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

Εξηγήσιμη Ομαδοποίηση σε Ευσταθή Στιγμιότυπα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΗΛΙΑΣ ΠΑΠΑΝΙΚΟΛΑΟΥ

Επιβλέπων : Δημήτριος Φωτάκης
Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2023

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

Εξηγήσιμη Ομαδοποίηση σε Ευσταθή Στιγμιότυπα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΗΛΙΑΣ ΠΑΠΑΝΙΚΟΛΑΟΥ

Επιβλέπων : Δημήτριος Φωτάκης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 4η Απριλίου 2023.

Δημήτριος Φωτάκης Καθηγητής Ε.Μ.Π.	Ευάγγελος Χατζηαφράτης Καθηγητής UC Santa Cruz	Αριστείδης Παγουρτζής Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2023

Ηλίας Παπανικολάου

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

Περίληψη

Αυτή η διπλωματική ασχολείται με το πρόβλημα της εξηγήσιμης ομαδοποίησης (explainable clustering) κάτω από παραδοχές "ευστάθειας" των εισόδων. Η εξηγήσιμη ομαδοποίηση είναι μια "ερμηνεύσιμη" διαδικασία που αναπτύχθηκε από τους Dasgupta κ.ά. και στοχεύει να παρέχει συνοπτικές εξηγήσεις για την συμπερίληψη κάθε σημείου στη συστάδα του (cluster). Το ερώτημα στο οποίο προσπαθούμε να απαντήσουμε είναι εάν το "Τίμημα της Εξηγησιμότητας", δηλαδή το εγγενές κόστος που προκύπτει εξαιτίας της περιορισμένης μορφής των λύσεων που εγγυώνται την εξηγησιμότητα, μπορεί να μειωθεί εάν υποθέσουμε ότι τα στιγμιότυπα ομαδοποίησης εισόδου ικανοποιούν είτε την ιδιότητα "εγγύτητας" είτε την ιδιότητα της "ευστάθειας σε διαταραχές". Αφού εισαγάγουμε διάφορες έννοιες ευστάθειας στο πλαίσιο της ομαδοποίησης και αναλύσουμε τους πιο σημαντικούς αλγόριθμους για την εξηγήσιμη ομαδοποίηση, αποδεικνύουμε ότι εάν τα στιγμιότυπα εισόδου ικανοποιούν την ιδιότητα της a-εγγύτητας με $a = \Omega\left(kd^{\frac{1}{p}}\right)$, μπορούμε να πάρουμε εξηγήσιμους αλγορίθμους σταθερού λόγου προσέγγισης. Στη συνέχεια, μελετάμε την ευστάθεια αρκετών δύσκολων στιγμιοτύπων της εξηγήσιμης ομαδοποίησης. Καταφέρνουμε να δείξουμε ότι αυτή η εξάρτηση στο πλήθος των διαστάσεων d και στο πλήθος των clusters k είναι αναγκαία, αφού υπάρχουν ορισμένα δύσκολα στιγμιότυπα ομαδοποίησης, των οποίων η συνάρτηση κόστους είναι η ℓ_p αντικειμενική συνάρτηση, που ικανοποιούν την a-εγγύτητα με $a = \Omega\left(kd^{\frac{1}{p}}\right)$, ενώ υπάρχουν στιγμιότυπα που είναι $\Omega(\sqrt{d})$-ευσταθή σε διαταραχές στην περίπτωση του k-median clustering ($p = 1$). Για να δείξουμε το δεύτερο αποτέλεσμα, αποδεικνύουμε ότι εάν ένα στιγμιότυπο ομαδοποίησης ικανοποιεί την ιδιότητα της a-εγγύτητας μαζί με μια ιδιότητα που διασφαλίζει ότι όλες οι συστάδες στη βέλτιστη ομαδοποίηση έχουν περίπου το ίδιο κόστος, τότε αυτό το στιγμιότυπο είναι $\Omega(\sqrt{a})$-ευσταθές σε διαταραχές. Συμπεραίνουμε ότι δεν είναι εύλογο να υποθέσουμε ότι τα στιγμιότυπα ομαδοποίησης που συναντάμε στην πράξη είναι αρκετά ευσταθή (με την έννοια των παραπάνω παραδοχών), ώστε να μειωθεί το Τίμημα της Εξηγησιμότητας.

Λέξεις Κλειδιά Ομαδοποίηση, Ερμηνεύσιμη Μηχανική Μάθηση, Ανάλυση πέρα από την χειρότερη περίπτωση, εξηγήσιμη ομαδοποίηση, ευστάθεια διαταραχών, μετρικοί χώροι

Abstract

This thesis is concerned with the explainable clustering problem under stability assumptions. Explainable Clustering is an interpretation method developed by Dasgupta et al. that aims to provide concise explanations for the inclusion of each data point in a cluster. The question that we try to answer is whether the Price of Explainability, i.e. the inherent cost due to the restricted solution format that guarantees explainability, can be reduced if we assume that the input clustering instances satisfy either the proximity or the perturbation stability property. After we introduce several stability notions in the context of clustering and analyze the most important algorithms for explainable clustering, we show that under a-center stability, with $a = \Omega(kd^{\frac{1}{p}})$ there are explainable algorithms with constant approximation ratio. Next, we study the stability of several hard explainable clustering instances and prove that this dependence on the number of dimensions d and clusters k is necessary. More specifically, we manage to show that there are some hard clustering instances with the ℓ_p objective that satisfy the a-proximity with $a = \Omega\left(kd^{\frac{1}{p}}\right)$ and there exist $\Omega(\sqrt{d})$-(metric) perturbation stable instances in the k-median case ($p = 1$), where d is the number of the dimensions of the dataset. To prove the second result, we show that if a clustering instance satisfies the a-proximity property along with a property that ensures that all clusters in the optimal clustering have roughly the same cost, then this instance is $\Omega(\sqrt{a})$-metric perturbation stable. We conclude that it is not reasonable to assume that practical instances are stable enough for the Price of Explainability to reduce, under these stability assumptions.

Keywords Clustering, Interpretable Machine Learning, Beyond the Worst-Case analysis, Explainable Clustering, Perturbation Stability, Metric Spaces

Acknowledgments / Ευχαριστίες

Θα ήθελα, καταρχάς, να ευχαριστήσω τον κύριο Δημήτρη Φωτάκη που μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα και για την πολύτιμη καθοδήγηση και τις συμβουλές του, τόσο στο πλαίσιο αυτής της διπλωματικής εργασίας, όσο και έξω από αυτό.
Επίσης, θέλω να ευχαριστήσω ιδιαίτερα τον Παναγιώτη Πατσιλινάκο και την Ελένη Ψαρουδάκη, για τις πολλές ώρες που μου αφιέρωσαν στις εβδομαδιαίες συναντήσεις μας και την υπομονή που έδειξαν. Ελπίζω η συνεργασία μας να τους φάνηκε το ίδιο ευχάριστη και ενδιαφέρουσα όσο και σε εμένα.
Θέλω, ακόμη, να εκφράσω την ευγνωμοσύνη μου στον κύριο Στάθη Ζάχο και τον κύριο Άρη Παγουρτζή για την βοήθειά τους στις αιτήσεις μου για διδακτορικό και για τις συμβουλές τους.
Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου Δημήτρη Δεσποτίδη, Μάριο Μάνταλο και την υπόλοιπη παρέα μας στη σχολή, χωρίς τους οποίους δε θα μπορούσα να φανταστώ τη ζωή μου τα προηγούμενα 5 χρόνια. Τους ευχαριστώ για τις ωραίες στιγμές που περάσαμε μαζί και που έδωσαν νόημα στα φοιτητικά μου χρόνια. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον φίλο μου Διονύση Αρβανιτάκη, ο οποίος τα τελευταία χρόνια και ειδικά στο δύσκολο αυτό τελευταίο εξάμηνο μου έχει σταθεί όσο κανείς άλλος. Τον ευχαριστώ για την πολύτιμη ψυχολογική υποστήριξη, τη συνεργασία μας σε άπειρες εργασίες και τις πολύ ενδιαφέρουσες συζητήσεις που έχουμε κάνει όλο αυτό τον καιρό. Τους εύχομαι ό,τι καλύτερο και ελπίζω να κρατάμε πάντα επαφή στα χρόνια που θα ακολουθήσουν.

