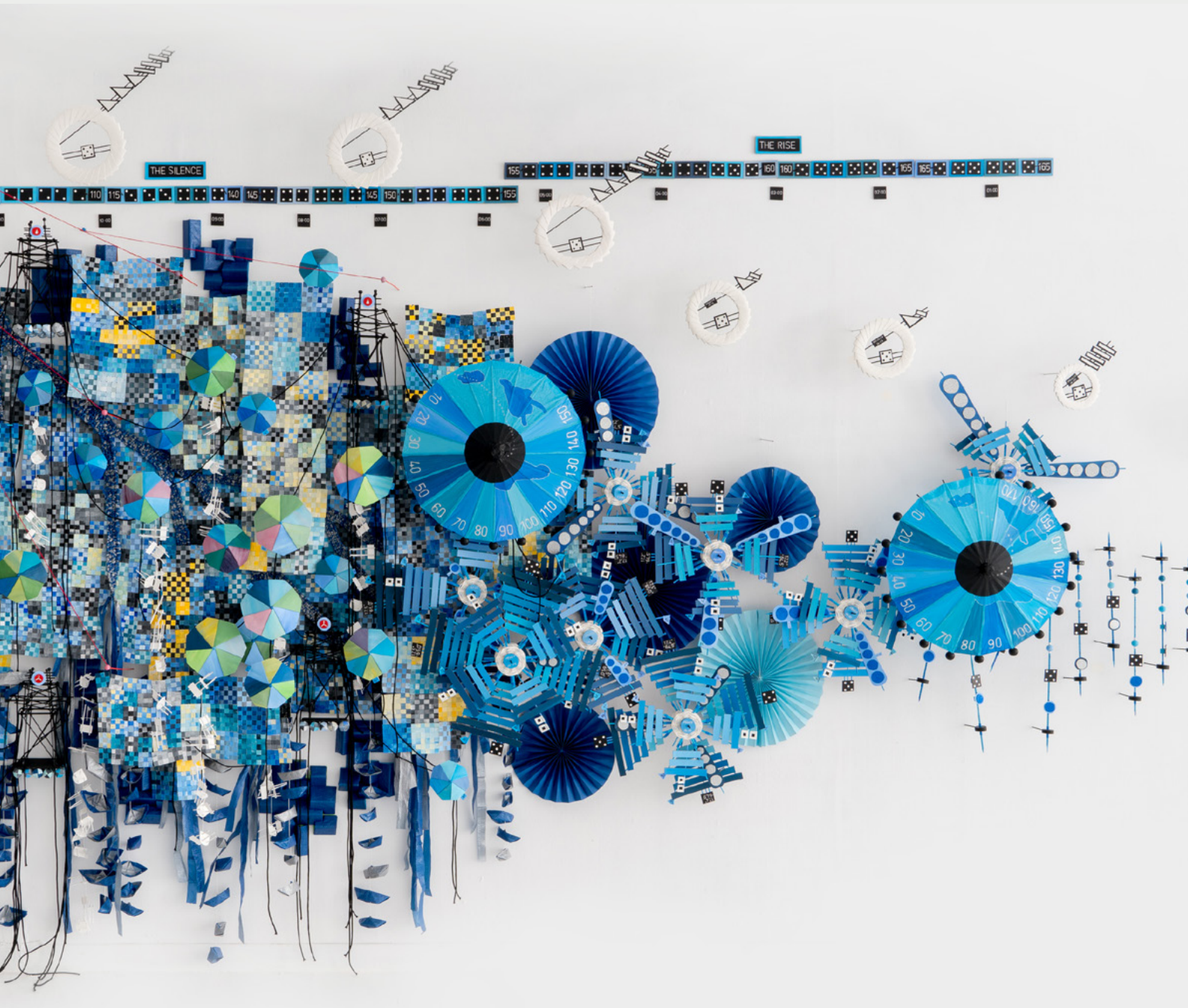
STUDBOOK

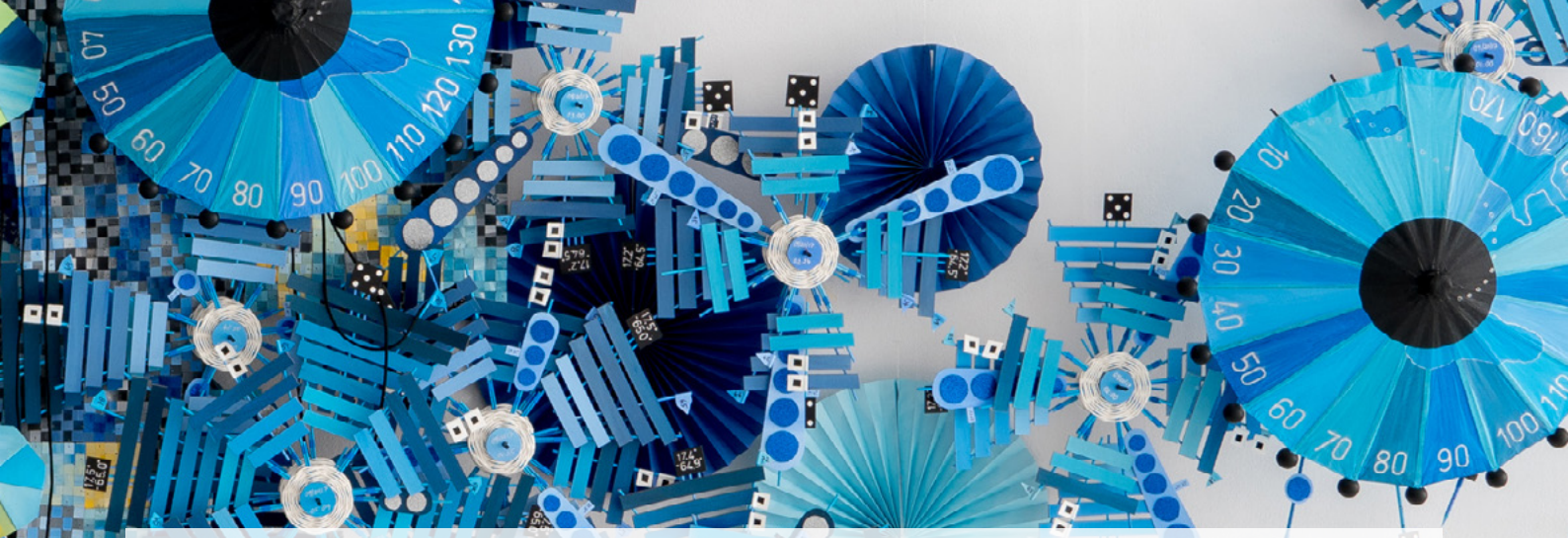
ΕΠΙΛΕΓΜΕΝΕΣ ΠΤΥΧΙΑΚΕΣ ΚΑΙ ΔΙΠΛΩΜΑΤΙΚΕΣ ΕΡΓΑΣΙΕΣ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

ΤΟΜΟΣ 17
2020



Εκδίδεται μία φορά το χρόνο από το:

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών,
Πανεπιστημιούπολη, 15784 Αθήνα

Επιμέλεια έκδοσης:

Επιτροπή Ερευνητικών και Αναπτυξιακών Δραστηριοτήτων

Θ. Θεοχάρης (υπεύθυνος έκδοσης), Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Η. Μανωλάκος, Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Γραφιστική επιμέλεια - Επιμέλεια κειμένων:

Λ. Χαλάτση, ΕΤΕΠ, Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ISSN 1792-8826

Εξώφυλλο: **Nathalie Miebach - The Burden of Every Drop**, 17’x10’x2’, 2018, Wood, Paper, Rope, Data
Photo Credit: Jean-Michael Seminaro

This is a story about Hurricane Maria – about the fierceness of the wind and rain, about the data silence as all electrical systems broke down, about the vastly underestimates death toll, about rebuilding and about people leaving the Puerto Rico. The piece combines weather and other numerical data with anecdotal information from news reports about the aftermath of the storm. This piece was written in conjunction to a musical score that uses the same base material as the wall piece. Read from right to left, it begins with lots of wind data, which comes to crescendo as it hits Puerto Rico, represented by an unraveling quilt.

website: https://nathaliemiebach.com, instagram: @miebachsculpture

Copyright © 2020, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Περιεχόμενα

Πρόλογος

Ο τόμος αυτός περιλαμβάνει περιλήψεις επιλεγμένων διπλωματικών και πτυχιακών εργασιών που εκπονήθηκαν στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών κατά το διάστημα **01/01/2019 - 31/12/2019**. Πρόκειται για τον 17^{ο} τόμο στη σειρά αυτή. Στόχος του θεσμού είναι η ενθάρρυνση της δημιουργικής προσπάθειας και η προβολή των πρωτότυπων εργασιών των φοιτητών του Τμήματος.

Η έκδοση αυτή είναι ψηφιακή, έχει δικό της ISSN και αναρτάται στην επίσημη ιστοσελίδα του Τμήματος ώστε να έχει μεγάλη προσβασιμότητα. Για το στόχο αυτό, σημαντική ήταν η συμβολή της Λήδας Χαλάτση που επιμελήθηκε και φέτος την ψηφιακή έκδοση και πέτυχε μια ελκυστική ποιότητα παρουσίασης, ενώ βελτίωσε και την ομοιογένεια των κειμένων.

