ otherwise.

We calculate the Kendall rank correlation coefficient (R) for the existing metrics and our TIS metric. The TIS metric has a high correlation with the user preference scores and can easily replace the existing quality metrics for quantitative evaluations of decolorization. The ranks for several images presented in the survey are listed in Table 3.

Table 3: Kendall rank correlation coefficient for the E-score metric with different thresholds, and our TIS metric

Image	$\tau = 5$	$\tau = 7$	$\tau = 9$	$\tau = 15$	$\tau = 25$	TIS
	0.333	0.333	0.333	0.667	0.667	0.667
	0.333	0.333	0.333	1	1	1
	0.667	0.333	0.333	0.333	0	0.333

6. Conclusion

This paper proposes a new TIS image quality metric for accurately evaluating image decolorization methods. The TIS quality metric is a blind, robust, monotonic, non-parametric metric and correlates with subjective preference scores. The quantitative and qualitative computer simulations on the Ĉadík's dataset demonstrate that the proposed metric outperforms the current state-of-the-art metrics. The TIS metric is also helpful in picking the best parameters of the grayscale algorithm.

Our future work will extend the proposed work to other types of distortion, generate new decolorization methods, and evaluate them on other databases.

References

[1] C. Saravanan, "Color image to grayscale image conversion", *Second Inter. Conference on Computer Engineering and Applications, IEEE*, vol. 2, pp. 196–199, March 2010.

[2] K. Panetta, C. Gao, and S. Agaian, "No reference color image contrast and quality measures", *IEEE trans. on Consumer Electronics*, vol. 59, no. 3, pp. 643–651, 2013.

[3] K. Panetta, C. Gao, and S. Agaian, “Human-visual-system-inspired underwater image quality measures”, *IEEE Journal of Oceanic Engineering*, vol. 41, no. 3, pp. 541–551, 2015.

[4] C. Lu, L. Xu, and J. Jia, “Contrast preserving decolorization with perception-based quality metrics”, *Inter. Journal of computer vision*, vol. 110, no. 2, pp. 222–239, 2014.

[5] C. Lu, L. Xu, and J. Jia, “Contrast preserving decolorization”, *Proc. of IEEE inter. conference on computational photography (ICCP)*, pp. 1–7, 2012.

[6] S. Agaian, “Visual Morphology”, *Proc. of SPIE, Nonlinear Image Processing X*, San Jose CA, vol. 3646, pp. 139–150, 1999.

[7] J. Cook, (2009) Three algorithms for converting color to grayscale. [Online]. Available: https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/

[8] R. Bala, and R. Eschbach, “Spatial color-to-grayscale transform preserving chrominance edge information”, *Color and Imaging Conference, Society for Imaging Science and Technology*, vol. 2004, no. 1, pp. 82–86, 2004.

[9] L. Neumann, M. Ĉadík, and A. Nemcsics, “An efficient perception-based adaptive color to gray transformation”, *Proc. of the Third Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging* pp. 73–80, 2007.

[10] Q. Liu, P. Liu, Y. Wang, and H. Leung, “Semiparametric decolorization with Laplacian-based perceptual quality metric”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 1856–1868, 2016.

[11] Q. Liu, and H. Leung, “Variable augmented neural network for decolorization and multi-exposure fusion”, *Information Fusion*, vol. 46, pp. 114–127, 2019.

[12] M. Ĉadík, “Perceptual Evaluation of Color-to-Grayscale Image Conversions”, *Computer Graphics Forum*, vol. 27, no. 7, Wiley Online Library, pp. 1745–54, 2008.

[13] V. Sowmya, D. Govind, and K. Soman, “Significance of incorporating chrominance information for effective color-to-grayscale image conversion”, *Signal, Image and Video Processing*, vol. 11, no. 1, pp. 129–136, 2017.

[14] A. Hoerl, and R. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems”, *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[15] R. Tibshirani, “Regression shrinkage and selection via the lasso”, *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[16] H. Abdi, “The Kendall rank correlation coefficient”, *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pp. 508–510, 2007.

Պատկերի գունազրկման որակի գնահատման նոր չափորոշիչ

Հրաչ Յու. Այունց[1] և Սոս Ս. Աղայան[2]

[1]Երևանի պետական համալսարան, Երևան, Հայաստան
[2]Սթեյթեն Այլենդի քոլեջ, Նյու Յորքի քաղաքային համալսարան, Նյու Յորք, ԱՄՆ
e-mail: hrach.ayunc@gmail.com, sos.agaian@csi.cuny.edu

Ամփոփում

Պատկերի գունազրկումը՝ գունավոր պատկերից մոնոխրոմ պատկերի փոխակերպման գործընթացը, կարևոր դեր է խաղում մեկ բաղադրիչով պատկերների մշակման, համակարգչային տեսողության, թվային տպագրության և մոնոխրոմ վիզուալիզացիայի մեջ: Այս գործընթացը առաջացնում է նոր աղավաղումներ, որոնց ազդեցությունը տեսողական որակի վրա պետք է բացահայտվի: Թեև տեսողական որակի գնահատումը եղել է բազմաթիվ ուսումնասիրությունների առարկա, դեռևս կան որոշ բաց հարցեր կապված փոխակերպման որակի նոր մետրիկաների հետ: Օրինակ՝ համակարգչային մոդելավորումը ցույց է տալիս, որ հաճախ օգտագործվող որակի չափորոշիչները, ինչպիսիք են՝ CCPR-ը, CCFR-ը և E-score-ը, կախված են պարամետրերից և կարող են ընտրել տարբեր լավագույն մեթոդներ՝ փոփոխելով պարամետրերը:

Հոդվածում առաջարկվում է պատկերների գունազրկման որակի գնահատման նոր չափորոշիչ, որը հիմնված է մարդու տեսողական հատկությունների վրա և հաշվվում է ռեգրեսիայի մեթոդի միջոցով: Փորձարարական արդյունքները ցույց են տալիս, որ առաջարկվող մետրիկան ավելի կայուն է, քան գոյություն ունեցողները, այն նաև ունի բարձր կորելացիա միջին կարծիքի գնահատականի (MOS) հետ, և դրա օգնությամբ հնարավոր է ընտրել լավագույն գունազերծման մեթոդները:

Բանալի բառեր՝ գունավոր պատկերների փոխակերպում մոնոխրոմ պատկերների, գունազրկում, մոնոխրոմ պատկեր, ռեգրեսիա, որակի չափորոշիչ:

Новая метрика качества оценки обесцвечивания изображения

Грач Ю. Аюнц[1] и Сос С. Агаян[2]

[1]Ереванский государственный университет, Ереван, Армения
[2]Колледж Статен-Айленда, Городской университет Нью-Йорка, Нью-Йорк, США
e-mail: hrach.ayunc@gmail.com, sos.agaian@csi.cuny.edu

Аннотация

Обесцвечивание изображения, процесс преобразования цветного изображения в монохромное, играет решающую роль в одноканальной обработке, компьютерном зрении, цифровой печати и монохромной визуализации. Этот процесс вызывает новые артефакты, влияние которых на визуальное качество должно быть определено. Несмотря на то, что оценка визуального качества была предметом многих исследований, все еще остается несколько открытых вопросов, касающихся новых показателей качества преобразования цветного изображения в серый. Например, компьютерное моделирование показывает, что обычно используемые показатели качества преобразовании, такие как CCPR, CCFR и E-score, зависят от параметров и могут выбирать различные наилучшие методы путем изменения параметров.

В этой статье предлагается новая метрика качества для оценки методов обесцвечивания изображения. Она использует информацию о зрительных свойствах человека и метод регрессии. Экспериментальные результаты также показывают сильную корреляцию между представленной метрикой качества обесцвечивания изображения и средней оценкой мнений (MOS), более надежную, чем существующие метрики качества, и помогают выбрать лучший из современных методов обесцвечивания с использованием представленной метрики и существующих метрик качества.

Ключевые слова: преобразования цветного изображения в монохромное, обесцвечивание, монохромное изображение, регрессия, метрика качества.

Mathematical Problems of Computer Science 57, 30–38, 2022.
doi: 10.51408/1963-0084

UDC 004.8

Compact N-gram Language Models for Armenian

Davit S. Karamyan[1] and Tigran S. Karamyan[2]

[1]Russian-Armenian University, Yerevan, Armenia
[2]Yerevan State University, Yerevan, Armenia
e-mail: davitkar98@gmail.com, t.qaramyan@ysu.am

Abstract

Applications such as speech recognition and machine translation use language models to select the most likely translation among many hypotheses. For on-device applications, inference time and model size are just as important as performance. In this work, we explored the fastest family of language models: the N-gram models for the Armenian language. In addition, we researched the impact of pruning and quantization methods on model size reduction. Finally, we used Bye Pair Encoding to build a subword language model. As a result, we obtained a compact (100 MB) subword language model trained on massive Armenian corpora.

Keywords: Armenian language, N-gram Language Model, Subword Language Model, Pruning, Quantization.
Article info: Received 31 March 2022; accepted 17 May 2022.

1. Introduction

Language modeling is a fundamental task of NLP. Models that assign probabilities to sequences of tokens are called language models or LMs. Here, tokens can be words, characters, or subwords. The N-gram is the simplest model that assigns probabilities to sentences and sequences of tokens. Although the N-gram models are much simpler than modern neural language models based on RNN[1, 2] and transformers[3, 4, 5], they are much faster than others since they perform constant-time lookups and scalar multiplications (instead of matrix multiplications in neural models). As always, trade-offs exist between time, space, and accuracy[6]. Hence, much recent work has focused on building faster and smaller N-gram language models[7, 8, 9].

N-gram language models are widely utilized in spelling correction[10], speech recognition[11] and machine translation[12] systems. In such systems, for each utterance/sentence translation, the system generates several alternative token sequences and scores them using N-gram LM to peek the most likely translation sequence. In addition, LM rescoring can be combined with beam search algorithms[13].

The Armenian language has a rich morphology: one word can have several tenses and surface forms. Moreover, one can form long words in Armenian by stringing word pieces together. The inclusion of every form in the vocabulary will make it intractable. Subword dictionaries, in which words are divided into frequent parts, can help reduce vocabulary size. Many efforts have been made to use word decomposition and subword LMs for dealing with out-of-vocabulary words in inflective languages such as Arabic[14], Finnish[15], Russian[16], and Turkish[17]. A review of the literature revealed that there have been no publicly available LM resources for the Armenian language. This work is devoted to the creation of a compact and fast N-gram LM for the Armenian language.

Summing up, we will give answers to the following practical questions: **Q1**. What order of N-grams is enough to build a good LM for the Armenian language? **Q2**. How much data is needed to build a model? **Q3**. How can pruning and quantization help reduce the size of the model? **Q4**. Can we build more compact models by using subwords?

In addition, we are going to release training codes and models.[1]

2. Background

Language Modeling (LM) is the task of predicting which token or word comes next. You might also think of an LM as a system that assigns probability to a piece of text. The probability of a sequence of n tokens $t_1^n\{t_1, ..., t_n\}$ is denoted as $P(t_1^n)$. Using the chain rule of probability we can decompose this probability:

$$P(\{t_1, ..., t_n\}) = \prod_{k=1}^{n} P(t_k | t_1^{k-1}).$$

Instead of computing the probability of a token given its entire history, it is usually conditioned on a window of N previous tokens. The assumption that the probability of a token depends only on the previous $N-1$ token is called a Markov assumption:

$$P(t_k|t_1^{k-1}) \approx P(t_k|t_{k-N+1}^{k-1}).$$

We can estimate the probabilities of an N-gram model by getting counts from a corpus and normalizing the counts so that they lie between 0 and 1. For example, to compute a particular N-gram probability of a token t_k given the previous tokens t_{k-N+1}^{k-1}, we'll compute the count of the N-gram t_{k-N+1}^{k} and normalize it by the sum of all the N-grams that share the same prefix t_{k-N+1}^{k-1}:

$$P(t_k|t_{k-N+1}^{k-1}) = \frac{Count(t_{k-N+1}^{k})}{\sum_t Count(t_{k-N+1}^{k-1}, t)} = \frac{Count(t_{k-N+1}^{k})}{Count(t_{k-N+1}^{k-1})}.$$

There are two major problems with N-gram language models: storage and sparsity. To compute N-gram probability we need to store counts for all N-grams in the corpus. As N increases or the corpus size increases, the model size increases as well. Pruning and Quantization may provide a partial solution to reduce the model size. Any N-gram that appeared a sufficient number of times might have a reasonable estimate for its probability. Since any corpus is limited, some perfectly acceptable tokens may never appear in the corpus. As a result of it, for any training corpus, there will be a substantial number of cases of putative zero probability N-grams. To keep an LM from assigning zero probability to these unseen events, we will have to shave off a bit of probability mass from some more frequent events and give it to the events we have never seen. This is called smoothing. There are many ways to do smoothing: add-one(add-k) smoothing, backoff, and Kneser-Ney smoothing[18].

[1] https://github.com/naymaraq/arm-n-gram

3. Experiments

Setup. We estimate N-gram probabilities on Armenian Wikipedia corpus[2] and CC-100 Web Crawl Data[3][19]. To test the language models, we compute perplexity on two test datasets: Armenian Paraphrase Detection Corpus[4] (ARPA[20]) and Universal Dependencies treebank[5] (UD). All datasets are normalized by removing punctuation marks and non-Armenian symbols. Table 1 provides some statistics of the data after all normalization steps have been performed. Table 2 shows unique N-gram counts presented in the training corpus.

We are going to measure the perplexity of corpus C that contains m sentences and N tokens. Let's the sentences $(s_1, s_2, ..., s_m)$ be part of C. Under assumption that those sentences are independent, the perplexity of the corpus is given by:

$$Perp(C) = \sqrt[N]{\frac{1}{p(s_1, s_2, ..., s_m)}} = \sqrt[N]{\frac{1}{\prod_{k=1}^{m} p(s_k)}}.$$

We use KenLM [21] to train language models. KenLM implements two data structures: Probing and Trie, for efficient language model queries, reducing both time and memory costs. KenLM estimates language model parameters from text using modified Kneser-Ney smoothing.

Table 1: Datasets statistics.

Dataset	Tokens (M)	Bytes	Split
CC-100	409	5.4Gb	train
Wiki	18.6	249Mb	train
ARPA	0.133	1.8Mb	test
UD	0.034	425Kb	test

Table 2: N-gram counts.

Order (N)	Count of unique N-grams
1	3648574
2	60190581
3	160796455
4	217396323
5	233510708

Q1. Order of Grams vs Perplexity

To determine what order of N-grams is sufficient to build a good LM for Armenian, we trained several LMs with different orders and calculated perplexity on the test datasets. Fig. 1 shows the trend between perplexity and order of N-gram. It also shows how the size of the model changes as N increases.

From Fig. 1 we can deduce that the effective orders are 5 and 6 grams. Although their sizes are quite large: 3.9GB and 5.5GB.