Η εργασία αυτή είναι αφιερωμένη στη μνήμη του μπαμπά μου.

Contents / Περιεχόμενα

List of Tables / Πίνακες

List of Figures / Εικόνες

Εκτεταμένη Ελληνική Περίληψη

Το έναυσμα πίσω από το ερώτημα που φιλοδοξούμε να απαντήσουμε προέρχεται από δύο σύγχρονα πεδία της Θεωρητικής Πληροφορικής. Η πρώτη είναι η "Ανάλυση Πέρα από την Χειρότερη Περίπτωση" (Beyond the Worst-Case Analysis), η οποία αποσκοπεί να παρέχει μια πιο ρεαλιστική μελέτη της απόδοσης των αλγορίθμων σε πρακτικά σενάρια, σε αντίθεση με την παραδοσιακή ανάλυση χειρότερης περίπτωσης, η οποία μερικές φορές μπορεί να είναι πολύ απαισιόδοξη και παραπλανητική. Το δεύτερο πεδίο είναι αυτό της Ερμηνεύσιμης Μηχανικής Μάθησης (Interpretable Machine Learning), στόχος της οποίας είναι να σχεδιάσει μοντέλα μηχανικής μάθησης που παρέχουν κατανοητές εξηγήσεις των αποφάσεών τους, οι οποίες μπορούν εύκολα να αξιοποιηθούν από τον άνθρωπο. Ο στόχος αυτής της διπλωματικής είναι να μελετήσει τη μέθοδο εξηγήσιμης ομαδοποίησης (Explainable Clustering), η οποία είναι ένα ερμηνεύσιμο μοντέλο μηχανικής μάθησης, από την οπτική της ανάλυσης που εκτείνεται πέρα από την ανάλυση της χειρότερης περίπτωσης, αξιολογώντας την απόδοση εξηγήσιμων αλγορίθμων ομαδοποίησης (explainable clustering algorithms) σε ευσταθή στιγμιότυπα εισόδου (stable clustering instances), δηλαδή σε προβλήματα που είναι πιο πιθανό να προκύψουν στην πράξη και ικανοποιούν ορισμένες υποθέσεις "ομαδοποιησιμότητας".
Ωστόσο, πριν μιλήσουμε για οτιδήποτε άλλο, πρέπει να προσφέρουμε κάποιο υπόβαθρο για το πρόβλημα της k-ομαδοποίησης (k-clustering). Πρώτα απ 'όλα, αφού εξοικειωθούμε με τα πιο σημαντικά αποτελέσματα για το πρόβλημα k-clustering, είναι πιο εύκολο να εκτιμήσουμε τη δυσκολία του και να κατανοήσουμε τον λόγο που πρέπει στρέψουμε την προσοχή μας στην ανάλυση πέρα από τη χειρότερη περίπτωση. Δεύτερον, αυτή η συζήτηση θα δικαιολογήσει την ανάγκη για επεξηγήσιμους αλγορίθμους ομαδοποίησης.

0.1 Ομαδοποίηση

Η Ομαδοποίηση (Clustering ή Cluster Analysis) είναι μία τεχνική *εκμάθησης χωρίς επίβλεψη* (unsupervised learning) που αποσκοπεί στην οργάνωση των δεδομένων εισόδου (μοτίβα) σε "εύλογες" ομάδες, που ονομάζονται *συστάδες* (clusters), έτσι ώστε να αποκαλυφθούν ομοιότητες και διαφορές μεταξύ αυτών των δεδομένων και να εξαχθούν χρήσιμα συμπεράσματα για αυτά. Η ιδέα της ομαδοποίησης όμοιων μοτίβων συναντάται σε πολλά επιστημονικά πεδία [22], όπως η βιολογία, η ζωολογία, οι επιστήμες υγείας (ψυχιατρική, παθολογία), οι κοινωνικές επιστήμες (κοινωνιολογία, αρχεολογία), η γεωγραφία, η γεολογία, η μηχανική. Εκτός αυτού, η ομαδοποίηση συνιστά μια πρωτόγονη πνευματική δραστηριότητα των ανθρώπων, ώστε να αποφεύγουν να επεξεργάζονται κάθε πληροφορία που λαμβάνουν ξεχωριστά, κατηγοριοποιώντας τα αντικείμενα που μοιράζονται κοινά χαρακτηριστικά στην ίδια ομάδα και αποδίδοντας σε κάθε μέλος της ομάδας αυτής τα γνωρίσματα αυτά. Στην πλειοψηφία των προβλημάτων ομαδοποίησης που προκύπτουν στην πράξη, τα δεδομένα εισόδου αναπαρίστανται από ένα σύνολο X, που είναι υποσύνολο του $\mathbb{R}^d$, όπου το $d \in \mathbb{N}^*$ ονομάζεται *διάσταση* των δεδομένων. Κάθε σημείο $x \in X$ είναι ένα d-διάστατο διάνυσμα *χαρακτηριστικών* (*features*) που κωδικοποιεί σημαντικές πληροφορίες για ένα συγκεκριμένο μοτίβο· για κάθε $i \in [d]$, x_i είναι η τιμή του i-στού χαρακτηριστικού του μοτίβου. Ο στόχος μας είναι να διαμερίσουμε το σύνολο X σε k μη κενά και ξένα μεταξύ τους σύνολα, ώστε να ελαχιστοποιείται μια συγκεκριμένη *συνάρτηση κόστους*(*cost function*) ή *αντικειμενική συνάρτηση*(*objective function*), η οποία είναι σχεδιασμένη με τέτοιο τρόπο, ώστε στις λύσεις χαμηλού κόστους, κάθε cluster περιέχει δεδομένα τα οποία είναι "κοντά" μεταξύ τους. Είναι, λοιπόν, σαφές, ότι για να ορίσουμε τυπικά το

στόχο αυτό, οφείλουμε να καθορίσουμε ένα *μέτρο εγγύτητας* (*proximity measure*), δηλαδή μία έννοια *απόστασης* μεταξύ των μοτίβων εισόδου που προσδιορίζει πόσο "όμοια" είναι. Παρ' όλα αυτά, υπάρχουν πολλά διαφορετικά μέτρα εγγύτητας που οδηγούν σε ικανοποιητικές διαμερίσεις, επομένως η επιλογή τους εξαρτάτε από την εκάστοτε εφαρμογή ομαδοποίησης που καλούμαστε να λύσουμε.
Αναμφισβήτητα, η πιο μελετημένη και συχνά χρησιμοποιούμενη συνάρτηση κόστους είναι η k-means αντικειμενική συνάρτηση (k-means objective). Εάν διαμερίσουμε το σύνολο X σε k clusters $C_1, C_2, ..., C_k$, τότε το k-means κόστος της ομαδοποίησης είναι:

$$\mathcal{H}_2(C_1, C_2, ..., C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} \delta^2(x, \mu_i)$$

