



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών

## Παραγοντοποίηση Τανυστή με Πληροφορίες Δικτύου για τον Εντοπισμό Ψευδών Ειδήσεων

*Διπλωματική εργασία*

Ευφροσύνη Παπαναστασίου

Επιβλέπων: Γιώργος Στάμου  
Αναπληρωτής Καθηγητής Ε.Μ.Π

*Αθήνα, Νοέμβριος 2019*





National Technical University of Athens  
School of Electrical and Computer Engineering  
Computer Science Division

# Network-based Tensor Factorization with Label Information for Fake News Detection

*Diploma Thesis*

Effrosyni Papanastasiou

**Supervisor:** Giorgos Stamou  
Associate Professor, NTUA

*Athens, November 2019*





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών

## Παραγοντοποίηση Τανυστή με Πληροφορίες Δικτύου για τον Εντοπισμό Ψευδών Ειδήσεων

*Διπλωματική εργασία*

Ευφροσύνη Παπαναστασίου

Επιβλέπων: Γιώργος Στάμου  
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25<sup>η</sup> Νοεμβρίου 2019

.....  
Γιώργος Στάμου  
Αναπληρωτής Καθηγητής Ε.Μ.Π.

.....  
Ανδρέας-Γεώργιος Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

.....  
Νικόλαος Παπασπύρου  
Καθηγητής Ε.Μ.Π.

*Αθήνα, Νοέμβριος 2019*

.....  
Ευφροσύνη Παπαναστασίου  
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ευφροσύνη Παπαναστασίου, 2019  
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Η διάδοση των "fake news", δηλαδή των ψευδών ειδήσεων που έχουν γραφτεί εσκεμμένα ώστε να παραπληροφορήσουν τους αναγνώστες, πρόκειται για ένα φαινόμενο που έχει αποκτήσει μεγάλες διαστάσεις τα τελευταία χρόνια εξαιτίας της εκτεταμένης χρήσης των μέσων κοινωνικής δικτύωσης για ενημερωτικούς σκοπούς. Λόγω των σοβαρών επιπτώσεων που επιφέρει η μαζική παραπληροφόρηση, θεωρείται επιτακτική η ανάγκη επίλυσης του προβλήματος, με τη χρήση κυρίως αυτοματοποιημένων μεθόδων. Οι περισσότερες μελέτες επικεντρώνονται στον αυτόματο εντοπισμό των ψευδών ειδήσεων μέσω ανάλυσης του κειμένου τους. Αντίθετα, σκοπός της παρούσας διπλωματικής εργασίας είναι η ανίχνευση των ψευδών ειδήσεων με τη χρήση στοιχείων που βασίζονται μόνο σε πληροφορίες δικτύου και πιο συγκεκριμένα στις φιλίες των χρηστών που αναδημοσιεύουν τις ειδήσεις. Θεωρούμε πως οι χρήστες που δημοσιεύουν ψευδείς ειδήσεις σχηματίζουν μεταξύ τους διακριτά δίκτυα σχέσεων που μπορούν να βοηθήσουν στην ταξινόμηση των ειδήσεων σε αληθείς και ψευδείς. Για τον λόγο αυτό, μοντελοποιούμε κάθε είδηση ως ένα δίκτυο φιλιών και αναπαριστάνουμε το σύνολό τους με τη χρήση τρισδιάστατων ταυσοτών (tensors). Έχοντας στη διάθεσή μας ένα σύνολο επισημασμένων ειδήσεων, προτείνουμε μία νέα τεχνική παραγοντοποίησης ή αλλιώς αποσύνθεσης ταυσοτών (tensor factorization) που, αντί να εκτελεί την ταξινόμηση σε ξεχωριστό βήμα μετά τη δημιουργία των πινάκων παραγόντων (factor matrices) όπως συνηθίζεται, ενσωματώνει έναν επιπλέον όρο σφάλματος ταξινόμησης στη διαδικασία της παραγοντοποίησης που μας οδηγεί κατευθείαν και με μεγαλύτερη ακρίβεια στην τελική πρόβλεψη. Έτσι, καταλήγουμε σε ένα ενιαίο πρόβλημα βελτιστοποίησης με επίγνωση της κλάσης για μερικά από τα δεδομένα τη στιγμή της παραγοντοποίησης. Αποτελέσματα σε πραγματικά δεδομένα επιβεβαιώνουν την αποτελεσματικότητα της προτεινόμενης μεθόδου και αποδεικνύουν πως αποδίδει εξίσου καλά με μεθόδους τελευταίας τεχνολογίας μέσω μίας πιο απλής προσέγγισης.

### Λέξεις Κλειδιά

παραγοντοποίηση ταυσοτή, ψευδείς ειδήσεις, κοινωνικά δίκτυα, μηχανική μάθηση





Social media usage for news consumption has grown excessively during the last years. Along with it came the spread of large amounts of "fake news" that are deliberately written to deceive readers. Due to the serious consequences of mass misinformation, fake news detection is considered an urgent technological issue in need of automated solutions. In related literature, we can find several attempts to detect fake news only from its content, but their performance is generally considered non-satisfactory. Hence, we need to incorporate auxiliary information, coming from the network, in order to be able to identify misinformation more accurately. Our intuition is that the network connections of users that share fake news can be discriminatory enough to facilitate the detection of fake news. Consequently, the aim of this thesis is to detect fake news using network-based information and some labeled data. Firstly, we present a novel tensor-based way of representing the news pieces using only the friendships between users who have shared the news. Then, we wish to exploit this representation and a set of labels we have in hand, by developing a tensor factorization method that associates the class information of some posts with their latent representations. Finally, we end up with a unified optimization process that integrates a classification error term inside the factorization itself. Results on real-world datasets demonstrate the effectiveness of our proposed method and suggest that it can become competitive against existing state-of-the-art methods, by employing an arguably simpler approach.

### Keywords

tensor factorization, fake news, social networks, joint optimization, machine learning, CP decomposition.



Η εργασία αυτή είναι το αποτέλεσμα συλλογικής δουλειάς και μίας συνεργασίας που σε όλα τα στάδιά της λειτούργησε κατά την άποψή μου με τρόπο υποδειγματικό. Οφείλω λοιπόν ένα μεγάλο ευχαριστώ στον Δρ. Γιώργο Παλιούρα και τον Γιώργο Κατσιμπρα, ερευνητές στο Ινστιτούτο Πληροφορικής Τηλεπικοινωνιών του Έ.Κ.Ε.Φ.Ε' Δημόκριτος, που με βοήθησαν στην εκπόνηση της διπλωματικής αυτής με τεράστια όρεξη και προθυμία, αφιερώνοντας άπλετο χρόνο και σκέψη για την παραγωγή ενός αποτελέσματος που να έχει αξία. Τους ευχαριστώ γιατί μέσα από τις συμβουλές και τις οδηγίες τους με έκαναν να νιώσω ισάξιο μέλος μιας πραγματικής ερευνητικής ομάδας και μέσα από την προφανή αγάπη που έχουν για το αντικείμενό τους, με έκαναν να αγαπήσω και εγώ με τη σειρά μου την όλη ερευνητική διαδικασία. Όποιο εμπόδιο κι αν προέκυπτε ήταν πάντα σε ετοιμότητα, στην κυριολεξία με χαρτί και με μολύβι ώστε να το ξεπεράσουμε, και με κύριο μέλημα να βοηθήσουν εμένα την ίδια να κατανοήσω σε βάθος όλες τις ερευνητικές και μαθηματικές έννοιες. Αυτή η συνεργασία μου έδωσε τεράστια ώθηση και ενθουσιασμό να συνεχίσω σε μία ανάλογη ερευνητική πορεία. Θα ήθελα επίσης να ευχαριστήσω τον αναπληρωτή καθηγητή του ΕΜΠ Δρ. Γιώργο Στάμου για την εμπιστοσύνη που μου έδειξε αναλαμβάνοντας αυτήν την εργασία και την επίσης μεγάλη προθυμία του να με βοηθήσει σε θέματα διαδικαστικά καθώς και στον τρόπο παρουσίασης της διπλωματικής.

Σίγουρα όλα αυτά δεν θα μπορούσαν να είχαν συμβεί χωρίς την αμέριστη συμπαράσταση, ενθάρρυνση και στήριξη, υλική και κυρίως συναισθηματική, από τους γονείς μου Κατερίνα και Γιώργο και τον αδερφό μου Παντελή, σε όλα τα στάδια της ζωής και των σπουδών μου και για αυτό τον λόγο η εργασία είναι αφιερωμένη σε αυτούς. Ό,τι έχω καταφέρει μέχρι στιγμής και ό,τι πρόκειται να καταφέρω θα είναι χάρη σε εκείνους. Φυσικά ευχαριστώ τις φίλες μου, που έκαναν υπέροχα τα φοιτητικά μου χρόνια, και τους συμφοιτητές μου, που με έμαθαν να μην σταματάω να παλεύω ποτέ για τις ιδέες μου και ένα καλύτερο μέλλον καθώς και για τις αμέτρητες ώρες διαβάσματος που μοιραστήσαμε παρέα βοηθώντας ο ένας τον άλλον. Τέλος, την εργασία αυτή θα ήθελα να την αφιερώσω και στην κολητή μου φίλη Όλγα που ήταν πάντα τόσα χρόνια δίπλα μου, και που τα λόγια δεν αρκούν για να περιγράψουν πόσο τυχερή είμαι που την έχω στη ζωή μου.



## Περιεχόμενα (Table of Contents)

---

Περίληψη	1
Abstract	3
Ευχαριστίες	5
Περιεχόμενα (Table of Contents)	8
Σχήματα (List of Figures)	9
Πίνακες (List of Tables)	11
<b>1 Εκτενής Ελληνική Περίληψη (Greek Summary)</b>	<b>13</b>
1.1 Αντικείμενο της Διπλωματικής . . . . .	13
1.2 Συνεισφορά . . . . .	14
1.3 Θεωρητικό Υπόβαθρο . . . . .	14
1.3.1 Παραγοντοποίηση τανυστή . . . . .	14
1.3.2 Σημειογραφία . . . . .	15
1.3.3 Παραγοντοποίηση RESCAL . . . . .	15
1.3.4 Παραγοντοποίηση CP . . . . .	17
1.4 Ορισμός του Προβλήματος . . . . .	17
1.5 Προτεινόμενη Μέθοδος 1: Παραγοντοποίηση NN-RESCAL . . . . .	17
1.6 Προτεινόμενη Μέθοδος 2: CLASS-CP . . . . .	19
1.7 Πειραματική Αξιολόγηση . . . . .	22
1.7.1 Δεδομένα . . . . .	22
1.7.2 Πειραματικές ρυθμίσεις . . . . .	22
1.7.3 Αποτελέσματα . . . . .	23
1.8 Συμπεράσματα και Μελλοντικές Προεκτάσεις . . . . .	24
<b>2 Introduction</b>	<b>27</b>
2.1 Thesis Objective . . . . .	28
2.2 Contribution . . . . .	28
2.3 Thesis Layout . . . . .	29
<b>3 Related Work</b>	<b>31</b>
3.1 Fake News Detection . . . . .	31
3.1.1 Content-based . . . . .	31
3.1.2 Social context-based . . . . .	32
3.2 Joint Factorization . . . . .	33
<b>4 Preliminaries</b>	<b>35</b>
4.1 Notation . . . . .	35
4.2 Tensor Factorization . . . . .	35
4.3 The CP Factorization . . . . .	36

---

4.4	The RESCAL Factorization . . . . .	37
4.5	Non-Negative RESCAL Factorization . . . . .	38
4.6	Problem Definition . . . . .	38
<b>5</b>	<b>Network-Based Tensor Factorization with Label Information</b>	<b>39</b>
5.1	Method Summary . . . . .	39
5.2	Model 1: NN-RESCAL Factorization . . . . .	39
5.3	Model 2: CLASS-CP Factorization . . . . .	41
5.3.1	Method overview . . . . .	42
5.3.2	Solving the joint optimization problem . . . . .	42
<b>6</b>	<b>Experiments &amp; Results</b>	<b>47</b>
6.1	Datasets . . . . .	47
6.2	Preliminary Results . . . . .	47
6.3	Experimental Settings . . . . .	48
6.4	Comparison with Other Methods . . . . .	50
6.5	Results . . . . .	51
<b>7</b>	<b>Conclusion</b>	<b>55</b>
	<b>References</b>	<b>57</b>

## Σχήματα (List of Figures)

---

4.1	CP Decomposition of a Three-Way Array [26] . . . . .	37
4.2	RESCAL Factorization into the Factor Matrix $A$ and the core Tensor $R_k$ [49]	37
5.1	The Conceptual Diagram of CLASS-CP . . . . .	42
6.1	The Accuracy Learning Curve of NN-RESCAL for $r=5,10,20$ on PolitiFact .	48
6.2	Behavior of CLASS-CP with Different Numbers of Factors for each Dataset	49
6.3	The F1-score Learning Curve on BuzzFeed and PolitiFact . . . . .	49
6.4	Learning Curves for Precision and Recall . . . . .	50
6.5	Performance Comparison for Different Metrics . . . . .	52





## Πίνακες (List of Tables)

---

4.1	Table of Notation . . . . .	36
6.1	Numbers of FakeNewsNet After Deletion of Nodes (engagements refer to posts, re-posts and replies related with a news article) . . . . .	47
6.2	Performance of NN-RESCAL for $r=5$ . . . . .	48
6.3	Performance Comparison on PolitiFact . . . . .	51
6.4	Performance Comparison on BuzzFeed . . . . .	51
6.5	Performance of Content-SVM . . . . .	51



Η αξιοσημείωτη αύξηση της χρήσης των μέσων κοινωνικής δικτύωσης (social media) ως βασική πηγή ενημέρωσης σε συνδυασμό με τον κλονισμό της εμπιστοσύνης στα παραδοσιακά μέσα μαζικής επικοινωνίας, τη μεγάλη ταχύτητα διάδοσης των ψηφιακών ειδήσεων καθώς και τον τεράστιο όγκο της πληροφορίας που κυκλοφορεί στο διαδίκτυο, έχουν επιδεινώσει το πρόβλημα των λεγόμενων ψευδών ειδήσεων (fake news). Πανερωπαϊκή έρευνα 26.000 ατόμων που διεξήχθη το 2018, επιβεβαιώνει ότι οι ψευδείς ειδήσεις εξαπλώνονται ταχύτατα σε ολόκληρη την Ευρώπη, με το 85% των συμμετεχόντων να θεωρεί τις ψευδείς ειδήσεις ένα σοβαρό πρόβλημα που χρήζει αντιμετώπισης [24]. Κατά αυτόν τον τρόπο, καταδεικνύεται η αναγκαιότητα ανάπτυξης αυτόματων μεθόδων ανίχνευσης των ψευδών ειδήσεων με βασικό στόχο την προστασία των αναγνωστών από την παραπληροφόρηση.

Η δυσκολία της αυτόματης ανίχνευσης ψευδών ειδήσεων στα social media οφείλεται κυρίως σε δύο συνιστώσες: (i) οι ψευδείς ειδήσεις είναι σκοπίμως γραμμένες κατά τρόπο που μπορεί να εξαπατήσει τους αναγνώστες (ii) το περιεχόμενο των ειδήσεων μπορεί να ποικίλλει όσον αφορά τη θεματολογία και τον τρόπο γραφής. Για παράδειγμα, ένα αληθινό γεγονός μπορεί να παρουσιαστεί μέσα σε ένα παραπλανητικό πλαίσιο για να υποστηριχθεί κάτι ψευδές [6, 7]. Κατά συνέπεια, ένα μοντέλο μηχανικής μάθησης που ανιχνεύει ψευδείς ειδήσεις αποκλειστικά μέσα από το περιεχόμενο, μπορεί να μην προσφέρει τα καλύτερα αποτελέσματα. Αντ' αυτού, μπορούμε να επωφεληθούμε από συμπληρωματικές πληροφορίες που παράγονται από το δίκτυο, όπως για παράδειγμα το προφίλ των χρηστών που αλληλεπιδρούν με τις ειδήσεις στα social media, για να εξάγουμε ακριβέστερα αποτελέσματα.

Το πρόβλημα της ανίχνευσης ψευδών ειδήσεων στα social media έχει μελετηθεί εκτενώς στη βιβλιογραφία. Μπορούμε να διακρίνουμε τρεις βασικές προσεγγίσεις οι οποίες βασίζονται: i) στο περιεχόμενο ii) στο δίκτυο και iii) στο περιεχόμενο και στο δίκτυο (υβριδικές προσεγγίσεις). Οι προσεγγίσεις που βασίζονται μόνο στο περιεχόμενο επικεντρώνονται στην ανάλυση του περιεχομένου των ειδήσεων (π.χ. λεξιλόγιο, σύνταξη) και στην ανίχνευση μοτίβων μέσω μεθόδων επεξεργασίας φυσικής γλώσσας (NLP) [8, 9, 28]. Ωστόσο, προκειμένου να παραχθούν ικανοποιητικά αποτελέσματα, απαιτείται προκαθορισμένο πεδίο ορισμού, κάτι που είναι δύσκολο να επιτευχθεί στην περίπτωση των ψευδών ειδήσεων εξαιτίας της ποικιλομορφίας τους [11].

Οι προσεγγίσεις που βασίζονται στο δίκτυο εξάγουν πληροφορίες από διαφορετικούς τύπους δικτύων τα οποία δημιουργούνται από τους χρήστες που αλληλεπιδρούν με τις ειδήσεις, όπως είναι τα δίκτυα διάδοσης (diffusion networks) ή τα δίκτυα σχέσεων (relationship networks) [7]. Γενικά, οι μέθοδοι που βασίζονται στο δίκτυο έχουν καλύτερες επιδόσεις σε δυναμικά περιβάλλοντα και επομένως είναι καταλληλότερες για τον εντοπισμό ψευδών ειδήσεων [12]. Τέλος, οι υβριδικές προσεγγίσεις στοχεύουν να συνδυάσουν τα πλεονεκτήματα των μοντέλων που βασίζονται στο περιεχόμενο και στο δίκτυο, ενσωματώνοντας σε ένα μοντέλο τόσο τα γλωσσικά χαρακτηριστικά όσο και τις πληροφορίες του δικτύου [13].