Q2. Training Corpus size vs Perplexity

The next question we would like to ask is about corpus size. If the training corpus is small, we will end up with a very sparse model, and all perfectly acceptable Armenian tokens will

[2] https://github.com/YerevaNN/word2vec-armenian-wiki

[3] https://data.statmt.org/cc-100/

[4] https://github.com/ivannikov-lab/arpa-paraphrase-corpus

[5] https://github.com/UniversalDependencies/UD_Armenian-ArmTDP

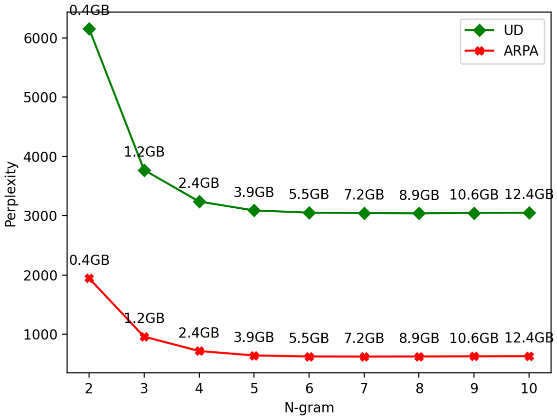


Fig. 1. *N*-gram order vs perplexity.

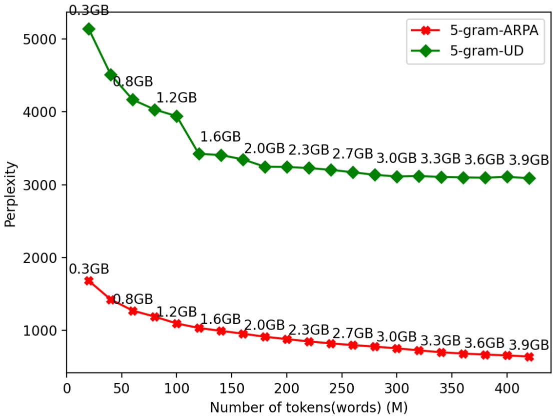


Fig. 2. Number of tokens in training corpus vs perplexity.

be considered unknown. To find out how much data is required, we shuffled and divided the entire training corpus into parts and trained a 5-gram LM for each part. Fig. 2 shows the trend between perplexity and corpus size.

It can be seen that the perplexity reaches saturation when the number of tokens exceeds 380M. Of course, there is always a trade-off between the corpus size, perplexity and the model size: the larger the corpus size, the less perplexity and the larger the model.

Q3. Quantization and Pruning

On-device applications should be as compact as possible. So, the next question we would like to raise concerns the size of the model. Can we build a smaller LM without sacrificing performance?

To reduce the size of the model, we prune all n-grams that appear in the training corpus less than or equal to a given threshold. In addition, we use quantized probabilities by setting fewer bits. For this experiment, we trained a 5-gram LM.

The effect of pruning and quantization is provided in Table 3. Quantization can help reduce the size of a model by a couple of megabytes without perplexity degradation. In contrast, pruning drastically reduces the size of the model at the cost of worsening perplexity. For example, removing all n-grams less than or equal to 4 can reduce the size of the model by more than 12 times with a relative perplexity degradation of 36% for the UD dataset and 100% for the ARPA dataset.

Q4. Subword Language Modeling

So far, we have considered text as a sequence of words separated by a space. Space tokenization is an example of word tokenization, which is defined as breaking sentences into words. The word tokenization method can lead to problems for massive text corpora and usually generates a very big vocabulary (e.g., our training corpus contains 3, 648, 574 unique tokens, see Table 1). Instead of using word tokenization, we will use subword tokenization, which is based on the principle that frequently used words should not be split into smaller subwords, but rare words should be decomposed into meaningful subwords. There are several subword

Table 3: The effect of pruning and quantization on the trade-off between size and perplexity.

Pruning threshold	N-bits	Size	UD	ARPA
0	5	3.44Gb	3043.47	631.58
0	6	3.59Gb	3068.62	638.84
0	7	3.74Gb	3075.99	641.57
0	8	3.9Gb	3089.41	642.93
2	5	481.28Mb	3781.29	1131.82
2	6	501.76Mb	3768.36	1128.14
2	7	512.0Mb	3767.81	1125.54
2	8	532.48Mb	3764.69	1125.0
4	5	296.96Mb	4252.71	1344.56
4	6	307.2Mb	4219.03	1335.89
4	7	317.44Mb	4218.13	1332.73
4	8	317.44Mb	4217.73	1332.84
6	5	245.76Mb	4473.19	1486.95
6	6	245.76Mb	4432.03	1474.89
6	7	256.0Mb	4435.29	1471.75
6	8	256.0Mb	4431.23	1471.89
8	5	215.04Mb	4694.73	1588.09
8	6	225.28Mb	4655.29	1576.21
8	7	225.28Mb	4652.95	1571.46
8	8	225.28Mb	4652.89	1571.94

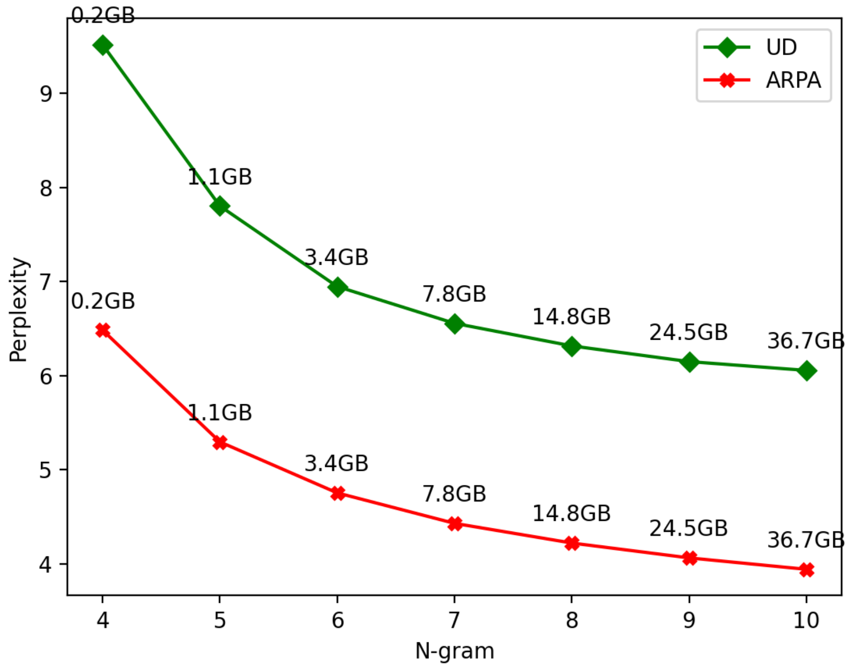


Fig. 3. N-gram order vs perplexity (subword).

tokenization algorithms: Byte-Pair Encoding[22] , WordPiece[23], and SentencePiece[24]. Subword tokenization allows the model to have a reasonable vocabulary size. In addition, subword tokenization enables the model to process words it has never seen before by decomposing them into known subwords. This is especially useful in agglutinative languages such as Armenian, where you can form long words by stringing subwords together.

We trained a BPE tokenizer with a vocabulary size of 128 using the SentencePiece package[6]. Next, we build several N-gram models on a tokenized corpus. Fig. 3 shows the trend between perplexity and order of N-gram for subword model. It also shows how the size of the model changes as N increases.

Table 4: Pruning effect for the subword model with 10-gram.

Pruning	Size	UD	ARPA
0	36.66Gb	6.055	3.941
2	1.11Gb	6.199	4.19
4	634.88Mb	6.323	4.306
6	440.32Mb	6.373	4.381
8	348.16Mb	6.435	4.44
10	286.72Mb	6.53	4.491
16	184.32Mb	6.781	4.619
20	153.6Mb	6.892	4.69
24	122.88Mb	7.02	4.751
30	102.4Mb	7.146	4.837

First, in Fig. 3 the perplexity (0-10) is significantly lower than the perplexity of the word-based tokenized model (0-7000, see Fig. 1). This is because we no longer have unknown tokens. In contrast to word-based models, subword models are much larger (e.g., 10-gram subword model is 3 times bigger).

Since the sequences no longer contain words, but contain subwords, in order to capture sufficient context, we need to consider higher order grams. From Fig. 3 it can be seen that the higher the order, the larger the model (for example, a subword model with 10-gram has a size of 36.7 GB). To reduce the size of the model, we use pruning again. Table 4 provides information about the pruning effect for the subword model with 10-gram. It can be seen that we can reduce the model size by a factor of 368 from 36.7 GB to 102 MB with a relative perplexity degradation of 18% for the UD dataset and 23% for the ARPA dataset.

4. Conclusions

In this article, we have explored N-gram language models for the Armenian language. Our experiments have shown that for word-based language models, the effective orders are 5 and 6. In contrast, the effective order for subword language models can be higher than 10.

We have also explored the impact of pruning and quantization on the trade-off between model size and perplexity. Quantization can help reduce the size of the model without

[6] https://github.com/google/sentencepiece

degrading perplexity significantly. Pruning, on the other hand, drastically reduces the size of the model at the expense of aggravating perplexity. For the subword language model, the perplexity degradation is much lower than for the word-based language model.

We have released compact N-gram language models built on very large corpora.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory. Neural computation", vol. 9, no. 8, pp. 1735-1780, 1997.

[2] J. Sarzyska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz and . Okruszek, "Detecting formal thought disorder by deep contextualized word representations", *Psychiatry Research*, vol. 304, pp. 114–135, 2021.

[3] J. Devlin, M. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P.J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer", *arXiv preprint arXiv:1910.10683*, 2019.

[5] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell and others, "Language models are few-shot learners", *arXiv preprint arXiv:2005.14165*, 2020.

[6] C. Buck, K. Heafield and B.V. Ooyen, "N-gram counts and language models from the common crawl", *In: LREC*, vol. 2, no. 4, 2014.

[7] A. Pauls and D. Klein, "Faster and smaller n-gram language models". *In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 258-267, 2011.

[8] D. Guthrie and M. Hepple, "Storing the web in memory: Space efficient language models with constant time retrieval", *In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 262-272, 2010.

[9] U. Germann, E. Joanis and S. Larkin, "Tightly packed tries: How to fit large models into memory, and make them load fast, too", *In: Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA- NLP 2009)*, pp. 31-39, 2009.

[10] S.D. Hernandez and H. Calvo, "Conll 2014 shared task: Grammatical error correction with a syntactic n-gram language model from a big corpora", *In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 53-59, 2014.

[11] A.Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates and others, "Deep speech: Scaling up end-to-end speech recognition", *arXiv preprint arXiv:1412.5567*, 2014.

[12] H. Schwenk, D. Dchelotte and J. Gauvain, "Continuous space language models for statistical machine translation", *In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 723-730, 2006.

[13] A.Y. Hannun, A.L. Maas, D. Jurafsky and A. Y. Ng, “First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns”, *arXiv preprint arXiv:1408.2873*, 2014.

[14] A.E.D. Mousa, H.J. Kuo, L. Mangu and H. Soltau, “Morpheme-based feature-rich language models using deep neural networks for lvcsr of egyptian arabic”, *In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8435-8439, 2013.

[15] V. Siivola, T. Hirsimki, M. Creutz and M. Kurimo, “Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner”, *In: Proc. Eurospeech*, vol. 3, pp. 2293-2296, 2003.

[16] I. Oparin, “Language models for automatic speech recognition of inflectional languages”, *University of West Bohemia*, 2008.

[17] D. Yuret and E. Bicici, “Modeling morphologically rich languages using split words and unstructured dependencies”, *In: Proceedings of the ACL-IJCNLP 2009 conference short papers*, pp.345–348, 2009.

[18] D. Jurafsky, “Speech and language processing”, *Pearson Education India*, 2000.

[19] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmn, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, “ Unsupervised cross-lingual representation learning at scale”, *arXiv preprint arXiv:1911.02116*, 2019.

[20] A. Malajyan, K. Avetisyan and T. Ghukasyan, “Arpa: Armenian paraphrase detection corpus and models”, *In: 2020 Ivannikov Memorial Workshop (IVMEM)*, pp. 35-39, 2020.

[21] K. Heafield, “Kenlm: Faster and smaller language model queries”, *In: Proceedings of the sixth workshop on statistical machine translation*, pp. 187-197, 2011.

[22] R. Sennrich, B. Haddow and A. Birch, “Neural machine translation of rare words with subword units”, *arXiv preprint arXiv:1508.07909*, 2015.

[23] M. Schuster and K. Nakajima, “Japanese and korean voice search”, *In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* pp. 5149-5152, 2012.

[24] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing”, *arXiv preprint arXiv:1808.06226*, 2018.

Կոմպակտ N-գրամ լեզվի մոդելներ հայերենի համար

Դավիթ Ս. Քարամյան[1] և Տիգրան Ս. Քարամյան[2]

[1] Ռուս-հայկական համալսարան, Երևան, Հայաստան
[2] Երևանի պետական համալսարան, Երևան, Հայաստան
e-mail: davitkar98@gmail.com, t.qaramyan@ysu.am

Ամփոփում

Այնպիսի հավելվածներ, ինչպիսիք են խոսքի ճանաչումը և մեքենայական թարգմանությունը, օգտագործում են լեզվի մոդելներ բազմաթիվ վարկածների մեջ ամենահավանական թարգմանությունն ընտրելու համար: Սարքերի վրա տեղադրված հավելվածների համար եզրակացության ժամանակը և մոդելի չափը նույնքան կարևոր են, որքան արտադրողականությունը: Այս աշխատանքում մենք ուսումնասիրել ենք լեզվական մոդելների ամենաարագ ընտանիքը՝ N-gram մոդելները հայերենի համար: Բացի այդ, մենք ուսումնասիրել ենք կտրման և քվանտացման մեթոդների ազդեցությունը մոդելի չափի կրճատման վրա: Ի վերջո, մենք օգտագործել ենք Bye Pair Encoding՝ ենթաբառերի լեզվի մոդել ստեղծելու համար: Արդյունքում ստացել ենք կոմպակտ (100 ՄԲ) ենթաբառերի լեզվի մոդել՝ պատրաստված հայկական զանգվածային կորպուսների վրա:

Բանալի բառեր՝ հայոց լեզու, N-gram լեզվի մոդել, ենթաբառերի լեզվի մոդել, էտում, քվանտացում:

Компактные языковые модели N-грамм для армянского языка

Давид С. Карамян[1] и Тигран С. Карамян[2]

[1] Российско-Армянский университет, Ереван, Армения
[2]Ереванский государственный университет, Ереван, Армения
e-mail: davitkar98@gmail.com, t.qaramyan@ysu.am

Аннотация

Такие приложения, как распознавание речи и машинный перевод, используют языковые модели для выбора наиболее вероятного перевода среди множества гипотез. Для приложений на устройстве время вывода и размер модели так же важны, как и производительность. В этой работе мы исследовали самое быстрое семейство языковых моделей: модели N-грамм для армянского языка. Кроме того, мы исследовали влияние методов обрезки и квантования на уменьшение размера модели. Наконец, мы использовали Bye Pair Encoding для построения модели языка подслов. В результате мы получили компактную (100 МБ) модель языка подслов, обученную на массивных армянских корпусах.