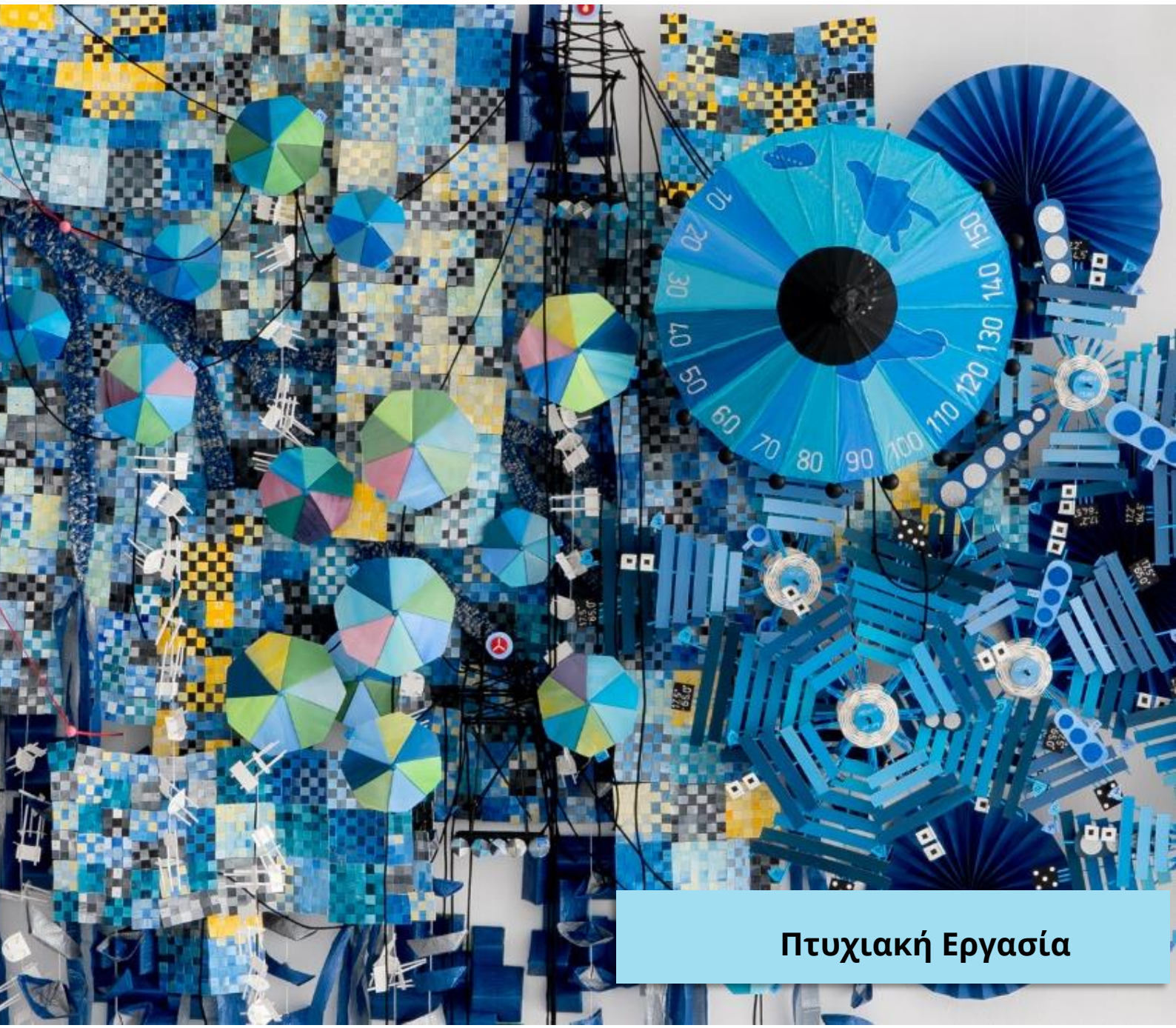
Η στάθμη των επιλεγμένων εργασιών είναι υψηλή και κάποιες από αυτές έχουν είτε δημοσιευθεί είτε υποβληθεί για δημοσίευση.

Θα θέλαμε να ευχαριστήσουμε τους φοιτητές για το χρόνο που αφιέρωσαν για να παρουσιάσουν τη δουλειά τους στα πλαίσια αυτού του θεσμού και να τους συγχαρούμε για την ποιότητα των εργασιών τους. Ελπίζουμε η διαδικασία αυτή να προσέφερε και στους ίδιους μια εμπειρία που θα τους βοηθήσει στη συνέχεια των σπουδών τους ή της επαγγελματικής τους σταδιοδρομίας.

Η Επιτροπή Ερευνητικών και Αναπτυξιακών Δραστηριοτήτων

Θ. Θεοχάρης (υπεύθυνος έκδοσης), Η. Μανωλάκος

Αθήνα, Ιούλιος 2020



Πτυχιακή Εργασία

Study of Probabilistic Topic Representations for the Classification of Genomic Elements

Nikolaos P. Gialitsis

ABSTRACT

In this study, we utilize probabilistic topic modeling and machine learning methods to develop an efficient pipeline for the representation and classification of DNA sequences ab-initio. By treating genomic sequences as natural language text, we infer their underlying topics which are formed by patterns in their sub-sequences. Subsequently, the genomic sequences are projected into a lower-dimensional space according to their topic compositions. A preliminary investigation was conducted via a wide experimental evaluation on a diverse dataset containing different types of genomic sequences originating from key-organisms. The results suggest that topic representations can prove beneficial to the classification process as they outperformed preceding methods and have been neglected in the literature.

Subject Area: Machine Learning

Keywords: genomics, topic-model, representation, bioinformatics, natural-language-processing

ADVISORS

George Giannakopoulos, Research Fellow (N.C.S.R. “Demokritos”),
Panagiotis Stamatopoulos, Assistant Professor (NKUA)

1 INTRODUCTION

DNA can be thought of as a manual containing the genetic instructions responsible for all functions in living organisms. A single strand of DNA is a biomolecule consisting of many linked smaller components, called nucleotides that come in four forms and are depicted with the letters A,T,C and G. Each DNA strand is typically represented as a character string containing combinations of these four letters. [2]

In this work, we examined topic-based representations for genomic sequences, which employ statistical methods widely used in the field of Natural Language Processing. We treated DNA as a natural language, by forming words, topics and documents, and we used the information stored in these topics to represent the DNA sequences.

We evaluated the effectiveness of our pipeline in 26 distinct binary classification experiments. Intuitively, when a classification is "harder", it means that the classes are more similar to each other. In evolutionary biology, the similarity between sequences usually reflects their evolutionary distance between the organisms. If two organisms' genomes are more similar to each other the organisms will be closer related than two organisms' genomic sequences which are divergent. As follows, by measuring the classification performance for each pair of organisms' genomes, it is theoretically possible to construct the phylogenetic history of the organisms. Thus, apart from performing the classification experiments and scoring high F-measures, we are also interested in understanding the biological driving forces behind the sequences. Even when we are only interested in one type of sequence, such as the conserved non-coding elements, there is value in experimenting on other types of sequences in order to develop a broader understanding of the problem.

2 RELATED WORK

2.1 Probabilistic Topic Modeling

Topics can be viewed as clusters of words that refer to a particular portion of reality. A document can refer to one or more distinct topics, which humans can often easily distinguish. For example, the words "fishing", "boat", "waves", have something in common, they are all affiliated with the sea. We can think of “sea” as one topic, which contains these three words. However, topics are not always unequivocally defined and there could exist a spectrum of broader or narrower topics within a document.

2.2 Latent Dirichlet Allocation (LDA)

Topic Models can infer topics by observing the distribution of words across documents. This can be accomplished with Latent Dirichlet Allocation (LDA) [70,5,4] , a generative statistical model that makes the hypothesis that there exists an underlying distribution of words, topics and documents, which generated the input text collection. Using probabilistic Topic Model jargon, the words of a document are called "observed variables", whereas the variables of the topic structure are called "hidden variables". Using an iterative process, the model estimates the posterior distribution of the hidden variables given the observed variables. However, the vast amount of topic structures that can exist result in exponential complexities of computation. For this reason, sampling-based algorithms have been developed, such as Gibbs sampling (introduced below). LDA is an instance of a more general class of models called grade of membership or mixed-membership models [15], because each document can be composed by multiple topics but the topics are shared between documents.

2.3 Gibbs Sampling

In Gibbs sampling [70], a Markov chain (i.e., a sequence of random variables, each only dependent on the previous) is constructed, using samples from the distribution of hidden variables. The assignment of words to topics is sampled iteratively until the Markov chain converges to the target distribution. In the

beginning of this procedure, each word is randomly assigned to a topic and in each subsequent iteration, the word-topic assignments are re-evaluated, which might result in words passing through multiple topics during the process.

2.4 Genomic Signatures (GS)

The idea of a 'Genomic Signature' is not a new idea [32]. GS is based on the observation that the dinucleotide odds ratios (e.g. 'AT' ratio) tend to be the same among organisms of the same species, and closely related organisms have substantial more similar dinucleotide ratios than those distantly related. Thus, these ratios can be thought to have a 1-to-1 relationship with species' genomes. Genomic Signatures consist an effective method of discriminating between sequences from different organisms.

2.5 N-Gram Graph (NGG)

NGG is a deterministic text classification model, opposed to other probabilistic models for text classification such as Hidden Markov Models [1], or Conditional Random Fields [13]. The main idea behind NGGs is that the neighborhood between sub-sequences in a sequence contains a crucial part of the sequence information. [51,19]. NGGs combine the benefits of n-gram flexibility with the well-structured representation of directed graphs [19,75]. Every extracted sequence from a text can be formed as a n-gram. Furthermore, the relation of these n-grams can be reflected using a graph. As follows, any text classification task can be reduced to a graph theory and pattern matching problem.

2.6 Applications to Genomics

The previous two methods (GS and NGG) were adopted by Polychronopoulos *et. al* [19,51] to investigate the relationships between genomic classes comprising of both non-coding and coding elements(exons). These sequences originated from key-species belonging in the taxonomic groups: humans, worms and insects. Their work has paved the way for this thesis, as their datasets and results have functioned as the baseline for our proposed method.

3 PROPOSED METHOD

In this section, we will give a high-level view of the methodology we followed, throughout the experiments.

3.1 Dataset Selection

Polychronopoulos *et. a*l [52] in their experiments described in the related work section, included data from various published and curated collections featuring coding and non-coding sequences of the organisms H.sapiens (human), D.melanogaster (insect, common fruit-fly) and C.elegans (worm). Because these data are heterogeneous, for the purpose of the classification experiments 1.000 elements were randomly selected from each genomic class.