όπου η συνάρτηση $\delta : \mathbb{R}^d \times \mathbb{R}^d \to [0, +\infty)$ ονομάζεται *μετρική* (*metric*), και υπολογίζει την απόσταση μεταξύ δύο μοτίβων εισόδου $x, y \in \mathbb{R}^d$, ενώ το $\mu_i = \arg\min_{\mu \in \mathbb{R}^d} \sum_{x \in C_i} \delta(x, \mu)^2$ καλείται *κέντρο* (*center*) του cluster C_i. Συνήθως, όταν μελετάμε τη συνάρτηση κόστους k-means σε αυτό το πλαίσιο, παίρνουμε τη μετρική να είναι: $\delta(x, y) = \|x - y\|_2$, δηλαδή η *Ευκλείδεια απόσταση* μεταξύ των x και y.
Παρομοίως, η k-median αντικειμενική συνάρτηση ορίζεται ως εξής:

$$\mathcal{H}_1(C_1, C_2, ..., C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} \delta(x, \mu_i)$$

όπου $\mu_i = \arg\min_{\mu \in \mathbb{R}^d} \sum_{x \in C_i} \delta(x, \mu)$ αι συνήθως επιλέγουμε we usually choose $\delta(x, y) = \|x - y\|_1$. Αυτές οι δύο συναρτήσεις κόστους ανήκουν σε μια ευρύτερη κατηγορία αντικειμενικών συναρτήσεων, όπου το κόστος της ομαδοποίησης υπολογίζεται αναθέτοντας ένα κέντρο σε κάθε cluster και ο στόχος είναι να βρούμε την k-διαμέριση C του X και ένα σύνολο κέντρων M τα οποία να ελαχιστοποιούν την αντικειμενική συνάρτηση.
Τα τελευταία χρόνια, λόγω της δημοτικότητας των παραπάνω διατυπώσεων του προβλήματος της ομαδοποίησης, ποικίλοι αλγόριθμοι έχουν προταθεί για την επίλυσή τους. Ωστόσο, έχει αποδειχθεί ότι για τα περισσότερα από τα προβλήματα ομαδοποίησης, συμπεριλαμβανομένων των k-means και k-median, είναι NP-hard να υπολογιστεί η βέλτιστη διαμέριση, στη χειρότερη περίπτωση [8] [21] [24]. Αν και μπορεί να γίνει σε $O\left(n^{kd}\right)$ χρόνο [9], δεδομένου ότι η ομαδοποίηση εκτελείται συνήθως για σύνολα δεδομένων υψηλών διαστάσεων με πολλά δεδομένα, αυτό είναι απαράδεκτη επίδοση για πρακτικές εφαρμογές και έτσι έχουμε καταφύγει στη χρήση ταχύτερων αλγορίθμων ομαδοποίησης που δεν επιστρέφουν απαραίτητα τη βέλτιστη λύση. Ένας τέτοιος αλγόριθμος για την περίπτωση του k-means είναι ο αλγόριθμος του Lloyd [5], που αλλιώς ονομάζεται αλγόριθμος k-means, ο οποίος έχει χρησιμοποιηθεί εκτενώς στην πράξη, λόγω της απλότητας και της ικανοποιητικής απόδοσής του για πολυάριθμες πρακτικές εφαρμογές. Είναι ένας αλγόριθμος *τοπικής αναζήτησης* (*local search*), δηλαδή ένας επαναληπτικός αλγόριθμος που βρίσκει μία αρχική ομαδοποίηση και σε κάθε επανάληψη βελτιώνει την τρέχουσα λύση βρίσκοντας μια καλύτερη στη "γειτονιά" αυτής της λύσης, έως ότου να μην μπορεί να υπάρξει βελτίωση (αυτό είναι εγγυημένο ότι θα συμβεί). Αν και έχει αποδειχθεί ότι ο αλγόριθμος του Lloyd μπορεί να παράγει αυθαίρετα κακές ομαδοποιήσεις στη χειρότερη περίπτωση, ακόμη και για σταθερό πλήθος δεδομένων n και clusters k [19], αποδίδει αξιοσημείωτα στην πράξη, βρίσκοντας ικανοποιητικές διαμερίσεις μετά από μερικά μόνο βήματα εκτέλεσης. Όπως θα δούμε στην επόμενη ενότητα, αυτό το φαινόμενο έχει παρακινήσει τους επιστήμονες να μελετήσουν αυτόν τον αλγόριθμο καθώς και την ομαδοποίηση γενικά, μέσα από το πρίσμα της ανάλυσης πέρα από τη χειρότερη περίπτωση.
Εκτός από τα παραπάνω, έχουν αναπτυχθεί αρκετοί αλγόριθμοι προσέγγισης για την ομαδοποίηση k-means. Ο Vassilvitskii κ.ά. δημιούργησαν τον αλγόριθμο k-means++[19], που είναι ίσως ο πιο συχνά χρησιμοποιούμενος αλγόριθμος k-means στην πράξη, κάνοντας μια μικρή τροποποίηση στον αλγόριθμο του Lloyd που αφορά την αρχική λύση της τοπικής αναζήτησης. Αυτός ο αλγόριθμος επιτυγχάνει λόγο προσέγγιστης $O(\log k)$, δηλαδή οι λύσεις που επιστρέφει κοστίζουν

το πολύ $O(\log k)$ φορές περισσότερο από τη βέλτιστη ομαδοποίηση. Μια σειρά εργασιών για τη μείωση του λόγου προσέγγισης των αλγορίθμων ομαδοποίησης οδήγησε σε αλγορίθμους *σταθερού λόγου προσέγγισης* [49] [23], με τον καλύτερο λόγο προσέγγισης μέχρι στιγμής να είναι 6.357, που αποδείχθηκε από τους Ahmadian, Norouzi-Fard, Svensson και Ward[49].
Όσον αφορά την περίπτωση του k-median, με μια τροποποίηση του αλγόριθμου k-means++ λαμβάνουμε $O(logk)$ λόγο προσέγγισης [19]. Επιπλέον, ο Li και ο Svensson σχεδίασαν έναν αλγόριθμο προσέγγισης $1+\sqrt{3}+\epsilon$ για το k-median πρόβλημα [32], η οποία αργότερα βελτιώθηκε σε $2,611+\epsilon$ από τους Byrka, Rybicki, Srinivasan και Trinh [43].
Αξίζει επίσης να αναφερθούμε σε μερικά ενδιαφέροντα αποτελέσματα σχετικά με τη δυσκολία πολλών προβλημάτων ομαδοποίησης. Όπως αναφέραμε προηγουμένως, ο υπολογισμός της βέλτιστης ομαδοποίησης είναι NP-hard και για τις συναρτήσεις κόστους k-means και για k-median. Οι Awasthi, Charikar, Krishnaswamy και Sinop [35] έδειξαν επίσης ότι είναι δύσκολο να προσεγγίσουμε την βέλτιστη k-means λύση με λόγο καλύτερο από $(1+\epsilon)$ για κάποια θετική σταθερά ϵ. Επιπλέον, οι Bhattacharya, Goyal και Jaiswal [54] έχουν αποδείξει ότι στο πρόβλημα k-median στον Ευκλείδειο χώρο δεν μπορούμε να πετύχουμε καλύτερο λόγο προσέγγισης $(1+\epsilon)$, για κάποιο $\epsilon > 0$, εάν αποδεχτούμε την εικασία Μοναδικών Παιχνιδιών (Unique Games Conjecture), ενώ στην *διακριτή* εκδοχή του k-median προβλήματος, όπου περιορίζουμε τα κέντρα του clustering να ανήκουν στο σύνολο εισόδου X, είναι NP-hard να προσεγγίσουμε τη βέλτιστη λύση με παράγοντα καλύτερο από $(1+\frac{2}{e})$ [11].