Ключевые слова: Армянский язык, модель языка N-грамм, модель языка подслов, обрезка, квантование.

Mathematical Problems of Computer Science 57, 39–46, 2022.
doi: 10.51408/1963-0085

UDC 004.05

Electronic Voting System Essentials and Problems

Arman A. Avetisyan

Russian-Armenian University
e-mail: armanavetisyan1997@gmail.com

Abstract

The development of reliable and safe e-voting systems is relevant because of the wide range of applications. This paper provides an analysis of modern electronic voting systems based on security criteria. An analysis was conducted based on the most popular modern e-voting system architectures. The analysis provides a baseline for developing a secure e-voting system.

Keywords: Electronic voting, Internet voting, Information security, Elections, System architecture, Voting systems.
Article info: Received 18 February 2022; received in revised form 16 September 2022; accepted 15 November 2022.
Acknowledgement: The work was supported by the Science Committee of RA, within the framework of the research project 21T-1B151.

1. Introduction

Electronic voting (e-voting) is a term that encompasses several different types of voting methods and electronic means of counting votes. E-voting systems include punched cards, optical voting systems and specialized voting kiosks (including stand-alone electronic systems for direct voting), as well as means for the transmission of ballots and votes by telephone, via a private computer network or via the Internet [1]. Such systems would speed up the counting of votes and make voting more accessible and transparent. However, weak e-voting systems could encourage electoral fraud. The advantages and disadvantages of modern e-voting solutions and technologies should be explored in order to create a secure system. This study focuses on the electronic systems through which the entire electoral process (voter registration, voting and counting votes) is conducted. The study distinguishes the standard functionality of e-voting systems.

Standard e-voting systems include the following modules [2]:

- electronic voter lists and a method of voter identification,
- interface for polling station staff,
- interface for voters,

- system for sending votes to count,
- interface to show results.

The e-voting system should correspond to a series of criteria, which can be divided into two important groups: primary - based on the security and safety of the system, and secondary - based on the user friendliness and accessibility.

System safety requirements are:

- **integrity of elections** (ensuring the accuracy of the elections, all the ballots should be accounted for and no changes must be made to them),
- **privacy of the vote** (make ballots indistinguishable from one another, as well as protect any information about the voter),
- **authenticity of the voters** (only eligable voters can take part in elections),
- **verifiability of the votes** (a person should be able to verify that his vote has been cast and counted),
- **protection against attacks** (the system should be secure against attacks in any phase of the elections, and the voters should be able to alert the committee if any fraud has been detected),
- **ensuring the confidentiality of personal data** (no voter should be able to prove to a third party that he voted for a particular person).

At the international level, the systems developed and tested today have some security problems. A great deal of scientific literature has been devoted to this study [2] - [5], but a number of questions still remain. Even the best e-voting systems today have some drawbacks.

In [6], the author reviewed electoral systems in some countries, where e-voting was used during elections. The comparative analysis was carried out on the basis of the main safety criteria of the most popular modern e-voting systems used in several countries. The study showed that the systems used nowadays are insecure against external attacks and thus raise questions about the integrity of elections they are used in. Most of the systems were implemented more than a decade ago, and the security protocols used have gone obsolete since then. Even the best systems that are steadily replacing paper voting, have been criticized by third-party studies due to massive vulnerabilities. The need for secure and reliable e-voting systems remains a relevant problem nowadays. This paper proposes a model based on the most popular practices in modern e-voting systems.

2. E-voting System Architecture

In the last twenty years there has been active research into the creation of secure voting systems. These systems are based on public-key cryptography and on the approach that the voter's vote is encrypted with a public key that corresponds to it. The private key is distributed among the members of the electoral commission, so the members of the electoral commission will be able to decrypt and count the votes together. In addition, special methods are used to ensure the secrecy of the ballots (MIX network, additive homomorphic encryption systems...). This study proposes a baseline system architecture based on e-voting systems and protocols used nowadays, as well as their known vulnerabilities.

First, we outline the basic voting procedure, divided into 5 key phases:

- **setup phase** - setting up the election system architecture,
- **e-ballot filling phase** - filling out an e-ballot on a device and casting the vote,
- **e-ballot registration phase** - checking if the e-ballot is eligible and storing it on a server,
- **anonymity phase** - making sure all the voter info is stripped away from the e-ballot,
- **counting phase** - counting all the anonymous ballots and providing the results to public.

2.1 3-Server Architecture

The proposed e-voting system architecture consists of 3 main servers (see Fig. 1):

- **Vote forwarding server** is the only publicly accessible server. It verifies the eligibility of the voter and acts as an intermediary to the backend vote storage server.
- **Vote storage server** is a backend server that stores signed, encrypted votes during the online voting period. Upon receiving a vote from the forwarding server, it confirms that the vote is formatted correctly and verifies the voter's digital signature.
- **Vote counting server** is never connected to a network and is only used during the final stage of the election to count the votes received from the storage server.

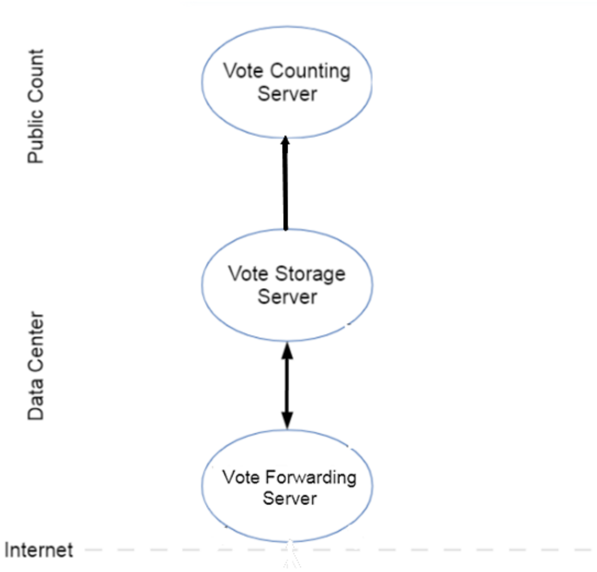


Fig. 1. 3-Server architecture.

2.2 Voting Process

During the initial voting stage, the voter uses client software to cast a vote. The software has a connection with a forwarding server, which is used to authenticate the voter and check their eligibility. All communication with elections servers is done via the vote forwarding server which is the only server accessible from the voter's device. The vote forwarding server is an intermediary between the client device and the storage server. After the voter is authenticated, the client receives a package with a set of candidates. When the voter picks a candidate, the software encrypts the information about the candidate and signs the data with the unique key of the voter.The software then sends the encrypted data package to the forwarding server which returns an ID of the package meaning the vote has been successfully casted. It is important to note that no information about the voter is sent to the election server other than their unique signature which can only be used to check the eligibility of the vote and not the identity of the voter.
The transfer of this data between the forwarding server and storage server remains a huge issue even nowadays because each part of the system trusts the channels through which the vote data is transferred. The transfer from client to storage is done via the Internet using secure protocols (see Fig. 2).

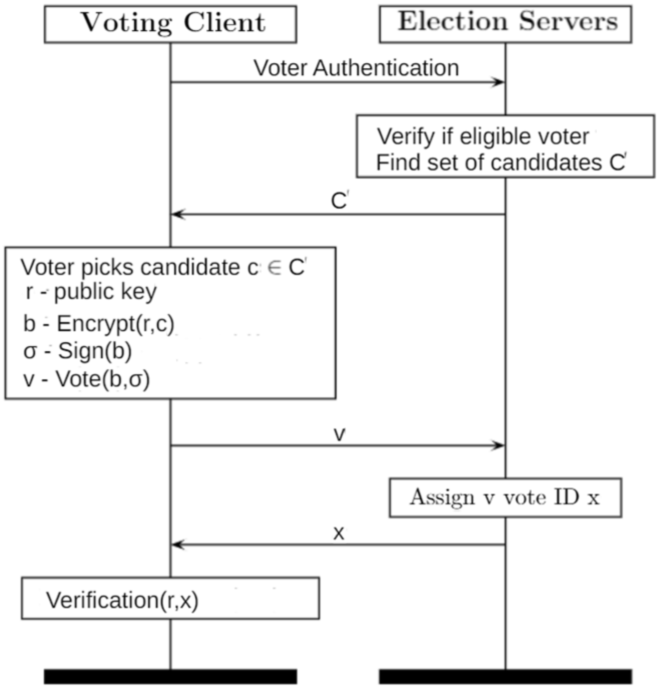


Fig. 2. Casting Vote.

When the vote data arrives at the vote storage server, it is once again checked and verified. Then the sensitive data like the unique signature is stripped from the votes to make them completely anonymous before sending to the counting server where the data (which contains only information about who the vote is for) is decrypted and tabulated (see Fig. 3).

Counting server then writes all the data into an accessible database from which the final results can be gathered.

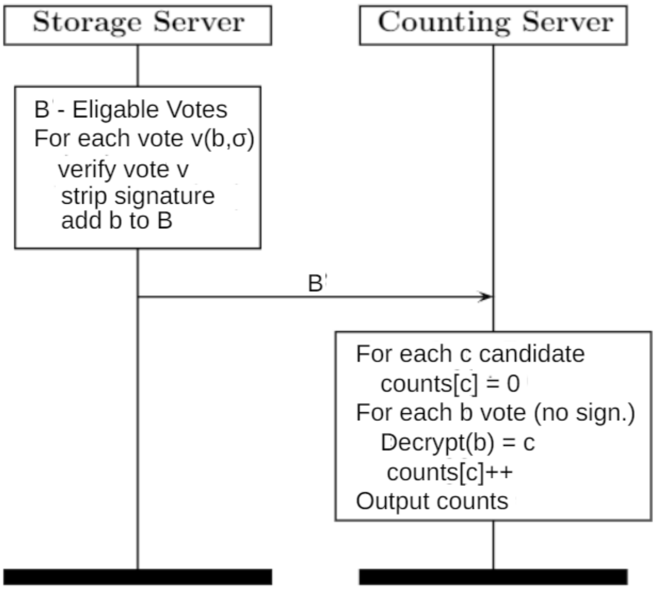


Fig. 3. Vote Counting.

The discussed system is a basic model of e-voting systems being used nowadays [7] - [10], but there are vulnerabilities and areas for improvement.

3. Discussion About Vulnerabilities

Although the system may seem straightforward and secure, its current implementations have raised serious concerns [11] - [13]. The main concern is the secure transfer of the vote data. Usage of client-side software is essential in e-voting systems so the process of transferring votes from the client device to the election server should be handled carefully. The inclusion of transfer server as a buffer between the client and the storage server is integral. We can differentiate two types of attacks: client-side and server-side.

Client-side attacks target the client device and exploit its vulnerabilities. The client software needs to be as secure as possible against this kind of attacks by not having any unwarranted communication with the device.

Server-side attacks target server architecture. The system must be minimally dependent on the person or people controlling it to be secure. The human factor plays a huge role in exploiting the election systems. Another major vulnerability is the code vulnerability. Due to the complex nature of the system, current implementations have been proven to have significant oversight in security against attacks like denial of service or shell injection.

Attacks target client software or server architecture and try to achieve the following:

- find out confidential information about voters like the candidate they voted for,
- try to alter the results of the election by changing or adding fake votes,
- altering/destroying enough votes to create mistrust in the election results.

To find out information about voters, the attack needs to take place before the anonymity phase, so client-side attacks mainly take place for this purpose. If the channel between the client software and the transfer server is not secure, the data may be intercepted and even altered. Modern cryptography methods help to encrypt data, so it is hard to decode, but adding noise to an insecure channel to alter or ruin the vote is easy if someone already has access. These attacks are generally low-scale as user devices need to be exploited one by one, even with techniques like botnets it is unlikely to cause too much harm to the overall security of elections.

The main damage to elections is caused by attacking storage and counting servers and the channel between them. As all the important vote databases are in those servers, if access is received by an attacker, the damage to elections will be massive. Having a good and secure architecture is the key to preventing that from happening. Currently, the systems use physical data transfer from one server to another, as well as a physical decryption device that holds the key. This relies too much on actual human beings to securely transfer a lot of essential data from one point to another, and the whole point of implementing e-voting is to get rid of the human factor. This also means that the storage server must be easily accessible for humans to put in data, which is not ideal.

We see that modern systems in use still use ”hybrid” systems partly run by people to make up for software vulnerabilities that need to be addressed in the future works when it comes to developing better e-voting systems.

4. Conclusion and Future Work

The paper discussed the basic architecture of the e-voting system, which is the baseline of systems used nowadays. Understanding the potential problems and vulnerabilities is essential to creating a secure e-voting system. The 3-server architecture is a good starting point to build a new system. It is evident that the physical transportation of data in modern systems is the key problem that has to be addressed first. The use of various steganographic models can help to reduce the risks of data corruption and tampering. More specifically, the steganography models with active adversary are very close to imitating attacks that can happen during elections. One of the main focus points of more secure systems will be the testability, it is essential to have an ability to quantify the level of security of the model. To achieve this goal, research has to be conducted in various fields of security, specifically steganography and differential privacy.

References

[1] M. Stenbro, *“Survey of Modern Electronic Voting Technologies”*, The Norwegian University of Science and Technology, Master Thesis, 2010.

[2] International IDEA *“Introducing Electronic Voting: Essential Considerations”*, https://www.corteidh.or.cr/tablas/28047, 2011.

[3] K. Sanjay and E. Walia, “Analysis of electronic voting system in various countries”, *International Journal on Computer Science and Engineering*, vol. 3, no. 5, pp. 1825–1830, May 2011.

[4] V. Martin, “Evaluation of internet voting systems based on requirements satisfaction”, *International Review of Social Sciences and Humanities*, vol. 6, no.1 , pp. 41-52, 2013.

[5] A. T. Sherman, R. A. Fink, R. Carback and D. Chaum, “Scantegrity III: Automatic trustworthy receipts, highlighting over/under votes, and full voter verifiability”, *In Proceedings of the Electronic Voting Technology/Workshop on Trustworthy Elections, EVT/WOTE*, 2011.

[6] A. Avetisyan, “Comparative analysis of modern E-voting systems based on security criteria”,*Proceedings of International Conference CSIT 2021*, Yerevan, Armenia, pp. 81-84, 2021.

[7] N. Goodman, J.H. Pammett and J. De Bardeleben, “A comparative assessment of electronic voting”, *Report Prepared for Elections Canada*, 2010.

[8] L. Loeber, “E-Voting in the Netherlands: from general acceptance to general doubt in two years, *3rd International Conference on Electronic Voting*, pp. 2130, 2008.