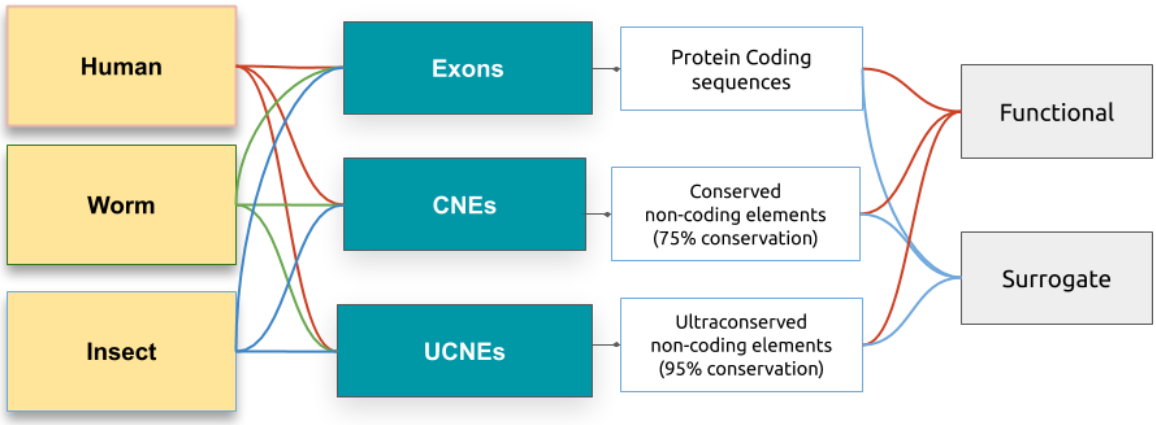


Figure 1: Depiction of genomic classes participating in the experiments

3.2 Formulation as a classification problem

The task consists of combining the sequences originating from two different genomic classes into one combined file containing both sets of sequences, and then predicting the class from which each sequence originated. Therefore, we run 26 different binary classification experiments. A high-level view of the pipeline can be seen in Figure 2.

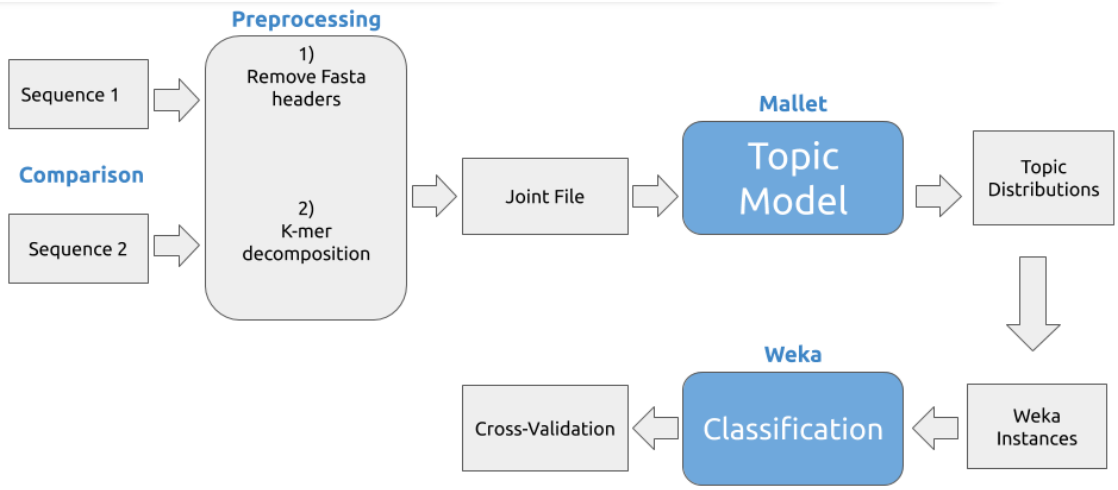


Figure 2: Proposed pipeline

3.3 Pre-processing and k-mer decomposition

Firstly, we eliminate all FASTA headers from the files involved, and we represent each sequence participating in the experiment as the vector of its k-mers in order to apply natural language processing techniques. The optimal value for k was estimated through an algorithm we developed around Zipf's law which originates from the field of computational linguistics. [53]

The "Zipf's" law makes the observation that in a long enough document, about 50% of the words only occur once. These words are called "Hapax legomena". This hypothesis has been reinforced in multiple cases, where large corpus of documents have been analyzed. Based on this phenomenon, we developed a simple algorithm and implemented it in Java(SDK 1.8) for calculating the word length k. The intuition behind our unorthodox approach is to estimate the k for which the genomic sequences mimic natural language text.

3.4 Topic Modeling

Next, we utilize Probabilistic Topic Modeling to infer topics from the corpus of the k-mer represented sequences. The next step involves projecting the sequences into the lower-dimensional space of their topic compositions based on the topics inferred in the previous step. The code for the representation is written in Java (JDK=1.8) and the topics inference was performed by MALLET. We

ran the experiments multiple times by varying the number of topics, and we compared the results through statistical testing.

3.5 Machine Learning

Subsequently, we ran multiple classification algorithms which cover a range of approaches in machine learning (trees:[Random Forest],Support Vector Classifier: [SMO],Neural Networks: [Multi-layer perceptron], Linear models: [Naive Bayes], Non-linear models : [Logistic]), on the topic-representations using 10-fold cross-validation. The Java software Weka [25] was used for this task.

3.6 Hypothesis Testing

Last but not least, we use hypothesis testing, to identify the dominant parameters for the representations, as well as the best performing algorithms and genomic classes in terms of F-measure. For all statistical testing, R-studio was used (version 3.6) . We deploy a set of tests from statistics, which are applied sequentially and are the following:

1. Shapiro-Wilk Normality Check [59,66]
2. Kruskal-Wallis non-parametric test 68]
3. Post-hoc analysis by Nemenyi 48]

4 RESULTS AND DISCUSSION

Through the extensive hypothesis testing and parameter tuning that took place, we reached some important conclusions on the selection of parameters' values for topic modeling.

4.1 Number of Topics

It seems that the effect of the number of topics on the classification reaches a plateau around 6 topics. After this number, increasing the quantity does not add

significant value to the experiments. However, if we value stability of speed, then 16 topics might be the wisest choice because of the lower overall-variance and slightly higher F-measures achieved.

2 Topics

	K = 3	K = 6	K = 8
Min	0.339	0.339	0.359
Max	0.856	0.882	0.898
Mean	0.6	0.625	0.62
StDev	0.123	0.138	0.125

16 Topics

	K = 3	K = 6	K = 8
Min	0.5	0.51	0.467
Max	0.904	0.929	0.89
Mean	0.725	0.749	0.699
StDev	0.103	0.101	0.1

64 Topics

	K = 6
Min	0.457
Max	0.912
Mean	0.753
StDev	0.1

4.2 Word Length

The length of the k-mers was inferred both experimentally, through classification results, and empirically, through application of Zipf's law (Figure 3). Both results suggest that $k = 6$ provides a good estimate for classification. However, exons are favored in the classification because $k = 6$ in this case is a multiple of 3 and is subject to codon bias.

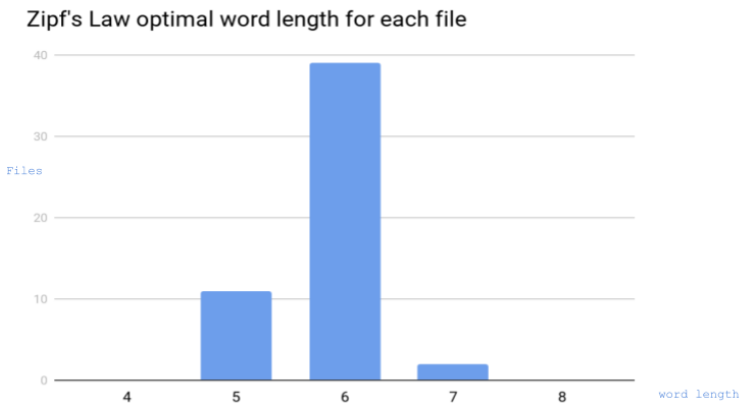


Figure 3: Results from Zipf's approach

4.3 Comparison with previous methods

We compare the F-measures between our topic representation and the methods implemented by Polychronopoulos *et.al*, consisting of N-gram graphs and Genomic Signatures in [51] which were applied on the same dataset as in our experiments. For a direct comparison between the methods, we have followed the same paradigm which divides the results into three distinct classes of experiments, involving the comparisons between:

1. surrogate DNA sequences
2. constrained DNA sequences and background surrogates
3. functional DNA sequences

and we display the results side-by-side in Figure 4, Figure 5 and Figure 6 for each class respectively. Overall, the Topic Model appears to perform considerably well, especially in comparisons involving exonic elements. This was to be expected, as mentioned before, as a result of codon bias. On the other hand, the topic model appears to follow the same ratios with the other methods. For example, the comparison between worm UCNEs and surrogates is the lowest-scoring experiment for both NGG and the Topic Model, and the same stands for the highest-scoring experiments of the two methods (worm exons vs human exons). It appears that the Topic Model performs especially well when classifying sequences originating from the worm or the insect, as well as when a functional class is compared with a surrogate class.