0.2 Πέρα από ανάλυση χειρότερης περίπτωσης και Ομαδοποίηση

Όπως αναφέραμε στην προηγούμενη ενότητα, ο αλγόριθμος k-means του Lloyd αποδίδει πολύ καλά στην πράξη, παρά τις απογοητευτικές εγγυήσεις του στη χειρότερη περίπτωση. Είναι ενδιαφέρον ότι η κατάσταση όπου ένας αλγόριθμος παράγει πολύ καλύτερες λύσεις από ό,τι περιμέναμε από την ανάλυση της απόδοσής του στη χειρότερη περίπτωση, είναι ένα κοινό φαινόμενο που προκύπτει κατά τη μελέτη πολλών προβλημάτων, εκτός από την ομαδοποίηση. Ως αποτέλεσμα, πολλοί επιστήμονες έχουν προσπαθήσει να αναπτύξουν εναλλακτικές μεθόδους ανάλυσης. Ποια είναι λοιπόν τα χαρακτηριστικά της ανάλυσης της χειρότερης περίπτωσης που την καθιστούν ξεπερασμένη, όταν μελετάμε ορισμένα προβλήματα, και ποιες είναι οι τεχνικές ανάλυσης που μπορούμε να χρησιμοποιήσουμε για να αντιμετωπίσουμε τα μειονεκτήματά της;
Στην *ανάλυση χειρότερης περίπτωσης*, ένας αλγόριθμος αξιολογείται με βάση τη χειρότερη απόδοσή του σε όλες τις εισόδους συγκεκριμένου μεγέθους. Πριν σπεύσουμε να καταδικάσουμε την ανάλυση χειρότερης περίπτωσης, αξίζει να σημειωθεί ότι υπάρχουν λόγοι για τους οποίους συνιστά την πιο συνηθισμένη μέθοδο ανάλυσης αλγορίθμων. Η χρησιμότητά της πηγάζει από το γεγονός ότι είναι ένας βολικός τρόπος να μιλήσουμε για την αποτελεσματικότητα ενός αλγορίθμου γιατί αν καταφέρουμε να αποδείξουμε ότι αποδίδει πολύ καλά ακόμα και στις πιο δύσκολες περιπτώσεις, είμαστε σίγουροι ότι θα έχει τις ίδιες ή και καλύτερες επιδόσεις στα πιθανώς πιο εύκολα προβλήματα που προκύπτουν στην πράξη. Εκτός αυτού, μια μεγάλη ποικιλία από κρίσιμα προβλήματα επιδέχονται αλγόριθμους, οι οποίοι συνοδεύονται από πολύ καλές εγγυήσεις χρόνου εκτέλεσης στη χειρότερη περίπτωση, ενώ είναι επίσης σύνηθες οι ”δύσκολες” είσοδοι ορισμένων προβλημάτων να είναι πολλές και εμφανίζονται συχνά στην πράξη.
Από την άλλη πλευρά, υπάρχουν περιπτώσεις όπου η ανάλυση της χειρότερης περίπτωσης αποτυγχάνει να εξηγήσει την εξαιρετική απόδοση ορισμένων αλγορίθμων, όπως ο αλγόριθμος Lloyd's στο πλαίσιο της ομαδοποίησης k-means, που τους αξιολογεί ως άχρηστους λόγω της κακής τους απόδοσης σε μη ρεαλιστικές εισόδους που δεν προκύπτουν ποτέ στην πράξη. Αξίζει να σημειωθεί ότι ο αλγόριθμος του Lloyd χρησιμοποιείται καθημερινά για την επίλυση εκατομμυρίων προβλημάτων clustering, τα οποία δεν θεωρούνται ιδιαίτερα δύσκολα, παρά το γεγονός ότι είναι δύσκολο να υπολογιστεί η βέλτιστη λύση ομαδοποίησης. Ένα άλλο παράδειγμα αυτού του φαινομένου είναι ο απλός αλγόριθμος του Dantzig για την επίλυση γραμμικών προγραμμάτων, όπου παρά τον εκθετικό χρόνο εκτέλεσής του στη χειρότερη περίπτωση, συνήθως υπερτερεί του ελλειψοειδούς

αλγόριθμου (Ellipsoid algorithm) [4], ο οποίος έχει πολυωνυμική πολυπλοκότητα στη χειρότερη περίπτωση [13]. Επιπροσθέτως, ο αλγόριθμος LRU για το πρόβλημα της online σελιδοποίησης εμφανίζει παρόμοια συμπεριφορά, καθώς ο αριθμός των σφαλμάτων σελίδας (page faults) της κρυφής μνήμης στη χειρότερη περίπτωση είναι μια πολύ απαισιόδοξη εκτίμηση της πραγματικής του απόδοσης σε εισόδους, οι οποίες χαρακτηρίζονται από *τοπικότητα αναφοράς* (locality of reference).
Υπάρχει πληθώρα μεθόδων αλγοριθμικής ανάλυσης εκτείνονται πέρα από την ανάλυση της χειρότερης περίπτωσης, όπως η *smoothed analysis* [13] [62] [16] [26] ή *ανάλυση μέσης περίπτωσης* (average-case analysis) [62] [7] [3] [1] [2] [10] Μάλιστα, έχει αποδειχθεί ότι τόσο ο αλγόριθμος του Lloyd [16][26] όσο και ο αλγόριθμος του Dantzig [13] του Lloyd έχουν πολυωνυμική smoothed complexity, ενώ εξηγείται και η απόδοση του LRU σε κατάλληλα παραμετροποιημένες εισόδους που εμφανίζουν τοπικότητα αναφοράς [14]. Ωστόσο, σε αυτήν τη διπλωματική, θα επικεντρωθούμε σε μια διαφορετική τεχνική ανάλυσης, όπου η απόδοση των αλγορίθμων αποτιμάται σε ένα υποσύνολο του χώρου των στιγμιοτύπων, το οποίο αντιστοιχεί στα πρακτικά στιγμιότυπα εισόδου, όπως τα ”ευσταθή” στιγμιότυπα.
Για να προσφέρουμε κάποια διαίσθηση πίσω από αυτή τη μέθοδο ανάλυσης αλγορίθμων, παρέχουμε το παράδειγμα των προβλημάτων ομαδοποίησης, με τα οποία ασχολείται και η εργασία αυτή. Παρατηρούμε ότι η ομαδοποίηση ενός συνόλου δεδομένων στοχεύει στην αποκάλυψη μιας δομής που υποθέτουμε σιωπηρά ότι υπάρχει στα δεδομένα. Πιο συγκεκριμένα, κάνουμε την υπόθεση ότι αυτή η δομή μπορεί να ανακτηθεί διαμερίζοντας το σύνολο δεδομένων σε συνεκτικές ομάδες και ότι υπάρχει μια τέτοια ”καλή” διαμέριση. Αντιστρόφως, εάν δεν υπάρχει μια καλή διαμέριση, θα μπορούσε κανείς να υποστηρίξει ότι η ομαδοποίηση δεν είναι το σωστό εργαλείο για την εξαγωγή των επιθυμητών πληροφοριών από τα δεδομένα. Επομένως, είναι λογικό να εστιάσουμε στον σχεδιασμό αποδοτικών αλγορίθμων μόνο για σύνολα δεδομένων τα οποία αποδέχονται μια τέτοια ομαδοποίηση. Όπως θα δούμε στο Κεφάλαιο 3, ο περιορισμός του χώρου των εισόδων σε ”ομαδοποιήσιμα” στιγμιότυπα καθιστά πολλά προβλήματα ομαδοποίησης ευεπίλυτα και διευκολύνει τη δημιουργία χρήσιμων αλγορίθμων που έχουν πολύ καλή απόδοση στην πράξη, αλλά θα μπορούσαν να παραβλεφθούν λόγω των κακών επιδόσεών τους σε τεχνητές, ”ασταθής” εισόδους. Κατά μία έννοια, η *Ομαδοποίηση είναι δύσκολη μόνο όταν δεν έχει σημασία* (Clustering is difficult only when it doesn't matter) [30]. Ο κύριος στόχος αυτής της διπλωματικής είναι να μελετήσει εάν αυτό ισχύει και για την περίπτωση της *εξηγήσιμης ομαδοποίησης* (explainable clustering), την οποία εισάγουμε τυπικά στο Κεφάλαιο 4.