[9] D. F. Aranha and J. van de Graaf, “The Good, the Bad, and the Ugly: Two decades of E-voting in Brazil”, *IEEE Security and Privacy*, vol. 16, no. 6, pp. 22-30, Nov.-Dec. 2018.

[10] M. Hapsara, A. Imran and T. Turner, ”E-Voting in developing countries”, *Electronic Voting. E-Vote-ID 2016. Lecture Notes in Computer Science*, vol. 10141, pp. 3655, 2017.

[11] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine and J. Halderman, ”Security analysis of the estonian internet voting system”, *Proceedings of the 21st ACM Conference on Computer and Communications Security*, pp. 703-715, 2014.

[12] T. Haines, S. J. Lewis, O. Pereira and V. Teague, “How not to prove your election outcome,” *IEEE Symposium on Security and Privacy (SP)*, pp. 644-660, 2020.

[13] M. A. Specter, J. Koppel and D. Weitzner, “The ballot is busted before the blockchain: a security analysis of voatz, the first internet voting application used in U.S. federal elections”, *Proceedings of the 29th USENIX Conference on Security Symposium. USENIX Association*, pp. 1535-1552, 2020.

Էլեկտրոնային քվեարկության հիմունքները և խնդիրները

Արման Ա. Ավետիսյան

Ռուս-հայկական համալսարան, Երևան, Հայաստան
e-mail: armanavetisyan1997@gmail.com

Ամփոփում

Հուսալի և անվտանգ էլեկտրոնային քվեարկության համակարգերի մշակումն արդիական է կիրառությունների լայն շրջանակի պատճառով։ Աշխատանքում ներկայացվում է ժամանակակից էլեկտրոնային քվեարկության համակարգերի վերլուծություն՝ հիմնված անվտանգության չափանիշների վրա։ Վերլուծությունն իրականացվել է՝ հիմնվելով էլեկտրոնային քվեարկության ամենահայտնի ժամանակակից համակարգերի վրա։ Վերլուծությունը հիմք է հանդիսանում անվտանգ էլեկտրոնային քվեարկության համակարգերի մշակման համար։

Բանալի բառեր՝ էլեկտրոնային քվեարկություն, ինտերնետ քվեարկություն, տեղեկատվական անվտանգություն, ընտրություններ, համակարգի ճարտարապետություն, քվեարկության համակարգեր։

Основы и проблемы электронного голосования

Арман А. Аветисян

Российско-Армянский университет, Ереван, Армения
e-mail: armanavetisyan1997@gmail.com

Аннотация

Разработка надежных и безопасных систем электронного голосования актуальна из-за широкого спектра применений. В данной работе был проведен анализ современных систем электронного голосования на основе критериев безопасности. Анализ проводился на основе наиболее популярных современных архитектур систем электронного голосования. Данный анализ является основой для разработки безопасной системы электронного голосования.

Ключевые слова: электронное голосование, интернет-голосование, информационная безопасность, выборы, архитектура системы, системы голосования.

Mathematical Problems of Computer Science 57, 47–55, 2022.
doi: 10.51408/1963-0086

UDC 510.64

On Sizes of Linear and Tree-Like Proofs for any Formulae Families in Some Systems of Propositional Calculus

Levon A. Apinyan[1] and Anahit A. Chubaryan[2]

[1]Russian-Armenian University, Yerevan, Armenia
[2]Yerevan State University, Yerevan, Armenia
e-mail: apinlev00@gmail.com, achubaryan@ysu.am

Abstract

The sizes of *linear and tree-like proofs* for any formulae families are investigated in some systems of propositional calculus: in different sequent systems (with quantifier rules, with the substitution rule, with the cut rule, without the cut rule, monotone) and in the generalization splitting system. The comparison of results obtained here with the bounds obtained formerly for the steps of proofs for the same formulas in the mentioned systems shows the importance of the *size of proof* among the other characteristics of proof complexities.

Keywords: The varieties of propositional sequent systems, The generalization splitting system, The proof size and number of proof steps, Exponential speed-up.

Article info: Received 15 March 2022; accepted 12 May 2022.

1. Introduction

The existence of a propositional proof system, which has polynomial-size proofs for all tautologies, is equivalent to saying that N P = co -N P [1]. This simple observation has drawn attention in recent years to the formalisms of propositional logic for the study of questions of computational complexity A hierarchy of propositional proof systems has been defined in terms of two main complexity characteristics (*size* and *lines*), and the relations between these systems are currently being analyzed. New systems are discovered and, as a consequence, the computational power of the old ones is better understood. It was shown in [2] that the addition of quantifier rules to the propositional sequent calculus induces, for some sequences of formulas, an exponential speed-up by lines over Substitution Frege systems when proofs are considered as *trees*. It was shown in [3] that the lines for *linear* proofs of the same formulae families both in quantifier systems and in the systems with substitution systems are the same by order. In this paper, we investigate the sizes of linear and tree-like proofs for the mentioned sequence of formulas in some sequent systems (QPK – the system with quantifier rules, SPK – the system with substitution rule, PK – the system with cut-rule, PK^{-} – the system without cut-rule, Pmon- the monotone system) and in the system GS, based on the generalized splitting method. The comparative analysis of our results shows that the size of proofs is a more important complexity characteristic of proofs and the linear proofs are preferable to the tree-like proofs.

2. Preliminaries

We will use the current concepts of a propositional formula, *quantified* propositional formula, a free variable in a quantified formula, sequent, different sequent systems and proof complexities. The language of the considered systems contains the propositional variables, logical connectives $\neg, \&, \vee, \supset$ and parentheses (,). Note that some parentheses can be omitted in generally accepted cases. In some systems, we can use the symbols T for «true» and $\perp$ for «false».

2.1 Definition of Considered Sequent Systems

The sequent system uses the denotation of sequent $\Gamma \to \Delta$, where Γ (antecedent) and Δ (succedent) are finite (may be empty) sequences of propositional formulas.

For every propositional variable p, the sequents p → p, → T are axioms of PK. For every formulas A , B, for any sequence of formulas Γ and sequence Δ, the logic rules are as follows:

$$\supset\to \frac{\Gamma \to \Delta, \ A \qquad B, \ \Gamma \to \Delta}{A \supset B, \ \Gamma \to \Delta} \qquad \to\supset \frac{A, \ \Gamma \to \ \ B, \ \Delta}{\Gamma \to A \supset B, \Delta}$$

$$\vee\to \frac{A, \ \ \Gamma \to \Delta \text{ and } \ B, \ \Gamma \to \Delta}{A \vee B, \ \ \Gamma \to \Delta} \qquad \to\vee \frac{\Gamma \to A, \ \ \Delta \text{ or } \ \ \Gamma \to B, \ \ \Delta}{\Gamma \to A \vee B, \ \Delta}$$

$$\& \to \frac{A, \ \ \Gamma \to \Delta \text{ or } \ B, \ \Gamma \to \Delta}{A\&B, \ \ \Gamma \to \Delta} \qquad \to \& \frac{\Gamma \to A, \Delta \ \text{ and } \Gamma \to B, \Delta}{\Gamma \to A\&B, \Delta}$$

$$\neg\to \frac{\Gamma \to \ \ \ A, \ \ \Delta}{\neg A, \ \ \Gamma \to \Delta} \qquad \to \neg \frac{A, \ \ \Gamma \to \ \ \ \Delta}{\Gamma \to \neg A, \ \ \Delta},$$

Structural rule is

$$\frac{\Gamma \to \ \ \Delta}{\Gamma' \to \Delta'},$$

where $\Gamma'(\Delta')$ contains $\Gamma(\Delta)$

Cut rule is

$$\frac{\Gamma \ \to \ \Delta, A \quad A, \Gamma \ \to \ \Delta}{\Gamma \ \to \ \Delta}.$$

The system **PK**$^{-}$ is obtained from the system **PK** by removing the cut rule**.** The system **SPK** is obtained from the system **PK** by adding a substitution rule:

$$S_p^{\mathrm{B}} \ \frac{\mathrm{C(p)}, \Gamma \ \to \ \Delta, \mathrm{A(p)}}{\mathrm{C(B)}, \Gamma \ \to \ \Delta, \mathrm{A(B)}},$$

where the variable p has no occurrences either in Γ or in Δ, B is the formula, which is substituted everywhere for the variable p.

The system **QPK** is obtained from the system **PK** by adding the following rules :

$$\frac{A(q), \Gamma \ \rightarrow\ \Delta}{(\exists p)A(p), \Gamma \ \rightarrow\ \Delta}(\exists \rightarrow) \qquad \frac{\Gamma \ \rightarrow\ \Delta, A(B)}{\Gamma \ \rightarrow\ \Delta, (\exists p)A(p)}(\rightarrow \exists)$$

$$\frac{A(B)\Gamma \rightarrow \Delta}{(\forall p)A(p),\Gamma \rightarrow \Delta}(\forall\rightarrow) \qquad \frac{\Gamma \rightarrow \Delta,A(q)}{\Gamma \rightarrow \Delta,(\forall p)A(p)}(\rightarrow \forall),$$

where B is any quantified propositional formula. The application of the rules ∃ → and → ∀ is restricted to the following requirements: the eigenvariable q does not occur free in the lower sequent of the rule, and all occurrences of q in A(q) are substituted by p. The rules → ∃ and ∀→ require B not to contain variables, which are under the scope of some quantifier.

All formulas in the antecedents and succedents of the system **Pmon** use only monotone logical functions, therefore the rules for implication and negation are not used here.

2.2 Definition of the System GS

Following the usual terminology, we call the variables and negated variables literals.

The following notions were introduced in [4]. Each of the under-mentioned trivial identities for a propositional formula ψ is called *a replacement rule:*

$$\begin{array}{llll}
0\,\&\,\psi = 0, & \psi\,\&\,0 = 0, & 1\,\&\,\psi = \psi, & \psi\,\&\,1 = \psi, \\
0 \vee \psi = \psi, & \psi \vee 0 = \psi, & 1 \vee\psi = 1, & \psi \vee 1 = 1, \\
0 \supset \psi = 1, & \psi \supset 0 = \bar{\psi}, & 1 \supset \psi = \psi, & \psi \supset 1 = 1, \\
\bar{0} = 1, & \bar{1} = 0, & \bar{\bar{\psi}} = \psi, & \\
0 \equiv \psi = \bar{\psi}, & \psi \equiv 0 = \bar{\psi}, & 1 \equiv \psi = \psi, & \psi \equiv 1 = \psi
\end{array}$$

Application of the replacement rule to some words consists in the replacing of some of its subwords having the form of the left-hand side of one of the above identities by the corresponding right-hand side.

The proof system GS. Let φ be some formula and p be some of its variables. Results of the splitting method of formula φ by the variable p (splinted variable) are the formulas $\varphi[p^{\boldsymbol{\delta}}]$ for every $\boldsymbol{\delta}$ from the set {0,1}, which are obtained from φ by assigning $\boldsymbol{\delta}$ to each occurrence of p and successively using replacement rules. The generalization of the splitting method allows every formula φ to associate some tree with a root, the nodes of which are labeled by formulas and edges, labeled by literals. The root is labeled by the formula φ itself. If some node is labeled by the formula v and α is some of its variable, then both edges outgoing from this node, are labeled by one of the literals $\alpha^{\boldsymbol{\delta}}$ for every $\boldsymbol{\delta}$ from the set {0,1}, and each of 2 “sons” of this node is labeled by the corresponding formula $v[\alpha^{\boldsymbol{\delta}}\,]$. Each of the tree’s leaves is labeled with some constant from the set {0,1}. The tree, which is constructed for the formula φ by the described method, we will call ***a splitting tree*** (s.t.) of φ. It is obvious, that by changing the order of splinted variables in the given formula φ, we can obtain different splitting trees of φ.

The ***GS*** proof system can be defined as follows: for every formula φ must be constructed some s.t. and if all the tree’s leaves are labeled by the value 1, then the formula φ is a tautology, and therefore we can consider the pointed constant 1 as an axiom, and for every formula v, which is a label of some s.t. node, and p is its splinted variable, then the following figure $v\,[p^0], v\,[p^1] \vdash v$

can be considered as some inference rule, hence, every above-described s.t. can be considered as some proof of φ in the system ***GS*** .

2.3 Proof Complexities

By $|\varphi|$ we denote the size of a formula φ, defined as the number of all logical signs in it. It is obvious that the full size of a formula, which is understood to be the number of all symbols, is bounded by some linear function in $|\varphi|$.

In the theory of proof complexity, the two main characteristics of the proof are: ***t-complexity*** (lines) defined as the number of proof steps, ***l-complexity*** (size) defined as the sum of sizes for all formulas (sequents) in the proof [1, 2].

Let ϕ be some proof system, φ be some tautology. By $\boldsymbol{t}^{\Phi}(\varphi)\left(\boldsymbol{l}^{\Phi}(\varphi)\right)$ is denoted the minimal possible value of *t*-complexity (*l*-complexity) for all Φ-proofs of φ (sequent → φ).

If for some sequence of sequents →ϕ_n in two systems ϕ_1 and ϕ_2 for sufficiently large n is valid $t^{\phi_1}(\varphi_n) = \Omega(2^{t^{\phi_2}(\varphi_n)})$, then we say that the system ϕ_2 has exponential sped-up by lines over the system ϕ_1.

2.4. Results of the Papers [2,3]

Some family of tautologies is introduced in [2]. For propositional variable p, the formula p^{m} is defined inductively as $p^0 \equiv p$ и $p^{i+1} \equiv (p^i \& p^i)$ for $i \geq 0$. It is easy to verify that the formula p^m has $2^m - 1$ logical signs and m distinct subformulas.

To simplify further notes, we introduce the following denotations. Let Φ be some sequent system, *t- complexity* (*l-complexity*) for tree-like proofs of the sequent $p \rightarrow p^m$ is denoted by $\boldsymbol{Tt}(\boldsymbol{m})\left(\boldsymbol{Tl}^{\Phi}(\boldsymbol{m})\right)$, and for linear proofs, accordingly, by $\boldsymbol{Lt}^{\Phi}(\boldsymbol{m})\left(\boldsymbol{Ll}^{\Phi}(\boldsymbol{m})\right)$.

Theorem 1: ([2]). *For sufficiently large n and sequence of sequents* $p \rightarrow p^{2^n}$ *the following holds*:

$$T\boldsymbol{t}^{\mathbf{QPK}}(2^n)=\mathrm{O}(\mathrm{n});\ T\boldsymbol{t}^{\mathbf{SPK}}(2^n)=\Omega(2^n);\ T\boldsymbol{t}^{\mathbf{PK}}(2^n)=\Omega(2^n);\ T\boldsymbol{t}^{\mathbf{PK-}}(2^n)=\Omega\left(2^{2^n}\right).$$

For the lines of linear proofs of the same sequence, the following was proved in [3]..