## 1.1 Αντικείμενο της Διπλωματικής

Τα παραπάνω σε συνδυασμό με το ότι i) οι χρήστες διαδραματίζουν καθοριστικό ρόλο στον τρόπο διάδοσης των ψευδών ειδήσεων [7] ii) οι χρήστες μοιράζονται πληροφορίες με

άλλους παρόμοιους χρήστες (φίλους) [14] και iii) τα χαρακτηριστικά του δικτύου είναι σημαντικά στην κατηγοριοποίηση ή ταξινόμηση των ψευδών ειδήσεων [15], υποδηλώνουν ότι η διερεύνηση των δικτύων σχέσεων μεταξύ των χρηστών θα μπορούσε να διευκολύνει σημαντικά την κατηγοριοποίηση των ψευδών ειδήσεων. Ως αποτέλεσμα, στην παρούσα διπλωματική εργασία, στοχεύουμε να εκμεταλλευτούμε τις πληροφορίες που προέρχονται από τις φιλίες μεταξύ των χρηστών, και να αποδείξουμε τη σπουδαιότητά τους στο πρόβλημα της ανίχνευσης των ψευδών ειδήσεων.

Στις περισσότερες περιπτώσεις, οι τυπικές μέθοδοι παραγοντοποίησης ή αποσύνθεσης τανυστή εκτελούνται σε μη-επιβλεπόμενο περιβάλλον και δεν επηρεάζονται από την πληροφορία της κλάσης που μπορεί να είναι διαθέσιμη για μερικά από τα δεδομένα. Αντιθέτως, δικός μας στόχος είναι με τη χρήση μεθόδων παραγοντοποίησης τανυστή, να κατασκευάσουμε πίνακες συντελεστών με επίγνωση κλάσης. Θεωρούμε πως η χρήση της πληροφορίας της κλάσης κατά τη διάρκεια, και όχι μετά, τη διαδικασία της παραγοντοποίησης, μπορεί να οδηγήσει στην παραγωγή ακριβέστερων αναπαραστάσεων για τον εντοπισμό των ψευδών ειδήσεων με μεγαλύτερη αξιοπιστία.

## 1.2 Συνεισφορά

Σε αυτή τη διπλωματική εργασία προτείνουμε μια νέα ημι-επιβλεπόμενη προσέγγιση για την ταξινόμηση των ψευδών ειδήσεων στα social media, την οποία ονομάζουμε CLASS-CP. Η μέθοδος αυτή λαμβάνει υπόψη τις σχέσεις φιλίας μεταξύ των χρηστών και την πληροφορία της κλάσης (ετικέτες κλάσης). Αντί να εκτελέσουμε την παραγοντοποίηση και την ταξινόμηση ξεχωριστά, όπως συνηθίζεται, η CLASS-CP τις ενσωματώνει σε μια ενιαία διαδικασία βελτιστοποίησης. Οι κύριες συμβολές αυτής της εργασίας είναι οι παρακάτω:

- Προτείνουμε ένα νέο τρόπο αναπαράστασης των ειδήσεων με τανυστές μέσω των φιλιών των χρηστών
- Αναπτύσσουμε μία διαδικασία παραγοντοποίησης τανυστή για την ανίχνευση ψευδών ειδήσεων που βασίζεται σε πληροφορίες δικτύου και σε ετικέτες κλάσης
- Εισάγουμε την ταξινόμηση στη διαδικασία της παραγοντοποίησης, ενσωματώνοντας μέσα σε αυτήν την πληροφορία της κλάσης
- Αξιολογούμε την απόδοση σε πραγματικά σύνολα δεδομένων και επιδεικνύουμε την αποτελεσματικότητα της μεθόδου

Ο κώδικας για την προτεινόμενη μέθοδο είναι ανοιχτός και διαθέσιμος στο <https://github.com/FrossoPap/class-cp/>.

## 1.3 Θεωρητικό Υπόβαθρο

### 1.3.1 Παραγοντοποίηση τανυστή

Ένας τανυστής είναι ένας πολυδιάστατος πίνακας ή ένα γενικευμένο διάνυσμα. Η τάξη (order) ενός τανυστή είναι ο αριθμός των διαστάσεών του, επομένως ένας τανυστής πρώτης τάξης είναι διάνυσμα, ένας τανυστής δεύτερης τάξης είναι διδιάστατος πίνακας, ενώ οι τανυστές τρίτης τάξης και άνω ονομάζονται τανυστές υψηλότερης τάξης. Ένας τανυστής  $r$ -τάξης ονομάζεται *πρώτου βαθμού* (rank-one) εάν αποσυντίθεται αυστηρά στο εξωτερικό γινόμενο  $r$  διανυσμάτων.

Η παραγοντοποίηση τανυστή χρησιμοποιείται εκτενώς σε προβλήματα μηχανικής μάθησης για την αναπαράσταση πολυδιάστατων δεδομένων. Μια τυπική προσέγγιση παραγοντοποίησης τανυστή είναι η Canonical/Parafac ή CP [16], που αποσυνθέτει τον τανυστή σε ένα σύνολο rank-one τανυστών, όπως περιγράφεται λεπτομερέστερα στην παράγραφο 1.3.4. Μια ακόμη παραγοντοποίηση που χρησιμοποιείται ευρέως, είναι η Tucker [17], η οποία αποσυνθέτει τον τανυστή σε ένα σύνολο ξεχωριστών πινάκων συντελεστών (factor matrices) για κάθε τάξη τανυστή και σε έναν τανυστή-πυρήνα (core tensor). Μια πιο πρόσφατη προσέγγιση είναι ο αλγόριθμος RESCAL [18] που παράγει μια μοναδική αναπαράσταση για κάθε διαφορετική σχέση που παρουσιάζουν τα δεδομένα μεταξύ τους και παρουσιάζεται αναλυτικότερα στην παράγραφο 1.3.3.

### 1.3.2 Σημειογραφία

Στην παρούσα διπλωματική εργασία θα χρησιμοποιήσουμε την ακόλουθη σημειογραφία: τα κεφαλαία καλλιγραφικά γράμματα δηλώνουν έναν τανυστή  $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ . Οι διδιάστατοι πίνακες αντιπροσωπεύονται από κεφαλαία γράμματα, όπως το  $A$ . Τα μικρά γράμματα, όπως το  $\mathbf{v}$ , υποδηλώνουν διανύσματα. Το στοιχείο  $(i, j)$  ενός πίνακα  $A$  σημειώνεται με  $a_{ij}$ . Για να ανατρέξουμε στη γραμμή  $i$  ενός πίνακα  $A$  χρησιμοποιούμε το  $\mathbf{a}_i$ . Ομοίως, ένα στοιχείο  $(i, j, k)$  ενός τανυστή  $\mathcal{T}$  θα σημειωθεί ως  $\mathcal{T}_{ijk}$ . Επιπλέον, το  $\text{vec}(A)$  είναι η διανυσματοποίηση του  $A$ , ο τελεστής  $\otimes$  δηλώνει το γινόμενο Kronecker και ο τελεστής  $\odot$  το γινόμενο Khatri-Rao.

#### Γινόμενο Kronecker

Αν ο  $A$  είναι ένας πίνακας  $m \times n$  και ο  $B$  ένας πίνακας  $p \times q$  τότε το γινόμενο Kronecker  $A \otimes B$  ορίζεται ως:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \quad (1.1)$$

#### Γινόμενο Khatri-Rao

Το γινόμενο Khatri-Rao ορίζεται ως εξής:

$$\mathbf{A} \odot \mathbf{B} = (\mathbf{A}_{ij} \otimes \mathbf{B}_{ij})_{ij} \quad (1.2)$$

και πιο αναλυτικά,

$$\mathbf{A} \odot \mathbf{B} = \left[ \begin{array}{c|c} \mathbf{A}_{11} \otimes \mathbf{B}_{11} & \mathbf{A}_{12} \otimes \mathbf{B}_{12} \\ \hline \mathbf{A}_{21} \otimes \mathbf{B}_{21} & \mathbf{A}_{22} \otimes \mathbf{B}_{22} \end{array} \right] \quad (1.3)$$

### 1.3.3 Παραγοντοποίηση RESCAL

Η παραγοντοποίηση RESCAL [18], αποτελεί ένα νέο μοντέλο παραγοντοποίησης τανυστή που κλιμακώνεται πολύ καλά σε μεγάλο αριθμό δεδομένων και μπορεί να παράγει state-of-the-art αποτελέσματα σε πολλά προβλήματα μηχανικής μάθησης [46, 62, 64]. Αναλυτικότερα, δεδομένου ενός τανυστή  $\mathcal{X}$ , κάθε slice του  $X_k$  παραγοντοποιείται σύμφωνα με το [18] ως εξής:

$$X_k \approx AR_kA^T, k = 1, \dots, m \quad (1.4)$$

όπου  $A$  είναι ένας πίνακας  $n \times r$  που περιέχει τις κρυφές αναπαραστάσεις των  $n$  οντοτήτων του προβλήματος και  $R_k$  είναι ένας πίνακας  $r \times r$  με τις κρυφές αλληλεπιδράσεις των  $r$  παραγόντων (factors) στη  $k$  διάσταση του ταυστή. Οι πίνακες  $A$  και  $R_k$  υπολογίζονται με την επίλυση του ακόλουθου προβλήματος ελαχιστοποίησης:

$$\min_{A, R_k} f(A, R_k) + g(A, R_k) \quad (1.5)$$

όπου

$$f(A, R_k) = \frac{1}{2} \sum_k \|X_k - AR_k A^T\|_f^2 \quad (1.6)$$

είναι το πρόβλημα ελαχίστων τετραγώνων της παραγοντοποίησης και

$$g(A, R_k) = \frac{1}{2} \lambda (\|A\|_f^2 + \sum_k \|R_k\|_f^2) \quad (1.7)$$

είναι ο όρος κανονικοποίησης που προστίθεται για να αποφευχθεί η υπερμοντελοποίηση (overfitting) του μοντέλου.

### Μη αρνητική παραγοντοποίηση RESCAL

Τα σχεσιακά δεδομένα όπως αυτά που παράγονται από τα κοινωνικά δίκτυα μπορούν γενικά να θεωρηθούν ως μη αρνητικά [49]. Επομένως, τα δεδομένα στο δικό μας πρόβλημα μπορούν να παραγοντοποιηθούν χρησιμοποιώντας μια μη αρνητική εκδοχή του RESCAL [49]. Αρχικά, αναπαριστάνουμε τα δεδομένα χρησιμοποιώντας έναν ταυστή τρίτης τάξης,  $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ . Κάθε slice του  $T_k$  παραγοντοποιείται ως εξής:

$$T_k \approx AR_k A^T, \quad k = 1, \dots, m \quad (1.8)$$

όπου  $A$  είναι ένας πίνακας  $n \times r$  που περιέχει τις κρυφές αναπαραστάσεις των  $n$  οντοτήτων και  $R_k$  είναι ένας πίνακας  $r \times r$  με τις κρυφές αλληλεπιδράσεις των  $r$  παραγόντων (factors) στη  $k$  διάσταση του ταυστή. Οι παράμετροι  $A$  και  $R_k$  υπολογίζονται με την επίλυση του ακόλουθου προβλήματος ελαχιστοποίησης:

$$\min_{A, R_k} f(A, R_k) + g(A, R_k) \quad (1.9)$$

όπου

$$f(A, R_k) = \sum_k \|T_k - AR_k A^T\|_f^2 \quad (1.10)$$

είναι το πρόβλημα ελαχίστων τετραγώνων της παραγοντοποίησης και

$$g(A, R_k) = \lambda_A \|A\|_f^2 + \lambda_R \sum_k \|R_k\|_f^2 \quad (1.11)$$

ο όρος κανονικοποίησης. Οι μη αρνητικές ενημερώσεις για τους πίνακες  $A$  και  $R_k$  αντίστοιχα είναι, σύμφωνα με [49]:

$$A \leftarrow A * \frac{\sum_k T_k A R_k^T + T_k^T A R_k}{A([\sum_k R_k A^T A R_k^T + R_k^T A^T A R_k] + \lambda_A I)} \quad (1.12)$$

$$R_k \leftarrow R_k * \frac{A^T T_k A}{A^T A R_k A^T A + \lambda_R R_k} \quad (1.13)$$

Παρόλα αυτά, οι μη αρνητικές παραγοντοποιήσεις ταυστών προσθέτουν επιπλέον περιορισμούς οι οποίοι μπορεί να μας οδηγήσουν σε πολύπλοκους πίνακες συντελεστών που απαιτούν περισσότερο χρόνο για ενημέρωση, προκαλώντας ζητήματα κλιμάκωσης [60].

### 1.3.4 Παραγοντοποίηση CP

Η παραγοντοποίηση Canonical/Parafac (CP) παραγοντοποιεί έναν τανυστή σε ένα άθροισμα τανυστών πρώτου βαθμού. Για παράδειγμα, δεδομένου ενός τανυστή τρίτης τάξης  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ , θέλουμε να τον εκφράσουμε ως:

$$\mathcal{X} \approx \sum_{r=1}^R a_r \circ b_r \circ c_r \quad (1.14)$$

όπου  $R$  είναι ένας θετικός ακέραιος, ο το εξωτερικό γινόμενο των διανυσμάτων και  $a_r \in \mathbb{R}^I, b_r \in \mathbb{R}^J, c_r \in \mathbb{R}^K$  για  $r = 1, \dots, R$ . Ακολουθώντας τους ορισμούς των Kolda και Bader [26], η (1.14) μπορεί να γραφεί σε μορφή πίνακα για κάθε μία από τις τρεις τάξεις του τανυστή:

$$\begin{aligned} X_{(1)} &\approx A(C \odot B)^T, \\ X_{(2)} &\approx B(C \odot A)^T, \\ X_{(3)} &\approx C(B \odot A)^T. \end{aligned} \quad (1.15)$$

## 1.4 Ορισμός του Προβλήματος

Σε αυτό το πρόβλημα έχουμε ένα σύνολο posts/ειδήσεων  $P = (p_1, p_2, \dots, p_p)$  που έχουν κοινοποιηθεί από χρήστες, ένα σύνολο δικτύων φιλίας  $G = (g_1, g_2, \dots, g_p)$  μεταξύ αυτών των χρηστών και ένα σύνολο ετικετών κλάσης  $Y = (y_1, y_2, \dots, y_l)$ , με  $l < p$ . Στόχος μας είναι να προβλέψουμε αποτελεσματικά την κλάση των ειδήσεων χωρίς ετικέτα. Η λύση αυτού του προβλήματος εγείρει τα ακόλουθα ερευνητικά ερωτήματα:

- (E1) Πώς μπορούμε να επωφεληθούμε από τα δίκτυα φιλίας που δημιουργήσαμε για την αποτελεσματική ταξινόμηση των ειδήσεων;
- (E2) Πώς μπορούν οι διαθέσιμες ετικέτες κλάσης των ειδήσεων να οδηγήσουν σε ακριβέστερα αποτελέσματα;

## 1.5 Προτεινόμενη Μέθοδος 1: Παραγοντοποίηση NN-RESCAL

Το πρώτο μοντέλο που αναπτύξαμε χρησιμοποιεί τη μη αρνητική παραγοντοποίηση RESCAL που παρουσιάστηκε στην παράγραφο 1.3.3. Έστω ότι διαθέτουμε  $2n$  αριθμό ειδήσεων (posts) που έχουν αναρτηθεί στα social media, οι  $2p$  από αυτές με ετικέτες κλάσης και οι υπόλοιπες  $2(n - p)$  χωρίς, όπου  $p < n$ . Υπάρχει ίσος αριθμός ψευδών και αληθών ειδήσεων. Αρχικά, δημιουργούμε δύο τανυστές τρίτης τάξης χρησιμοποιώντας δυαδική αναπαράσταση για τη μοντελοποίηση των φιλιών των χρηστών:  $\mathcal{X}_{(fake)} \in \mathbb{R}^{u \times u \times p}$  που περιέχει το σύνολο των posts-ειδήσεων που έχουν επισημανθεί ως ψευδείς και  $\mathcal{X}_{(real)} \in \mathbb{R}^{u \times u \times p}$  με τις ειδήσεις που έχουν επισημανθεί ως αληθείς. Τα posts-ειδήσεις αναπαριστώνται με πίνακες γειτνίασης. Με τη στοίβαξη αυτών των πινάκων, τον έναν μετά το άλλο, διαμορφώνουμε τους τανυστές με τον ακόλουθο τρόπο:

$$\mathcal{X}_{(fake)}_{ijk} = \begin{cases} 1, & \text{αν ο χρήστης } j \text{ έχει αλληλεπιδράσει με το fake post } k \\ & \text{και ο χρήστης } i \text{ ακολουθεί τον χρήστη } j \\ 0, & \text{αλλιώς} \end{cases} \quad (1.16)$$

ομοίως,

$$\mathcal{X}_{(real)_{ijk}} = \begin{cases} 1, & \text{αν ο χρήστης } j \text{ έχει αλληλεπιδράσει με το real post } k \\ & \text{και ο χρήστης } i \text{ ακολουθεί τον χρήστη } j \\ 0, & \text{αλλιώς} \end{cases} \quad (1.17)$$