Description	NGG	GS	Topic Model T=16 K=6
surrogates for human exons surrogates for worm exons	83.86	85.98	88.62
surrogates for human UCNEs surrogates for insect UCNEs	73.98	84.05	83.04
surrogates for insect exons surrogates for human exons	80.48	87.49	87.77
surrogates for worm exons surrogates for insect exons	73.50	70.35	81.90
surrogates for human UCNEs surrogates for worm UCNEs	80.35	83.75	80.07
surrogates for worm UCNEs surrogates for insect UCNEs	58.79	64.00	64.35

Figure 4: Comparison between the Topic Model representation and the N-gram graph and Genomic Signature methods in terms of F-measure on experiments involving only surrogates

Description	NGG	GS	Topic Model T=16 K=6
worm exons surrogates	57.7	61.05	66.92
human exons surrogates	74.3	63.41	69.22
insect exons surrogates	52.36	59.45	62.84
worm UCNEs surrogates	56.76	55.62	60.37
human UCNEs surrogates	82.63	72.00	78.86
human EUnx CNEs surrogates	76.43	63.75	72.78
amniotic CNEs surrogates	78.62	63.00	74.12
mammalian CNEs surrogates	75.85	55.65	67.96
insect UCNEs surrogates	64.15	62.65	67.02

Figure 5: Comparison between the Topic Model representation and the N-gram graph and Genomic Signature methods in terms of F-measure on experiments involving the surrogate's dataset

Description	NGG	GS	Topic Model T=16 K=6
worm exons human exons	74.68	82.21	83.90
worm UCNEs human UCNEs	77.77	82.29	83.04
worm UCNEs insect UCNEs	70.89	74.95	77.38
human UCNEs insect UCNEs	82.08	86.70	85.47
insect exons human exons	70.03	81.29	81.90
worm exons insect exons	71.39	72.35	82.89

Figure 6: Comparison between the Topic Model representation and the N-gram graph and Genomic Signature methods in terms of F-measure on experiments involving elements with known functions

A factor that is important to consider, is that topic-representations do not encompass any kind of syntax related to the order of the sub-sequences in a sequence. The fact that they performed reasonably well in the various experiments involving genomic sequences, is an indicator that the true order of nucleotides in a DNA sequence does not play a major role in its functionality.

Furthermore, running the Topic Model multiple times varying the number of topics, can be a good way to capture both close-distance and long-distance interactions between the nucleotides. The intuition behind this is that if the number of topics is small, each topic will on average, span a larger area of the sequence than if the number of topics was very big. This might explain the high classification scores achieved in the experiments.

4.4 Verification with scientific literature

Chiang *et al.*in the published work [10] have shown that vertebrate CNEs are enriched in the subsequence *"TAATTA"*. We set up an experiment in order to examine if our model is able to capture this fact, in experiments involving vertebrate CNEs. For each comparison file, we ran the Topic Model (T=16, k=6).

Then we searched for the word *"TAATTA"* in the top 10 most frequent words of each one of the 16 topics. The result for this experiment is depicted in Figure 7, in a confusion matrix. Thus, the accuracy rate of our model is $(7 + 14)/26 = 80.76\%$. This result suggests that the model truly was able to infer this known fact about these genomic sequences automatically without any a-priori knowledge. This Is a promising sign that the topic-representation was chosen correctly for the types of genomic data that we worked with.

Record observations and construct confusion matrix.

Class A : comparison containing vertebrate CNEs

Class B : all other comparisons

Confusion Matrix for "TAAATTA"observations

class A	**7**	**3**
class B	**2**	**14**
	found	not found

Figure 7: Confusion matrix obtained from model verification according to literature

4.5 Future directions

Firstly, the topics produced by the Topic Model themselves consist an important information source, since by analyzing the sub-strings they contain, we might be able to identify new sequential motifs for each genomic class. Furthermore, more experiments can be performed on other genomic classes, to assess the model's consistency when ran on different data. Additionally, statistical testing can also be applied on the selection fork, which might also influence the overall score of the classification. Also, more variations of Topic Models can be tested on the data, such as the Hierarchical Topic Models which strive to learn hierarchies from data [23]. Lastly, methods could be adopted into the topic-modeling pipeline which automatically asses the ideal number of topics based on the topics' stability [21].

REFERENCES

[1] Hagai Almagor. A Markov analysis of DNA sequences. Journal of Theoretical Biology, 104(4):633 – 645, 1983.

[2] D. Anastassiou. Genomic signal processing. IEEE Signal Processing Magazine, 18(4):8–20, July 2001.

[3] Michele Banko and Lucy Vanderwende. Using n-grams to understand the nature of summaries. In Proceedings of HLT-NAACL 2004: Short Papers, pages 1–4. Association for Computational Linguistics, 2004.

[4] David M. Blei. Introduction to Probabilistic Topic Models. 2010.

[5] David M. Blei. Probabilistic Topic Models. Commun. ACM, 55(4):77–84, April 2012.

[6] David M Blei, Michael I Jordan, and others. Variational inference for Dirichlet process mixtures. Bayesian analysis, 1(1):121–143, 2006.

[7] Philipp Bucher. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. Journal of molecular biology, 212(4):563–578, 1990.

[8] William B Cavnar, John M Trenkle, and others. N-gram-based text categorization. In Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval, volume 161175. Citeseer, 1994.

[9] Betty Yee Man Cheng, Jaime G Carbonell, and Judith Klein-Seetharaman. Protein classification based on text document classification techniques. Proteins: Structure, Function, and Bioinformatics, 58(4):955–970, 2005.

[10] Charleston W. K. Chiang, Adnan Derti, Daniel Schwartz, Michael F. Chou, Joel N. Hirschhorn, and C.- ting Wu. Ultraconserved Elements: Analyses of Dosage Sensitivity, Motifs and Boundaries. Genetics, 180(4):2277–2293, December 2008.

[11] William W Cohen. Fast effective rule induction. In Machine learning proceedings 1995, pages 115–123. Elsevier, 1995.

[12] Terry Copeck and Stan Szpakowicz. Vocabulary agreement among model summaries and source documents. In ACL Text Summarization Workshop, 2004.

[13] Aron Culotta, David Kulp, and Andrew McCallum. Gene prediction with conditional random fields. University of Massachusetts, Amherst, Tech. Rep. UM-CS-2005-028, 2005.

[14] P. J. Deschavanne, A. Giron, J. Vilain, G. Fagot, and B. Fertil. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. Molecular Biology and Evolution, 16(10):1391–1399, October 1999.

[15] Elena A Erosheva. Bayesian estimation of the grade of membership model. Bayesian Statistics, 7:501–510, 2003.

[16] Madhavi Ganapathiraju, Deborah Weisser, Roni Rosenfeld, Jaime Carbonell, Raj Reddy, and Judith Klein-Seetharaman. Comparative n-gram analysis of whole-genome protein sequences. In Proceed- ings of the second international conference on Human Language Technology Research, pages 76–81. Morgan Kaufmann Publishers Inc., 2002.

[17] Zoubin Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. Philo- sophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371(1984):20110553, 2013.

[18] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. Nature, 521(7553):452, 2015.

[19] George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. ACM Transactions on Speech and Language Processing (TSLP), 5(3):5, 2008.

[20] R. Grantham, C. Gautier, M. Gouy, R. Mercier, and A. Pavé. Codon catalog usage and the genome hypothesis. Nucleic Acids Research, 8(1):197–197, 1980.

[21] Derek Greene, Derek O'Callaghan, and Pádraig Cunningham. How many topics? stability analysis for topic models. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 498–513. Springer, 2014.

[22] Philippa C. Griffin, Jyoti Khadake, Kate S. LeMay, Suzanna E. Lewis, Sandra Orchard, Andrew Pask, Bernard Pope, Ute Roessner, Keith Russell, Torsten Seemann, Andrew Treloar, Sonika Tyagi, Jef- frey H. Christiansen, Saravanan

Dayalan, Simon Gladman, Sandra B. Hangartner, Helen L. Hayden, William W. H. Ho, Gabriel Keeble-Gagnère, Pasi K. Korhonen, Peter Neish, Priscilla R. Prestes, Mark F. Richardson, Nathan S. Watson-Haigh, Kelly L. Wyres, Neil D. Young, and Maria Victoria Schneider. Best Practice Data Life Cycle Approaches for the Life Sciences. July 2017.

[23] Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, and David M Blei. Hierarchical topic models and the nested Chinese restaurant process. In Advances in neural information processing systems, pages 17–24, 2004.

[24] Thomas L Griffiths and Mark Steyvers. A probabilistic approach to semantic representation. In Pro- ceedings of the annual meeting of the cognitive science society, volume 24, 2002.

[25] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. SIGKDD Explor. Newsl., 11(1):10–18, November 2009.

[26] Ross C Hardison. Comparative Genomics. PLOS Biology, 1(2), 2003.

[27] Zellig S. Harris. Distributional Structure. WORD, 10(2-3):146–162, 1954.

[28] Nathaniel D Heintzman, Rhona K Stuart, Gary Hon, Yutao Fu, Christina W Ching, R David Hawkins, Leah O Barrera, Sara Van Calcar, Chunxu Qu, Keith A Ching, and others. Distinct and predictive chro- matin signatures of transcriptional promoters and enhancers in the human genome. Nature genetics, 39(3):311, 2007.

[29] John Houvardas and Efstathios Stamatatos. N-gram feature selection for authorship identification. In International conference on artificial intelligence: Methodology, systems, and applications, pages 77–86. Springer, 2006.

[30] Edwin T Jaynes. Probability theory: The logic of science. Cambridge university press, 2003.

[31] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. page 9, 2004.

[32] Samuel Kariin and Chris Burge. Dinucleotide relative abundance extremes: a genomic signature. Trends in Genetics, 11(7):283 – 290, 1995.

[33] S Karlin, J Mrázek, and A M Campbell. Compositional biases of bacterial genomes and evolutionary implications. Journal of Bacteriology, 179(12):3899–3913, June 1997.

[34] Samuel Karlin and Lon R Cardon. Computational DNA sequence analysis. Annual review of microbi- ology, 48(1):619–654, 1994.

[35] Donna Karolchik, Robert Baertsch, Mark Diekhans, Terrence S Furey, Angie Hinrichs, YT Lu, Krishna M Roskin, Matthias Schwartz, Charles W Sugnet, Daryl J Thomas, and others. The UCSC genome browser database. Nucleic acids research, 31(1):51–54, 2003.

[36] Jong Yong Kim and John Shawe-Taylor. Fast string matching using an n-gram algorithm. Software: Practice and Experience, 24(1):79–88, 1994.

[37] Min-Soo Kim, Kyu-Young Whang, Jae-Gil Lee, and Min-Jae Lee. n-gram/2l: A space and time efficient two-level n-gram inverted index structure. In Proceedings of the 31st international conference on Very large data bases, pages 325–336. VLDB Endowment, 2005.

[38] Su Yeon Kim and Jonathan K Pritchard. Adaptive evolution of conserved noncoding elements in mam- mals. PLoS genetics, 3(9):e147, 2007.