Theorem 2: ([3]). *For sufficiently large n and sequence of sequents* $p \rightarrow p^{2^n}$ *the following holds:*

$$\boldsymbol{Lt}^{\mathbf{QPK}}(2^n)=\mathrm{O}(\mathrm{n});\ L\boldsymbol{t}^{\mathbf{SPK}}(2^n)=\mathrm{O}(\mathrm{n});\ L\boldsymbol{t}^{\mathbf{PK}}(2^n)=\theta(2^n);\ L\boldsymbol{t}^{\mathbf{PK-}}(2^n)=\theta(2^n).$$

The comparative analysis results of both above theorems shows that the system **QPK** has no preference by lines of proof over the system SPK, and the latter system has a well-known speed-up by lines over PK. Analogous sped-up was first fixed in [5].

3. The Main Results

3.1. The *l-complexities* of **linear** proofs for the same family of sequents $p \rightarrow p^{2^n}$ in above-mentioned sequent systems are investigated here.

Theorem 3: *For sufficiently large n and sequence of sequents* $p \to p^{2^n}$ *the following holds:*

$$Ll^{\mathbf{QPK}}(2^n) = \theta(2^{2^n});\quad Ll^{\mathbf{SPK}}(2^n) = \theta(2^{2^n});\quad Ll^{\mathbf{PK}}(2^n) = \theta(2^{2^n});\ Ll^{\mathbf{PK-}}(2^n) = \theta(2^{2^n}) \text{ and } Ll^{\boldsymbol{Pmon}}(2^n) = \theta\left(2^{2^n}\right).$$

To prove the mentioned results, we should evaluate the *sizes* of proofs for the sequents $p \to p^{2^n}$ in all the mentioned systems. Note that $|\,p^{2^n}| = 2^{2^n} - 1$ and as the derivable sequent itself must be in every proof, then the lower bounds $\Omega(2^{2^n})$ are obvious for all systems. To prove the upper bounds, we investigate the “good” linear proofs in the mentioned systems.

Linear proof in QPK

We use the tree-like proofs of $p \to p^{2^n}$ in the system **QPK** with O(n) lines [2]. At first, we consider the provable sequent $\forall q(q \supset q^k) \to \forall q(q \supset q^{2k})$, where k is an arbitrary integer and $q^{2k} = (q^k)^k$. The proof of this sequent will not depend on k and can be obtained in a constant number of lines as follows (note, that their sizes are written to the right of every sequent):

$p \to p \quad p^k \to p^k$	$2^{k+2}+2$
$p \supset p^k, p \to p^k$	$2^{k+2}+2$
$\forall q(q \supset q^k), p \to p^k \quad p^{2k} \to p^{2k}$	$2^{k+2}+2^{2k+2}+5$
$\forall q(q \supset q^k), p^k \supset p^{2k}, p \to p^{2k}$	$2^{k+2}+2^{2k+2}+5$
$\forall q(q \supset q^k), p^k \supset p^{2k} \to p \supset p^{2k}$	$2^{k+2}+2^{2k+2}+6$
$\forall q(q \supset q^k), \forall q(q \supset q^k) \to p \supset p^{2k}$	$2^{k+2}+2^{2k+1}+8$
$\forall q(q \supset q^k) \to p \supset p^{2k}$	$2^{k+1}+2^{2k+1}+5$
$\forall q(q \supset q^k) \to \forall q(q \supset q^{2k})$	$2^{k+1}+2^{2k+1}+7$

Note, that this proof is also *linear*. By combining the above sequents *n* times, one obtains

$$\forall q(q \supset q^2) \to \forall q\left(q \supset q^{2^n}\right),$$

and since $\forall q(q \supset q^2)$ is provable in constant lines, one infers $\forall q(q \supset q^{2^n})$, and therefore $\to p \supset p^{2^n}$ in O(n) lines. The number of all logical signs in the pointed part of the proof is

$7 \cdot 2^{k+2} + 9 \cdot 2^{2k+1} + 40$, and as such steps are repeated n times with $k = 2^i$, for i = 0, 1, 2, …, n, then the size of all proofs must be $\sum_{i=0}^{n} (7 \cdot 2^{2^i+2} + 9 \cdot 2^{2^{i+1}+1} + 40)$. The bound of the major addendum is $7\sum_{i=0}^{n} 2^{2^i+2} \le 7\sum_{i=0}^{2^n+2} 2^i \le 7 \cdot (2^{2^n+3} - 1)$, and hence the upper bound is $O(2^{2^n})$. So, $Ll^{\mathbf{QPK}}(2^n) = \theta(2^{2^n})$.

Linear proof in SPK

1	$p^0 \to p^0$ axs.	0	
2	$p^0 \to p^1$ (→ &)	2^1 - 1	
3	$p^1 \to p^2$ subst.	$2^1 - 1 + 2^2 - 1$	
4	$p^0 \to p^2$ cut	$2^2 - 1$	
5	$p^2 \to p^4$ subst.	$2^2 - 1 + 2^4 - 1$	(1)
6	$p^0 \to p^4$ cut	2^4 - 1	

…		
2n + 1	$p^{2^{n-1}} \to p^{2^n}$ subst.	$2^{2^{n-1}} - 1 + 2^{2^n} - 1$
2n + 2	$p^0 \to p^{2^n}$ cut	$2^{2^n} - 1$

It is not difficult to see that the size cannot be more, than $3\sum_{i=0}^{n} 2^{2^i} \leq 3\sum_{i=0}^{2^n} 2^i \leq 2^{2^n+3}$, hence the upper bound is O(2^{2^n}). So, $Ll^{\mathbf{SPK}}(2^n) = \theta(2^{2^n})$.

Linear proof in PK

1	$p^0 \to p^0$	0	
2	$p^0 \to p^1$ (→ &)	2^1 - 1	
3	$p^0 \to p^2$ (→ &)	$2^2 - 1$	
4	$p^0 \to p^3$ (→ &)	$2^3 - 1$	
...			(2)
2^n	$p^0 \to p^{2^n-1}$ (→ &)	$2^{2^n-1} - 1$	
2^n+1	$p^0 \to p^{2^n}$ (→ &)	$2^{2^n} - 1$	

The size of such linear proof must be no more, than $\sum_{i=0}^{2^n} 2^i \leq 2^{2^n+1}$, hence the upper bound is O(2^{2^n}). So, $Ll^{\mathbf{PK}}(2^n) = \theta(2^{2^n})$.

As in this proof we do not use the cut rule, but only the rule (→ &), then the bounds both in **PK— and in Pmon** are analogous.

Theorem 1 is proved.

3.2. The *l-complexities* of **tree-like** proofs for the same family of sequents $p \to p^{2^n}$ in the above-mentioned sequent systems are investigated here.

Theorem 4: *For sufficiently large n and sequence of sequents $p \to p^{2^n}$ the following holds:*

$$Tl^{\mathbf{QPK}}(2^n) = \theta(2^{2^n});\quad Tl^{\mathbf{SPK}}(2^n) = \theta(2^{2^n});$$
$$\log_2(Tl^{\mathbf{PK}}(2^n)) = \theta(2^n);\ \log_2(Tl^{\mathbf{PK}—}(2^n)) = \theta(2^n) \text{ и } \log_2(Tl^{\mathbf{Pmon}}(2^n)) = \theta(2^n).$$

To **prove** these results, we transform every linear proof above into a tree-like proof in the same system.

The size of tree-like proof in QPK: As we noted above, the proof in **QPK** is linear and tree-like simultaneously, hence the bound is the same.

The size of tree-like proof in SPK: We should transform the above proof (1) into tree-like. It is enough to change every part « p^0→ p^{2^i}, $p^{2^i} \to p^{2^{i+1}}$ $(substitution)$, p^0→ $p^{2^{i+!}}$ (cut)» for $0 \leq i \leq n-1$ of linear proof with the part «tree-like proof of p^0→ p^{2^i},$p^{2^i} \to p^{2^{i+1}}$ $(substitution)$, tree-like proof of p^0→ p^{2^i}, p^0→ $p^{2^{i+!}}$ (cut)». After such transformation we have

$$Tl^{\mathbf{SPK}}(2^i) \leq 2Tl^{\mathbf{SPK}}(2^{i-1}) + \left|p^{2^{i-1}} \to p^{2^i}\right| + \left|p^0 \to p^{2^i}\right| \ for\ 1 \leq i \leq n,$$

hence

$$Tl^{\mathrm{SPK}}(2^{\mathrm{n}}) \leq 2^{n} l\text{д}^{\mathrm{SPK}}(2^{0}) + \sum_{i=1}^{n} 2^{n-i}(2^{2^{i}+2}) \leq 2^{n} + 2^{n+2}2^{2^{n}+2}.$$

So, the upper bound is O($2^{2^{n}}$), hence $Tl^{\mathbf{SPK}}(2^{n})= \theta(2^{2^{n}})$.

The size of tree-like proof in PK: Here we should transform the above proof (2) into tree-like. It is enough to change every part « $\mathrm{p}^{0}\rightarrow p^{i}$, $\mathrm{p}^{0}\rightarrow p^{i+1}$ (→ &)» for $0\leq i \leq n-1$ of linear proof with the part « tree-like proof of $\mathrm{p}^{0}\rightarrow p^{i}$, tree-like proof of $\mathrm{p}^{0}\rightarrow p^{i}$, $\mathrm{p}^{0}\rightarrow p^{i+1}$ (→ &)», then it is obvious that

$$Tl^{\mathbf{PK}}(\boldsymbol{i}) \leq 2Tl^{\mathbf{PK}}(\boldsymbol{i}-\mathbf{1}) + \left|p^{0} \rightarrow p^{i}\right| for\ 1 \leq i \leq 2^{n},$$

hence we have

$$Tl^{\mathrm{PK}}(2^{\mathrm{n}}) \leq 2^{2^{n}} Tl^{\mathrm{PK}}(2^{0}) + \sum_{i=1}^{2^{n}} 2^{i}(2^{n-i}) \leq 2^{2^{n}} + 2^{n}2^{2^{n}(2^{n}+1)/2}.$$

So, the upper bound for $\log_2(Tl^{\mathrm{PK}}(2^{\mathrm{n}}))$ is O(2^{n}), and hence $\log_2(Tl^{\mathbf{PK}}(2^{n}))= \theta(2^{n})$.

As above, in this proof we do not use the cut rule, but only the rule (→ &), then the bounds both in **PK⁻** and in **Pmon** are analogous.

Theorem 4 is proved.

Note that we do not have any exponential speed-up here (it may only be quadratic).

The size of linear and tree-like proofs in GS:

Theorem 5: *For sufficiently large n and sequence of formulas* $p \supset p^{2^{n}}$ *the following holds:*

$$Lt^{\boldsymbol{GS}}(2^{n})= \theta(1); \quad Tt^{\boldsymbol{GS}}(2^{n})= \theta(1);$$

$$Ll^{\boldsymbol{GS}}(2^{n})= \theta(2^{2^{n}}); \ Tl^{\boldsymbol{GS}}(2^{n})= \theta(2^{2^{n}}).$$

Note that every formula $p \supset p^{2^{n}}$ has only one variable for split, hence the **proof** of Theorem 5 is obvious.

4. Conclusion

The analysis of all the results shows that in the theory of proof complexity, the investigations of *l-complexity in linear proofs* are important.

References

[1] S. A. Cook and A. R. Reckhow, “The relative efficiency of propositional proof systems”, *Symbolic Logic*, vol. 44, pp. 36-50, 1979.

[2] A. Carbone, “Quantified propositional logic and the number of lines of tree-like proofs”, *Studia Logica*, vol. 64, pp. 315-321, 2000.

[3] А. А. Тамазян и А. А. Чубарян, “Об отношениях сложностей выводов в ряде систем исчисления высказываний”, *Математические вопросы кибернетики и вычислстельной техники*, vol. 54, pp. 138-146, 2020.

[4] Ан. А. Чубарян и Арм. А. Чубарян, “Оценки некоторых сложностных характеристик выводов в системе обобщенных расщеплений”, НАУ, *Отечественная наука в эпоху изменений: постулаты прошлого и теории нового времени*, часть 10, 2(7), стр.11-14, 2015.

[5] Г. Цейтин и Ан. Чубарян, “Некоторые оценки длин логических выводов в классическом исчислении высказываний”, *ДАН Арм. ССР*, том 55, но.1, стр. 10-12, 1972.

Ասույթային հաշվի մի շարք համակարգերում բանաձևերի որոշ ընտանիքների գծային և ծառատիպ արտածումների երկարությունների մասին

Լևոն Ա. Ափինյան[1] և Անահիտ Ա. Չուբարյան[2]

[1]Ռուս-հայկական համալսարան, Երևան, Հայաստան
[2]Երևանի պետական համալսարան, Երևան, Հայաստան
e-mail: apinlev00@gmail.com, achubaryan@ysu.am

Ամփոփում

Բանաձևերի մի քանի ընտանիքների համար ուսումնասիրված են *գծային և ծառատիպ արտածումների* երկարությունները ասույթային հաշվի մի քանի համակարգերում՝ սեկվենցիալ համակարգերի տարատեսակներում /ծավալիչներով, տեղադրման կանոնով, հատույթի կանոնով, առանց հատույթի կանոնի, մոնոտոն/, ինչպես նաև ընդհանրացված տրոհումների համակարգում: Սույն աշխատանքում ստացված արդյունքների համեմատումը նախկինում ստացված նույն բանաձևերի նշված համակարգերում արտածումների քայլերի համար ստացված արդյունքների հետ փաստում են *արտածումների երկարության*՝ որպես բարդության բնութագրիչի արժեքավորումը:

Բանալի բառեր՝ ասույթային հաշվի սեկվենցիալ համակարգերի տարատեսակներ, ընդհանրացված տրոհումների համակարգ, արտածման քայլերի քանակ և երկարություն, էքսպոնենցիալ արագացում:

О длинах линейных и древовидных выводов некоторых семейств формул в ряде систем исчисления высказываний

Левон А. Апинян[1] и Анаит А. Чубарян[2]

[1]Российско-Армянский университет, Ереван, Армения
[2]Ереванский государственный университет, Ереван, Армения
e-mail: apinlev00@gmail.com, achubaryan@ysu.am

Аннотация

Для некоторых семейств формул исследованы *длины* линейных и древовидных выводов в ряде систем исчисления высказываний: в разновидностях секвенциальных систем (с кванторами, с правилом подстановки, с правилом сечения, без правила сечения, монотонных), а также в системе обобщенных расщеплений. Сравнение полученных результатов с ранее полученными оценками для *шагов* тех же разновидностей выводов тех же формул и в тех же системах указывают на определенную значимость именно длины вывода как основной сложностной характеристики выводов.

Ключевые слова: разновидности секвенциальных систем исчисления высказываний; система обобщенных расщеплений; количество шагов и длина вывода; экспоненциальное ускорение.