Σημειώνεται πως όταν δηλώνουμε ότι ένας χρήστης έχει αλληλεπιδράσει με ένα post, αυτό σημαίνει ότι έχει κάνει κοινοποίηση το αντίστοιχο post/είδηση στο προφίλ του στα social media, στην περίπτωση μας στο Twitter. Στη συνέχεια, εφαρμόζουμε τη μη αρνητική παραγοντοποίηση RESCAL στους τανυστές  $\mathcal{X}_{(fake)}$  και  $\mathcal{X}_{(real)}$ , με τον τρόπο που παρουσιάστηκε στην παράγραφο 1.3.3. Ως αποτέλεσμα της παραγοντοποίησης, καταλήγουμε στους πίνακες  $A_{fake} \in \mathbb{R}^{u \times r}$  και  $A_{real} \in \mathbb{R}^{u \times r}$  αντίστοιχα, μέσω των ακόλουθων μη αρνητικών ενημερώσεων που προκύπτουν σύμφωνα με τις εξισώσεις (1.12) και (1.13):

$$A_{fake} \leftarrow A_{fake} * \frac{\sum_k X_{(fake)k} A_{fake} R_k + X_{(fake)k}^T A_{fake} R_k}{A_{fake}([\sum_k R_k (A_{fake})^T A_{fake} R_k^T + R_k (A_{fake})^T A_{fake} R_k] + \lambda_{A_{fake}} I)} \quad (1.18)$$

όπου

$$R_k \leftarrow R_k * \frac{\sum_k A_{fake}^T X_{(fake)k} A_{fake}}{A_{fake}^T A_{fake} R_k (A_{fake})^T A_{fake} + \lambda_R R_k} \quad (1.19)$$

και

$$A_{real} \leftarrow A_{real} * \frac{\sum_k X_{(real)k} A_{real} R_k + X_{(real)k}^T A_{real} R_k}{A_{real}([\sum_k R_k (A_{real})^T A_{real} R_k^T + R_k (A_{real})^T A_{real} R_k] + \lambda_{A_{real}} I)} \quad (1.20)$$

όπου

$$R_k \leftarrow R_k * \frac{\sum_k A_{real}^T X_{(real)k} A_{real}}{A_{real}^T A_{real} R_k (A_{real})^T A_{real} + \lambda_R R_k} \quad (1.21)$$

Τα posts που δεν χρησιμοποιήθηκαν στο train set ως slices των τανυστών, ανήκουν στο test set και δεν έχουν ακόμα ετικέτα. Έστω ότι έχουμε ένα σύνολο πινάκων  $M = \{P_1, P_2, \dots, P_{2(n-p)}\}$ , με κάθε πίνακα να αντιστοιχεί σε ένα διαφορετικό post χωρίς ετικέτα, όπου  $P_{idx} \in \mathbb{R}^{u \times u}$  και  $idx \in [1, 2(n-p)]$ . Ο πίνακας  $P_{idx}$  δημιουργείται για κάθε post  $idx$  ως εξής:

$$P_{(idx)_{ij}} = \begin{cases} 1, & \text{αν ο χρήστης } j \text{ έχει αλληλεπιδράσει με το post } idx \\ & \text{και ο χρήστης } i \text{ ακολουθεί τον χρήστη } j \\ 0, & \text{αλλιώς} \end{cases} \quad (1.22)$$

Στη συνέχεια, προσθέτουμε το post  $P_{idx} \in \mathbb{R}^{u \times u}$  από το test set στους τανυστές  $\mathcal{X}_{(fake)}$  και  $\mathcal{X}_{(real)}$  ως ένα νέο slice στο τέλος του κάθε τανυστή. Τελικά, καταλήγουμε στους νέους τανυστές  $\mathcal{X}'_{(fake)} \in \mathbb{R}^{u \times u \times (p+1)}$  και  $\mathcal{X}'_{(real)} \in \mathbb{R}^{u \times u \times (p+1)}$  αντίστοιχα. Μετά την προσθήκη του test post και στους δύο τανυστές, εκτελούμε τη μη αρνητική παραγοντοποίηση RESCAL στους τανυστές  $\mathcal{X}'_{(fake)}$  και  $\mathcal{X}'_{(real)}$ , για να παραγάγουμε τους νέους πίνακες παραγόντων  $A'_{fake} \in \mathbb{R}^{u \times r}$  και  $A'_{real} \in \mathbb{R}^{u \times r}$  μέσω των ακόλουθων μη αρνητικών ενημερώσεων που προκύπτουν ξανά από τις εξισώσεις (1.12) και (1.13):

$$A'_{fake} \leftarrow A'_{fake} * \frac{\sum_k X'_{(fake)k} A'_{fake} R_k + X_{(fake)k}^T A'_{fake} R_k}{A'_{fake}([\sum_k R_k A_{fake}^T A'_{fake} R_k^T + R_k A_{fake}^T A'_{fake} R_k] + \lambda_{A'_{fake}} I)} \quad (1.23)$$



όπου

$$R_k \leftarrow R_k * \frac{\sum_k A'_{fake}{}^T X'_{(fake)k} A'_{fake}}{A'_{fake}{}^T A'_{fake} R_k A'_{fake}{}^T A'_{fake} + \lambda_R R_k} \quad (1.24)$$

και

$$A'_{real} \leftarrow A'_{real} * \frac{\sum_k X'_{(real)k} A'_{real} R_k + X'_{(real)k}{}^T A'_{real} R_k}{A'_{real}([\sum_k R_k A'_{real}{}^T A'_{real} R_k^T + R_k A'_{real}{}^T A'_{real} R_k] + \lambda_{A'_{real}} I)} \quad (1.25)$$

όπου

$$R_k \leftarrow R_k * \frac{\sum_k A'_{real}{}^T X'_{(real)k} A'_{real}}{A'_{real}{}^T A'_{real} R_k A'_{real}{}^T A'_{real} + \lambda_R R_k} \quad (1.26)$$

Δεδομένου ότι η ευκλείδεια απόσταση αποτελεί γνωστή μετρική για τον υπολογισμό της απόστασης που χρησιμοποιείται τακτικά σε παρόμοια προβλήματα υπολογισμού διαφοράς [50, 51, 52], υπολογίζουμε την πρόβλεψη με τον ακόλουθο τρόπο: Έστω  $l$  η ακόμα άγνωστη ετικέτα κάθε post, με  $l \in [0, 1]$ , όπου το 0 δείχνει ένα αληθές post και το 1 ένα ψευδές. Η ετικέτα του προκύπτει ως εξής:

$$l = \begin{cases} 1, & \text{εάν } \|A_{fake} - A'_{fake}\| < \|A_{real} - A'_{real}\| \\ 0, & \text{αλλιώς} \end{cases}$$

Τέλος, αφαιρούμε τον πίνακα  $P_{idx}$  από τους  $\mathcal{X}'_{(fake)}$  και  $\mathcal{X}'_{(real)}$ , προσθέτουμε τον επόμενο πίνακα  $P_{idx+1}$  από το σύνολο πινάκων  $M$  και στους δύο τανυστές και επαναλαμβάνουμε τη διαδικασία για τα υπόλοιπα  $2(n - p)$  posts. Ολόκληρος ο αλγόριθμος για τη διαδικασία φαίνεται στον Αλγόριθμο 1.

Ωστόσο, όπως φαίνεται στην παράγραφο 1.7, το μοντέλο αυτό δεν καταφέρνει να ανιχνεύσει με ακρίβεια τις ψευδείς ειδήσεις στα μέσα κοινωνικής δικτύωσης. Επιπλέον, για την παραγωγή των τελικών αποτελεσμάτων πρέπει να τρέξουμε τέσσερις φορές τον αλγόριθμο παραγοντοποίησης NN-RESCAL, γεγονός που αυξάνει την πολυπλοκότητα και τον συνολικό χρόνο υπολογισμού της τελικής πρόβλεψης. Συνεπώς, συνεχίζουμε με την ανάπτυξη ενός δεύτερου, και όπως θα δείξουμε πιο αποτελεσματικού, μοντέλου που παρουσιάζεται στην επόμενη παράγραφο.

## 1.6 Προτεινόμενη Μέθοδος 2: CLASS-CP

Με βάση τη σημειογραφία που παρουσιάστηκε στην παράγραφο 1.3.2, για τη μοντελοποίηση των δεδομένων θα χρησιμοποιήσουμε την ακόλουθη δυαδική αναπαράσταση με τανυστές:

$$\mathcal{T}_{ijk} = \begin{cases} 1, & \text{αν ο χρήστης } k \text{ έχει αλληλεπιδράσει με το post } i \\ & \text{και ο χρήστης } j \text{ ακολουθεί τον χρήστη } k \\ 0, & \text{αλλιώς} \end{cases}$$

Όπως και προηγουμένως, όταν ένας χρήστης αλληλεπιδρά με ένα post, αυτό σημαίνει ότι έχει κάνει κοινοποίηση το αντίστοιχο post/είδηση στο προφίλ του στα social media, στην περίπτωση μας στο Twitter.

Η προτεινόμενη μέθοδος πρόκειται για συνδυασμό δύο βασικών διαδικασιών που απαντούν στις ερωτήσεις (E1) και (E2) αντίστοιχα: (i) παραγοντοποίηση τανυστή και (ii) βελτιστοποίηση της απόδοσης με ενσωμάτωση ετικετών κλάσης. Σε αυτή την εργασία, χρησιμοποιούμε και επεκτείνουμε την πρότυπη παραγοντοποίηση CP [16]. Το διάγραμμα της CLASS-CP απεικονίζεται στο σχήμα 5.1. Καταρχήν, κατασκευάζουμε έναν τρισδιάστατο τανυστή από τα

δίκτυα φιλίας των χρηστών που έχουν αλληλεπιδράσει με τις ειδήσεις, δηλαδή  $\mathcal{T} \in \mathbb{R}^{p \times u \times u}$ . Η παραγοντοποίηση CP ενός τέτοιου τανυστή υπολογίζεται με την ακόλουθη μέθοδο των ελαχίστων τετραγώνων:

$$\min_{A,B,C} \|\mathcal{T} - \llbracket A, B, C \rrbracket\|^2,$$

όπου ο  $A \in \mathbb{R}^{p \times r}$  αναπαριστά τον πίνακα παραγόντων (factors) των ειδήσεων,  $B \in \mathbb{R}^{u \times r}$  και  $C \in \mathbb{R}^{u \times r}$  αναπαριστούν τους πίνακες παραγόντων των χρηστών και  $r$  είναι ο βαθμός (rank) της παραγοντοποίησης.

Για την εξαγωγή πληροφοριών που σχετίζονται με την κλάση ενσωματώνουμε στη διαδικασία βελτιστοποίησης έναν όρο σφάλματος κατηγοριοποίησης που βασίζεται στους παράγοντες του πίνακα  $A$ . Όπως θα δείξουμε, ο πίνακας συντελεστών  $A$  επηρεάζεται από έναν ταξινομητή. Συνδέοντας τις γραμμές του  $A$  (που αντιστοιχούν στις ειδήσεις) με τις ετικέτες κλάσης των αντίστοιχων ειδήσεων, λαμβάνουμε μια προσέγγιση του τανυστή  $\mathcal{T}$  που βασίζεται σε αυτές τις ετικέτες.

Η υπόθεσή μας είναι ότι οι ειδήσεις με ίδια κλάση θα μοιράζονται παρόμοιους παράγοντες [20]. Επομένως, ενσωματώνουμε την πληροφορία σχετικά με την κλάση στη διαδικασία της παραγοντοποίησης του τανυστή, ώστε να μετακινήσουμε ειδήσεις της ίδιας κλάσης πιο κοντά στη λανθάνουσα (latent) περιοχή του προβλήματος. Υποθέτουμε ότι υπάρχει ένας πίνακας συντελεστών  $W \in \mathbb{R}^{r \times c}$ , όπου  $r$  είναι ο αριθμός των παραγόντων και  $c$  είναι ο αριθμός των ετικετών κλάσης. Αυτός ο πίνακας συντελεστών αποδίδει ετικέτες στις ειδήσεις, βασιζόμενος στις γραμμές του πίνακα συντελεστών  $A$ , όπως συμβαίνει σε ένα κοινό πρόβλημα παρεμβολής (regression), δηλαδή  $Y = AW$ .

### Επίλυση του προβλήματος κοινής βελτιστοποίησης

Σε συνέχεια των παραπάνω, έχοντας έναν τανυστή  $\mathcal{T} \in \mathbb{R}^{p \times u \times u}$  και ένα σύνολο ετικετών  $Y \in \mathbb{R}^{1 \times c}$ , όπου  $l$  ο αριθμός των ειδήσεων με ετικέτες με  $l < p$ , και  $c$  ο αριθμός των ετικετών, λύνουμε το πρόβλημα βελτιστοποίησης με την παρακάτω εξίσωση:

$$\min_{A,B,C,W} f(A, B, C) + g(A, W) \quad (1.27)$$

όπου

$$f(A, B, C) = \|\mathcal{T} - \llbracket A, B, C \rrbracket\|^2$$

το πρόβλημα των ελαχίστων τετραγώνων για την παραγοντοποίηση του τανυστή και

$$g(A, W) = \lambda_g \|Y - DAW\|^2 \quad (1.28)$$

είναι το σφάλμα πρόβλεψης του ταξινομητή, όπου  $D = [I^{l \times l}, 0^{l \times (p-l)}] \in \mathbb{R}^{1 \times p}$  και  $\lambda_g$  είναι μία υπερ-παραμέτρος για τον έλεγχο της επίδρασης του σφάλματος ταξινόμησης στη βελτιστοποίηση.

Λύνουμε το πρόβλημα ελαχιστοποίησης της εξίσωσης (1.27), χρησιμοποιώντας την αποδοτική μέθοδο εναλλασσόμενων ελαχίστων τετραγώνων (ALS). Στην παραγοντοποίηση CP, η μέθοδος ALS διατηρεί σταθερούς τους πίνακες παραγόντων εκτός από έναν και λύνει ως προς αυτόν. Η  $f$  σε μορφή πίνακα (ένας πίνακας για κάθε τάξη) μπορεί να λυθεί ως εξής:

$$\begin{aligned} T_{(1)} &\approx A(C \odot B)^T, \\ T_{(2)} &\approx B(C \odot A)^T, \\ T_{(3)} &\approx C(B \odot A)^T. \end{aligned} \quad (1.29)$$

### Ενημέρωση πινάκων συντελεστών B και C

Για την ενημέρωση των πινάκων  $B$  και  $C$ , αγνοούμε τον δεύτερο όρο της εξίσωσης (1.27) και λύνουμε τις αντίστοιχες εξισώσεις όπως στην εξίσωση (1.29). Η προσέγγιση ALS κρατάει σταθερούς τους  $A$ ,  $C$  λύνοντας ως προς τον  $B$ , καθώς και τους  $A$ ,  $B$  λύνοντας ως προς τον  $C$ . Αν  $Z_{(2)} = (C \odot A)$  και  $Z_{(3)} = (B \odot A)$ , τότε από τις (1.27) και (1.29) μπορούμε να γράψουμε:

$$\min_B \|T_{(2)} - BZ_{(2)}^T\|^2 \quad (1.30)$$

και

$$\min_C \|T_{(3)} - CZ_{(3)}^T\|^2. \quad (1.31)$$

Η λύση στις (1.30) και (1.31) δίνεται από [26] οπότε οι ενημερώσεις των  $B$  και  $C$  μπορούν να γραφτούν ως:

$$B = (Z_{(2)}^T Z_{(2)})^{-1} Z_{(2)}^T T_{(2)} \quad (1.32)$$

και

$$C = (Z_{(3)}^T Z_{(3)})^{-1} Z_{(3)}^T T_{(3)} \quad (1.33)$$

### Ενημέρωση πίνακα συντελεστών A

Έστω  $Z_{(1)} = (C \odot B)$ . Για την ενημέρωση του πίνακα συντελεστών  $A$ , συνδυάζουμε (1.27) και (1.29) και γράφουμε:

$$\min_A \|T_{(1)} - AZ_{(1)}^T\|^2 + \|Y - DAW\|^2. \quad (1.34)$$

Η εξίσωση (1.34) μπορεί να λυθεί πιο εύκολα αν τη λύσουμε στη διανυσματοποιημένη μορφή της. Τότε η (1.34) μπορεί να γραφτεί ως:

$$\min_A \left\| \text{vec}(T_{(1)}) - (Z_{(1)} \otimes I_p) \text{vec}(A) \right\|^2 + \left\| \text{vec}(Y) - (W^T \otimes D) \text{vec}(A) \right\|^2. \quad (1.35)$$

Έστω  $G = (Z_{(1)} \otimes I_p)$  και  $L = (W^T \otimes D)$ , τότε η λύση της (1.35) μπορεί να υπολογιστεί εύκολα και η ενημέρωση του  $A$  δίνεται από:

$$\text{vec}(A) = (G^T G + L^T L)^{-1} (G^T \text{vec}(T_{(1)}) + L^T \text{vec}(Y)) \quad (1.36)$$

### Ενημέρωση πίνακα συντελεστών W

Για να βρούμε την ενημέρωση του  $W$ , αγνοούμε τον πρώτο όρο της (1.27) οπότε και λύνουμε:

$$\min_W \|Y - DAW\|^2. \quad (1.37)$$

Η επίλυση της (1.37) ως προς  $W$  μας δίνει την ενημέρωση του  $W$ :

$$W = (A^T D^T D A)^{-1} A^T D^T Y \quad (1.38)$$

Ο αλγόριθμος της μεθόδου CLASS-CP φαίνεται στον Αλγόριθμο 2. Για να υπολογίσουμε τα  $A$ ,  $B$ ,  $C$  και τον πίνακα συντελεστών  $W$  ο αλγόριθμος εκτελεί εναλλασσόμενες ενημερώσεις μέχρι να συγκλίνει σε ένα κριτήριο ή να φτάσει σε ένα μέγιστο αριθμό επαναλήψεων. Ως κριτήριο χρησιμοποιούμε τη *σχετική αλλαγή* η οποία μπορεί να υπολογιστεί σε κάθε επανάληψη ως εξής:

$$\frac{|(f_{new} + g_{new}) - (f_{old} + g_{old})|}{f_{old} + g_{old}} \quad (1.39)$$

## 1.7 Πειραματική Αξιολόγηση

### 1.7.1 Δεδομένα

Για να αξιολογήσουμε την προτεινόμενη μέθοδο, πραγματοποιήσαμε πειράματα με δύο δημόσια σύνολα δεδομένων που έχουν ήδη χρησιμοποιηθεί στη βιβλιογραφία [21, 22]. Τα σύνολα δεδομένων προέρχονται από δύο πλατφόρμες, το BuzzFeed και το PolitiFact, και περιλαμβάνουν τόσο το περιεχόμενο των ειδήσεων όσο και πληροφορίες δικτύου, μαζί με ετικέτες κλάσης που δείχνουν για κάθε είδηση ξεχωριστά εάν είναι ψευδής ή αληθής. Το περιεχόμενο περιλαμβάνει πληροφορίες που σχετίζονται με το περιεχόμενο του ειδησεογραφικού άρθρου, όπως ο τίτλος, ο συντάκτης και το κείμενο, ενώ οι πληροφορίες δικτύου περιλαμβάνουν στοιχεία όπως το προφίλ των χρηστών, οι φιλίες τους και η δραστηριότητά τους.