0.2.1 Μερικές δημοφιλείς έννοιες ”ευστάθειας” στιγμιοτύπων ομαδοποίησης

Για να βεβαιωθούμε ότι η ομαδοποίηση είναι δύσκολη μόνο σε εισόδους που δεν εμφανίζονται στην πράξη, θα πρέπει να εντοπίσουμε ορισμένες ιδιότητες που έχουν τα περισσότερα πρακτικά στιγμιότυπα ομαδοποίησης και να αποδείξουμε ότι το πρόβλημα της k-ομαδοποίησης είναι ευκολότερο, εάν περιορίσουμε τον χώρο εισόδων μόνο σε εκείνες που ικανοποιούν αυτές τις ιδιότητες. Με άλλα λόγια, θέλουμε να βρούμε κάποιες παραδοχές ”ευστάθειας” στιγμιοτύπων ομαδοποίησης που να πληρούν τις ακόλουθες απαιτήσεις [37]:

1. Η ομαδοποίηση είναι ευεπίλυτη, εάν αποδεχτούμε αυτές τις παραδοχές, δηλαδή υπάρχουν αλγόριθμοι πολυωνυμικού χρόνου που λαμβάνουν καλές προσεγγίσεις της βέλτιστης ομαδοποίησης, εάν η είσοδος είναι ευσταθής.

2. Οι παραδοχές δεν είναι πολύ αυστηρές, επομένως τα περισσότερα από τα στιγμιότυπα των προβλημάτων ομαδοποίησης που προκύπτουν στην πράξη τις ικανοποιούν.

Αυτές είναι, φυσικά, οι ελάχιστες απαιτήσεις που πρέπει να πληρούνται από τις παραδοχές ευστάθειας. Σε ένα ιδανικό σενάριο, θα θέλαμε οι υποθέσεις να ικανοποιούν και την ακόλουθη ιδιότητα:

3. Υπάρχει ένας αποδοτικός αλγόριθμος (πολυωνυμικού χρόνου) που ελέγχει εάν ένα δεδομένο στιγμιότυπο ομαδοποίησης ικανοποιεί την υπόθεση ευστάθειας ή όχι.

Ας προσπαθήσουμε τώρα να βρούμε τέτοιες παραδοχές ευστάθειας. Μια ιδιότητα που θα μπορούσαν να ικανοποιούν πολλά πρακτικά στιγμιότυπα είναι ότι οποιαδήποτε βέλτιστη ομαδοποίηση «ξεχωρίζει», δηλαδή, οι βέλτιστες συστάδες είναι πολύ καλά διαχωρισμένες, έτσι ώστε κάθε στοιχείο να βρίσκεται πολύ πιο κοντά στο κέντρο του δικού του cluster παρά σε οποιοδήποτε άλλο κέντρο. Αυτή η ιδέα οδηγεί φυσικά στην παραδοχή ***α*-ευστάθειας-κέντρων** (a-center-stability), που διαφορετικά ονομάζεται ***α*-εγγύτητα** (a-proximity), όπου $a \geq 1$ είναι η παράμετρος που ελέγχει τον βαθμό διαχωρισμού της βέλτιστης ομαδοποίησης. Όσο υψηλότερο είναι το a, τόσο πιο καλά διαχωρισμένη είναι η βέλτιστη ομαδοποίηση. Η ευστάθεια κέντρων είναι μια κεντρική έννοια στην ανάλυση των αλγορίθμων ομαδοποίησης πέρα από τη χειρότερη περίπτωση, επειδή, όχι μόνο ικανοποιεί την ιδιότητα 1, όπως θα δούμε στο κεφάλαιο 3, αλλά είναι και από τις πιο "ασθενείς" παραδοχές, αφού άλλες έννοιες ευστάθειας συνεπάγονται την ευστάθεια κέντρων, όπως θα δούμε παρακάτω.

Για να εκτιμήσουμε την ιδέα πίσω από τη δεύτερη παραδοχή ευστάθειας που μελετάμε σε αυτή την εργασία, θα πρέπει πρώτα να κατανοήσουμε τον βοηθητικό ρόλο της αντικειμενικής συνάρτησης σε ένα πρόβλημα ομαδοποίησης. Πιο συγκεκριμένα, όπως έχουμε εξηγήσει στην προηγούμενη ενότητα, ο στόχος της ομαδοποίησης είναι να διαμερίσει το σύνολο δεδομένων σε "συνεκτικές" ομάδες. Επομένως, η αντικειμενική συνάρτηση είναι απλώς ένα μέσο για να ποσοτικοποιήσουμε την επιθυμία μας να ομαδοποιήσουμε τα δεδομένα, έτσι ώστε κάθε συστάδα να περιέχει παρόμοια σημεία και τα σημεία που ανήκουν σε διαφορετικά συμπλέγματα να είναι ανόμοια. Σε ορισμένες περιπτώσεις, η δομή και τα χαρακτηριστικά των δεδομένων υποδεικνύουν ότι μόνο μερικές αντικειμενικές συναρτήσεις είναι κατάλληλες για τη συγκεκριμένη εφαρμογή. Ωστόσο, είναι πολύ συνηθισμένο ότι δεν μας ενδιαφέρει πραγματικά η ελαχιστοποίηση μιας συγκεκριμένης συνάρτησης κόστους. Με αφορμή αυτή την παρατήρηση, θα μπορούσαμε να αναμένουμε ότι για πρακτικές περιπτώσεις ομαδοποίησης, η βέλτιστη λύση δεν εξαρτάται σε μεγάλο βαθμό από τις ιδιαιτερότητες του μέτρου εγγύτητας που χρησιμοποιείται για την ποσοτικοποίηση της ομοιότητας των μοτίβων εισόδου. Αυτή η συλλογιστική πορεία οδήγησε στον ορισμό της **Bilu-Linial ευστάθειας** ή ***α*-ευστάθειας διαταραχών** (a-perturbation stability) ή ***α*-ελαστικότητα διαταραχών** (a-perturbation resilience). Συγκεκριμένα, ένα στιγμιότυπο ομαδοποίησης με ένα συγκεκριμένο μέτρο εγγύτητας δ είναι a-ευσταθές στις διαταραχές εάν μικρές αλλαγές στον ορισμό του μέτρου εγγύτητας, που ονομάζονται a-διαταραχές, δεν αλλάζουν τη *μοναδική* βέλτιστη ομαδοποίηση του στιγμιότυπου. Και πάλι, η παράμετρος a καθορίζει το μέγεθος της διαταραχής. Όσο μεγαλύτερο είναι το a, τόσο περισσότερο μπορούμε να διαταράξουμε το αρχικό στιγμιότυπο χωρίς να αλλάξει η βέλτιστη ομαδοποίηση. Όπως θα δούμε στο Κεφάλαιο 3, η a-ευστάθεια διαταραχών συνεπάγεται την a-εγγύτητα.