Mathematical Problems of Computer Science 57, 56–64, 2022.
doi: 10.51408/1963-0087

UDC 519.7

Complete Caps in Affine Geometry $AG(n, 3)$

Karen I. Karapetyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: karen-karapetyan@iiap.sci.am

Abstract

We consider the problem of constructing complete caps in affine geometry $AG(n, 3)$ of dimension n over the field F_3 of order three. We will take the elements of F_3 to be 0, 1 and 2. A cap is a set of points, no three of which are collinear. Using the concept of P_n −set, we give two new methods for constructing complete caps in affine geometry $AG(n, 3)$. These methods lead to some new upper and lower bounds on the possible minimal and maximal cardinality of complete caps in affine geometry $AG(n, 3)$.

Keywords: Affine geometry, Projective geometry, Cap, Complete cap.
Article info: Received 28 February 2022; received in revised form 2 May 2022; accepted 16 May 2022.

1. Introduction

A cap in an affine geometry $AG(n, q)$ or in a projective geometry $PG(n, q)$ over a finite field F_q is a set of points no three of which are collinear. A cap is called complete when it cannot be extended to a large cap. The central problem in the theory of caps is to find the maximal and minimal sizes of caps in the affine geometry $AG(n, q)$ or in the projective geometry $PG(n, q)$. In this paper, $s_{n,q}$ and $s'_{n,q}$ denote the size of the largest caps in $AG(n, q)$ and $PG(n, q)$, respectively. Presently, only the following exact values are known: $s_{n,2} = s'_{n,2} = 2^n$, $s_{2,q} = s'_{2,q} = q + 1$ if q is odd, $s_{2,q} = s'_{2,q} = q + 2$ if q is even, and $s'_{3,q} = q^2 + 1, s_{3,q} = q^2$ [1, 2]. Aside from these general results, the precise values are known only in the following cases: $s_{4,3} = s'_{4,3} = 20$ [3], $s'_{5,3} = 56$ [4], $s_{5,3} = 45$ [5], $s'_{4,4} = 41$ [6], $s_{6,3} = 112$ [7]. In the other cases, only lower and upper bounds on the sizes of caps in $AG(n, q)$ and $PG(n, q)$ are known. Finding the exact value for $s_{n,q}$ and $s'_{n,q}$ in the general case seems to be a very hard problem [8–10]. The only complete cap in $AG(n, 2)$ is the whole $AG(n, 2)$. The trivial lower bound for the size of the smallest complete cap in $AG(n,q)$ is $\sqrt{2}q^{\frac{n-1}{2}}$. For even q there exist complete caps in geometry $AG(n,q)$ with less than $q^{\frac{n}{2}}$ points. But for odd q complete caps in $AG(n,q)$ with less than $q^{\frac{n}{2}}$ points are known to exist [11, 12] only for $n = 0(mod\,4)$, $n = 2(mod\,4)$. For more information about complete caps, for small values n and q, we refer the reader to [10–13]. Note that the problem of determining the minimum size of a complete cap in a given geometry is of particular interest in Coding theory. Using the concept of $a\ P_n$-set, which was introduced by the author in 2015 [14], we give two new methods for constructing complete caps in the affine geometry $AG(n,3)$. These methods yield some new upper and lower bounds on the possible minimal and maximal sizes of complete caps in the affine geometry $AG(n,3)$.

2. Main Results

We will write the points of $AG(n,q)$ in the following way: $\boldsymbol{x} = (x_1,\cdots,x_n)$, and let us denote by $\mathbf{0} = (0,\cdots,0)$ the origin point of the geometry $AG(n,3)$. It is easy to check that if $\boldsymbol{S}$ is a cap in $AG(n,3)$, then $\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma} \neq \mathbf{0}\ (mod\ 3)$ for every triple of distinct points $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma} \in \boldsymbol{S}$. Let's denote by $B_n = \{\boldsymbol{\alpha} = (\alpha_1,\cdots,\alpha_n) | \alpha_i = 1,2\}$ and by P_n the set of points of $AG(n,3)$ satisfying the following two conditions:

i) for any two distinct points $\boldsymbol{\alpha},\boldsymbol{\beta} \in P_n$, there exists $i\ (1 \leq i \leq n)$ such that $\alpha_i = \beta_i = 0$,

ii) for any triple of distinct points $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma} \in P_n$, $\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma} \neq \mathbf{0}(mod\ 3)$.

We say P_n to be complete when it cannot be extended to a larger one. We will define the concatenation of the points of the sets in the following way. Let $A \subset AG(n,3)$ and $B \subset AG(m,3)$. We form a new set $AB \subset AG(n+m,3)$ consisting of all points $\boldsymbol{\alpha} = (\alpha_1,\cdots,\alpha_n, \alpha_{n+1},\cdots,\alpha_{n+m})$, where $\boldsymbol{\alpha}^{(1)} = (\alpha_1,\cdots,\alpha_n) \in A$ and $\boldsymbol{\alpha}^{(2)} = (\alpha_{n+1},\cdots,\ \alpha_{n+m}) \in B$. In a similar way, one can define the concatenation of the points for any number of sets.

Claim 1. Note that if $x,y,z \in F_3$, then $x+y+z = 0\ (mod\ 3)$ if and only if $x=y=z$ or they are pairwise distinct numbers.
The following two theorems, which we need, are proven in [16, 17].

Theorem 1: *The following recurrence relation* $P_n = P_{n_1}P_{n_2}B_{n_3} \cup P_{n_1}B_{n_2}P_{n_3} \cup B_{n_1}P_{n_2}P_{n_3}$, *with initial sets* $P_1 = \{(0)\}$, $P_2 = \{(0,1),(0,2)\}$ *and* $n = \sum_{j=1}^{3} n_j$, *yields a complete* P_n *set.*

Having the sets $P_{n_1}, P_{n_2}, P_{n_3}, P_{n_4}, P_{n_5}, P_{n_6}$ and $B_{n_1}, B_{n_2}, B_{n_3}, B_{n_4}, B_{n_5}, B_{n_6}$, let us form the following ten sets, by concatenation of the points of the sets.

$$A_1 = P_{n_1}P_{n_2}B_{n_3}B_{n_4}B_{n_5}P_{n_6}, \qquad A_2 = B_{n_1}P_{n_2}P_{n_3}P_{n_4}B_{n_5}B_{n_6},$$
$$A_3 = P_{n_1}B_{n_2}P_{n_3}B_{n_4}P_{n_5}B_{n_6}, \qquad A_4 = B_{n_1}B_{n_2}P_{n_3}P_{n_4}B_{n_5}P_{n_6},$$
$$A_5 = B_{n_1}B_{n_2}P_{n_3}B_{n_4}P_{n_5}P_{n_6}, \qquad A_6 = B_{n_1}P_{n_2}B_{n_3}P_{n_4}P_{n_5}B_{n_6},$$
$$A_7 = B_{n_1}P_{n_2}B_{n_3}B_{n_4}P_{n_5}P_{n_6}, \qquad A_8 = P_{n_1}B_{n_2}B_{n_3}P_{n_4}P_{n_5}B_{n_6},$$
$$A_9 = P_{n_1}B_{n_2}B_{n_3}P_{n_4}B_{n_5}P_{n_6}, \qquad A_{10} = P_{n_1}P_{n_2}P_{n_3}B_{n_4}B_{n_5}B_{n_6}.$$

Theorem 2: *The following recurrence relation $P_n = \bigcup_{i=1}^{10} A_i$, with initial sets $P_1 = \{(0)\}$, $P_2 = \{(0,\ 1),(0,2)\}$ and $n = \sum_{i=1}^{6} n_i$ yields a complete P_n set.*

Claim 2. Note that from the construction of P_n in both theorems it follows that for every i ($1 \leq i \leq n$), if the point $\boldsymbol{p} = (p_1, \dots, p_i, \dots, p_n) \in P_n$ and $p_i \neq 0$, then, also, the point $\boldsymbol{p}' = (p_1, \dots, p_i^{-1}, \dots, p_n) \in P_n$, where p_i^{-1} is the additive inverse of p_i in the field F_3.

The following two main theorems without proofs were first presented at CSIT 2015 in a weak form [14], that they yield caps. But at CSIT 2017 they were presented with a strong conclusion that they yield complete caps [15]. In this paper, we give their complete proofs.

Theorem 3: *If P_n and P_m are constructed either by Theorem 1 or by Theorem 2, then for the given natural numbers n and m, the set $S = P_nB_m \cup B_nP_m$ is a complete cap in the geometry $AG(n+m,3)$.*

Proof. First of all we will prove that the set $S = P_nB_m \cup B_nP_m$ is a cap. Suppose, to the contrary, that S is not a cap. Then there is a triple of distinct points $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma} \in S$, such that $\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma} = \mathbf{0}(mod\ 3)$. Let's represent the points $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}$ as $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}$, $\boldsymbol{\beta} = \boldsymbol{\beta}^{(1)}\boldsymbol{\beta}^{(2)}$ and $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(1)}\boldsymbol{\gamma}^{(2)}$, respectively, where $\boldsymbol{\alpha}^{(1)} = (\alpha_1,\cdots,\alpha_n)$, $\boldsymbol{\alpha}^{(2)} = (\alpha_{n+1},\cdots,\alpha_{n+m})$, $\boldsymbol{\beta}^{(1)} = (\beta_1,\cdots,\beta_n)$, $\boldsymbol{\beta}^{(2)} = (\beta_{n+1},\cdots,\beta_{n+m})$, $\boldsymbol{\gamma}^{(1)} = (\gamma_1,\cdots,\gamma_n)$ and $\boldsymbol{\gamma}^{(2)} = (\gamma_{n+1},\cdots,\gamma_{n+m})$. Thus, we obtain $\boldsymbol{\alpha}^{(1)}+\boldsymbol{\beta}^{(1)}+\boldsymbol{\gamma}^{(1)} = \mathbf{0}(mod\ 3)$ and $\boldsymbol{\alpha}^{(2)}+\boldsymbol{\beta}^{(2)}+\boldsymbol{\gamma}^{(2)} = \mathbf{0}(mod\ 3)$. If all three points $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma} \in P_nB_m$, then it follows that $\boldsymbol{\alpha}^{(1)},\boldsymbol{\beta}^{(1)},\boldsymbol{\gamma}^{(1)} \in P_n$ and $\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(2)},\boldsymbol{\gamma}^{(2)} \in B_m$. The definition of the set P_n implies that $\boldsymbol{\alpha}^{(1)} = \boldsymbol{\beta}^{(1)} = \boldsymbol{\gamma}^{(1)}$ and Claim 1 implies that $\boldsymbol{\alpha}^{(2)} = \boldsymbol{\beta}^{(2)} = \boldsymbol{\gamma}^{(2)}$. Therefore, $\boldsymbol{\alpha} = \boldsymbol{\beta} = \boldsymbol{\gamma}$, which contradicts that $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are pairwise distinct points. In the same manner, one can prove the case, when all three points $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma} \in B_nP_m$, is impossible. Now let us assume that two of these points belong to one set (say $\boldsymbol{\alpha},\boldsymbol{\beta} \in P_nB_m$) and the third point $\boldsymbol{\gamma}$ belongs to the other set (say $\gamma \in B_nP_m$). By definition of P_n there is i, $1 \leq i \leq n$, so that $\alpha_i = \beta_i = 0$. But, by definition of B_n, $\gamma_i = 1\ or\ 2$. Hence, $\alpha_i + \beta_i + \gamma_i \neq 0(mod\ 3)$, which contradicts that $\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma} = \mathbf{0}(mod\ 3)$. In a similar way, one can prove the case when two points belong to B_nP_m and the third one belongs to P_nB_m is impossible. Therefore, S is a cap.
We will prove the completeness of S again by contradiction. Suppose that there is a point $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n, \alpha_{n+1}, \dots, \alpha_{n+m})$, such that $\boldsymbol{\alpha} \notin$ S and S $\cup \{\boldsymbol{\alpha}\}$ is a cap. Let's represent the point $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}$, where $\boldsymbol{\alpha}^{(1)} = (\alpha_1,\cdots,\alpha_n)$, $\boldsymbol{\alpha}^{(2)} = (\alpha_{n+1},\cdots,\alpha_{n+m})$. The following two cases are possible.

Case 1. At least one of the sets $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ or $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ satisfies the condition i). Assume that the set $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ satisfies the condition i). If $\boldsymbol{\alpha}^{(1)} \in P_n$, then we can choose two points $\boldsymbol{x}$, $\boldsymbol{y} \in B_m$ in the following way. If $\alpha_i = 0$, then we will assume that $x_i = 1$ and $y_i = 2$, otherwise $x_i = y_i = \alpha_i$, $n+1 \leq i \leq n+m$. Therefore, $\boldsymbol{\alpha}^{(2)} \notin B_m$, since $\boldsymbol{\alpha} \notin$ S and $\boldsymbol{\alpha}^{(1)} \in P_n$. Hence, $\boldsymbol{\alpha}^{(2)}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ are pairwise distinct points. It is not difficult to see that $\boldsymbol{\alpha}^{(1)}\boldsymbol{x}$, $\boldsymbol{\alpha}^{(1)}\boldsymbol{y} \in P_nB_m$. Claim 1 implies that $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)} + \boldsymbol{\alpha}^{(1)}\boldsymbol{x} + \boldsymbol{\alpha}^{(1)}\boldsymbol{y} = \mathbf{0}(mod\ 3)$, which contradicts the assumption that $\mathrm{S} \cup \{\boldsymbol{\alpha}\}$ is a cap. If $\boldsymbol{\alpha}^{(1)} \notin P_n$, then the completeness of the P_n implies that there are two distinct points $\boldsymbol{\beta}, \boldsymbol{\gamma} \in P_n$, such that $\boldsymbol{\alpha}^{(1)} + \boldsymbol{\beta} + \boldsymbol{\gamma} = \mathbf{0}(mod\ 3)$. Now, as described above, we will choose two points $\boldsymbol{x}, \boldsymbol{y} \in B_m$ in the following way. If $\alpha_i = 0$, then we will take $x_i = 1$ and $y_i = 2$, otherwise $x_i = y_i = \alpha_i$, $n + 1 \le i \le n + m$. The choice of the points $\boldsymbol{x}, \boldsymbol{y}$ implies that $\boldsymbol{x}, \boldsymbol{y} \in B_m$ and $\boldsymbol{\alpha}^{(2)} + \boldsymbol{x} + \boldsymbol{y} = \mathbf{0}(mod\ 3)$. Therefore, $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)} + \boldsymbol{\beta}\boldsymbol{x} + \boldsymbol{\gamma}\boldsymbol{y} = \mathbf{0}(mod 3)$, which contradicts the assumption that $\mathrm{S} \cup \{\boldsymbol{\alpha}\}$ is a cap. Similarly, one can prove the case, when the set $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ satisfies the condition i), is impossible.