[39] Daphne Koller and Nir Friedman. Probabilistic graphical models: principles and techniques. MIT press, 2009.

[40] Nikos Kostagiolas, Nikiforos Pittaras, Christoforos Nikolaou, and George Giannakopoulos. Exploring different sequence representations and classification methods for the prediction of nucleosome positioning. bioRxiv, page 482612, 2018.

[41] Maxwell W. Libbrecht and William Stafford Noble. Machine learning applications in genetics and gen- omics. Nature Reviews Genetics, 16(6):321–332, June 2015.

[42] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence stat- istics. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pages 150–157, 2003.

[43] RN Mantegna, SV Buldyrev, AL Goldberger, S Havlin, C-K Peng, M Simons, and HE Stanley. System- atic analysis of coding and noncoding DNA sequences using methods of statistical linguistics. Physical Review E, 52(3):2939, 1995.

[44] Andrew Kachites McCallum. MALLET: A Machine Learning for Language Toolkit. 2002.

[45] Páll Melsted and Jonathan K. Pritchard. Efficient counting of k-mers in DNA sequences using a bloom filter. BMC Bioinformatics, 12(1):333, August 2011.

[46] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In Pro- ceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[47] G.E. Moore. Cramming More Components Onto Integrated Circuits. Proceedings of the IEEE, 86(1):82–85, January 1998.

[48] Peter Nemenyi. Distribution-free multiple comparisons. In Biometrics, volume 18, page 263. International Biometric Soc 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1962.

[49] Uwe Ohler, Guo-chun Liao, Heinrich Niemann, and Gerald M Rubin. Computational analysis of core promoters in the Drosophila genome. Genome biology, 3(12):research0087–1, 2002.

[50] Dimitris Polychronopoulos, James W. D. King, Alexander J. Nash, Ge Tan, and Boris Lenhard. Con- served non-coding elements: developmental gene regulation meets genome organization. Nucleic Acids Research, 45(22):12611–12624, December 2017.

[51] Dimitris Polychronopoulos, Anastasia Krithara, Christoforos Nikolaou, Giorgos Paliouras, Yannis Almirantis, and George Giannakopoulos. Analysis and Classification of Constrained DNA Elements with N- gram Graphs and Genomic Signatures. In Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe,editors, Algorithms for Computational Biology, pages 220–234, Cham, 2014. Springer International Publishing.

[52] Dimitris Polychronopoulos, Diamantis Sellis, and Yannis Almirantis. Conserved Noncoding Elements Follow Power-Law-Like Distributions in Several Genomes as a Result of Genome Dynamics. PLOS ONE, 9(5):1–12, 2014.

[53] David M. W. Powers. Applications and Explanations of Zipf's Law. In Proceedings of the Joint Con- ferences on New Methods in Language Processing and Computational Natural Language Learning, NeMLaP3/CoNLL '98, pages 151–160, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics. event-place: Sydney, Australia.

[54] David T Pride, Richard J Meinersmann, Trudy M Wassenaar, and Martin J Blaser. Evolutionary im- plications of microbial genome tetranucleotide frequency biases. Genome research, 13(2):145–158, 2003.

[55] Kim D Pruitt, Tatiana Tatusova, and Donna R Maglott. NCBI reference sequences (RefSeq): a cur- ated non-redundant sequence database of genomes, transcripts and proteins. Nucleic acids research, 35(suppl_1):D61–D65, 2006.

[56] Ji Qi, Hong Luo, and Bailin Hao. CVTree: a phylogenetic tree reconstruction tool based on whole genomes. Nucleic acids research, 32(suppl_2):W45–W47, 2004.

[57] Aaron R Quinlan and Ira M Hall. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics, 26(6):841–842, 2010.

[58] Lawrence R Rabiner and Biing-Hwang Juang. An introduction to hidden Markov models. ieee assp magazine, 3(1):4–16, 1986.

[59] Nornadiah Mohd Razali, Yap Bee Wah, and others. Power comparisons of shapiro-wilk, kolmogorov- smirnov, lilliefors and anderson-darling tests. Journal of statistical modeling and analytics, 2(1):21–33, 2011.

[60] Dorota Retelska, Emmanuel Beaudoing, Cédric Notredame, C Victor Jongeneel, and Philipp Bucher. Vertebrate conserved non coding DNA regions have a high persistence length and a short persistence time. BMC genomics, 8(1):398, 2007.

[61] Peter Rice, Ian Longden, and Alan Bleasby. EMBOSS: the European molecular biology open software suite. Elsevier current trends, 2000.

[62] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5):513–523, January 1988.

[63] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620, 1975.

[64] Yutaka Sasaki and others. The truth of the F-measure. Teach Tutor mater, 1(5):1–5, 2007.

[65] Sophie Schbath, Bernard Prum, and Elisabeth de Turckheim. Exceptional motifs in different Markov chain models for a statistical analysis of DNA sequences. Journal of Computational Biology, 2(3):417– 437, 1995.

[66] Samuel S Shapiro and RS Francia. An approximate analysis of variance test for normality. Journal of the American Statistical Association, 67(337):215–216, 1972.

[67] Victor V Solovyev and Kira S Makarova. A novel method of protein sequence classification based on oligopeptide frequency analysis and its application to search for functional sites and to domain localiz- ation. Bioinformatics, 9(1):17–24, 1993.

[68] John D Spurrier. On the null distribution of the Kruskal–Wallis statistic. Nonparametric Statistics, 15(6):685–691, 2003.

[69] Stuart Stephen, Michael Pheasant, Igor V Makunin, and John S Mattick. Large-scale appearance of ultraconserved elements in tetrapod genomes and slowdown of the molecular clock. Molecular biology and evolution, 25(2):402–408, 2007.

[70] Mark Steyvers and Tom Griffiths. Probabilistic Topic Models. page 15, 2017.

[71] Efi Karra Taniskidou, George Papadakis, George Giannakopoulos, and Manolis Koubarakis. Compar- ative Analysis of Content-based Personalized Microblog Recommendations [Experiments and Analysis]. arXiv preprint arXiv:1901.05497, 2019.

[72] Daniel Tauritz. Application of n-Grams. Department of Computer Science, 2002.

[73] RStudio Team and others. RStudio: integrated development for R. RStudio, Inc., Boston, MA URL http://www. rstudio. com, 42:14, 2015.

[74] P. D. Turney and P. Pantel. From Frequency to Meaning: Vector Space Models of Semantics. Journal of Artificial Intelligence Research, 37:141–188, February 2010.

[75] John Violos, Konstantinos Tserpes, Iraklis Varlamis, and Theodora Varvarigou. Text Classification Us- ing the N-Gram Graph Representation Model Over High Frequency Data Streams. Frontiers in Applied Mathematics and Statistics, 4:41, 2018.

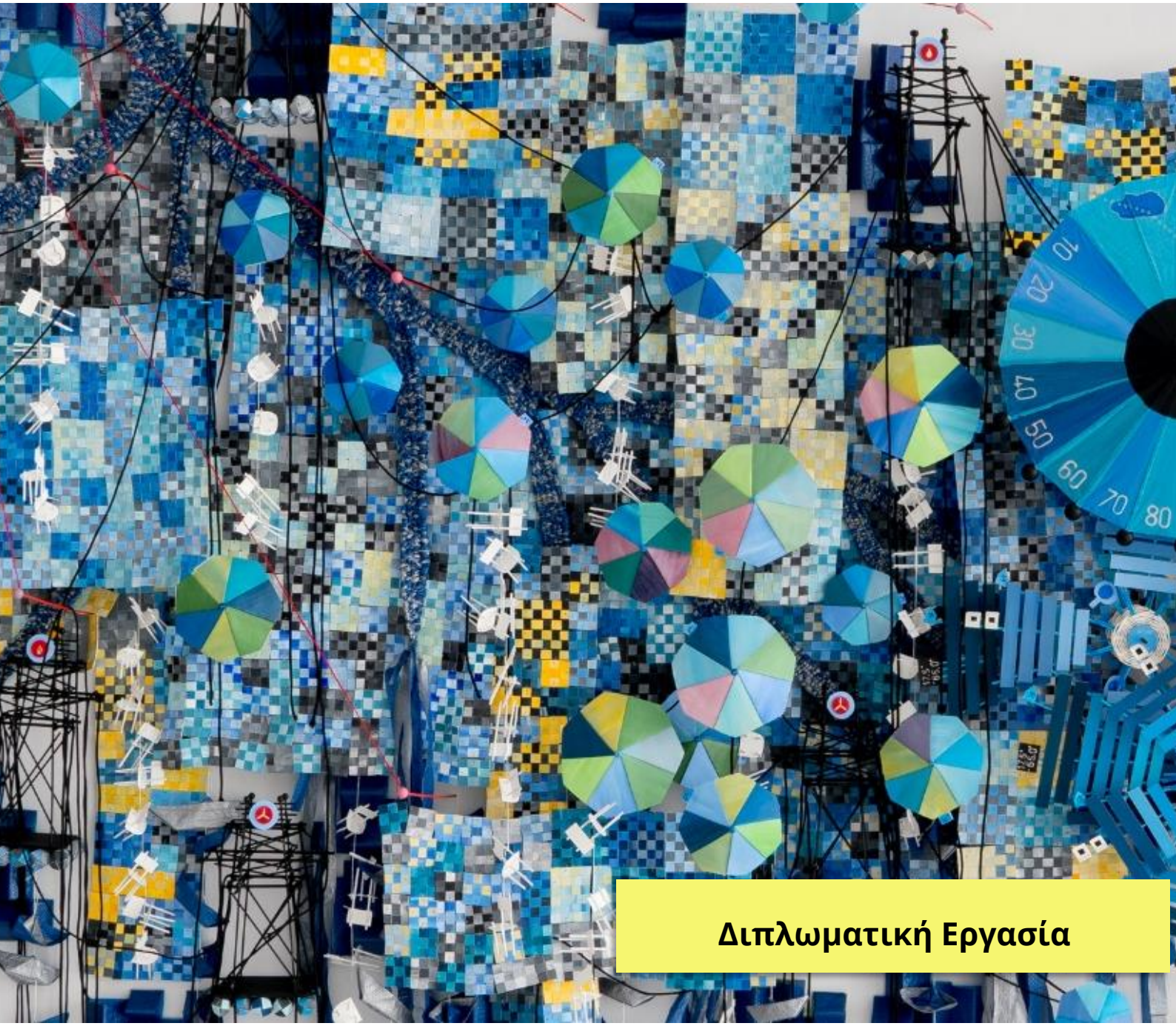
[76] Klaudia Walter, Irina Abnizova, Greg Elgar, and Walter R. Gilks. Striking nucleotide frequency pattern at the borders of highly conserved vertebrate non-coding sequences. Trends in Genetics, 21(8):436 – 440, 2005.