Για την αξιολόγησή μας χρησιμοποιούμε μόνο τις πληροφορίες του δικτύου, και συγκεκριμένα τα δίκτυα φιλίας μεταξύ των χρηστών που έχουν αναρτήσει τις σχετικές ειδήσεις. Προκειμένου να μειωθεί το μέγεθος και η σποραδικότητα των δεδομένων, καταργήσαμε τους χρήστες με βαθμό κόμβου  $< 3$ . Στο τέλος, καταλήγουμε με δύο ταυστές διαστάσεων  $182 \times 1449 \times 1449$  και  $240 \times 1697 \times 1697$  για τα σύνολα δεδομένων BuzzFeed και PolitiFact αντίστοιχα. Τα στατιστικά για κάθε σύνολο δεδομένων παρουσιάζονται λεπτομερώς στον πίνακα 6.1.

### 1.7.2 Πειραματικές ρυθμίσεις

Για να μετρήσουμε την απόδοση της μεθόδου επιλέγουμε τις μετρικές αξιολόγησης precision, recall, F1-score και accuracy οι οποίες χρησιμοποιούνται συχνά σε παρόμοια προβλήματα [21, 11, 7]. Το πρώτο 80% των ειδήσεων είναι το σύνολο εκπαίδευσης (train set) και το υπόλοιπο 20% είναι το σύνολο ελέγχου (test set). Ο αριθμός των ειδήσεων που χαρακτηρίζονται ως αληθείς ισούται με τον αριθμό αυτών που χαρακτηρίζονται ως ψευδείς. Εκτελούμε τα πειράματα 10 φορές, ανεξάρτητα για κάθε σύνολο δεδομένων και καταγράφουμε τα μέσα αποτελέσματα.

#### Αριθμός συντελεστών

Προτού συγκρίνουμε την CLASS-CP με άλλες προσεγγίσεις, μετράμε την επίδραση του αριθμού των παραγόντων στα αποτελέσματα. Δοκιμάζουμε για  $r = 2, 3, 5, 10, 15, 20$ . Ένα μεγαλύτερο  $r$  αυξάνει τον χρόνο υπολογισμού της παραγοντοποίησης. Η καλύτερη συνολικά επίδοση επιτυγχάνεται για  $r = 5$  όπως φαίνεται στο σχήμα 6.2 και το αφήνουμε έτσι για τα υπόλοιπα πειράματα.

#### Επίδραση μεγέθους συνόλου εκπαίδευσης

Επιπλέον, εξετάζουμε τις επιδόσεις της CLASS-CP σε σχέση με τον αριθμό των δεδομένων με ετικέτες που χρησιμοποιούνται στο στάδιο της εκπαίδευσης. Το σχήμα 6.3 παρουσιάζει τις καμπύλες μάθησης για τη μετρική F1 σε σχέση με την αναλογία δεδομένων εκπαίδευσης με ετικέτες σε αμφότερα τα σύνολα δεδομένων. Από το γράφημα, μπορούμε να δούμε ότι όσο περισσότερα επισημασμένα δεδομένα εκπαίδευσης χρησιμοποιούμε, τόσο καλύτερη η απόδοση. Το ίδιο μοτίβο παρατηρείται και στις άλλες μετρικές.

### Υπερ-παράμετρος $\lambda_g$

Η τιμή της υπερ-παραμέτρου ορίζεται ως  $\lambda_g = 1$ , οπότε το σφάλμα της κατηγοριοποίησης συμβάλλει το ίδιο στη διαδικασία βελτιστοποίησης.

### Χρόνος εκτέλεσης

Κάθε επανάληψη στον CLASS-CP ολοκληρώνεται σε λιγότερο από ένα λεπτό και χρειάζονται κατά μέσο όρο λιγότερα από 10 λεπτά για να φτάσουμε σε σύγκλιση.

### 1.7.3 Αποτελέσματα

Συγκρίνουμε τον προτεινόμενο αλγόριθμο CLASS-CP με τις ακόλουθες μεθόδους:

- **NN-RESCAL:** Το πρώτο μοντέλο που αναπτύξαμε με τη χρήση της μη αρνητικής παραγοντοποίησης RESCAL όπως περιγράψαμε στην παράγραφο 1.5.
- **SVM:** Πρότυπο μοντέλο που βασίζεται σε Μηχανές Διανυσμάτων Υποστήριξης (SVM) στις οποίες εισάγουμε τις γραμμές του ταυυστή  $\mathcal{T}$  σε μορφή διανυσμάτων.
- **CP+SVM:** Η κλασική παραγοντοποίηση CP με έναν γραμμικό ταξινομητή SVM. Πρόκειται για μία διαδικασία δύο βημάτων. Στο πρώτο βήμα υπολογίζουμε τους πίνακες συντελεστών που μας δίνει η παραγοντοποίηση CP (χωρίς επίβλεψη) και στο δεύτερο βήμα εκπαιδεύουμε έναν ταξινομητή SVM με είσοδο τους συντελεστές του πρώτου βήματος.
- **CP+k-NN:** Η παραγοντοποίηση CP με ταξινομητή  $k$ -κοντινότερων γειτόνων ( $k$ -NN). Ακολουθούμε την ίδια διαδικασία όπως στην CP+SVM αλλά αντί για ταξινομητή SVM χρησιμοποιούμε ταξινομητή  $k$ -NN. Όπως προτείνουν οι [25, 27], ορίζουμε το  $k$  ίσο με την τετραγωνική ρίζα του μεγέθους του συνόλου εκπαίδευσης δηλαδή  $k = \sqrt{p}$ .
- **TriFN:** Πρόκειται για μια σύγχρονη μέθοδο ανίχνευσης ψευδών ειδήσεων η οποία χρησιμοποιεί και συνδυάζει πληροφορίες από τις αλληλεπιδράσεις των χρηστών και τους εκδότες των ειδήσεων. Τα αποτελέσματα παράγονται υπολογίζοντας την αλληλεξάρτηση έξι ετερογενών παραμέτρων. Χρησιμοποιούμε τις ίδιες ρυθμίσεις αξιολόγησης που προτείνονται από τους Shu et al. [23] ώστε να μπορέσουμε να συγκρίνουμε τα αποτελέσματά μας με αυτά της TriFN.
- **Content-SVM:** Πρότυπο μοντέλο σάκου λέξεων (bag-of-words) που βασίζεται στο περιεχόμενο του κειμένου ώστε να συγκριθεί με τις προαναφερθείσες μεθόδους που βασίζονται σε πληροφορίες δικτύου. Αρχικά χωρίζουμε το κείμενο κάθε είδησης σε λέξεις. Στη συνέχεια, υπολογίζουμε τον αριθμό των φορών που κάθε λέξη/όρος εμφανίζεται σε κάθε έγγραφο. Κάθε όρος θεωρείται ως ένα ξεχωριστό χαρακτηριστικό και καταλήγουμε να κατασκευάζουμε έναν πίνακα εγγράφων-λέξεων ο οποίος τελικά δίνεται ως είσοδος σε έναν ταξινομητή SVM.

### Συγκριτικά αποτελέσματα

Το σχήμα 6.5 και οι Πίνακες 6.3 και 6.4 δείχνουν τη σύγκριση μεταξύ της CLASS-CP και των προαναφερόμενων μεθόδων για τις διαφορετικές μετρικές. Ο πίνακας 6.5 δείχνει τα αποτελέσματα για τη μέθοδο Content-SVM. Από αυτά, μπορούμε να κάνουμε τις παρακάτω παρατηρήσεις:

- Η NN-RESCAL αποδίδει χειρότερα κι από την τυχαία πρόβλεψη οπότε την αφαιρούμε από τα γραφήματα σύγκρισης.
- Η απλή προσέγγιση με SVM δίνει τα υψηλότερα αποτελέσματα στο precision αλλά είναι πιο αδύναμη στις άλλες μετρικές.
- Οι πίνακες 6.3 και 6.4 δείχνουν πως η CLASS-CP αποδίδει σημαντικά καλύτερα από τις SVM, CP+SVM και CP+k-NN και στα δύο σύνολα PolitiFact και BuzzFeed.
- Η CLASS-CP υπερέχει της TriFN στη μετρική precision και για τα δύο σύνολα.
- Η χρήση του CP δεν βελτιώνει την απόδοση του SVM στην περίπτωση του PolitiFact. Αντίθετα, για το σύνολο δεδομένων BuzzFeed, το μοντέλο CP+SVM φαίνεται να έχει καλύτερη απόδοση σε σχέση με το βασικό μοντέλο SVM. Συνεπώς, μπορούμε να πούμε ότι υπάρχει μια αναντιστοιχία στην τάση απόδοσης μεταξύ των δύο συνόλων δεδομένων, όταν χρησιμοποιείται η απλή παραγοντοποίηση CP. Από την άλλη, το μοντέλο CLASS-CP φαίνεται να παράγει παρόμοια αποτελέσματα και στα δύο σύνολα δεδομένων.
- Για το PolitiFact, το CP σε συνδυασμό με το k-NN φαίνεται να έχει καλύτερη απόδοση από το CP+SVM. Στην περίπτωση του BuzzFeed παρατηρούμε το αντίθετο. Και πάλι παρατηρείται αναντιστοιχία στην τάση απόδοσης μεταξύ των δύο συνόλων δεδομένων. Υποθέτουμε πως αυτό οφείλεται στις τιμές των παραμέτρων που είναι ενσωματωμένες στον SVM. Πιθανώς με την προσαρμογή (tuning) αυτών των παραμέτρων να παίρναμε παρόμοια αποτελέσματα και για τα δύο datasets.
- Σε σύγκριση με τις άλλες μεθόδους, η CLASS-CP παράγει υψηλότερα νούμερα απόκλισης (μεταξύ 0.058 και 0.146). Αυτό υποδηλώνει ότι τα σημεία δεδομένων (data points) που παράγονται από την κλάση-CLASS-CP μπορούν να εξαπλωθούν σε μεγαλύτερη απόσταση από τον μέσο όρο, σε σύγκριση, για παράδειγμα, με το TriFN (που παράγει απόκλιση μεταξύ 0.013 και 0.040). Σε κάποιο βαθμό αυτό μπορεί να αποδοθεί στο σχετικά μικρότερο μέγεθος δείγματος που χρησιμοποιείται στα πειράματά μας (βλ. Ενότητα 6.1) που μπορεί να οδηγήσει σε μεγαλύτερες αποκλίσεις μεταξύ των σημείων δεδομένων.
- Το μοντέλο Content-SVM παράγει ικανοποιητικά αποτελέσματα σε σχέση με τις υπόλοιπες πρότυπες μεθόδους που βασίζονται σε πληροφορίες δικτύου. Επομένως, η προσθήκη της πληροφορίας του περιεχομένου στο μοντέλο CLASS-CP που αναπτύξαμε, θεωρείται μελλοντική επέκταση αυτής της εργασίας.

Τα παραπάνω αποδεικνύουν πως (i) χρησιμοποιώντας μόνο δεδομένα δικτύου και μερικές ετικέτες κλάσης, (ii) προσθέτοντας την πληροφορία κλάσης εν μέσω και όχι μετά τη διαδικασία παραγοντοποίησης του τανυστή και (iii) με μικρό αριθμό παραγόντων που οδηγούν σε σύντομους χρόνους υπολογισμού, μπορούμε να επιτύχουμε εξίσου καλές επιδόσεις με state-of-the-art τεχνικές, όπως είναι η TriFN, που απαιτεί τον συνδυασμό πολλών ετερογενών πληροφοριών και πολύπλοκους υπολογισμούς. Λαμβάνοντας αυτά υπόψη, μπορούμε να επιβεβαιώσουμε την αρχική μας υπόθεση ότι η διερεύνηση των δικτύων μεταξύ των χρηστών μπορεί να είναι χρήσιμη στη διαδικασία ανίχνευσης ψευδών ειδήσεων.

## 1.8 Συμπεράσματα και Μελλοντικές Προεκτάσεις

Σε αυτή τη διπλωματική εργασία, αντί να εφαρμόσουμε χωριστά την παραγοντοποίηση και την ταξινόμηση, όπως συνηθίζεται, προτείνουμε μια μέθοδο που τις συνδυάζει σε μια

κοινή διαδικασία εκμάθησης για την ανίχνευση ψευδών ειδήσεων στα social media. Αυτή η προσέγγιση χρησιμοποιεί τα δίκτυα φιλίας των χρηστών που έχουν αλληλεπιδράσει με τις ειδήσεις, καθώς και ένα σύνολο ετικετών κλάσης που είναι διαθέσιμες για ορισμένες από τις ειδήσεις. Προκειμένου να συνδυάσουμε αυτές τις πληροφορίες, προτείνουμε μια επέκταση της πρότυπης μεθόδου παραγοντοποίησης τανυστή CP, η οποία ενσωματώνει τις ετικέτες κλάσης στην ίδια την παραγοντοποίηση. Με αυτό τον τρόπο καταλήγουμε με επιτυχία σε μία ημι-επιβλεπόμενη παραγοντοποίηση τανυστή η οποία έχει επίγνωση της κλάσης.

Για την αξιολόγηση της μεθόδου, διεξήγαμε πειράματα με δύο δημόσια σύνολα δεδομένων. Τα αποτελέσματα καταδεικνύουν τη σημασία της ενσωμάτωσης των πληροφοριών της κλάσης στη φάση της παραγοντοποίησης, σαν μία ενιαία διαδικασία. Επιβεβαίωσαν επίσης την αρχική μας υπόθεση, ότι δηλαδή, ο τρόπος με τον οποίο οι χρήστες συνδέονται μεταξύ τους έχει να κάνει άμεσα με την εγκυρότητα των ειδήσεων με τις οποίες αλληλεπιδρούν στα social media.

Ως μελλοντική προέκταση, επιθυμούμε να διερευνήσουμε τον βαθμό στον οποίο η CLASS-CP βελτιώνεται όταν προστίθενται σε αυτήν περισσότερες πληροφορίες, τόσο από το δίκτυο όσο και από το περιεχόμενο των ειδήσεων. Επιπλέον, σχεδιάζουμε να αξιολογήσουμε την απόδοση της προσέγγισής μας σε περισσότερες βάσεις δεδομένων καθώς και να διερευνήσουμε την επίδραση του μεγέθους των δεδομένων στη δυνατότητα κλιμάκωσης του αλγορίθμου. Σε μεθοδολογικό επίπεδο, θα ήταν ενδιαφέρον να διερευνηθούν νέοι τρόποι αναπαράστασης των διαθέσιμων πληροφοριών με τανυστές, προκειμένου να ενσωματωθούν στην προτεινόμενη μέθοδο.





Over the last years, there has been a remarkable increase in global social media usage for news consumption: 68% U.S. adults got news on social media in 2018, as opposed to 2012, when only 49% of them had seen any news at all on social media [1]. In Greece, 71% of adults prefer social media as a source of news to TV or print and similar patterns are observed in the rest of Europe [3]. In an attempt to benefit from these numbers, many online platforms are generating large volumes of *fake news*, i.e., news with misleading information, both for political and financial gain [5]. A European survey of 26,000 adults in 2018, confirmed that fake news is widely spread across Europe and 85% of the participants believed that fake news is a serious problem [24]. All the above, illustrate the great need for truthfulness in online news articles and raise the challenge of developing automatic fake news detection methods to protect readers from misinformation.

The difficulty of automatically detecting fake news on social media lies mainly in the following two factors: (i) fake news is deliberately written to deceive readers and (ii) fake news content can vary in topics, writing styles and mediums. For instance, a truthful fact may be presented in the wrong context to back up a deceptive claim [6, 7]. Consequently, a model solely based on the news content may not give the best results. Instead, we could benefit from supplementary information generated by the network, e.g., the users interacting with the posts, to extract more accurate results.

The problem of detecting fake news on social media has been studied extensively in the literature. We can distinguish three main approaches: i) content-based, ii) network-based, and iii) hybrid. Content-based approaches focus on analyzing news content and detecting language patterns via natural language processing (NLP) tasks [8, 9, 28]. News content features can be further categorized into linguistic-based and visual-based. Linguistic-based features derive from the text content itself, such as the news headline, vocabulary, syntax, frequency of words, unique words and punctuation [7]. Visual-based cues come from visual elements such as images or videos that are often manipulated to provoke a certain reaction. Such cues were exploited by Huh et al. in [57], where they proposed a learning algorithm for identifying visual image alterations, trained by using only a large dataset of real photographs. The most common issue with this approach, however, is that it requires closed domains, with pre-defined users and context to produce satisfying results. This can become a challenge when dealing with fake news and may explain the rather limited use of content-based methods alone for fake news detection on social media [11].