Case 2. Both sets $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ and $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ do not satisfy the condition i). Therefore, the condition i) for the set $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ follows that there is a point $\boldsymbol{\beta} \in P_n$, such that if $\alpha_i = 0$, then $\beta_i \neq 0$ and if $\beta_i = 0$, then $\alpha_i \neq 0$, $1 \le i \le n$. We will choose the point $\boldsymbol{x} \in B_n$ in the following way. If $\alpha_i = 0$, then $x_i = \beta_i^{-1}$ and if $\beta_i = 0$, then $x_i = \alpha_i^{-1}$, otherwise, using Claim 2, we can assume that $x_i = \beta_i = \alpha_i$, $1 \le i \le n$. By the same reason, the condition i) for the set $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ implies that there is a point $\boldsymbol{\gamma} \in P_m$, so that if $\alpha_i = 0$, then $\gamma_i \neq 0$ and if $\gamma_i = 0$, then $\alpha_i \neq 0, n + 1 \le i \le n + m$. In the same manner, we will choose the point $\boldsymbol{y} \in B_m$. If $\alpha_i = 0$, then $y_i = \gamma_i^{-1}$ and if $\gamma_i = 0$, then $y_i = \alpha_i^{-1}$, otherwise, by Claim 2, we can assume that $y_i = \gamma_i = \alpha_i$, $n + 1 \le i \le n + m$). It is obvious that $\boldsymbol{\beta}\boldsymbol{y} \in P_nB_m$ and $\boldsymbol{x}\boldsymbol{\gamma} \in B_nP_m$. The choice of the points $\boldsymbol{x}, \boldsymbol{y}$ implies that $\boldsymbol{\alpha}^{(1)} + \boldsymbol{\beta} + \boldsymbol{x} = \mathbf{0}(mod\ 3)$ and $\boldsymbol{\alpha}^{(2)} + \boldsymbol{\gamma} + \boldsymbol{y} = \mathbf{0}(mod\ 3)$. Therefore, $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)} + \boldsymbol{\beta}\boldsymbol{y} + \boldsymbol{x}\boldsymbol{\gamma} = \mathbf{0}(mod\ 3)$, which again contradicts the assumption that $\mathrm{S} \cup \{\boldsymbol{\alpha}\}$ is a cap.

□

Corollary 1: *For the given natural numbers n and m,* $s_{n+m,3} \ge |P_n||B_m| + |B_n||P_m|$.

Corollary 2: *For every natural number n,* $s_{n+1,3} \ge 2|P_n| + |B_n|$.

Theorem 4: *If P_n and P_m are constructed by Theorem 1 or by Theorem 2, then for the given natural numbers n and m, $S = P_nP_m\{0\} \cup P_nB_m\{1\} \cup B_nP_m\{1\} \cup B_{n+m}\{2\}$ is a complete cap in the geometry $AG(n + m + 1, 3)$.*

Proof. First we will prove that the set $S = P_nP_m\{0\} \cup P_nB_m\{1\} + B_nP_m\{1\} + B_{n+m}\{2\}$ is a cap by contradiction. Assume that there are three distinct points $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n, \alpha_{n+1}, \dots, \alpha_{n+m}, \alpha_{n+m+1})$, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_n, \beta_{n+1}, \dots, \beta_{n+m}, \beta_{n+m+1})$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n, \gamma_{n+1}, \dots, \gamma_{n+m}, \gamma_{n+m+1}) \in S$, such that $\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} = \mathbf{0}(mod\ 3)$. Therefore, $\boldsymbol{\alpha}^{(1)} + \boldsymbol{\beta}^{(1)} + \boldsymbol{\gamma}^{(1)} = \mathbf{0}(mod\ 3)$, $\boldsymbol{\alpha}^{(2)} + \boldsymbol{\beta}^{(2)} + \boldsymbol{\gamma}^{(2)} = \mathbf{0}(mod\ 3)$ and $\alpha_{n+m+1} + \beta_{n+m+1} + \gamma_{n+m+1} = \mathbf{0}(mod\ 3)$, where $\boldsymbol{\alpha}^{(1)} = (\alpha_1, \cdots, \alpha_n)$, $\boldsymbol{\alpha}^{(2)} = (\alpha_{n+1}, \cdots, \alpha_{n+m})$, $\boldsymbol{\beta}^{(1)} = (\beta_1, \cdots, \beta_n)$, $\boldsymbol{\beta}^{(2)} = (\beta_{n+1}, \cdots, \beta_{n+m})$, $\boldsymbol{\gamma}^{(1)} = (\gamma_1, \cdots, \gamma_n)$ and $\boldsymbol{\gamma}^{(2)} = (\gamma_{n+1}, \cdots, \gamma_{n+m})$. Claim 1 implies that $\alpha_{n+m+1} = \beta_{n+m+1} = \gamma_{n+m+1}$ or α_{n+m+1}, β_{n+m+1}, and γ_{n+m+1} are pairwise distinct numbers. Hence, the following four cases are possible.

Case 1. $\alpha_{n+m+1} = \beta_{n+m+1} = \gamma_{n+m+1}= 0$. Therefore, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in P_nP_m\{0\}$, $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\beta}^{(1)}, \boldsymbol{\gamma}^{(1)} \in P_n$ and $\boldsymbol{\alpha}^{(2)}, \boldsymbol{\beta}^{(2)}, \boldsymbol{\gamma}^{(2)} \in P_m$. From the definition of P_n and P_m and the two relations $\boldsymbol{\alpha}^{(1)} + \boldsymbol{\beta}^{(1)} +$

$\boldsymbol{\gamma}^{(1)} = 0(mod\ 3)$, $\boldsymbol{\alpha}^{(2)} + \boldsymbol{\beta}^{(2)} + \boldsymbol{\gamma}^{(2)} = \mathbf{0}(mod\ 3)\ it$ follows that $\boldsymbol{\alpha}^{(1)} = \boldsymbol{\beta}^{(1)} = \boldsymbol{\gamma}^{(1)}$ and $\boldsymbol{\alpha}^{(2)} = \boldsymbol{\beta}^{(2)} = \boldsymbol{\gamma}^{(2)}$. Hence, $\boldsymbol{\alpha} = \boldsymbol{\beta} = \boldsymbol{\gamma}$, which contradicts the assumption that $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are pairwise distinct points.

Case 2. $\alpha_{n+m+1} = \beta_{n+m+1} = \gamma_{n+m+1}$=1. Assume that $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in P_n B_m\{1\}$. Then $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\beta}^{(1)}, \boldsymbol{\gamma}^{(1)} \in P_n$ and $\boldsymbol{\alpha}^{(2)}, \boldsymbol{\beta}^{(2)}, \boldsymbol{\gamma}^{(2)} \in B_m$. The definition of P_n implies that $\boldsymbol{\alpha}^{(1)} = \boldsymbol{\beta}^{(1)} = \boldsymbol{\gamma}^{(1)}$**,** since $\boldsymbol{\alpha}^{(1)} + \boldsymbol{\beta}^{(1)} + \boldsymbol{\gamma}^{(1)} = \mathbf{0}(mod\ 3)$**.** Because $\boldsymbol{\alpha}^{(2)} + \boldsymbol{\beta}^{(2)} + \boldsymbol{\gamma}^{(2)} = \mathbf{0}(mod\ 3)$, Claim 1 implies that $\boldsymbol{\alpha}^{(2)} = \boldsymbol{\beta}^{(2)} = \boldsymbol{\gamma}^{(2)}$**.** Therefore, $\boldsymbol{\alpha} = \boldsymbol{\beta} = \boldsymbol{\gamma}$**,** which, again contradicts the assumption that $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are pairwise distinct points. Similarly, one can prove that the case is impossible, when $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in B_n P_m\{1\}$. Therefore, two points, say $\boldsymbol{\alpha}, \boldsymbol{\beta} \in P_n B_m\{1\}$ and $\boldsymbol{\gamma} \in B_n P_m\{1\}$. The definition of P_n implies that there is i, such that $\alpha_i = \beta_i = 0,\ 1 \le i \le n,$. But by the definition of B_n, $\gamma_i = 1\ or\ 2$. Hence, $\alpha_i + \beta_i + \gamma_i \neq 0(mod\ 3)$, which contradicts that $\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} = \mathbf{0}(mod\ 3)$. In a similar manner, one can prove that the case is impossible, when two points from $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ belong to $B_n P_m$ and the third one belongs to $P_n B_m$. Therefore, S is a cap.

Case 3. $\alpha_{n+m+1} = \beta_{n+m+1} = \gamma_{n+m+1} = 2$. Therefore $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in B_{n+m}\{2\}$. Hence, $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}$, $\boldsymbol{\beta}^{(1)}\boldsymbol{\beta}^{(2)}, \boldsymbol{\gamma}^{(1)}\boldsymbol{\gamma}^{(2)} \in B_{n+m}$ and $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)} + \boldsymbol{\beta}^{(1)}\boldsymbol{\beta}^{(2)} + \boldsymbol{\gamma}^{(1)}\boldsymbol{\gamma}^{(2)} = \mathbf{0}(mod\ 3)$. Claim 1 implies that $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)} = \boldsymbol{\beta}^{(1)}\boldsymbol{\beta}^{(2)} = \boldsymbol{\gamma}^{(1)}\boldsymbol{\gamma}^{(2)}$. This yields $\boldsymbol{\alpha} = \boldsymbol{\beta} = \boldsymbol{\gamma}$**,** which, again contradicts the assumption that $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are pairwise distinct points.

Case α_{n+m+1}, β_{n+m+1} and γ_{n+m+1} are pairwise distinct numbers. Without loss of generality, let us assume that $\alpha_{n+m+1} = 0$, $\beta_{n+m+1} = 1$ and $\gamma_{n+m+1} = 2$. Therefore, $\boldsymbol{\alpha} \in P_n P_m\{0\}, \boldsymbol{\beta} \in P_n B_m\{1\}$ or $\boldsymbol{\beta} \in B_n P_m\{1\}$ and $\boldsymbol{\gamma} \in B_{n+m}\{2\}$. If $\boldsymbol{\beta} \in P_n B_m\{1\}$, then $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\beta}^{(1)} \in P_n$. Hence, the definition of P_n implies that there is i, such that $\alpha_i = \beta_i = 0,\ 1 \le i \le n$. But, by the definition of $B_n, \gamma_i = 1$ or 2. Therefore, $\alpha_i + \beta_i + \gamma_i \neq 0(mod\ 3)$, which contradicts that $\boldsymbol{\alpha}^{(1)} + \boldsymbol{\beta}^{(1)} + \boldsymbol{\gamma}^{(1)} = \mathbf{0}(mod\ 3)$. The last relation, in turn, implies that $\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} \neq \mathbf{0}(mod\ 3)$. In a similar manner, one can prove the case when $\boldsymbol{\beta} \in B_n P_m\{1\}$ is impossible. Hence, S is a cap.

Now we will prove the completeness of S also by contradiction. Let us assume that there is a point $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n, \alpha_{n+1}, \dots, \alpha_{n+m}, \alpha_{n+m+1})$, such that $\boldsymbol{\alpha} \notin S$ and $S \cup \{\boldsymbol{\alpha}\}$ is a cap. The following three cases are possible.

Case $\alpha_{n+m+1} = 2$. Since $\boldsymbol{\alpha} \notin S$, we have $(\alpha_1, \dots, \alpha_n, \alpha_{n+1}, \dots, \alpha_{n+m}) \notin B_{n+m}$. We can choose two points $\boldsymbol{x}$**,** $\boldsymbol{y} \in B_{n+m}\{2\}$, such that, if $\alpha_i = 0$ then $x_i = 2$ and $y_i = 1$, otherwise $x_i = y_i = \alpha_i$, $1 \le i \le n + m$. It is obvious that $\boldsymbol{x}\{2\}$**,** $\boldsymbol{y}\{2\} \in B_{n+m}\{2\}$ and $\boldsymbol{\alpha}, \boldsymbol{x}\{2\}$, $\boldsymbol{y}\{2\}$ are pairwise distinct points. Claim 1 implies that $\boldsymbol{x}\{2\} + \boldsymbol{y}\{2\} + \boldsymbol{\alpha} = \mathbf{0}(mod\ 3)$, which contradicts the assumption that $S \cup \{\boldsymbol{\alpha}\}$ is a cap.

Case $\alpha_{n+m+1} = 1$. Let's represent the point $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}\{1\}$**,** where $\boldsymbol{\alpha}^{(1)} = (\alpha_1, \cdots, \alpha_n)$ and $\boldsymbol{\alpha}^{(2)} = (\alpha_{n+1}, \cdots, \alpha_{n+m})$. Assume that at least one of the sets $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ or $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ satisfies the condition i), say $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$**.** First, suppose that $\boldsymbol{\alpha}^{(1)} \notin P_n$. Then the completeness of the set P_n follows that there are two points $\boldsymbol{\beta}$**,** $\boldsymbol{\gamma} \in P_n$, such that $\boldsymbol{\beta} + \boldsymbol{\gamma} + \boldsymbol{\alpha}^{(1)} = \mathbf{0}(mod\ 3)$**.** We will choose two points $\boldsymbol{x}$**,** $\boldsymbol{y} \in B_m$ in the following way. If $\alpha_i = 0$, then $x_i = 1$ and $y_i = 2$, otherwise $x_i = y_i = \alpha_i$, $n+1 \le i \le n+m$. From the choice of the points $\boldsymbol{x}, \boldsymbol{y}$ it follows that $\boldsymbol{x}, \boldsymbol{y} \in B_m$ and $\boldsymbol{\alpha}^{(2)} + \boldsymbol{x} + \boldsymbol{y} = \boldsymbol{0}(mod\ 3)$. Therefore, $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}\{1\} + \boldsymbol{\beta x}\{1\} + \boldsymbol{\gamma y}\{1\} = \boldsymbol{0}(mod\ 3)$, which contradicts the assumption that $S \cup \{\boldsymbol{\alpha}\}$ is a cap. Otherwise, if $\boldsymbol{\alpha}^{(1)} \in P_n$, then $\boldsymbol{\alpha}^{(2)} \notin B_m$, because $\boldsymbol{\alpha} \notin S$. Then it is easy to see that $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}\{1\} + \boldsymbol{\alpha}^{(1)}\boldsymbol{x}\{1\} + \boldsymbol{\alpha}^{(1)}\boldsymbol{y}\{1\} = \boldsymbol{0}(mod\ 3)$, which, again contradicts the assumption that $S \cup \{\boldsymbol{\alpha}\}$ is a cap. Similarly, one can prove the case, when the set $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ satisfies the condition i) is impossible. Therefore, both sets $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ and $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ do not satisfy the condition i). Hence, there is a point $\boldsymbol{\beta} \in P_n$, (respectively, $\boldsymbol{\gamma} \in P_m$), such that if $\alpha_i = 0$, then $\beta_i \ne 0$ and if $\beta_i = 0$, then $\alpha_i \ne 0, 1 \le i \le n$ (respectively, if $\alpha_i = 0$, then $\gamma_i \ne 0$ and if $\gamma_i = 0$, then $\alpha_i \ne 0, n+1 \le i \le n+m$). First, let's choose the point $\boldsymbol{x} \in B_n$ in the following way. If $\alpha_i = 0$, then $x_i = \beta_i^{-1}$ and if $\beta_i = 0$, then $x_i = \alpha_i^{-1}$, otherwise, by Claim 2, we can assume that $x_i = \beta_i = \alpha_i$, $1 \le i \le n$. In the same manner, we will choose the point $\boldsymbol{y} \in B_m$. If $\alpha_i = 0$, then $y_i = \gamma_i^{-1}$ and if $\gamma_i = 0$, then $y_i = \alpha_i^{-1}$, otherwise, using Claim 2, we can assume that $y_i = \gamma_i = \alpha_i$, $n+1 \le i \le n+m$). The choice of the points $\boldsymbol{x}$ and $\boldsymbol{y}$ implies that $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}\{1\} + \boldsymbol{\beta y}\{1\} + \boldsymbol{x\gamma}\{1\} = \boldsymbol{0}(mod\ 3)$, which again contradicts the assumption that $S \cup \{\boldsymbol{\alpha}\}$ is a cap.