[77] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz

Bonino da Silva Santos, Philip E Bourne, and others. The FAIR Guiding Principles for scientific data management and stewardship. Scientific data, 3, 2016.

[78] Adi Yannai, Sophia Katz, and Ruth Hershberg. The Codon Usage of Lowly Expressed Genes Is Subject to Natural Selection. Genome Biology and Evolution, 10(5):1237–1246, April 2018.

[79] Richard Zens and Hermann Ney. Discriminative reordering models for statistical machine translation. In Proceedings of the Workshop on Statistical Machine Translation, pages 55–63. Association for Computational Linguistics, 2006.



Διπλωματική Εργασία

Autonomic Tackling of Unknown Obstacles in Navigation of Robotic Platform

Nefeli K. Prokopaki Kostopoulou

ABSTRACT

The goal of the present thesis is to develop a method for a robotic outdoor platform. The robot should discover by itself, based on its sensors and its previous knowledge, how to approach an obstacle that stands in front of it, whether it is capable of driving over the obstacle or should avoid it. Obstacle avoidance ensures the safety and integrity of both the robotic platform and the people and objects present in the same space. That is one of the reasons why current approaches mainly concentrate on maneuver to avoid obstacles rather than yield autonomous systems with the ability to self improve. There is not much work done on curiosity-driven exploration, in which there is no explicit goal, but the abstract need for the robot to learn a new environment.

In the current thesis we introduce a system that not only autonomously classifies its environment to areas that can or cannot be driven over, but also has the capacity for self- improvement. To do so, we use a pre-trained neural network for whole scene semantic segmentation. We implement a program that accepts as input images extracted from the neural network mentioned above and predicts whether the illustrated scenes can be traversed or not. The program trains itself and then evaluates its effectiveness. Our results are quite satisfactory and the error rate can be explained by the fact that the environment is not evenly distributed in obstacles and paths, while at the same time it is not always clear which one is dominant. Furthermore, we show that our model can be easily optimized with just a few modifications.

SUPERVISORS

Stasinos Konstantopoulos, Research Associate NCSR Demokritos

Panagiotis Stamatopoulos, Assistant Professor NKUA

1 INTRODUCTION

During the last years the focus of research for robotic applications evolved from well structured indoor environments to unstructured outdoor environments. With this expansion of interest, it is a crucial prerequisite to reliably classify traversable ground in the environment, especially when it comes to truly autonomous (or else self-supervised) systems. This topic is typically referred to as *traversability analysis* or *obstacle detection* [1]. The verb traverse is defined as "*to pass or move over, along, or through*". Hence *traversability* refers to the affordance of being able to traverse [2]. Failing on this task can cause great damage or restrict the robots movement unnecessarily.

In this thesis we will tackle on how an autonomous mobile robot can improve its traversability estimation method in natural environments, meaning not only on bare ground-like environment but also on terrain containing vegetation. On contrast, we will rule out high-risk applications where a single accident can be fatal to the robot like planetary or volcano exploration. We will concentrate in everyday practical situations. We will determine how to introduce a learning capability to the robot that will enable it to decide for itself the traversability of the terrain around it, based on input from its sensors and its experience of traveling over similar terrain in the past. We would also like our robot to plan further ahead and avoid entering traps that prevent it from reaching its goal.

2 AUTONOMOUS NAVIGATION

The goal is for the robot to be able to autonomously navigate in natural environments. To do so, we could use a pre-trained neural network in order to be able to distinguish traversable from non-traversable terrain.

For the purposes of this thesis we will basically deal with pre-trained models. By borrowing a little story from Gupta [46] we are going to explain why. Imagine two people, Mr. Potato and Mr. Athlete. They sign up for soccer training at the same time. Neither of them has ever played soccer and the skills like dribbling, passing, kicking etc. are new to both of them. Mr. Potato does not move much

but Mr. Athlete does. That is the core difference between the two even before the training has even started. As you can imagine, the skills Mr. Athlete has developed as an athlete (e.g. stamina, speed and even sporting instincts) are going to be very useful for learning soccer even though Mr. Athlete has never trained for soccer. Mr. Athlete benefits from his pre-training. Mr. Potato on the other hand will have to develop all these skills from scratch, something that will cost him much more energy and time.

In this chapter we describe the primary thoughts and the actual strategy on how to reach the goal mentioned above.

2.1. Selecting a convolutional neural network

Primarily, a neural network is needed in order to convert images that depict the environment seen by the robot, to a form more recognizable by it. Ideally, after the conversion, all objects would be distinguished from all traversable areas. But the idea of finding such a neural network is probably not realistic. Thus, the goal for this section is to find a neural network that distinguishes all objects from one another.

Object classification, localization, detection

The first attempt was to use one of the most-well known models from ILSVRC. With a little help from the code of Rosebrocke [47] we experimented on pre-trained VGGNet, ResNet, Inception and Xception. We fed them images and, as expected, they returned classification predictions about them. With the contribution of ImageNet a list of human-readable labels and the probability associated with them was printed.

Images containing just one object (e.g. soccer ball, couch) led to predictions that were satisfactory in their entirety. But being fed with images containing multiple objects (e.g. book and glasses) the networks got confused. They gave some decent predictions (like envelope or book jacket, in the previous example) but

also some that were a little bit off (like lighter or birdhouse), within their top-5 list.

Images that contain vegetation, which are of particular interest to us in this work, were the worst case scenario. For example, the networks above, after being fed with an image of a tree, gave as most likely predictions kinds of seeds like lemons. This probably happened because the networks concentrated on the one part of the whole image that was most recognizable by them. Finding out what the predictions with smaller probabilities were, did not help. As the networks kept predicting, they got desperate and started giving all sorts of predictions.

Even though these networks are proven to be great on object classification, when it comes to scene recognition there is no trivial way to make them work correctly.

Similarly, models specialized in object detection like SSD or YOLO, do not seem to be able to generalize on scene segmentation issues. As dictated by their name, they detect all objects within an image, but ignore the rest of the scene.

Related work

Many papers have been published regarding robot navigation approaches that use already existing deep neural networks, or modify them in order to meet their needs. Unfortunately, some of these researchers had not made their code publicly available, in order for us to rely on part of their work and extend it with our own ideas. And while others kindly released their code online, their networks were not trained compatible to the needs of this thesis.

Some papers considering traversability estimation concentrate on go or no-go situations, [17, 48]. They use generative adversarial networks and train them to estimate whether the space seen through the given image is traversable or not. They are trained in indoor environments, which means that we would have to train them from scratch to work in natural outdoor environments.

Likewise, Tai et al. [22] further extended the concept of deep neural networks to not only perception but also decision-making. Basically, they used a structure that fuses several convolutional neural network layers with decision-making process, in order to explore an unknown environment. According to the authors, in traditional computer vision applica- tions each label of the output represents either an object or scene categories. The outputs of their model are control commands that show the platform which route to follow.

Other researchers use neural networks to classify the area in front of the robot according to traversability and level of confidence [49, 50]. These neural networks assign class la- bels to parts of the input image. Classes which show that "*only ground or only obstacle is seen in the area*", are of high confidence. While the rest inspire lower confidence. These classes are separated to "*ground and obstacle may be seen*", "*obstacle is seen but does not fill the area*", "*location where an obstacle meets the ground*".

Some approaches identify only a few class labels to classify the whole image [51, 52, 53]. While others have as their main goal to find and follow a path [54, 55]. The first category usually includes scene labels that can be used in outdoor environments (such as sky, road, tree, grass, building), as their title declares. The second one, while also being able to recognize such class labels, identifies them as obstacles.

Scene segmentation

So, the aim is on total scene segmentation rather than single or even multiple object categorization. Semantically meaningful image understanding is a relatively recent topic in computer vision. That explains why, compared to recognition, far fewer papers address scene segmentation in neural networks [42]. A general semantic segmentation architecture can be broadly thought of as an *encoder network* followed by a *decoder network* [41]. The encoder is usually a pre-trained classification network like VGGNet or ResNet that outputs a feature map. The task of the decoder is to semantically project the lower resolution

features learned by the encoder, onto the higher resolution (pixel space) to get the best closest match to the original input.

Given a visual scene of, let us say, a living room, a robot equipped with a trained convolutional network can accurately predict the scene category. However, to freely navigate in the scene and manipulate the objects inside, the robot has far more information to digest [44]. It needs to recognize and localize not only the objects like sofa, table, and television but also to segment the stuff like floor, wall and ceiling for spatial navigation. It probably needs to recognize also object parts, e.g. a seat of a chair or a handle of a cup, to allow proper interaction.

Following the instructions of Le [41], on his guide on how to do semantic segmentation using deep learning, we attempted to implement the most popular architecture for semantic segmentation, fully convolutional networks. We had in mind that the encoder, VGGNet in this case, would be pre-trained, but we would have to train the decoder from the beginning on KITTI [56]. But fully convolutional networks, at least those trained on KITTI dataset, do semantic segmentation only on foreground and ignore the background. In order for us to be able to decide terrain traversability, whole scene segmentation is necessary.

Datasets

Scene parsing, or recognizing and segmenting objects in an image, remains one of the key problems in scene understanding [44]. Going beyond the image-level recognition, scene parsing requires a much denser annotation of scenes with a large set of objects.