Network-based approaches extract information by constructing different types of networks, where the nodes can be users, news articles or social media posts. For example, *friendship networks* show the follower/followee structure of users [7]. In *diffusion networks*, nodes represent users and edges show the way information is spread between them [66]. Another type of network that has been used in fake news detection is the *stance network* in which nodes represent all the social media posts relevant to the news and edges show the similarity of stances [34, 12]. *Interaction networks* are built of nodes representing publishers, users or news, and the edges show the interactions among them [15]. Existing network metrics, such as degree or clustering coefficient [66] can then apply to these networks as feature representations. Alternatively, the latent node embeddings

can be learned with factorization techniques [54] or network propagation approaches [34]. Network-based methods perform better in dynamic environments, thus they are better suited for identifying fake news which expands across topics and domains [12].

Finally, hybrid approaches aim to combine the advantages of content-based and network-based models by incorporating both linguistic and network features in a complementary way. In [13], Ruchansky et al. incorporated in a single model the articles' text, the temporal behavior of users along with temporal textual and user features by using neural networks. Tri-relationship embedding framework, called TriFN [23], models publisher-news relations and user-news interactions simultaneously, with the help of news content embeddings for fake news classification. Additionally, Conroy et al. [2] proposed guidelines for a hybrid fake news classification approach that combines linguistic features with network-based behavioral data.

## 2.1 Thesis Objective

The above challenges combined with the facts that i) users play a decisive role in the way fake news spread [7], ii) users share information with other similar users (friends) [14] and iii) network features are important for the classification of fake news [15], suggest that the investigation of relationship networks between users could facilitate in debunking a fake story. This thesis aims to exploit the information deriving from users' friendship networks and prove their importance in detecting fake news.

In most cases, the standard tensor factorization methods perform in an unsupervised setting and are not aware of labeled data that might be available. Through tensor factorization methods, our goal is to build class-aware factor matrices and reveal the hidden information behind users' friendship networks that can differentiate fake from real news. We believe that incorporating class information during the factorization process and not in a separate step could help produce discriminatory representations that identify fake news with higher reliability.

## 2.2 Contribution

In this thesis, we propose a new semi-supervised approach for classifying fake news posts found on social media, called CLASS-CP, that takes into account the follower/followee structure of users and the class label information simultaneously. Specifically, instead of using tensor factorization and classification separately, as is often the case, this method integrates them in a single optimization process. The main contributions of this thesis are:

- A novel tensor-based way of modeling news through users' friendships
- A simple but competitive model for detecting fake news based only on network information and some available labeled data
- We integrate classification into the tensor factorization process, achieving class-aware modeling of data for fake news classification
- We evaluate the performance on real-world datasets and show the effectiveness of our method

As we will show below, our proposed technique introduces an alternative tensor-based model for representing news posts only through structural information; thus, it can become useful for several other domains where we have little to no content information (e.g., biology and medicine). The code for the proposed method is publicly available at <https://github.com/FrossoPap/class-cp>.

## 2.3 Thesis Layout

Chapter 3 presents the literature work related to fake news detection and joint optimization. Chapter 4 introduces tensor factorization techniques and presents the notation that we will use throughout the work. Chapter 5 describes the problem and presents the proposed factorization method. Following this is Chapter 6 that presents the datasets used, the results from the experiments, and the comparison with other methods. Finally, Chapter 7 summarizes what we have achieved in this thesis.



### 3.1 Fake News Detection

Fake news is not a new problem. The media ecosystem of fake news has evolved through the years from newsprint to radio, television, online news and more recently, social media that have led to an information environment where they can spread uncontrollably. Because of the serious impact and social consequences of receiving and sharing of false information, fake news detection has drawn a lot of attention in recent literature. Different types of automated methods have been developed to tackle the problem. On online news media, fake news is identified with the help of methods that mainly rely on the news content. On social media, auxiliary information coming from the social context can be exploited in order to identify a fake news piece with greater accuracy. In this section we will present fake news detection techniques that can typically be *content-based*, *social context-based* or both.

#### 3.1.1 Content-based

Content-based approaches focus on finding patterns coming from news content. Generally, in terms of its content a news piece consists of the following basic elements [7]:

- **Source:** the publisher/news platform or author of the news piece
- **Headline:** a short text title that describes the main subject of the article
- **Body text:** the main text content that describes the details of the news story
- **Image/video:** provides visual cues to support the main body text

Given these features, several types of representations can be employed to discriminate between fake and real news. In most cases, news content features that we can usually work with, come either from *linguistic cues* (e.g., vocabulary, syntax, writing style, etc) or from *visual cues* (e.g., videos, images, etc).

#### Linguistic cues

Fake news is *intentionally* written in a way that can easily confuse and delude the readers. Thus, it comes naturally to look into the linguistic patterns that characterize the article's text and may give hints of misinformation. Linguistic-based features are obtained from the text content in terms of article structure on different levels (e.g., characters, words or sentences). For example, Hosseinimotlagh et al. [28] proposed an unsupervised tensor modeling of the problem, based on term frequency and spatial relations between terms and articles. Guacho et al. [29] introduced a semi-supervised model via tensor embeddings that use spatial/contextual information about news articles. Gupta et al. [30] proposed a classifier to estimate tweet credibility from features such as the number of words, URLs, hashtags, emojis, presence of swear words and pronouns. Horne et al. [33] employed

an SVM-based algorithm using stylistic, complexity and psychological features of both title and body text to classify real, fake and satire news. However, good performance of linguistic-only methods requires closed domains with predefined context and users. This can become quite restrictive, given that fake news expands across many topics and domains.

## Visual cues

Visual cues come from visual elements such as video and images that support the main body text in news articles [7]. When maliciously altered, they can become very popular and facilitate in the fake news propagation. Visual and statistical features can be extracted to identify patterns in fake images either visually (clarity score, diversity score, similarity distribution histogram, etc) or statistically (image ratio, multi-image ratio, hot image ratio, etc) [56, 7]. In another work, classification models that rely on user-based and tweet-based features were used to distinguish fake from real images of Hurricane Sandy [55, 7]. In a more alternative approach, Papadopoulou et al. [59] recently developed an online tool that collects and performs calculations on information surrounding the video, rather than analyzing the visual elements of the video itself, and then creates a verification report for fake video detection.

### 3.1.2 Social context-based

Aside from the use of text that derives directly from the content of news articles or posts, supplementary social context features coming from the surrounding environment of fake news can be extracted, such as the users' engagements or the networks they create with each other.

We can further divide social context features into *user-based*, *post-based* and *network-based* features. User-based features come from user profiles and are used to calculate their credibility or other user-level characteristics [66]. For example, Castillo et al. [35] exploited features from users' posting and reposting behavior, in combination with text and citations to external sources. Shu et al. [43] suggested measuring users' sharing behaviors on social media and then comparing explicit and implicit profile features, to reveal their potential to detect fake news. Post-based features are extracted from users' responses and engagements to social media posts in terms of stance [34], topics [36], or credibility [35]. Tacchini et al. [12] identified fake news based on users who "liked" them using logistic regression and a Boolean crowd-sourcing algorithm for classification. Then, Della Vedova et al. [45] extended this work and complemented it efficiently with a content-based method when little social context information is available. On the other hand, network-based features derive from the construction of different networks, such as diffusion networks [66], co-occurrence networks [13], etc.

Existing models that take into account these context-based features can be classified into *stance-based* and *propagation-based* [7]. Stance-based models make use of the users' implicit or explicit stance towards a news post. Explicit stances are extracted from the users' direct response to a post such as the "like" button on Facebook. Implicit stances are automatically extracted with stance detection tasks that can decide from a post whether the user is in favor of, neutral toward, or against the main idea or event presented in a news article [4]. On the other hand, propagation-based models are based on the assumption that news credibility is highly relevant to the credibility of social media posts, on which several propagation methods can be applied [34, 7]. For example, Gupta et al. [31] proposed a PageRank-like credibility propagation algorithm on a multi-type network consisting of

events, tweets, and users by encoding users' credibility and tweets' implications. Jin et al. [32] proposed to detect news' sub-events, then build a three-level credibility network consisting of events, sub-events, and messages, propagate the credibility value of each entity on the network and formulate the process as a graph optimization problem to infer the events' credibility.

## 3.2 Joint Factorization

We can find several joint factorization techniques in the literature that are *context-aware* or *class-aware*. Context-aware approaches integrate contextual information into factorization to solve mostly recommendation tasks. For example, Li et al. [37] suggested seeing context as a user feature that is dynamic and developed a probabilistic latent relation model that combines context and the user's long-term preferences. In the same way, Rendle et al. [38] proposed the use of Factorization Machines (FMs) to model contextual information for context-aware rating predictions.

Class-aware approaches add information of class into the factorization process. For instance, Katsimpras and Paliouras [41] extended RESCAL factorization to produce a semi-supervised factorization technique that adds a classification error term in the standard factor optimization process for node classification. Cao et al. [39] proposed semiBAT, a semi-supervised brain network analysis approach based on constrained tensor factorization. In the same concept, Xiao et al. [40] introduced a semi-supervised non-negative matrix factorization method that associates a class label with each base vector, for image representation.

As stated above, all these methods are optimized to perform in domains different from fake news detection. Our framework extends the class-aware methods and aims to build a less complex but still effective fake news detection system that exploits the friendships among users and the class-label information available for some news, in a joint factor optimization process.





## 4.1 Notation

A tensor is a multidimensional or N-way array. The *order* of a tensor, also known as *ways* or *modes*, is the number of its dimensions, therefore a first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors. An  $r$ -order tensor is of rank-one if it can be strictly decomposed into the outer product of  $r$  vectors. Throughout this thesis, we will use the following notation: the uppercase calligraphy letters denote tensors, e.g.,  $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ . Matrices are represented by uppercase italic letters like  $A$ . Lowercase bold letters, like  $\mathbf{v}$ , denote vectors. The  $(i,j)$  element of a matrix  $A$  is denoted by  $a_{ij}$ . To refer to the  $i$ -th row of a matrix  $A$  we use  $\mathbf{a}_i$ . Similarly, an element  $(i,j,k)$  of a tensor  $\mathcal{T}$  will be denoted as  $\mathcal{T}_{ijk}$ .

### Kronecker product

If  $A$  is an  $m \times n$  matrix and  $B$  a  $p \times q$  matrix then the Kronecker product  $A \otimes B$  is defined as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \quad (4.1)$$

A known matrix operation property that involves the Kronecker product is:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}. \quad (4.2)$$

### Khatri-Rao product

Proceeding with the same matrices, the Khatri-Rao product is defined as:

$$\mathbf{A} \odot \mathbf{B} = (\mathbf{A}_{ij} \otimes \mathbf{B}_{ij})_{ij} \quad (4.3)$$

and more analytically:

$$\mathbf{A} \odot \mathbf{B} = \left[ \begin{array}{c|c} \mathbf{A}_{11} \otimes \mathbf{B}_{11} & \mathbf{A}_{12} \otimes \mathbf{B}_{12} \\ \hline \mathbf{A}_{21} \otimes \mathbf{B}_{21} & \mathbf{A}_{22} \otimes \mathbf{B}_{22} \end{array} \right] \quad (4.4)$$

## 4.2 Tensor Factorization

Tensor factorization or decomposition provides a scheme for expressing a tensor as a sequence of elementary operations between new, often simpler, tensors or arrays. The domain of social media networks is characterized by high dimensional, sparse data that challenge the process of extracting valuable information efficiently. Tensor factorization

Table 4.1: Table of Notation

$p$	number of posts
$u$	number of users
$\mathcal{T}$	tensor $\in \mathbb{R}^{p \times u \times u}$
$\mathcal{T}_{ijk}$	an element $(i, j, k)$ of tensor $\mathcal{T}$
$\mathbf{v}$	vector
$A$	matrix
$\mathbf{a}_i$	the $i$ -th row of a matrix
$\otimes$	Kronecker product
$\odot$	Khatri-Rao product

has brought significant performance results in such cases, compared to traditional distance measures that are used to compute the similarity of entities, such as Euclidean distance or cosine similarity [61, 62]. It can be applied in many ways on data arrays for extracting and better comprehending their hidden properties. Thus, tensor factorization has wide applications in latent variable learning and can be extensively used in machine learning problems to calculate hidden features of multidimensional data.

A standard tensor factorization approach is the Canonical/Parafac, or CP [16], that decomposes a tensor into a sum of rank-one tensors, as described in more detail in Section 4.3. Another widely used factorization called Tucker [17], decomposes a tensor into a set of separate factor matrices for each tensor mode and one core tensor. A more recent approach is RESCAL tensor factorization [18] that generates a unique representation for every relation in the data.

### 4.3 The CP Factorization

The Canonical/Parafac (CP) decomposition factorizes a tensor into a sum of rank-one tensors or the sum of outer product of vectors. According to Kolda [26], given a third-order tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  we may as well write it as:

$$\mathcal{X} \approx \sum_{r=1}^R a_r \circ b_r \circ c_r \quad (4.5)$$

where  $R$  is a positive integer,  $\circ$  is the vector outer product and  $a_r \in \mathbb{R}^I, b_r \in \mathbb{R}^J, c_r \in \mathbb{R}^K$  for  $r = 1, \dots, R$ . Tensor  $\mathcal{X}$  can then be written in matricized form [26]:

$$\begin{aligned} X_{(1)} &\approx A(C \odot B)^T, \\ X_{(2)} &\approx B(C \odot A)^T, \\ X_{(3)} &\approx C(B \odot A)^T. \end{aligned} \quad (4.6)$$

The standard algorithm for computing the CP decomposition is CP-ALS, that guarantees a good trade-off between convergence rate (number of iterations) and iteration cost [26, 65]. In cases of big sparse tensors, computing the CP decomposition can get very costly. Consequently, the efficient computation of CP by exploiting the sparsity of a tensor has gained recently significant research interest [65].

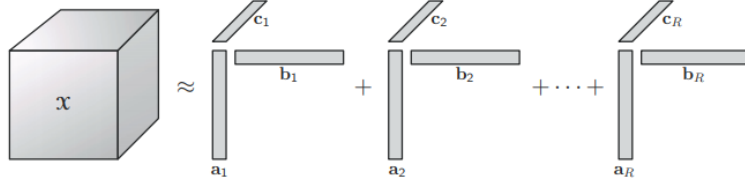


Figure 4.1: CP Decomposition of a Three-Way Array [26]

## 4.4 The RESCAL Factorization

RESCAL factorization, proposed by [18], is a tensor factorization model that takes the inherent structure of relational data into account. RESCAL scales very well to a large number of data and has produced state-of-the-art results mainly in link prediction and entity resolution tasks [46, 62, 64]. RESCAL expects the first two tensor modes to be symmetric as a constraint to decompose a three-way tensor [62, 63].

In more detail, given a tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ , where the first two modes are identically created by linking together the entities of our problem (e.g., users) and the third mode holds the relations between them, each slice  $X_k$  is factorized according to [18] as:

$$X_k \approx AR_k A^T, \text{ for } k = 1, \dots, m \quad (4.7)$$

where  $A$  is an  $n \times r$  matrix that contains the latent-component representation of the entities in the domain and  $R_k$  is an asymmetric  $r \times r$  matrix that models the interactions of the latent components in the  $k$ -th predicate. The factor matrices  $A$  and  $R_k$  are computed by solving the following regularized minimization problem:

$$\min_{A, R_k} f(A, R_k) + g(A, R_k) \quad (4.8)$$

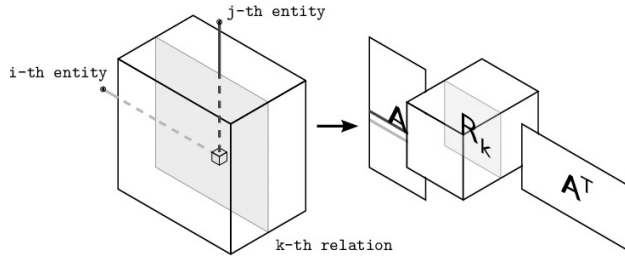
where

$$f(A, R_k) = \frac{1}{2} \sum_k \|X_k - AR_k A^T\|_f^2 \quad (4.9)$$

is the tensor factorization least-squares problem and

$$g(A, R_k) = \frac{1}{2} \lambda (\|A\|_f^2 + \sum_k \|R_k\|_f^2) \quad (4.10)$$

is the regularization term that is added to prevent overfitting of the model.

Figure 4.2: RESCAL Factorization into the Factor Matrix  $A$  and the core Tensor  $R_k$  [49]

## 4.5 Non-Negative RESCAL Factorization

Relational data like social network data can generally be considered as non-negative [49]. For this reason, tensorial data can be factorized using a non-negative version of RESCAL factorization which was developed by [49]. Following the notation described in section 4.1, given a tensor  $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ , each slice  $T_k$  is factorized as:

$$T_k \approx AR_k A^T, \text{ for } k = 1, \dots, m \quad (4.11)$$

where  $A$  is an  $n \times r$  matrix that contains the latent-component representation of the entities in the domain and  $R_k$  is an asymmetric  $r \times r$  matrix that models the interactions of the latent components in the  $k$ -th dimension. The factor matrices  $A$  and  $R_k$  are computed by solving the following regularized minimization problem:

$$\min_{A, R_k} f(A, R_k) + g(A, R_k) \quad (4.12)$$

where

$$f(A, R_k) = \sum_k \|T_k - AR_k A^T\|_f^2 \quad (4.13)$$

is the tensor factorization least-squares problem and

$$g(A, R_k) = \lambda_A \|A\|_f^2 + \lambda_R \sum_k \|R_k\|_f^2 \quad (4.14)$$

is the regularization term that is added to prevent overfitting of the model.