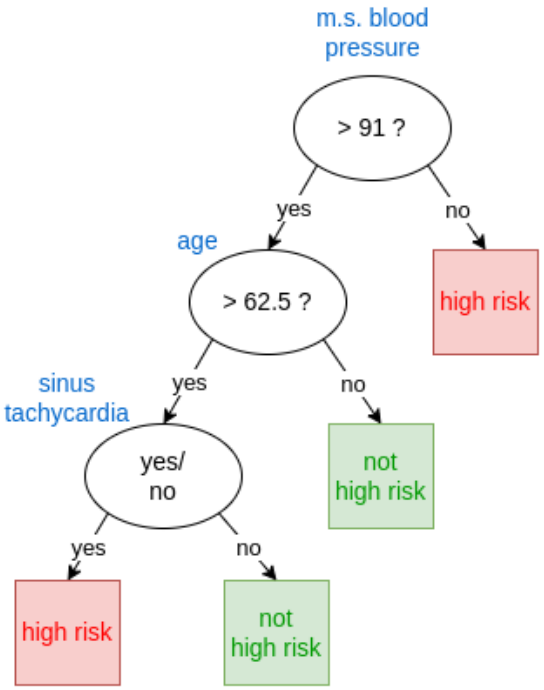
Οι πρώτοι που εισήγαγαν την έννοια της ευστάθειας διαταραχών ήταν οι Bilu και Linial στο [28], οι οποίοι έδωσαν τον ορισμό ενός ευσταθούς στιγμιοτύπου για προβλήματα διακριτής βελτιστοποίησης και σχεδίασαν έναν αποδοτικό αλγόριθμο για το πρόβλημα της Μέγιστης Τομής (Max Cut) σε $O(n)$-ευσταθείς εισόδους (αυτό το αποτέλεσμα αργότερα βελτιώθηκε σε $O(\sqrt{n})$ από τον Bilu κ.ά [29] και μειώθηκε περαιτέρω σε $O(\sqrt{\log n}\log\log n)$ από τον Makarychev κ.ά [34]). Έκτοτε, πολλά προβλήματα έχουν μελετηθεί στο πλαίσιο της ευστάθειας διαταραχών, όπως το πρόβλημα του πλανώδιου πωλητή (Traveling Salesman Problem) [27], το Minimum Multiway Cut [34] [42], το πρόβλημα εύρεσης του Μέγιστου Ανεξάρτητου Συνόλου (Maximum Independent Set) [46] και άλλα.

Η ευστάθεια διαταραχών μελετήθηκε στο πλαίσιο της ομαδοποίησης αρχικά από τους Awasthi, Blum και Sheffet στο [25], όπου σχεδίασαν έναν αλγόριθμο πολυωνυμικό χρόνου για 3-ευσταθή στιγμιότυπα. Αργότερα, οι Balcan και Liang χαλάρωσαν την απαίτηση ευστάθειεας σε $(1+\sqrt{2})$-ευσταθή στιγμιότυπα [39] και τελικά ο Angelidakis κ.ά [42] έδειξε ότι μια παραλλαγή του αλγορίθμου ομαδοποίησης *single linkage* εξάγει τη βέλτιστη ομαδοποίηση σε πολυωνυμικό χρόνο ακόμη και για 2-ευσταθή στιγμιότυπα, όπως θα δούμε στο Κεφάλαιο 3. Αυτό το αποτέλεσμα είναι ουσιαστικά βέλτιστο, αφού οι Ben-David και Reyzin στο [33] απέδειξαν ότι είναι NP-hard να βρεθεί η βέλτιστη λύση στην k-median ομαδοποίηση, εάν τα στιγμιότυπα ικανοποιούν την ιδιότητα της

$(2-\epsilon)$-εγγύτητας, ενώ στην περίπτωση k-centers ο Balcan κ.ά απέδειξε στο [36] ότι δεν υπάρχει αλγόριθμος πολυωνυμικού χρόνου που να μπορεί να λύνει $(2-\epsilon)$-ευσταθή στιγμιότυπα του προβλήματος k-centers, εκτός εάν $NP = RP$, το οποίο θεωρείται πως δεν ισχύει. Αξίζει να αναφέρουμε ότι, για να αποδειχθούν τα παραπάνω αποτελέσματα, αρκεί να υποθέσουμε την ιδιότητα της a-εγγύτητας, δηλαδή δεν χρειάζεται η πιο αυστηρή παραδοχή της a-ευστάθειας διαταραχών. Καταλήγοντας, σημειώνουμε ότι μέχρι στιγμής δεν υπάρχει αποδοτικός αλγόριθμος που να ελέγχει εάν ένα στιγμιότυπο ομαδοποίησης ικανοποιεί τις παραδοχές της a-εγγύτητας ή της a-ευστάθειας διαταραχών, επομένως δεν ικανοποιούν την ιδιότητα 3. Ακόμη πιο ανησυχητικό είναι το γεγονός ότι πολλοί θεωρούν ότι οι τιμές της παραμέτρου ευστάθειας a που απαιτούνται για τα προαναφερθέντα αποτελέσματα είναι πολύ μεγάλες, επομένως υπάρχει επίσης αμφιβολία εάν αυτές οι παραδοχές ικανοποιούν την ιδιότητα 2 [37].