Case $\alpha_{n+m+1} = 0$. Assume that at least one of the sets $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ or $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ does not satisfy the condition i), say the set $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$**.** Therefore, the condition i) implies that there is a point $\boldsymbol{\beta} \in P_n$, such that, if $\alpha_i = 0$, then $\beta_i \ne 0$ and if $\beta_i = 0$, then $\alpha_i \ne 0, 1 \le i \le n$. We will choose the points $\boldsymbol{z}^{(1)} \in B_n$ and $\boldsymbol{z}^{(2)}$, $\boldsymbol{y} \in B_m$ in the following way. First let's choose $\boldsymbol{z}^{(1)}$**.** If $\alpha_i = 0$, then $z_i = \beta_i^{-1}$ and if $\beta_i = 0$, then $z_i = \alpha_i^{-1}$, otherwise, using Claim 2, we will assume that $z_i = \beta_i = \alpha_i$, $1 \le i \le n$. Now we will choose the points $\boldsymbol{z}^{(2)}$, $\boldsymbol{y} \in B_m$ in the following way. If $\alpha_i = 0$, then we will assume that $z_i = 1$ and $y_i = 2$, otherwise $z_i = y_i = \alpha_i$, $n+1 \le i \le n+m$. It is easy to see that $\boldsymbol{\beta y}\{1\} \in P_n B_m\{1\}, \boldsymbol{z}^{(1)}\boldsymbol{z}^{(2)}\{2\} \in B_{n+m}\{2\}$. The choice of the points $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}$ and $\boldsymbol{y}$ imply that $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}\{0\} + \boldsymbol{\beta y}\{1\} + \boldsymbol{z}^{(1)}\boldsymbol{z}^{(2)}\{2\} = \boldsymbol{0}(mod\ 3)$, which contradicts the assumption that $S \cup \{\boldsymbol{\alpha}\}$ is a cap. Similarly, one can prove the case is impossible, when the set $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ does not satisfy the condition i). Therefore, both sets $P_n \cup \{\boldsymbol{\alpha}^{(1)}\}$ and $P_m \cup \{\boldsymbol{\alpha}^{(2)}\}$ are satisfying the condition i). Since $\boldsymbol{\alpha} \notin S$, therefore either $\boldsymbol{\alpha}^{(1)} \notin P_n$ or $\boldsymbol{\alpha}^{(2)} \notin P_m$. If $\boldsymbol{\alpha}^{(1)} \notin P_n$ and $\boldsymbol{\alpha}^{(2)} \in P_m$, then the completeness of P_n follows that there are two points $\boldsymbol{x}, \boldsymbol{y} \in P_n$, so that $\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{\alpha}^{(1)} = \boldsymbol{0}(mod\ 3)$. Since $\boldsymbol{x}, \boldsymbol{y} \in P_n$ and $\boldsymbol{\alpha}^{(2)} \in P_m$, we have $\boldsymbol{x\alpha}^{(2)}$, $\boldsymbol{y\alpha}^{(2)} \in P_n P_m$ and $\boldsymbol{x\alpha}^{(2)}\{0\} + \boldsymbol{y\alpha}^{(2)}\{0\} + \boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}\{0\} = \boldsymbol{0}(mod\ 3)$, which contradicts the assumption that $S \cup \{\boldsymbol{\alpha}\}$ is a cap. The case, when $\boldsymbol{\alpha}^{(2)} \notin P_m$ and $\boldsymbol{\alpha}^{(1)} \in P_n$ is analogous to the above described one and therefore is impossible. Hence, $\boldsymbol{\alpha}^{(1)} \notin P_n$ and $\boldsymbol{\alpha}^{(2)} \notin P_m$. Therefore, from the completeness of P_n and P_m it follows that there are points $\boldsymbol{\beta}, \boldsymbol{\gamma} \in P_n$ and $\boldsymbol{\delta}, \boldsymbol{\theta} \in P_m$, so that $\boldsymbol{\beta} + \boldsymbol{\gamma} + \boldsymbol{\alpha}^{(1)} = \boldsymbol{0}(mod\ 3)$ and $\boldsymbol{\delta} + \boldsymbol{\theta} + \boldsymbol{\alpha}^{(2)} = \boldsymbol{0}(mod\ 3)$. The last two relations imply that $\boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)}\{0\} + \boldsymbol{\beta\delta}\{0\} + \boldsymbol{\gamma\theta}\{0\} = \boldsymbol{0}(mod\ 3)$, which contradicts the assumption that $S \cup \{\boldsymbol{\alpha}\}$ is a cap.

□

Corollary 3: *For the given natural numbers n and m*, $s_{n+m+1,3} \ge |P_n||P_m| + |P_n||B_m| + |B_n||P_m| + |B_{n+m}|$.

Corollary 4: $s_{5,3} \geq 42$.

Proof. By definition P_1={(0)}. From Theorem 1 it follows that $P_3 = P_{1+1+1} = P_1P_1B_1 \cup P_1B_1P_1 \cup B_1P_1P_1$ $=\{(0,0,1),(0,0,2),(0,1,0),(0,2,0),(1,0,0),(2,0,0)\}$. It is easy to see that $|B_n| = 2^n$. Therefore, $s_{5,3} \geq |P_3||P_1| + |P_3||B_1| + |B_3||P_1| + |B_4|$ $=6 \times 1 + 6 \times 2 + 8 \times 1 + 16 = 42$.

□

3. Conclusion

Notice that the cardinality of P_n obtained by Theorem 1 (Theorem 2) [16, 17], essentially depends on the representation of n as the sum of three (six) natural numbers. Presenting the natural numbers as the sum of six natural numbers and applying Theorem 2, for some $n \geq 6$ in some cases, one can obtain larger complete P_n sets than those, which are constructed by Theorem 1. It is easy to check that $|P_1| = 1$, $|P_2| = 2$, and $|P_{1+1+1}| = 6$. $|P_{2+1+1}| = 12$, $|P_{3+1+1}| = 32$, $|P_{1+1+1+1+1+1}| = 80$, $|P_7| = |P_{3+3+1}| = 168$, $|P_8| = |P_{1+1+1+1+1+3}| = 400$, $|P_9| = |P_{3+3+3}| = 864$… It is not difficult to see that the maximal size $|P_n| > 2^n$, if $n > 5$. Therefore, to construct large complete caps it is convenient to use Corollary 2, but for small complete caps one can use Theorem 4.

References

[1] R. C. Bose, "Mathematical theory of the symmetrical factorial design", *Sankhya*, vol. 8, pp. 107-166, 1947.

[2] B. Qvist, "Some remarks concerning curves of the second degree in a finite plane", *Ann Acad. Sci. Fenn*, Ser. A, vol. 134, p. 27. 1952.

[3] G. Pellegrino, "Sul Massimo ordine delle calotte in $S_{4,3}$", *Matematiche* (Catania), vol. 25, pp. 1-9, 1970.

[4] R. Hill, "On the largest size of cap in $S_{5,3}$", *Atti Accad Naz.Lincei Rendicondi*, vol. 54, pp. 378-384, 1973.

[5] Y. Edel, S. Ferret, I. Landjev and L. Storme, "The classification of the largest caps in $AG(5,3)$", *Journal of Combinatorial Theory*, ser. A, vol. 99, pp. 95-110, 2002.

[6] Y. Edel and J. Bierbrauer, "41 is the largest size of a cap in $PG(n,3)$", *Designs, Codes and Cryptography*, vol. 16, pp. 151-160, 1999.

[7] A. Potechin, "Maximal caps in $AG(6,3)$", *Designs, Codes and Cryptography*, vol. 46, pp. 243-259, 2008.

[8] J.W. Hirschfeld and L. Storme, ''The packing problem in statistics, coding theory and finite projective spaces'', *Journal of Statistical Planning and Inference* 72, pp. 355-380, 1998.

[9] J.W. Hirschfeld and L. Storme, "The packing problem in statistics, coding theory and finite projective spaces'', *Proceeding of the Fourth Isle of Thorns Conference*, pp. 201-246, July 16-21, 2000.

[10] J. Bierbrauer and Y. Edel, “Large caps in projective Galois spaces”, In: Current topics in Galois geometry, Editors J. De Beule and L.Storm, pp. 87-104, 2012.

[11] A. A. Davidov, G. Faina, S. Marcugini and F. Pambianco, “Computer search in projective planes for the sizes of complete arcs”, *J. Geometry*, vol. 82, pp. 50-62, 2005.

[12] A. A. Davidov and P. R. J. Ostergard, “Recursive constructions of complete caps”, *J. Statist. Planning Infer*, vol. 95, pp. 167-173, 2001.

[13] M. Geuletti, “Small complete caps in Galois affine spaces”, *J. Algebr. Comb.* Vol. 25, pp.149-168, 2007.

[14] K. Karapetyan, “Large Caps in Affine Space”, *Proceedings of International Conference Computer Science and Information Technologies*, Yerevan, Armenia, pp. 82-83, 2015.

[15] K. Karapetyan, “On the complete caps in Galois affine space $AG(n,3)$”, *Proceedings of International Conference Computer Science and Information Technologies*, Yerevan, Armenia, p. 205, 2017.

[16] I.A. Karapetyan and K.I. Karapetyan. “The Complete Caps in Projective Geometry $PG(n,3)$”, *«Լրաբեր» գիտական հոդվածների ժողովածու (ՀԱՊՀ*), հատոր 1, էջեր 35-44, 2021.

[17] I. Karapetyan and K. Karapetyan, “Complete Caps in Projective Geometry $PG(n,3)$”, *Proceedings of International Conference Computer Science and Information Technologies*, Yerevan, Armenia, pp. 57-60, 2021.

Լրիվ գլխարկներ $AG(n,3)$ աֆինական երկրաչափությունում

Կարեն Ի. Կարապետյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ
e-mail: karen-karapetyan@iiap.sci.am

Ամփոփում

Դիտարկվում է n չափանի $AG(n,3)$ աֆինական երկրաչափությունում լրիվ գլխարկների կառուցման խնդիրը $F_3 = \{0,1,2\}$ դաշտի վրա: Գլխարկը այն կետերի բազմությունն է, որոնցից ոչ մի երեքը համագիծ չեն: Օգտագործելով P_n բազմության հասկացությունը, մշակվել են լրիվ գլխարկների կառուցման երկու նոր մեթոդներ:

Բանալի բառեր՝ աֆինական երկրաչափություն, պրոյեկտիվ երկրաչափություն, կետեր, գլխարկներ, լրիվ գլխարկներ:

Полные шапки в аффинной геометрии $AG(n,3)$

Карен И. Карапетян

Институт проблем информатики и автоматизации НАН РА
e-mail: karen-karapetyan@iiap.sci.am

Аннотация

Рассматривается задача построения полных шапок в аффинной геометрии $AG(n,3)$ размерности n над полем $F_3 = \{0,1,2\}$. Шапка — это набор точек, никакие три из которых не коллинеарны. С помощью понятия множества P_n, разработаны две новые конструкции построения полных шапок.

Ключевые слова: аффинная геометрия, проективная геометрия, точки, шапки, полные шапки.

Կանոններ հեղինակների համար

ՀՀ ԳԱԱ ԻԱՊԻ “Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ” պարբերականը տպագրվում է 1963 թվականից: Պարբերականում հրատարակվում են նշված ոլորտին առնչվող գիտական հոդվածներ, որոնք պարունակում են նոր` չհրատարակված արդյունքներ:

Հոդվածները ներկայացվում են անգլերեն՝ ձևավորված համապատասխան “ոճով” (style): Հոդվածի ձևավորման պահանջներին ավելի մանրամասն կարելի է ծանոթանալ պարբերականի կայքէջում՝ http://mpcs.sci.am/:

Rules for authors

The periodical “Mathematical Problems of Computer Science” of IIAP NAS RA has been published since 1963. Scientific articles related to the noted fields with novel and previously unpublished results are published in the periodical.

Papers should be submitted in English and prepared in the appropriate style. For more information, please visit the periodical's website at http://mpcs.sci.am/.

Правила для авторов

Журнал «Математические проблемы компьютерных наук» ИПИА НАН РА издается с 1963 года. В журнале публикуются научные статьи в указанной области, содержащие новые и ранее не опубликованные результаты.

Статьи представляются на английском языке и оформляются в соответствующем стиле. Дополнительную информацию можно получить на веб-сайте журнала: http://mpcs.sci.am/.

The electronic version of the periodical “Mathematical Problems of Computer Science” and rules for authors are available at

http://mpcs.sci.am/

Phone:	(+37460) 62-35-51
Fax:	(+37410) 28-20-50
E-mail:	mpcs@sci.am
Website:	http://mpcs.sci.am/

Ստորագրված է տպագրության՝ 27.05.2022
Թուղթը՝ օֆսեթ։
Հրատարակված է ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման
պրոբլեմների ինստիտուտի կողմից
Ծավալը՝ 66 էջ։ Տպաքանակը՝ 100
ՀՀ ԳԱԱ ԻԱՊԻ Համակարգչային պոլիգրաֆիայի լաբորատորիա
Երևան, Պ. Սևակի 1
Հեռ. +(374 60) 623553
Գինը՝ անվճար

Подписано в печать 27.05.2022
Офсетная бумага.
Опубликовано Институтом проблем
информатики и автоматизации НАН РА
Объём: 66 страниц. Тираж: 100
Лаборатория компьютерной
полиграфии ИПИА НАН РА.
Ереван, П. Севака 1
Тел.: +(374 60) 623553
Цена: бесплатно

Signed in print 27.05.2022
Offset paper
Published by Institute for Informatics
and Automation Problems of NAS RA
Volume: 66 pages
Circulation: 100
Computer Printing Lab
of IIAP NAS RA
Yerevan, 1, P. Sevak str.
Phone: +(374 60) 623553
Free of charge