According to [49], the non-negative updates with the least-squares cost function for matrices  $A$  and  $R_k$  respectively are:

$$A \leftarrow A * \frac{\sum_k T_k A R_k^T + T_k^T A R_k}{A([\sum_k R_k A^T A R_k^T + R_k^T A^T A R_k] + \lambda_A I)} \quad (4.15)$$

$$R_k \leftarrow R_k * \frac{A^T T_k A}{A^T A R_k A^T A + \lambda_R R_k} \quad (4.16)$$

However, non-negative tensor decomposition brings forth additional limitations that may result in sparse factor matrices that require more time to update, introducing a scalability issue [60].

## 4.6 Problem Definition

Following the above notation, we can model the problem of detecting fake news on social media that was presented in Chapter 2 with a set of posts  $P = (p_1, p_2, \dots, p_p)$ , a set of user-user friendship networks  $G = (g_1, g_2, \dots, g_p)$  and a set of labels  $Y = (y_1, y_2, \dots, y_l)$ , with  $l < p$ . Our goal is to efficiently predict the labels of the unknown posts. The solution to this problem raises the following challenges or questions:

- (Q1) How can we take advantage of the friendship networks we created to efficiently classify the nodes?
- (Q2) How can we exploit the available label information of some nodes to enhance the analysis?

## 5.1 Method Summary

Our initial goal is to develop a method that successfully answers to question (Q1). Hence, we first try to generate discriminatory factor matrices using two different factorization techniques: (i) the non-negative RESCAL factorization and (ii) the CP Factorization. RESCAL factorization has produced state-of-the-art results in recent classification problems [18, 48, 46]. However, experimental results presented in Chapter 6 suggest that in our case the RESCAL factorization does not provide indicative enough representations for the tensorial data. Therefore, we proceed with the CP factorization that, as shown, interprets the available friendship networks more accurately. Finally, we answer to question (Q2) by developing an extended version of CP that combines a classification error term with the standard factorization in a unified optimization process.

## 5.2 Model 1: NN-RESCAL Factorization

The first model we developed uses the non-negative RESCAL factorization that was presented in Chapter 4. Say that we have  $2n$  number of posts available,  $2p$  of them are labeled and the rest  $2(n - p)$  is unlabeled, with  $p < n$ . The dataset is balanced, i.e., there is an equal number of fake and real posts in the dataset.

First, we create two third-order tensors by using a binary representation for the modeling of the users' friendships:  $\mathcal{X}_{(fake)} \in \mathbb{R}^{u \times u \times p}$  that contains the train set with the posts labeled as fake and  $\mathcal{X}_{(real)} \in \mathbb{R}^{u \times u \times p}$  that contains the train set with the posts labeled as real. Posts are represented as user-user adjacency matrices. Fake news can be considered as a distortion bias on information manipulated by its authors and is usually modeled as a binary classification problem [47]. Therefore, in order to decide whether or not a news article is fake, we form our tensors in the following way:

$$\mathcal{X}_{(fake)_{ijk}} = \begin{cases} 1, & \text{if user } j \text{ has interacted with fake post } k \\ & \text{and user } i \text{ follows user } j \\ 0, & \text{otherwise} \end{cases} \quad (5.1)$$

likewise,

$$\mathcal{X}_{(real)_{ijk}} = \begin{cases} 1, & \text{if user } j \text{ has interacted with real post } k \\ & \text{and user } i \text{ follows user } j \\ 0, & \text{otherwise} \end{cases} \quad (5.2)$$

Note that an interaction with a post means that the user has shared or posted the respective news pieces on their social media profile, in our case, on Twitter.

Next, we apply the non-negative RESCAL factorization on tensors  $\mathcal{X}_{(fake)}$  and  $\mathcal{X}_{(real)}$ , the way it was described in Chapter 4. As a result of the factorization, we come up with matrices  $A_{fake} \in \mathbb{R}^{u \times r}$  and  $A_{real} \in \mathbb{R}^{u \times r}$  respectively through the following non-negative updates according to (4.15):

$$A_{fake} \leftarrow A_{fake} * \frac{\sum_k X_{(fake)k} A_{fake} R_k^T + X_{(fake)k}^T A_{fake} R_k}{A_{fake}([\sum_k R_k A_{fake}^T A_{fake} R_k^T + R_k^T A_{fake}^T A_{fake} R_k] + \lambda_{A_{fake}} I)} \quad (5.3)$$

where

$$R_k \leftarrow R_k * \frac{A_{fake}^T X_{(fake)k} A_{fake}}{A_{fake}^T A_{fake} R_k A_{fake}^T A_{fake} + \lambda_R R_k} \quad (5.4)$$

and

$$A_{real} \leftarrow A_{real} * \frac{\sum_k X_{(real)k} A_{real} R_k^T + X_{(real)k}^T A_{real} R_k}{A_{real}([\sum_k R_k A_{real}^T A_{real} R_k^T + R_k^T A_{real}^T A_{real} R_k] + \lambda_{A_{real}} I)} \quad (5.5)$$

where

$$R_k \leftarrow R_k * \frac{A_{real}^T X_{(real)k} A_{real}}{A_{real}^T A_{real} R_k A_{real}^T A_{real} + \lambda_R R_k} \quad (5.6)$$

The remaining posts that were not used in the train set form our test set, in which posts have no labels. Suppose the test set is a set of matrices  $M = \{P_1, P_2, \dots, P_{2(n-p)}\}$ , each matrix corresponding to a different unlabeled post, where  $P_{idx} \in \mathbb{R}^{u \times u}$  and  $idx \in [1, 2(n-p)]$ . A matrix  $P_{idx}$  is created for each post  $idx$  as follows:

$$P_{(idx)_{ij}} = \begin{cases} 1, & \text{if user } j \text{ has interacted with the post } idx \\ & \text{and user } i \text{ follows user } j \\ 0, & \text{otherwise} \end{cases} \quad (5.7)$$

Next, we add a post,  $P_{idx} \in \mathbb{R}^{u \times u}$ , from the test set to the  $\mathcal{X}_{(fake)}$  and  $\mathcal{X}_{(real)}$  tensors as a new tensor slice. Finally, we end up with new tensors  $\mathcal{X}'_{(fake)} \in \mathbb{R}^{u \times u \times (p+1)}$  and  $\mathcal{X}'_{(real)} \in \mathbb{R}^{u \times u \times (p+1)}$  respectively. After the addition of the test instance in both tensors, we run the non-negative RESCAL factorization on tensors  $\mathcal{X}'_{(fake)}$  and  $\mathcal{X}'_{(real)}$ , to produce the new factor matrices  $A'_{fake} \in \mathbb{R}^{u \times r}$  and  $A'_{real} \in \mathbb{R}^{u \times r}$  through the following non-negative updates according to (4.15):

$$A'_{fake} \leftarrow A'_{fake} * \frac{\sum_k X'_{(fake)k} A'_{fake} R_k^T + X'_{(fake)k}{}^T A'_{fake} R_k}{A'_{fake}([\sum_k R_k A'_{fake}{}^T A'_{fake} R_k^T + R_k^T A'_{fake}{}^T A'_{fake} R_k] + \lambda_{A'_{fake}} I)} \quad (5.8)$$

where

$$R_k \leftarrow R_k * \frac{A'_{fake}{}^T X'_{(fake)k} A'_{fake}}{A'_{fake}{}^T A'_{fake} R_k A'_{fake}{}^T A'_{fake} + \lambda_R R_k} \quad (5.9)$$

and

$$A'_{real} \leftarrow A'_{real} * \frac{\sum_k X'_{(real)k} A'_{real} R_k^T + X'_{(real)k}{}^T A'_{real} R_k}{A'_{real}([\sum_k R_k A'_{real}{}^T A'_{real} R_k^T + R_k^T A'_{real}{}^T A'_{real} R_k] + \lambda_{A'_{real}} I)} \quad (5.10)$$

where

$$R_k \leftarrow R_k * \frac{A'_{real}{}^T X'_{(real)k} A'_{real}}{A'_{real}{}^T A'_{real} R_k A'_{real}{}^T A'_{real} + \lambda_R R_k} \quad (5.11)$$

Given that the Euclidean distance is a regularly used distance metric for numerical data [50, 51, 52], we calculate the prediction in the following fashion: Suppose  $l$  is each

post's yet unknown label, with  $l \in [0, 1]$ , where 0 signifies a real post and 1 a fake post. The label is derived as follows:

$$l = \begin{cases} 1, & \text{if } \|A_{fake} - A'_{fake}\| < \|A_{real} - A'_{real}\| \\ 0, & \text{otherwise} \end{cases}$$

Finally, we remove the matrix  $P_{idx}$  from  $\mathcal{X}'_{(fake)}$  and  $\mathcal{X}'_{(real)}$ , we add the next matrix from the matrices set  $M$  in both tensors, and we repeat the same procedure for the rest  $2(n-p)$  posts. The full algorithm for this model is shown in Algorithm 1.

However, as shown in Chapter 6, this model does not succeed in accurately detecting fake news on social media. In addition, to produce the final results we need to run four times the factorization algorithm which increases complexity and computation time. Therefore, we proceed with a second model that is presented in the following section.

---

**Algorithm 1** NN-RESCAL: Given  $2n$  number of posts,  $2p$  of them are labeled, with  $p < n$ , and a set of matrices  $M = \{P_1, P_2, \dots, P_{2(n-p)}\}$ , where  $P_{idx} \in \mathbb{R}^{u \times u}$  and  $idx \in [1, 2(n-p)]$ , compute  $A_{fake}, A_{real}, A'_{fake}, A'_{real}$

---

**Require:** Tensors  $\mathcal{X}_{(fake)}$  and  $\mathcal{X}_{(real)}$  according to (5.1),(5.2) and set of matrices  $M = \{P_1, P_2, \dots, P_{2(n-p)}\}$  created according to (5.7)

- 1: Initialize  $idx = 1$
  - 2: **repeat**
  - 3:   update  $A_{fake}$  using Eq. (5.3)
  - 4:   update  $A_{real}$  using Eq. (5.5)
  - 5:   add  $P_{idx}$  to  $\mathcal{X}_{(fake)}$  as a new tensor slice
  - 6:   add  $P_{idx}$  to  $\mathcal{X}_{(real)}$  as a new tensor slice
  - 7:   update  $A'_{fake}$  using Eq. (5.8)
  - 8:   update  $A'_{real}$  using Eq. (5.10)
  - 9:   n1 = norm of  $(A_{fake} - A'_{fake})$
  - 10:   n2 = norm of  $(A_{real} - A'_{real})$
  - 11:    $l = (n1 < n2)$
  - 12:   remove  $P_{idx}$  from  $\mathcal{X}_{(fake)}$
  - 13:   remove  $P_{idx}$  from  $\mathcal{X}_{(real)}$
  - 14: **until**  $idx > 2(n-p)$
- 

## 5.3 Model 2: CLASS-CP Factorization

Experimental results have shown that CP factorization is able to (i) represent the data in a more compact way compared to RESCAL, with little loss of information and (ii) improve classification accuracy, especially in a low dimensional feature space [44]. Therefore, as a next step, we choose to build an extension of the standard CP factorization [16]. Our goal is to associate the class labels of data samples with their latent representations for better classification results. We suggest that if we make the factorization aware of the class information, the prediction model will become more accurate.

Following the notation described in 4.1, we are going to represent the available friendship networks with tensors. For our modeling, we are going to use a binary representation

with a third-order tensor,  $\mathcal{T} \in \mathbb{R}^{p \times u \times u}$ :

$$\mathcal{T}_{ijk} = \begin{cases} 1, & \text{if user } k \text{ has interacted with post } i \\ & \text{and user } j \text{ follows user } k \\ 0, & \text{otherwise} \end{cases}$$

Again, when users interact with a post, it means they shared or posted it on their social media profile, in our case, on Twitter.

### 5.3.1 Method overview

The conceptual diagram of CLASS-CP is depicted in Figure 5.1. Firstly, we build a three-dimensional tensor from the friendship networks between users that have engaged with the news posts, namely  $\mathcal{T} \in \mathbb{R}^{p \times u \times u}$  (note that CLASS-CP is able to accept higher modes as well). The CP decomposition of such a tensor is computed by the following least-squares loss:

$$\min_{A,B,C} \|\mathcal{T} - \llbracket A, B, C \rrbracket\|^2,$$

where  $A \in \mathbb{R}^{p \times r}$  represents the posts factor matrix,  $B \in \mathbb{R}^{u \times r}$  and  $C \in \mathbb{R}^{u \times r}$  represent the users factor matrices and  $r$  is the rank of the factorization.

To extract class-related information we integrate into the optimization process a classification error term that is based on the latent factor  $A$ . As we will show, in the learned latent space, a classifier biases the posts-factor matrix  $A$ . By linking the post vectors in  $A$  to class labels, we get an approximation of the 3rd-order tensor  $\mathcal{T}$  that is based on these labels.

With a set of labels available for some of the posts, our intuition is that similarly labeled posts are going to share similar factors [20]. Based on this intuition, we integrate class-label information into the tensor factorization process itself, in order to move posts of the same class closer in the latent space. We assume that there is a matrix of coefficients  $W \in \mathbb{R}^{r \times c}$ , where  $r$  is the number of factors and  $c$  is the number of class labels. This coefficient matrix assigns labels to posts based on the factor matrix  $A$  as in a common regression problem, i.e.,  $Y = AW$ .

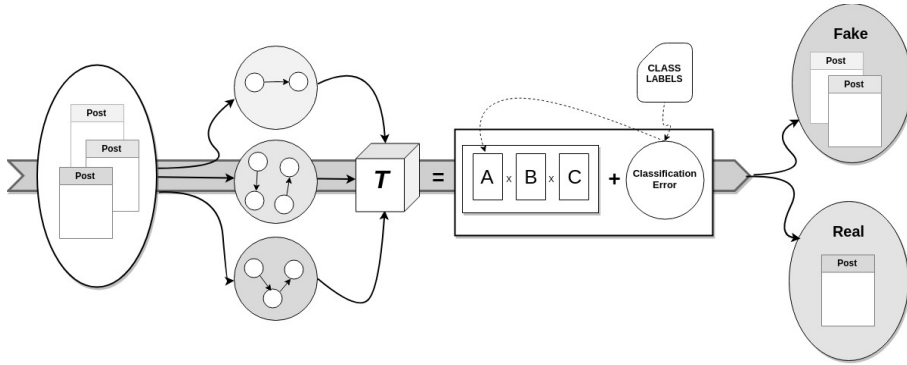


Figure 5.1: The Conceptual Diagram of CLASS-CP

### 5.3.2 Solving the joint optimization problem

Following the above, given a tensor  $\mathcal{T} \in \mathbb{R}^{p \times u \times u}$  and a set of labels  $Y \in \mathbb{R}^{l \times c}$ , where  $l$  is the number of labeled posts with  $l < p$ , and  $c$  is the number of class labels, we solve



the optimization problem presented in (5.12).

$$\min_{A,B,C,W} f(A, B, C) + g(A, W) \quad (5.12)$$

where

$$f(A, B, C) = \|\mathcal{T} - \llbracket A, B, C \rrbracket\|^2$$

is the tensor factorization least-squares problem,

$$g(A, W) = \lambda_g \|Y - DAW\|^2 \quad (5.13)$$

is the prediction error of the classifier, where  $D = [I^{l \times l}, 0^{l \times (p-l)}] \in \mathbb{R}^{1 \times p}$  and  $\lambda_g$  is a hyperparameter to control the influence of the classification error in the optimization. Note that  $g(A, W)$  is produced with respect to the current values of the post factor matrix  $A$ .

We solve the minimization problem in (5.12), using the efficient alternating least-squares method (ALS). This approach alternately fixes and solves factor matrices following update rules. The update rules are derived by setting the gradient of (5.12) with respect to each factor matrix to zero. For CP, the ALS method fixes every factor matrix except one and solves for it. The matricized form of  $f$  (one per mode) can be written as:

$$\begin{aligned} T_{(1)} &\approx A(C \odot B)^T, \\ T_{(2)} &\approx B(C \odot A)^T, \\ T_{(3)} &\approx C(B \odot A)^T. \end{aligned} \quad (5.14)$$

Recall that the  $\odot$  operator denotes the Khatri-Rao product.