0.3 Ερμηνεύσιμα μοντέλα μηχανικής μάθησης

Για να καταλάβουμε τη σημασία του προβλήματος της εξηγήσιμης ομαδοποίησης που μελετάμε σε αυτή την εργασία, είναι απαραίτητο να κατανοήσουμε την ανάγκη για ερμηνεύσιμα μοντέλα μηχανικής μάθησης. Είναι γνωστό ότι τα τελευταία χρόνια, η μηχανική μάθηση είχε μεγάλη επιτυχία σε μια ποικιλία εφαρμογών, που κυμαίνονται από τη σύσταση προϊόντων έως την αναγνώριση εικόνας και την ανάλυση συναισθήματος [17] [40] [**Duda2000-hc**]. Αυτή η επιτυχία μπορεί να αποδοθεί στην ικανότητα των μοντέλων μηχανικής μάθησης να λαμβάνουν αξιόπιστες και μη τετριμμένες αποφάσεις και προβλέψεις με βάση την εμπειρία που έχουν αποκτήσει από την εκπαίδευσή τους σε τεράστια σύνολα δεδομένων. Ωστόσο, είναι συχνό φαινόμενο η ακριβής συλλογιστική πίσω από τις προβλέψεις του μοντέλου να είναι πολύπλοκη και να μη γίνεται διαισθητικά κατανοητή από τους ανθρώπους. Ως αποτέλεσμα, προκύπτουν φυσικά τα ακόλουθα ερωτήματα: «Πώς μπορούμε να είμαστε σίγουροι ότι το εκπαιδευμένο μοντέλο έχει μάθει αυτό που υποτίθεται ότι πρέπει να μάθει;» και «μπορούμε να σχεδιάσουμε μοντέλα που να παρέχουν εξηγήσεις για τις αποφάσεις τους, οι οποίες μπορούν να ελεγχθούν από ειδικούς και να χρησιμοποιηθούν σε συνδιασμό με την ανθρώπινη εμπειρία με αποτέλεσμα τη βαθύτερη κατανόηση αυτών των δεδομένων;”. Προκειμένου να αναδείξουμε τη σημασία της ερμηνευσιμότητας στη μηχανική μάθηση, παρέχουμε το ακόλουθο παράδειγμα από το κλασικό έργο των Leo Breiman, Jerome Friedman, Charles J Stone και R A Olshen [6]. Θεωρείστε το πρόβλημα κατηγοριοποίησης (classification) όπου θέλουμε να αποφασίσουμε εάν ένας ασθενής που έπαθε καρδιακή προσβολή έχει υψηλό κίνδυνο να πεθάνει μέσα στις επόμενες 30 ημέρες μετά τη νοσηλεία του στο νοσοκομείο. Για την επίλυση αυτού του προβλήματος, εκπαιδεύτηκε ένα *δέντρο απόφασης* (decision tree) σε δεδομένα ασθενών, δίνοντας το δέντρο που εμφανίζεται στο Σχήμα 0.1. Αυτό το δέντρο επέτρεψε στους γιατρούς να προσδιορίσουν τον κίνδυνο θανάτου εξετάζοντας τρεις απλές μετρήσεις: την ελάχιστη συστολική αρτηριακή πίεση του/της ασθενούς, την ηλικία του/της και αν πάσχει από φλεβοκομβική ταχυκαρδία. Αν και ήταν γνωστό ότι αυτοί οι παράγοντες ήταν σημαντικοί στο πλαίσιο των καρδιακών προσβολών, οι γιατροί δεν μπορούσαν να βρουν ούτε τα σωστά όρια για κάθε ερώτηση ούτε την ακριβή αλληλουχία αυτών των ερωτήσεων. Επομένως, αυτό το εύληπτο μοντέλο μπορεί να συνδυαστεί με την τεχνογνωσία του ιατρικού προσωπικού για να γίνουν χρήσιμες προβλέψεις και να σωθούν ζωές. Τώρα, φανταστείτε ότι, αντί αυτού του δέντρου αποφάσεων, χρησιμοποιούνταν ένα σύνθετο νευρωνικό δίκτυο για αυτό το πρόβλημα και ότι, λόγω ενός προγραμματιστικού σφάλματος ή μιας κακής επιλογής των επιπέδων (layers) του δικτύου, το μοντέλο κατέληγε να μάθει ότι όσο υψηλότερη είναι η αρτηριακή πίεση , τόσο χαμηλότερος είναι ο κίνδυνος θανάτου. Φυσικά, αυτό έρχεται σε αντίθεση με τα επιστημονικά δεδομένα, αλλά λόγω της πολυπλοκότητας του μοντέλου, θα ήταν αδύνατο για τους γιατρούς να ανακαλύψουν το λάθος. Ένα παρόμοιο πρόβλημα συζητείται στο [52], όπου οι συγγραφείς αναφέρουν μια μελέτη [38], η οποία είχε ως στόχο να σχεδιάσει ένα μοντέλο που προοριζόταν να χρησιμοποιηθεί σε νοσοκομείο για να δώσει προτεραιότητα στη φροντίδα ασθενών που πάσχουν από πνευμονία, αλλά τελικά έμαθαν ότι το άσθμα μειώνει τον κίνδυνο θανάτου από πνευμονία, ενώ στην πραγματικότητα ισχύει το αντίθετο. Μέχρι τώρα είναι