However, the current datasets have limited number of objects, e.g. COCO [45], PASCAL VOC [57]. In many cases those objects are not the most common objects one encounters in the world like frisbees or baseball bats. Or the datasets only cover a limited set of scenes, like urban sceneries, e.g. Cityscapes [58] and KITTI [56]. Most of the large-scale datasets typically only contain labels at the image level or provide bounding boxes, e.g. ImageNet [31], PASCAL VOC and KITTI. ImageNet has the largest set of classes, but contains relatively simple scenes.

Compared to the largest annotated datasets COCO and ImageNet, ADE20K [59] comprises of much more diverse scenes.

Finally, existing datasets with pixel-level labels typically provide annotations only for a subset of foreground objects, and no background, e.g. PASCAL VOC and COCO. That is probably why fully convolutional networks trained on KITTI, as mentioned before, do not give class labels to the background pixels. But, generally, pixel appearance features al- low to perform well on classifying (amorphous) background classes. ADE20K categorizes semantic classes present in the scene into three super classes: stuff (sky, road, building, etc), foreground objects (car, tree, sofa, etc), and object parts (car wheels and door, people head and torso, etc).

Pre-trained models on scene segmentation

After finding some pre-trained models we experimented on different algorithms and training datasets. The backbone network in all those cases was ResNet. Be reminded that successful deep neural network architectures for image level classification like AlexNet, VGGNet and ResNet are a natural precursor to, and often a direct part, of semantic segmentation architectures.

The algorithms mentioned previously are the fully convolutional network [43], *Pyramid Scene Parsing Network (PSP)* [60] and DeepLab [61]. All three of them when trained on COCO or PASCAL VOC do not perform complete scene semantic segmentation. When they are fed with indoor images they recognize only foreground objects. No class labels are given to the background pixels. Our hypothesis that the part of the image theyignore is the background is further intensified by the way outdoor images are handled. Their result on outdoor input is a black image. On the contrary, when trained on ADE20K all tree networks transact semantic segmentation on the whole scene. The outputs from all tree algorithms trained on each of the three datasets are depicted in Figure 2.1 for outdoor inputs.

Many deep learning architectures have been proposed for image segmentation. As far as we know, in order to be able to semantically segment the whole image we can use any of the three algorithms mentioned above as long as they are pre-trained on the ADE20K dataset. But how can we use them to help us distinguish traversable from non-traversable terrain?

(a) Original image **(b) COCO or VOC**

(c) FCN with ADE **(d) PSP with ADE** **(e) DeepLab with ADE**

Figure 2.1: Results given from outdoor input. Fully convolutional network, Pyramid scene parsing network and DeepLab trained on COCO, PASCAL VOC and ADE20K dataset.

2.2. Estimating traversability

We implemented a program that when given an image predicts whether the illustrated scene is traversable or not. The program trains itself and then evaluates its effectiveness. All the input values are images extracted from a neural network for whole image semantic segmentation. These images consist of colors depicting objects and parts of the scene. Each color is recognized by the computer as a triad of numbers representing the amount of red, green and blue present to produce the original color.

For humans to be able to observe the program's functionality we keep a default correspondence between colors and class labels. Also, to ease comprehension, all operations are rounded to the second decimal place.

During the program training period, it reads previously annotated images. These annotations are penetrable region or obstacle. In this thesis, the term *penetrable* will be used interchangeably with the term traversable. The program keeps track of how many times each color is found in traversable and how many in non-traversable images. So, it can compute the traversability percentage of each color. For example, let's suppose that brown corresponds to "earth". Brown color exists in 70 images from which 56 are traversable and the other 14 non-traversable. So the traversability percentage of the earth is

$$56 \times 100 \div 70 = 80$$

Therefore, the earth (and, consequently, the color brown) has 80% chance of being traversable.

During evaluation we implement two different calculating methods. Both of them are responsible for deciding whether the given images are traversable or not. Every image with chance of being traversable greater or equal to 50% is considered to be traversable (and therefore with chance less that 50% is considered to be non-traversable). Let's take Figure 2.2 and try to explain the two methods. For the purpose of this example, we suppose that brown has 80% chance of being traversable, as is found to be penetrable 8 times out of the total of 10. Green 9 out of 30 (30%) and blue 9 out of 20 (45%). To ease our understanding let's say that brown corresponds to "earth", green to "tree" and blue to "sky".

(a) Original image **(b) Image resulting from PSP**

Figure 2.2: Example for explaining the differences between the two calculating methods - earth (80% chance of being traversable, found to be penetrable 8 times out of the total of 10), tree (30%, 9 out of 30), sky (45%, 9 out of 20)

Intuitively, the difference between the two methods is that the former considers all classes as equal, while the latter pays attention to how often a class has been seen. More technically:

1. the first method determines the probability of an image to be penetrable, as the average of the traversability percentages for each color contained within. Figure 2.2, which contains brown, green and blue, has a probability of 51.67% being traversable.

$$(80\% + 30\% + 45\%) \div 3 = 51.67\%$$

2. the second method uses the number of times each color was found in traversable and in non-traversable images. It assumes that the likelihood of an image being traversable is in direct dependence of the number of times each color found within, is penetrable. In other words, it specifies that the traversability percentage of an image, is the weighted average of the traversability percentage of all the colors within it. This time, the probability of Figure 2.2 being traversable is 43.33%.

$$(8 + 9 + 9) \times 100\% \div (10 + 30 + 20) = 43.33\%$$

The first method is an obvious way to find the probability of an image being penetrable. The second one, however, emphasizes on statistics, in the sense that a one-time event may be a coincidence, but the more often it happens the more

secure it is to become a rule. In the previous example class earth was found to be traversable 8 times out of the total of 10. This means that 2 out of 10 times, it was found to be non-traversable. Let's suppose the first given image containing earth was non-traversable. So, the traversability percentage of earth would be 0%. While the first method will use it as a certainty, the second one will have low confidence on it.

It is known in advance whether the testing images are penetrable. It is trivial to discover whether the previously described methods' decisions are right or wrong. In the example above the first method decides that the image is non-traversable, while the second method determines the opposite. Note that in other cases the results of the two methods may concur.

As we have introduced a program that decides whether the input images are traversable or not, we will proceed in the next section to evaluate the effectiveness of our approach.

3 EXPERIMENTAL VALIDATION AND COMPARISON

We evaluate our machine learning model with a procedure called *cross validation* [62]. It is also known as *rotation estimation* or *out-of-sample testing*. Cross validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions. In a prediction problem, a model is usually given two datasets. First a dataset of known data on which training is run (*training set*). And then a dataset of unknown data (or first seen data) against which the model is tested (*validation set* or *testing set*). The goal of cross validation is to test the model's ability to predict data not used during the training of the model.

The procedure has a single parameter N that refers to the number of folds that a given data sample is to be split into. As such, the procedure is often called *N-fold cross validation*.

In this thesis our data sample consists of twenty directories containing images. Ten of these directories contain 778 traversable images and the other ten include 945 non- traversable.

In order to evaluate our model we split our data sample in five folds and perform 5-fold cross validation. Because we have twenty directories, each fold has an equal number of four directories. During each fold we use two traversable and two non-traversable directories for testing and the rest for training. In Table 2.1 one can see the numbers of images used for training and testing for each fold.

As mentioned before we implemented a program that predicts whether a scene is traversable or not. It accepts as input images outputted by a neural network for semantic segmentation. Previously we established that the ADE20K dataset is the best fit for whole scene semantic segmentation. And that, in theory, any of the fully convolutional network (FCN), pyramid scene parsing network (PSP) and DeepLab can be used equally effectively. So we run our program with images that have emerged by each one of them to check which has the best results.

Table 2.1: Number of images for each testing and training set in 5-fold cross validation. Total number of images: 778 traversable and 945 non-traversable.

Fold	Traversable images		Non-traversable images	
	Testing set	Training set	Testing set	Training set
1	153	625	184	761
2	169	609	221	724
3	164	614	225	720
4	109	669	165	780
5	183	595	150	795

We gave as input to each of the three neural network models the images contained in the directories mentioned previously. And then fed our program with their outputs. The results of the first calculating method, described in Section 2.2 are shown in Table 2.2. And the results of the second method in Table 2.3. In both tables it is quite obvious that the PSP has clearly better results than the other two models, for both individual and global trials.

Table 2.2: 1st calculating method - Percentage of evaluation images whose traversability was found correctly. In the second column are the results from images derived from the FCN, the third from PSP and the forth from DeepLab**.**

Success on 1st calculating method			
Fold	with FCN (%)	with PSP (%)	with DeepLab (%)
1	57.57	77.45	56.97
2	65.9	94.36	56.67
3	66.58	87.15	58.87
4	84.31	86.86	72.26
5	75.98	94.89	75.08
Average	69.3	88.33	63.26

Table 2.3: 2nd calculating method - Percentage of evaluation images whose traversability was found correctly. In the second column are the results from images derived from the FCN, the third from PSP and the fourth from DeepLab.

Success on 2nd calculating method			
Fold	with FCN (%)	with PSP (%)	with DeepLab (%)
1	54.6	62.61	54.6
2	57.69	93.33	56.67
3	57.84	80.46	58.35
4	63.5	79.2	60.58
5	45.05	45.05	45.05
Average	55.6	72.84	55.02

Consequently, we decided to continue our research using the PSP model. In Table 2.4 we gathered the results from both methods when using data from the PSP algorithm. Even though we think that the second calculating method is more representative of the overall sample, in this case the first one gives better results. That happens probably because the non-traversable images of the training set are much more that the traversable ones, as shown in Table 2.1.

Table 2.4: PSP - Percentage of testing images whose traversability was found correctly with the 1st and the 2nd calculating method.

PSP trained on ADE20K dataset		
Fold	Success with 1st method (%)	Success with 2nd method (%)
1	77.45	62.61
2	94.36	93.33
3	87.15	80.46
4	86.86	79.2
5	94.89	45.05
Average	88.33	72.84

Therefore, as is obvious, we chose to deepen into the first method of deciding on image traversability. In Table 2.5 we describe how the success rates of the first method arose.