### Updating factor matrices B and C

To find the updates for  $B$  and  $C$ , we ignore the second term of (5.12) and we solve the respective equations as presented in (5.14). The ALS approach fixes  $A$  and  $C$  to solve for  $B$ , and  $A$ ,  $B$  to solve for  $C$ . Let  $Z_{(2)} = (C \odot A)$  and  $Z_{(3)} = (B \odot A)$ , then by combining (5.12) and (5.14) we can write:

$$\min_B \|T_{(2)} - BZ_{(2)}^T\|^2 \quad (5.15)$$

and

$$\min_C \|T_{(3)} - CZ_{(3)}^T\|^2. \quad (5.16)$$

The solution to (5.15) and (5.16) is given by [26], so the update of  $B$  can be found as follows:

$$\begin{aligned} \|T_{(2)} - BZ_{(2)}^T\| &= (T_{(2)} - BZ_{(2)}^T)^T (T_{(2)} - BZ_{(2)}^T) = \\ &= T_{(2)}^T T_{(2)} - T_{(2)}^T BZ_{(2)}^T - Z_{(2)} B^T T_{(2)} + Z_{(2)} B^T B Z_{(2)}^T \end{aligned} \quad (5.17)$$

Given (5.17), the solution to (5.15) that gives us the update of  $B$  is:

$$\begin{aligned} \frac{\partial(5.17)}{\partial B} = 0 &\Rightarrow -T_{(2)}^T Z_{(2)}^T - (Z_{(2)} T_{(2)})^T + B^T (Z_{(2)} Z_{(2)}^T + Z_{(2)} Z_{(2)}^T) = \\ &= -2T_{(2)}^T Z_{(2)}^T + 2B^T Z_{(2)} Z_{(2)}^T = 0 \Rightarrow B = (Z_{(2)}^T Z_{(2)})^{-1} Z_{(2)}^T T_{(2)} \end{aligned} \quad (5.18)$$

and in the same way, the update of  $C$  is:

$$C = (Z_{(3)}^T Z_{(3)})^{-1} Z_{(3)}^T T_{(3)}. \quad (5.19)$$

### Updating factor matrix A

Let  $Z_{(1)} = (C \odot B)$ . To find the update for factor matrix  $A$ , we can combine (5.12) and (5.14) and write:

$$\min_A \left\| T_{(1)} - AZ_{(1)}^T \right\|^2 + \|Y - DAW\|^2. \quad (5.20)$$

Equation (5.20) can be solved more easily in vectorized form. Then (5.20) is written as:

$$\min_A \left\| \text{vec}(T_{(1)}) - (Z_{(1)} \otimes I_p) \text{vec}(A) \right\|^2 + \left\| \text{vec}(Y) - (W^T \otimes D) \text{vec}(A) \right\|^2. \quad (5.21)$$

where  $I_p$  is the  $p \times p$  identity matrix. Let  $G = (Z_{(1)} \otimes I_p)$  and  $L = (W^T \otimes D)$ , then (5.21) can be written as:

$$\min_A \left\| \text{vec}(T_{(1)}) - G \text{vec}(A) \right\|^2 + \left\| \text{vec}(Y) - L \text{vec}(A) \right\|^2 \quad (5.22)$$

which is a common linear regression problem. Then, (5.22) can be solved as follows:

$$\begin{aligned} & (\text{vec}(T_{(1)})^T - \text{vec}(A)^T G^T) (\text{vec}(T_{(1)}) - G \text{vec}(A)) + \\ & + (\text{vec}(Y)^T - \text{vec}(A)^T L^T) (\text{vec}(Y) - L \text{vec}(A)) = \\ & = \text{vec}(T_{(1)})^T \text{vec}(T_{(1)}) - \text{vec}(T_{(1)})^T G \text{vec}(A) - \\ & - \text{vec}(A)^T G^T \text{vec}(T_{(1)}) + \text{vec}(A)^T G^T G \text{vec}(A) + \text{vec}(Y)^T \text{vec}(Y) - \\ & - \text{vec}(Y)^T L \text{vec}(A) - \text{vec}(A)^T L^T \text{vec}(Y) + \text{vec}(A)^T L^T L \text{vec}(A) \end{aligned} \quad (5.23)$$

The solution that gives us the update of A is:

$$\begin{aligned} \frac{\partial(5.23)}{\partial 1} = 0 & \Rightarrow -2 \text{vec}(T_{(1)})^T G + 2 \text{vec}(A)^T (G^T G) - 2 \text{vec}(Y)^T L + 2 \text{vec}(A)^T (L^T L) = 0 \\ & \Rightarrow \text{vec}(A)^T (G^T G + L^T L) = \text{vec}(T_{(1)})^T G + \text{vec}(Y)^T L \\ & \Rightarrow \text{vec}(A) (G^T G + L^T L) = G^T \text{vec}(T_{(1)}) + L^T \text{vec}(Y) \end{aligned} \quad (5.24)$$

then, the update of  $A$  can easily be found as:

$$\text{vec}(A) = (G^T G + L^T L)^{-1} (G^T \text{vec}(T_{(1)}) + L^T \text{vec}(Y)) \quad (5.25)$$

where

$$\text{vec}(T_{(1)}) = \text{vec}(AZ_{(1)}^T) = (Z_{(1)} \otimes I_p) \text{vec}(A) = G \text{vec}(A) \quad (5.26)$$

and

$$G^T \text{vec}(T_{(1)}) = G^T G \text{vec}(A) \quad (5.27)$$

Product  $G^T G$  can be efficiently computed using property (4.2):

$$G^T G = (Z_{(1)} \otimes I_p)^T (Z_{(1)} \otimes I_p) = (Z_{(1)}^T \otimes I_p^T) (Z_{(1)} \otimes I_p) = (Z_{(1)}^T Z_{(1)}) \otimes (I_p^T I_p) \quad (5.28)$$

### Updating factor matrix $W$

To find the update of  $W$ , we ignore the first term of (5.12) and so we solve:

$$\min_W \|Y - DAW\|^2. \quad (5.29)$$

The solution of (5.29) with respect to  $W$  gives as the update of  $W$  in the following way:

$$\|Y - DAW\|^2 = Y^T Y - Y^T DAW - W^T A^T D^T Y + W^T A^T D^T DAW \quad (5.30)$$

So, the solution to (5.29) that gives the update of  $W$  can be found through the following equation:

$$\begin{aligned} \frac{\partial(5.30)}{\partial W} = 0 &\Rightarrow -A^T D^T Y - A^T D^T Y + (A^T D^T DA + A^T D^T DA)W = 0 \Rightarrow \\ &\Rightarrow -2A^T D^T Y + 2A^T D^T DAW = 0 \Rightarrow W = (A^T D^T DA)^{-1} A^T D^T Y. \end{aligned} \quad (5.31)$$

The algorithm of CLASS-CP is depicted in Algorithm 2. To compute the factor matrices  $A$ ,  $B$ ,  $C$ , and the coefficient matrix  $W$ , the algorithm performs alternating updates until it converges to a criterion or it reaches a maximum number of iterations. As a criterion we use the *relative change* which can be calculated at each iteration as:

$$\frac{|(f_{new} + g_{new}) - (f_{old} + g_{old})|}{f_{old} + g_{old}} \quad (5.32)$$

---

**Algorithm 2** CLASS-CP: Given a tensor  $\mathcal{T}$  and a set of labels  $Y$ , approximate  $A, B, C$  and  $W$

---

**Require:** tensor  $\mathcal{T}$ , labels  $Y$

**Ensure:** factor matrices  $A, B, C$  and coefficients  $W$

- 1: Initialize  $A, B, C, W$  and hyperparameter  $\lambda_g$
  - 2: **repeat**
  - 3:   updateA using Eq. (5.25)
  - 4:   updateB using Eq. (5.18)
  - 5:   updateC using Eq. (5.19)
  - 6:   updateW using Eq. (5.31)
  - 7: **until** convergence
-



## 6.1 Datasets

For the evaluation of our proposed method CLASS-CP, we conducted experiments with two real-world public datasets that have already been used in the literature [21, 7, 22]. The dataset, called FakeNewsNet, is collected with fact-checking from two platforms, BuzzFeed, via its journalist experts, and PolitiFact. In PolitiFact, journalists and domain experts review the political news and provide fact-checking evaluation results to identify news articles as fake or real. FakeNewsNet includes news content, social context features, and user engagements with fact-checked ground truth labels for the news articles. News content features include meta-information related to the article’s content, such as its title and body text. In contrast, social context features include users’ profile information, relationships, and posts. For our evaluation, we use only social context features, such as the friendship network, that shows the follower/followee structure of users who publish related posts.

Table 6.1: Numbers of FakeNewsNet After Deletion of Nodes (engagements refer to posts, re-posts and replies related with a news article)

Media Platform	BuzzFeed	PolitiFact
# Users	1449	1697
# Engagements	8598	10249
# Social Links	6571	3093
# True news	91	120
# Fake news	91	120

To decrease the size and sparsity of the data, we removed users with a node degree  $< 3$ . In the end, we come up with two tensors of size  $182 \times 1449 \times 1449$  and  $240 \times 1697 \times 1697$  for the BuzzFeed and PolitiFact datasets respectively. Statistics on the datasets after the deletion of the nodes are shown in detail in Table 6.1.

## 6.2 Preliminary Results

As mentioned in Section 5.2, NN-RESCAL does not manage to produce satisfactory results in terms of our chosen evaluation metrics. Specifically, as depicted in Table 6.2, on the PolitiFact dataset, for  $r = 5$  number of factors with the first 80% of news as our train set and the remaining 20% as our test set, performance for all the metrics is lower than random choice. Figure 6.1 also presents the accuracy score achieved with NN-RESCAL in relation to different numbers of labeled instances used in the training stage (20, 40, 60, 80, 100), for a different number of factors ( $r=5,10,20$ ) on PolitiFact. We can see that no other number of labeled instances or factors can achieve better results. Similar patterns were observed for the other metrics, as well as on BuzzFeed. On top of that, NN-RESCAL requires the non-negative factorization algorithm to run four times to produce the result,

and therefore a large number of factors may increase the computation time. Due to all the above, we proceeded with the development of a second model, CLASS-CP. We present its performance analysis in the following sections.

Table 6.2: Performance of NN-RESCAL for  $r=5$

Dataset	Precision	Recall	F1-score	Accuracy
PolitiFact	$0.462 \pm 0.148$	$0.350 \pm 0.123$	$0.394 \pm 0.126$	$0.469 \pm 0.104$
BuzzFeed	$0.440 \pm 0.033$	$0.672 \pm 0.092$	$0.530 \pm 0.048$	$0.408 \pm 0.052$

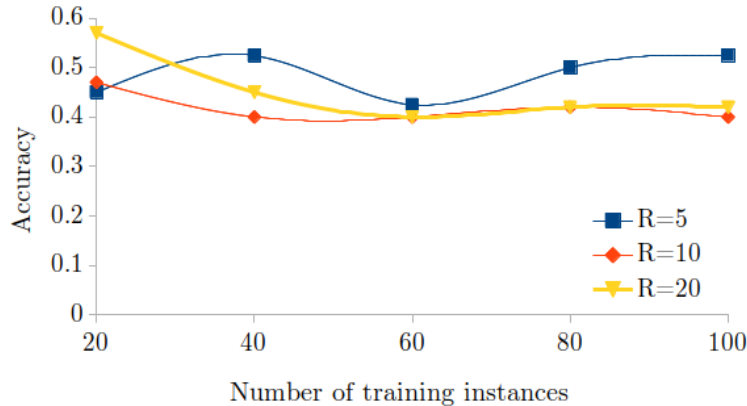


Figure 6.1: The Accuracy Learning Curve of NN-RESCAL for  $r=5,10,20$  on PolitiFact

### 6.3 Experimental Settings

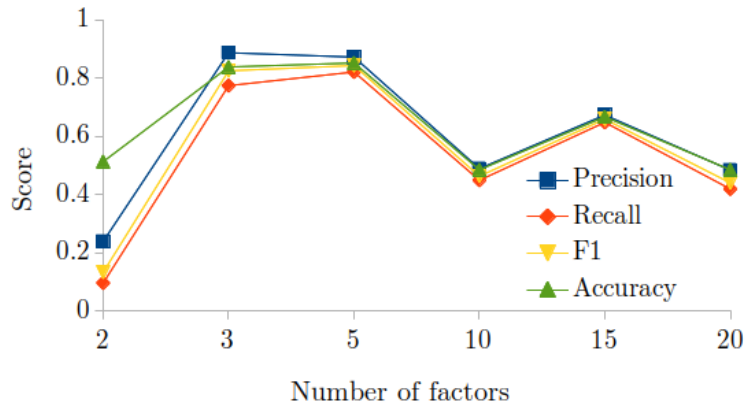
To measure the performance of CLASS-CP, we choose as evaluation metrics accuracy, precision, recall, and F1-score that are commonly used in similar problems [21, 11, 7]. The first 80% of news is our training set and the remaining 20% is for testing. The number of news labeled as true is equal to the number of news labeled as fake. We run the experiments 10 times, independently for each dataset, and we record the average results.

#### Number of factors

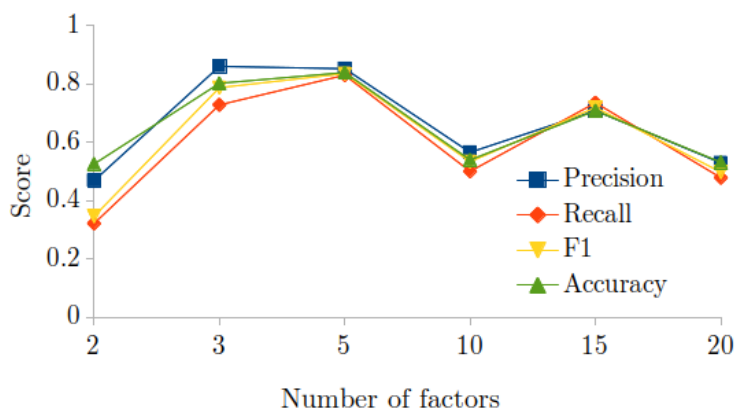
Generally, a large number of factors  $r$  increases the computation time of the factorization. Before the methods' comparison, we wish to choose the optimal number of factors for the CLASS-CP factorization. Figure 6.2 shows the scores of precision, recall, F1-score and accuracy for  $r = 2, 3, 5, 10, 15, 20$ . As demonstrated, we can get the best overall performance for  $r = 5$ . For a small number of factors, CLASS-CP performs slightly worse, whereas a large number of factors lowers performance significantly. This may suggest that less than five of the latent features we extracted are associated with the news veracity.

#### Impact of training set size

We also explore the performance of CLASS-CP with respect to the size of the labeled data used in the training stage. Figure 6.3 presents the F1-score achieved with CLASS-CP in relation to various levels of labeled instances used in the training stage (5%, 10%, 20%,



(a) PolitiFact



(b) BuzzFeed

Figure 6.2: Behavior of CLASS-CP with Different Numbers of Factors for each Dataset

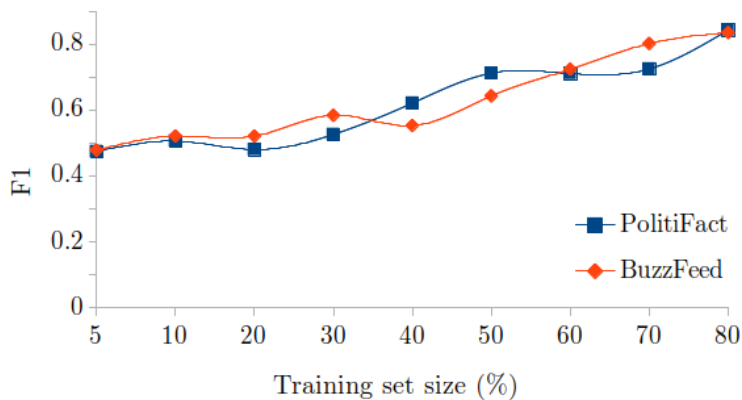


Figure 6.3: The F1-score Learning Curve on BuzzFeed and PolitiFact

30%, 40%, 50%, 60%, 70%, 80%) on both platforms. From the plot, it is evident that the more labeled instances are used for training, the better the performance. We can also see that there is no point of convergence to an upper limit for the F1-score. This can be very

convenient when dealing with data that have a small number of labeled instances available [42]. As shown in Figure 6.4, the same pattern is observed for precision and recall.

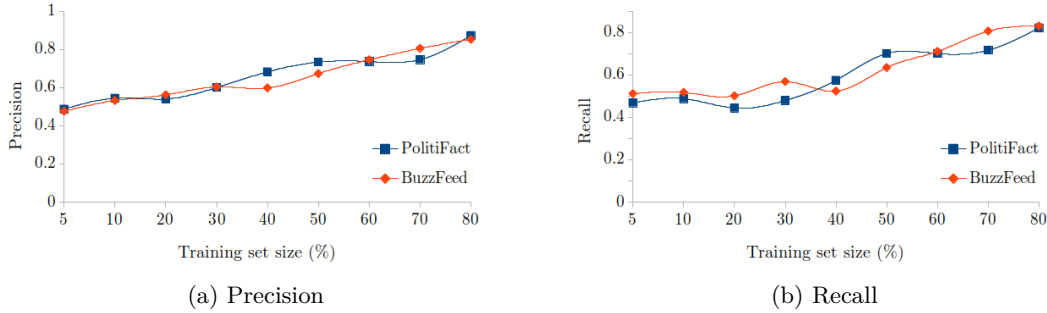


Figure 6.4: Learning Curves for Precision and Recall

### Hyperparameter $\lambda_g$

The value of the hyperparameter is set as  $\lambda_g = 1$  so that the classification loss will contribute equally to the optimization process.

### Execution time

Each CLASS-CP iteration takes less than a minute to finish and we need on average a few minutes ( $< 10$ ) to reach convergence.

## 6.4 Comparison with Other Methods

We compare the proposed CLASS-CP algorithm with the following methods:

- **NN-RESCAL:** The model that utilizes the non-negative RESCAL factorization as described in detail in Chapter 5.
- **SVM:** A baseline SVM model that is built by using the rows of the tensor  $\mathcal{T}$  as input feature vectors.
- **CP+SVM:** A two-step process that combines the original CP factorization with a linear SVM classifier. Firstly, we use CP to calculate the tensor embeddings of posts in an unsupervised manner. Secondly, we train an SVM classifier using the embeddings that were produced by CP in the first step.
- **CP+k-NN:** The CP factorization with a  $k$ -NN classifier. We follow the same procedure as in CP+SVM but instead of an SVM classifier, we use a  $k$ -NN classifier. As suggested by [25, 27], we set  $k$  equal to the square root of the size of the training data, i.e.,  $k = \sqrt{p}$ .
- **TriFN:** A state-of-the-art fake news detection framework that calculates and combines five components: news contents embedding, user embedding, user-news interaction embedding, publisher-news relation embedding, and news classification. We use the same evaluation settings proposed in [23] to compare the CLASS-CP results with the ones demonstrated in their work.



- **Content-SVM:** A baseline content-based bag-of-words model to stand as a comparison against the above-mentioned network-based methods. Firstly, each news piece is segmented into words from the article’s text. Then, we count the number of times each word/term occurs in each document. Each term is considered a feature and we end up with a document-term matrix. These features are then given as input into a baseline SVM classifier.

## 6.5 Results

Table 6.3: Performance Comparison on PolitiFact

Metric	NN-RESCAL	SVM	CP+SVM	CP+ $k$ -NN	TriFN	CLASS-CP
Precision	$0.462 \pm 0.148$	1.000	$0.804 \pm 0.253$	$0.708 \pm 0.033$	$0.867 \pm 0.034$	$0.872 \pm 0.058$
Recall	$0.350 \pm 0.123$	0.670	$0.522 \pm 0.380$	$0.766 \pm 0.066$	$0.893 \pm 0.023$	$0.821 \pm 0.122$
F1	$0.394 \pm 0.126$	0.800	$0.490 \pm 0.168$	$0.735 \pm 0.039$	$0.880 \pm 0.017$	$0.843 \pm 0.089$
Accuracy	$0.469 \pm 0.104$	0.833	$0.577 \pm 0.055$	$0.725 \pm 0.034$	$0.878 \pm 0.020$	$0.852 \pm 0.078$

Table 6.4: Performance Comparison on BuzzFeed

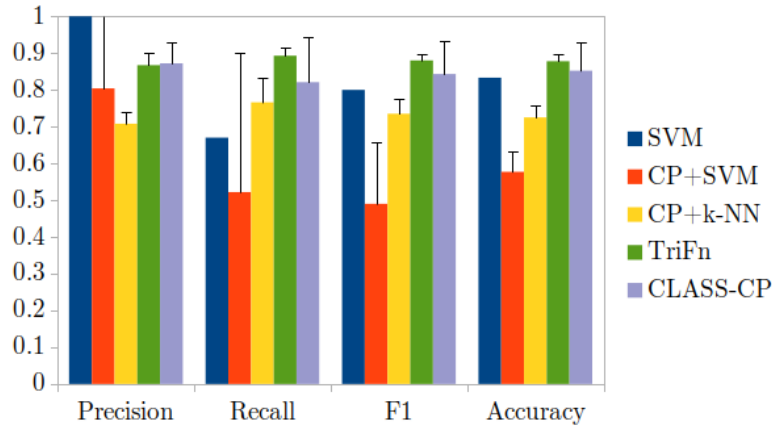
Metric	NN-RESCAL	SVM	CP+SVM	CP+ $k$ -NN	TriFN	CLASS-CP
Precision	$0.440 \pm 0.033$	0.910	$0.671 \pm 0.058$	$0.805 \pm 0.041$	$0.849 \pm 0.040$	$0.852 \pm 0.143$
Recall	$0.672 \pm 0.092$	0.560	1.000	$0.614 \pm 0.137$	$0.893 \pm 0.013$	$0.830 \pm 0.146$
F1	$0.530 \pm 0.048$	0.690	$0.803 \pm 0.044$	$0.689 \pm 0.056$	$0.870 \pm 0.019$	$0.835 \pm 0.127$
Accuracy	$0.408 \pm 0.052$	0.750	$0.750 \pm 0.071$	$0.728 \pm 0.032$	$0.864 \pm 0.026$	$0.839 \pm 0.118$

Table 6.5: Performance of Content-SVM

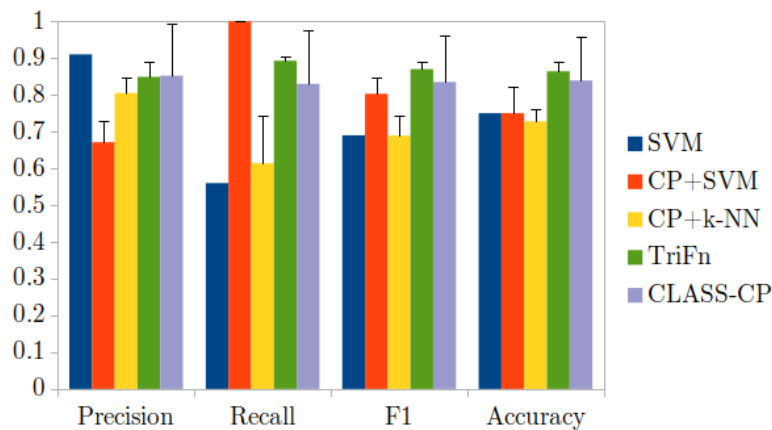
Dataset	Precision	Recall	F1-score	Accuracy
PolitiFact	0.89	0.71	0.79	0.82
BuzzFeed	0.68	0.72	0.70	0.70

Figure 6.5 and Tables 6.3 and 6.4 show the comparison of CLASS-CP with the aforementioned methods on the chosen metrics. Table 6.5 shows the performance of the baseline content-SVM method. From these, we can make the following observations:

- The NN-RESCAL performs worse than chance (random guessing), so we exclude it from the comparison figures.
- The model using the CP factorization is preferable to RESCAL in terms of simplicity and performance. Bad performance of RESCAL may be attributed to the fact that it requires tensors made up of square slices ( $\mathcal{T} \in \mathbb{R}^{p \times u \times u}$ ). This way, RESCAL ends up factorizing users, instead of posts, which makes us assume that the final factorized data is not aware of the posts’ authenticity. On the other hand, CP does not require square slices, thus we can factorize in terms of posts and end up with more posts-specific representations.
- Generally, the SVM baseline provides the highest precision results and performs decently on the other metrics, especially for PolitiFact.



(a) PolitiFact



(b) BuzzFeed

Figure 6.5: Performance Comparison for Different Metrics

- The use of CP does not improve the performance of the SVM in the case of PolitiFact. In contrast, for the BuzzFeed dataset, the CP+SVM configuration seems to improve the performance over the baseline SVM model. Consequently, we can say that there is a mismatch in the performance trend between the two datasets when utilizing the simple CP configuration. However, the CLASS-CP approach seems to produce similar results on both datasets.
- For PolitiFact, CP combined with  $k$ -NN seems to perform better than CP+SVM. Interestingly, in the case of BuzzFeed, we can notice the opposite. Again there is a mismatch in the performance trend. This could be attributed to the values of SVM's and  $k$ -NN's built-in parameters. We assume that by tuning them we are going to get similar results for both datasets.
- CLASS-CP significantly outperforms SVM, CP+SVM and CP+  $k$ -NN both on PolitiFact and BuzzFeed datasets.
- CLASS-CP outperforms TriFN in terms of precision on both datasets.
- CLASS-CP performs comparably to TriFN for all metrics in both datasets.

- In comparison with the other methods, CLASS-CP produces higher deviation numbers (between 0.058 and 0.146). This suggests that data points produced by the class-aware CLASS-CP may spread in a higher distance from the mean, compared, for example, to the state-of-the-art TriFN (that produces deviation between 0.013 and 0.040). To some extent, this may be attributed to the relatively smaller sample size that is used in our experiments (see section 6.1) that may lead to bigger differences between the data points.
- The content-SVM model performs decently compared to the network-based baselines. Consequently, the integration of content into our CLASS-CP framework is considered future work.

From these observations, we can conclude that CLASS-CP, through a simple approach, a small number of factors and by using only one source of information from the network, performs equally well with the state-of-the-art framework TriFN that needs the combination of five components coming both from the network and the news content to produce satisfying results. In other words, (i) users' friendships on social media are a useful indicator of an article's accuracy, and (ii) the integration of the class-label information inside the factorization process can noticeably improve the model's performance. These conclusions provide a promising new direction for the development of network-based class-aware frameworks that are more accurate, rely on fewer parameters, and are less complicated and quicker to produce.



In this thesis, we present a tensor-based method for fake news detection on social media that relies only on the network of users that interact with the news. At first, we propose a novel tensor-based representation of the news, that utilizes the users' follower/followee structure. In order to exploit this information, after rejecting the RESCAL factorization due to unsatisfactory performance, we develop an extension of the standard CP factorization that integrates the available class-label information into the tensor factorization itself. Finally, we end up with a class-aware semi-supervised tensor factorization.

Our detailed experiments on two real-world public datasets show that our method, CLASS-CP, through a relatively simpler computation approach, can become competitive in terms of performance when compared with state-of-the-art network-based methods. The results confirm our initial intuition, that the friendship networks of users can accurately identify the truthfulness of the news they share on social media. The results also prove the significance of the class-label information when it is incorporated into the factorization process, as a unified optimization step.

As an extension, we aim to investigate the performance of CLASS-CP when more features, both from the network and the content, are included. Moreover, we intend to apply our approach to more datasets as well as to investigate its scalability in relation to the size of the data used. Finally, another interesting direction would be to consider new ways of representing the available data using tensors in order to incorporate them into the proposed framework.



## Βιβλιογραφία (References)

---

- [1] Pew Research Center. News Use Across Social Media Platforms. <https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/>, 2018.
- [2] Niall J Conroy, Victoria L Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. *Association for Information Science and Technology*., 2015.
- [3] Reuters Institute for the Study of Journalism. Reuters Institute Digital News Report. <http://media.digitalnewsreport.org/wp-content/uploads/2018/06/digital-news-report-2018.pdf?x89475>, 2018.
- [4] Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and sentiment in tweets. *ACM Transactions on Internet Technology (TOIT)*., 2017.
- [5] Hunt Allcott and Matthew Gentzkow. Social Media and Fake News in the 2016 Election. *Technical Report. National Bureau of Economic Research*, 2017.
- [6] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, 2012.
- [7] Song Feng, Ritwik Banerjee, and Yejin Choi. Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations Newsletter*, 2017.
- [8] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. In search of credible news. *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, 2016.
- [9] Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. *Proceedings of the second workshop on computational approaches to deception detection*, 2016.
- [10] Seyedmehdi Hosseini-motlagh and Evangelos E. Papalexakis. Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles, 2018.
- [11] Marco L. Della Vedova, Eugenio Tacchini, Stefano Moret, Gabriele Ballarin, Massimo DiPierro, and Luca de Alfaro. Automatic Online Fake News Detection Combining Content and Social Signals. *2018 22nd Conference of Open Innovations Association (FRUCT)* , 2018.
- [12] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *CoRR*, 2017.
- [13] Natali Ruchansky, Sungyong Seo, and Yan Liu. CSI: A Hybrid Deep Model for Fake News Detection. *In Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM)*, 2017.

- [14] Bing Tian Dai, Freddy Chong Tat Chua, and Ee-Peng Lim.. Structural analysis in multi-relational social networks. *Proceedings of the 2012 SIAM international conference on data mining (CIKM)*, 2012.
- [15] Kai Shu, H Russell Bernard, and Huan Liu. Studying fake news via network analysis: detection and mitigation. *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining (CIKM)*, 2019.
- [16] Richard A. Harshman and Margaret E. Lundy. PARAFAC: Parallel factor analysis. *Computational Statistics & Data Analysis*, 1994.
- [17] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 1966.
- [18] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. *ICML'11 Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011.
- [19] Tamara Gibson Kolda. Multilinear operators for higher-order decompositions, 2006.
- [20] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.
- [21] Kai Shu, Suhang Wang, and Huan Liu. Exploiting Tri-Relationship for Fake News Detection. *CoRR*, 2017.
- [22] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. *CoRR*, 2018.
- [23] Kai Shu, Suhang Wang and Huan Liu. Beyond News Contents: The Role of Social Context for Fake News Detection Fake News on Social Media. *Proceedings of 12th ACM International Conference on Web Search and Data Mining (WSDM 2019)*, 2019.
- [24] European Commission, Directorate-General for Communication. Flash Eurobarometer on Fake News and Online Disinformation. <https://ec.europa.eu/digital-single-market/en/news/final-results-eurobarometer-fake-news-and-online-disinformation>, 2018.
- [25] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification, 2nd Edition*, 2001.
- [26] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 2009.
- [27] Ahmad Basheer Hassanat, Mohammad Ali Abbadi, Ghada Awad Altarawneh and Ahmad Ali Alhasanat. Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach. *International Journal of Computer Science and Information Security, International Journal of Computer Science and Information Security*, 2014.
- [28] Seyedmehdi Hosseinimotlagh and Evangelos E. Papalexakis. Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles, 2018.



- [29] Gisel Bastidas Guacho, Sara Abdali, Neil Shah and Evangelos E. Papalexakis. Semi-supervised Content-based Detection of Misinformation via Tensor Embeddings. *CoRR*, 2018.
- [30] Aditi Gupta and Ponnurangam Kumaraguru. Credibility ranking of tweets during high impact event. *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, 2012.
- [31] Manish Gupta, Peixiang Zhao, and Jiawei Han. Evaluating event credibility on twitter. *PSDM*, 2012.
- [32] Zhiwei Jin, Juan Cao, Yu-Gang Jiang, and Yongdong Zhang. News credibility evaluation on microblog with a hierarchical propagation model. *ICDM*, 2014.
- [33] Benjamin D. Horne and Sibel Adali. This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. *Proceedings of the 2nd International Workshop on News and Public Opinion at ICWSM*, 2017.
- [34] Zhiwei Jin, Juan Cao, Yongdong Zhang and Jiebo Luo. News verification by exploiting conflicting social viewpoints in microblogs. *AAAI*, 2016.
- [35] Carlos Castillo, Marcelo Mendoza and Barbara Poblete. Information Credibility on Twitter. *Proceedings of the 20th international conference on World wide web*, 2011.
- [36] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015.
- [37] Yize Li, Jiazhong Nie and Yi Zhang. Contextual Recommendation based on Text Mining. *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.
- [38] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.
- [39] Bokai Cao, Chun-Ta Lu, Xiaokai Wei, Yu S Philip and Alex D Leow. Semi-supervised tensor factorization for brain network analysis. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016.
- [40] Yan-Hui Xiao, Zhen-Feng Zhu, Yao Zhao and Yun-Chao Wei. Class-driven non-negative matrix factorization for image representation. *Journal of Computer Science and Technology*, 2013.
- [41] Georgios Katsimpras and Georgios Paliouras. Semi-Supervised Tensor Factorization for Identifying Nodes in Complex Social Networks. *The Joint International Workshop on Social Influence Analysis and Mining Actionable Insights from Social Networks*, 2018.
- [42] Georgios Katsimpras and Georgios Paliouras. Class aware tensor factorization for multi-relational classification. *Information Processing & Management*, 2019.

- [43] Kai Shu, Suhang Wang and Huan Liu. Understanding User Profiles on Social Media for Fake News Detection. *Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018.
- [44] Mohamad Jouni, Mauro Dalla Mura, Pierre Comon. Classification of Hyperspectral Images as Tensors Using Nonnegative CP Decomposition. *hal-01998121*, 2019.
- [45] Marco L. Della Vedova, Eugenio Tacchini, Stefano Moret, Gabriele Ballarin, Massim DiPierro and Luca de Alfaro. Automatic Online Fake News Detection Combining Content and Social Signals. *Proceedings of the 22nd Conference of Open Innovations Association (FRUCT)*, 2018.
- [46] Maximilian Nickel, Volker Tresp. and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. *Proceedings of the 21st international conference on World Wide Web*, 2012.
- [47] Matthew Gentzkow, Jesse M Shapiro, and Daniel F Stone. Media bias in the marketplace: Theory. *Technical report, National Bureau of Economic Research*,, 2014.
- [48] Denis Krompaß, Maximilian Nickel, and Volker Tresp. Large-Scale Factorization of Type-Constrained Multi-Relational Data. *Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA)*, 2014.
- [49] Denis Krompaß, Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Non-negative tensor factorization with rescal. *Tensor Methods for Machine Learning, ECML workshop*, 2013.
- [50] Kimberly Elmore and Michael Richman. Euclidean Distance as a Similarity Metric for Principal Component Analysis. *Monthly Weather Review*, 2001.
- [51] Archana Singh, Avantika Yadav, and Ajay Rana. K-means with Three different Distance Metrics. *International Journal of Computer Applications*, 2013.
- [52] Liwei Wang, Yan Zhang, and Jufu Feng. On the Euclidean Distance of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.
- [53] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. Prominent features of rumor propagation in online social media. *ICDM*, 2013.
- [54] Shashank Gupta, Raghuv eer Thirukovalluru, Manjira Sinha and Sandya Mannarswamy. CIMTDetect: A Community Infused Matrix-Tensor Coupled Factorization Based Method for Fake News Detection. *ASONAM*, 2018.
- [55] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. *WWW*, 2013.
- [56] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia*, 2017.
- [57] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting Fake News: Image Splice Detection via Learned Self-Consistency. *ECCV 2018: Computer Vision*, 2018.

- 
- [58] Wee Yong Lim, Mong Li Lee, and Wynne Hsu. End-to-End Time-Sensitive Fact Check. *ROME: Workshop on Reducing Online Misinformation Exposure*, 2019.
- [59] Olga Papadopoulou, Dimitrios Giomelakis, Lazaros Apostolidis, Symeon Papadopoulos, and Yiannis Kompatsiaris. Context Aggregation and Analysis: A Tool for User-Generated Video Verification. *ROME: Workshop on Reducing Online Misinformation Exposure*, 2019.
- [60] Ankur Padia, Konstantinos Kalpakis, and Tim Finin. Inferring Relations in Multi-relational Knowledge Graphs with Tensor Decomposition. *IEEE Big Data*, 2016.
- [61] Kai-Wei Chang, Wen-tau Yih, and Chris Meek. MultiRelational Latent Semantic Analysis. *EMNLP*, 2013.
- [62] Melkamu Beyene, Pierre-Edouard Portier, Solomon Atnafu, and Sylvie Calabretto. Tensor factorization for cross lingual entity co-reference resolution in the linked open data. *MEDES*, 2015.
- [63] Anh Huy Phan and Andrzej Cichocki. Tensor Decompositions for Very Large Scale Problems. 2010.
- [64] Gustavo De Assis Costa and José M. Parente de Oliveira. A Relational Learning Approach for Collective Entity Resolution in the Web of Data. *COLD*, 2014.
- [65] Oguz Kaya. A Parallel Nonzero CP Decomposition Algorithm for Higher Order Sparse Data Analysis. *INFOCOMP*, 2017.
- [66] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen and Yajun Wang. Prominent features of rumor propagation in online social media. *IEEE 13th International Conference on Data Mining*, 2013.