- In column 1 each fold is shown.
- In columns 2 and 3 we give the number of traversable images and the percentage of those that were successfully predicted traversable.
- In columns 3 and 4 the same for non traversable images.
- Finally, column 5 gives the total success rate of the first method, summarizing the results of both traversable and non-traversable images.

Table 2.5: PSP 1st method - Number of traversable and non-traversable images for each testing set, with their success rate in finding traversability. Observe that the success rate for non-traversable images is always 100%, while in traversable images is much lower.

Testing images on 1st method with PSP					
Fold	Traversable		Non-traversable		Average (%)
	Number	Success (%)	Number	Success (%)	
1	153	50.33	184	100	77.45
2	169	86.98	221	100	94.36
3	164	69.51	225	100	87.15
4	109	66.97	165	100	86.86
5	183	90.71	150	100	94.89
Average	778	74.16	945	100	88.33

4 CONCLUSIONS AND FUTURE WORK

In this thesis we concentrated on traversability estimation methods. We implemented a program that when given images, predicts whether they are traversable or not. These images are extracted from a convolutional neural network for total scene semantic segmentation. We chose Pyramid Scene Parsing Network (PSP) pre-trained on ADE20K dataset for this purpose, as it seemed to have the best results. We fed PSP with images from natural outdoor environments, containing vegetation. It resulted with images consist of colors depicting objects and parts of the scene. Lastly, we evaluated our implementation and explained how the traversable images are more difficult to predict than the non-traversable. The main reason for this is that the data

sample used for training, consists of many more non-traversable than traversable images. That is something we are not able to control because we want our system to be autonomous.

We believe that our contribution has been noteworthy. As said before, it has become clear to researchers in robotics that current approaches are yielding systems with limited autonomy and ability for self-improvement. We managed to create a system that not only autonomously learns the traversability of its environment, but also has the capacity to self-improve. With only a few modifications our program could be ready to be used on an actual robotic platform to explore its surroundings and improve its knowledge of traversability.

REFERENCES

[1] Benjamin Suger, Bastian Steder, and Wolfram Burgard. Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3d-lidar data. IEEE International Conference on Robotics and Automation (ICRA), pages 3941–3946, 2015.

[2] Emre Ugur and Erol Şahin. Traversability: A case study for learning and perceiving affordances in robots. Adaptive Behavior, 2010.

[3] Panagiotis Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. Engineering Applications of Artificial Intelligence, pages 1373–1385, 2013.

[4] Jean-François Lalonde, Nicolas Vandapel, Daniel F. Huber, and Martial Hebert. Natural terrain classification using three-dimensional ladar data for ground robot mobility. Journal of Field Robotics, 2006.

[5] David Droeschel, Max Schwarz, and Sven Behnke. Continuous mapping and localization for autonomous navigation in rough terrain using a 3d laser scanner. Robotics and Autonomous Systems, pages 104–115, 2017.

[6] Maggie Wigness, John G. Rogers, and Luis E. Navarro-Serment. Robot navigation from human demonstration: Learning control behaviors. IEEE International Conference on Robotics and Automation (ICRA), pages 1150–1157, 2018.

[7] Honggu Lee, Kiho Kwak, and Sungho Jo. An incremental nonparametric bayesian clustering- based traversable region detection method. Autonomous Robots, pages 795–810, 2017.

[8] Dongshin Kim, Jie Sun, Sang Min Oh, James M. Rehg, and Aaron F. Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. IEEE International Conference on Robotics and Automation (ICRA), pages 518–525, 2006.

[9] Jahanzaib Shabbir and Tarique Anwer. A survey of deep learning techniques for mobile robot applications. arXiv preprint arXiv:1803.07608, 2018.

[10] Michael Shneier, Tommy Chang, Tsai Hong, Will Shackleford, Roger Bostelman, and James S. Albus. Learning traversability models for autonomous mobile vehicles. Autonomous Robots, page 69–86, 2008.

[11] Patrick Pfaff, Rudolph Triebel, and Wolfram Burgard. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. International Journal of Robotics Research, 2007.

[12] In So Kweon and Takeo Kanade. High-resolution terrain map from multiple sensor data. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), pages 278–292, 1992.

[13] Hans P. Moravec and Alberto Elfes. High resolution maps from wide angle sonar. IEEE International Conference on Robotics and Automation (ICRA), pages 116–121, 1985.

[14] François Denis, Rémi Gilleron, and Marc Tommasi. Text classifation from positive and unla- beled examples. 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), 2002.

[15] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. 14th International Conference on Knowledge Discovery and Data Mining (SIGKDD), pages 213–220, 2008.

[16] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. IEEE/RSJ International

Conference on Intelligent Robots and Systems (IROS), pages 2276–2282, 2006.

[17] Noriaki Hirose, Amir Sadeghian, Marynel Vázquez, Patrick Goebel, and Silvio Savarese. Gonet: A semi-supervised deep learning approach for traversability estimation. arXiv preprint arXiv:1803.03254, 2018.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. 27th International Conference on Neural Information Processing Systems (NIPS), pages 2672–2680, 2014.

[19] Oleksii Zhelo, Jingwei Zhang, Lei Tai, Ming Liu, and Wolfram Burgard. Curiosity- driven exploration for mapless navigation with deep reinforcement learning. arXiv preprint arXiv:1804.00456, 2018.

[20] S. Z. Li. Markov random field models in computer vision. European Conference on Computer Vision (ECCV), pages 361–370, 1994.

[21] Noriaki Hirose, Amir Sadeghian, Patrick Goebel, and Silvio Savarese. To go or not to go? a near unsupervised learning approach for robot navigation. arXiv preprint arXiv:1709.05439, 2017.

[22] Lei Tai, Shaohua Li, and Ming Liu. Autonomous exploration of mobile robots through deep neural networks. International Journal of Advanced Robotic Systems (IJARS), 2017.

[23] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458, 2015.

[24] Fjodor Van Veen. The neural network zoo, 2016. URL http://www.asimovinstitute.org/ neural-network-zoo/.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. 25th International Conference on Neural Information Process- ing Systems (NIPS), pages 1097–1105, 2012.

[27] Adit Deshpande. The 9 deep learning papers you need to know about (understanding cnns part 3), 2016. URL https://adeshpande3.github.io/ The-9-Deep-Learning-Papers-You-Need-To-Know-About.html.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2016.

[29] Adit Deshpande. A beginner's guide to understanding convolutional neural net- works, 2016. URL https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner' s-Guide-To-Understanding-Convolutional-Neural-Networks/.

[30] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. IEEE Conference on Computer Vi- sion and Pattern Recognition (CVPR), pages 2036–2043, 2009.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255, 2009.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhi- heng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei- Fei. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, pages 211–252, 2015.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–9, 2015.

[34] François Chollet. Xception: Deep learning with depthwise separable convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1251–1258, 2017.

[35] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. arXiv preprint arXiv:1602.07360, 2016.

[36] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural net- works for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[37] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014.

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Uni- fied, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779–788, 2016.

[39] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. European Conference on Computer Vision (ECCV), pages 21–37, 2016.

[40] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. International Journal of Computer Vision, pages 154–171, 2013.

[41] James Le. How to do semantic segmentation using deep learning, 2018. URL https://medium.com/nanonets/ how-to-do-image-segmentation-using-deep-learning-c673cc5862ef.

[42] DeLiang Wang. Visual scene segmentation. Handbook of Brain Theory and Neural Networks, pages 1215–1219, 2003.

[43] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for se- mantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3431–3440, 2015.

[44] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision, pages 302–321, 2019.

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. European Conference on Computer Vision (ECCV), pages 740–755, 2014.

[46] Vikas Gupta. Keras tutorial : Using pre-trained imagenet models, 2017. URL https://www. learnopencv.com/keras-tutorial-using-pre-trained-imagenet-models/.

[47] Adrian Rosebrocke. Imagenet: Vggnet, resnet, inception, and xcep- tion with keras, 2017. URL https://www.pyimagesearch.com/2017/03/20/ imagenet-vggnet-resnet-inception-xception-keras/.

[48] Noriaki Hirose, Amir Sadeghian, Fei Xia, Roberto Martin-Martin, and Silvio Savarese. Vunet: Dynamic scene view synthesis for traversability estimation using an rgb camera. IEEE Robotics and Automation Letters, pages 2062–2069, 2019.

[49] Pierre Sermanet, Raia Hadsell, Marco Scoffier, Matt Grimes, Jan Ben, Ayse Erkan, Chris Crudele, Urs Muller, and Yann LeCun. A multi-range architecture for collision-free off-road robot navigation. Journal of Field Robotics, 2009.

[50] Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. Journal of Field Robotics, 2009.

[51] Christopher J. Holder, Toby P. Breckon, and Xiong Wei. From on-road to off: Transfer learn- ing within a deep convolutional neural network for segmentation and classification of off-road scenes. European Conference on Computer Vision (ECCV), pages 149–162, 2016.

[52] Bosch, X. Muñoz, and J. Freixenet. Segmentation and description of natural outdoor scenes. Image and Vision Computing, pages 727–740, 2007.

[53] Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), pages 1915–1929, 2013.

[54] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3684–3692, 2018.

[55] Marin Oršić, Ivan Krešo, Petra Bevandić, and Siniša Šegvić. In defense of pre-trained ima- genet architectures for real-time semantic segmentation of road-driving images. arXiv preprint arXiv:1903.08469, 2019.

[56] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3354–3361, 2012.

[57] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, pages 303–338, 2010.

[58] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Ro- drigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele.

The cityscapes dataset for se- mantic urban scene understanding. IEEE Conference on Computer Vision and Pattern Recog- nition (CVPR), pages 3213–3223, 2016.

[59] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 633–641, 2017.

[60] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2881–2890, 2017.

[61] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), pages 834–848, 2018.

[62] Jason Brownlee. A gentle introduction to k-fold cross-validation, 2018. URL https:// machinelearningmastery.com/k-fold-cross-validation/.

[63] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. 27th International Conference on Neural Information Processing Systems (NIPS), pages 487–495, 2014.

[64] Personal communication with Katerina Maria Oikonomou. Institute of Informatics and Telecommunications, NCSR ”Demokritos”.