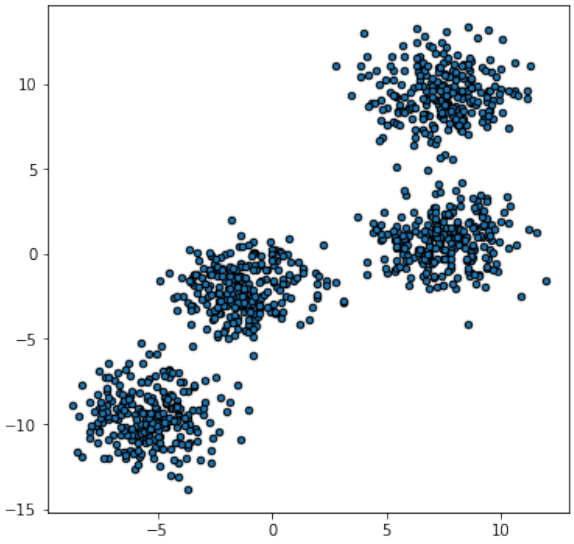
Σχήμα 0.1: Δέντρο απόφασης

σαφές ότι η ερμηνευσιμότητα είναι μια κρίσιμη απαίτηση σε πολλές εφαρμογές. Στο [52] οι συγγραφείς προτείνουν τον ακόλουθο ορισμό:
"Ερμηνεύσιμη μηχανική μάθηση είναι η χρήση μοντέλων μηχανικής μάθησης για την εξαγωγή σχετικών *πληροφοριών που αφορούν τις σχέσεις που περιέχονται στα δεδομένα. Εδώ, θεωρούμε τη γνώση σχετική εάν παρέχει πληροφορίες σε ένα συγκεκριμένο κοινό για ένα επιλεγμένο πρόβλημα σε κάποιο συγκεκριμένο πεδίο ενδιαφέροντος. Αυτές οι πληροφορίες χρησιμοποιούνται συχνά για την καθοδήγηση της επικοινωνίας, των ενεργειών, και της ανακάλυψης."*
Σημειώστε ότι η συγκεκριμένη μορφή με την οποία παρουσιάζονται οι σχετικές πληροφορίες στο κοινό εξαρτάται από τα χαρακτηριστικά αυτού του κοινού (πχ. του ιατρικού προσωπικού στο προηγούμενο παράδειγμα) και τη φύση της εφαρμογής.
Υπάρχουν δύο κύριες κατηγορίες μεθόδων ερμηνείας: ερμηνευσιμότητα *σε επίπεδο μοντέλου* και *εκ των υστέρων* ερμηνευτικότητα [52]. Η πρώτη εστιάζει στο σχεδιασμό μοντέλων μηχανικής μάθησης με περιορισμένη μορφή, έτσι ώστε να παρέχουν εύκολα χρήσιμες πληροφορίες για τις άγνωστες σχέσεις που θέλουμε να ανακαλύψουμε. Με άλλα λόγια, αυτές οι μέθοδοι περιορίζουν το χώρο των πιθανών μοντέλων σε εκείνα που ικανοποιούν ένα πλήθος επιθυμητών ιδιοτήτων, οι οποίες μπορούν να χρησιμοποιηθούν για να εξηγήσουν τις αποφάσεις τους. Αυτός ο περιορισμός θα πρέπει να εκτελείται με προσοχή γιατί μπορεί να οδηγήσει σε χαμηλότερη ακρίβεια πρόβλεψης. Επομένως, οι μέθοδοι που βασίζονται σε μοντέλα είναι προτιμότερες όταν η υποκείμενη σχέση είναι σχετικά απλή. Από την άλλη πλευρά, η εκ των υστέρων ερμηνευτικότητα στοχεύει στην εξαγωγή πληροφοριών σχετικά με τις μαθημένες σχέσεις ενός ήδη εκπαιδευμένου (πιθανώς μη-ερμηνεύσιμου) μοντέλου και χρησιμοποιείται κυρίως όταν η υποκείμενη σχέση είναι περίπλοκη.
Υπάρχουν πολλά πλεονεκτήματα και μειονεκτήματα και στους δύο αυτούς τύπους μεθόδων σχεδίασης ερμηνεύσιμων μοντέλων. Αναμφισβήτητα, το μεγαλύτερο μέρος της δουλειάς σε επεξηγήσιμα μοντέλα αφορά την εκ των υστέρων ερμηνευσιμότητα [50] [51] [55] [65] [45] [48] [58]. Ωστόσο, στο [53] η συγγραφέας ισχυρίζεται ότι τα πλεονεκτήματα του σχεδιασμού εγγενώς ερμηνεύσιμων μοντέλων υπερτερούν των μειονεκτημάτων τους και υποστηρίζει ότι τα "εξηγήσιμα μαύρα κουτιά" πρέπει να αποφεύγονται σε αποφάσεις υψηλού στοιχήματος. Η εξηγήσιμη ομαδοποίηση που προτάθηκε από τους Dasgupta κ.ά στο [57] είναι, εξ όσων γνωρίζω, η πρώτη μέθοδος ερμηνευσι-

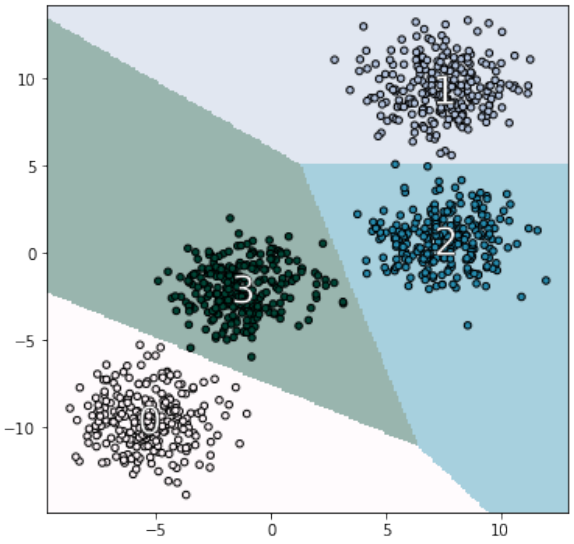
μότητας σε επίπεδο μοντέλου στο πλαίσιο της ομαδοποίησης, που συνοδεύεται από θεωρητικές εγγυήσεις για την επίδοση των εξηγήσιμων αλγορίθμων συγκριτικά με την βέλτιστη ομαδοποίηση χωρίς περιορισμούς.

0.3.1 Εξηγήσιμη ομαδοποίηση

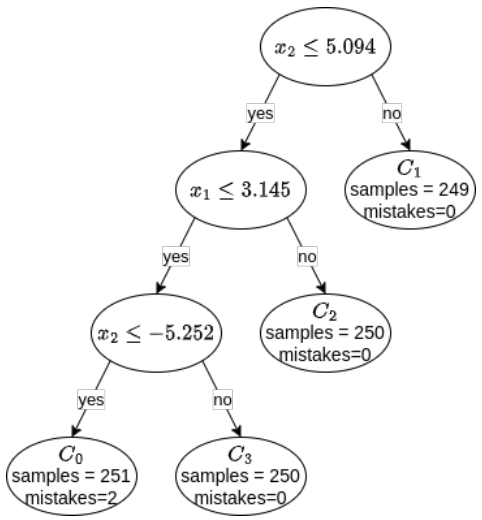
Όπως είδαμε προηγουμένως, η ομαδοποίηση είναι ένα σημαντικό πρόβλημα, με πολλές εφαρμογές και έχει μελετηθεί διεξοδικά από επιστήμονες υπολογιστών, οι οποίοι κατάφεραν να σχεδιάσουν αποτελεσματικούς αλγόριθμους με πολύ καλούς λόγους προσέγγισης. Ωστόσο, οι ομαδοποιήσεις που παράγονται από αυτούς τους αλγόριθμους μπορεί να είναι δύσκολο να ερμηνευτούν, επειδή είναι δύσκολο να δικαιολογηθεί η συμπερίληψη ενός δεδομένου σε μία συστάδα (cluster), καθώς τα επαγόμενα όρια απόφασης εξαρτώνται πιθανώς από όλα τα χαρακτηριστικά των δεδομένων (Σχήμα **??**). Αυτό είναι ιδιαίτερα προβληματικό όταν έχουμε να κάνουμε με δεδομένα υψηλών διαστάσεων, καθώς είναι πρακτικά αδύνατο να εξηγήσουμε γιατί κάθε σημείο ανήκει στο cluster του.
Γι' αυτό οι Dasgupta κ.ά στο [57] προσπάθησαν να δημιουργήσουν έναν αποδοτικό αλγόριθμο ομαδοποίησης που στοχεύει στην εύρεση μιας λύσης που ελαχιστοποιεί κάποια δημοφιλή συνάρτηση κόστους ομαδοποίησης (για παράδειγμα την k-median ή την k-means), ενώ ταυτόχρονα παρέχει μια *συνοπτική εξήγηση* αυτής της ομαδοποίησης, έτσι ώστε η συμπερίληψη οποιουδήποτε σημείου στη συστάδα του να είναι εύκολα επαληθεύσιμη και διαισθητικά κατανοητή. Για το σκοπό αυτό, εισήγαγαν την *ομαδοποίηση μέσω δέντρων κατωφλίου*, η οποία είναι μια μέθοδος ερμηνευσιμότητας σε επίπεδο μοντέλου, η οποία παρέχει έναν συνοπτικό χαρακτηρισμό για κάθε συστάδα με τη μορφή δέντρου, καθιστώντας εύκολο για κάποιον να ελέγξει γιατί ένα συγκεκριμένο σημείο ανήκει σε κάποιο cluster. Όπως μπορούμε να δούμε στο σχήμα 0.2γ΄, κάθε εσωτερικός κόμβος του δέντρου περιέχει ένα ζεύγος χαρακτηριστικού-κατωφλίου (i, θ) που παραπέμπει στη συνθήκη $(x_i \geq \theta)$, ενώ τα φύλλα του δέντρου επάγουν την ομαδοποίηση που επιστρέφει ο αλγόριθμος ως εξής: κάθε φύλλο συνιστά ένα cluster και περιέχει τα σημεία που συμφωνούν στις συνθήκες του μονοπατιού από τη ρίζα προς αυτό το φύλλο, όπως φαίνεται στο Σχήμα 0.2δ΄.
Όμως γιατί να χρησιμοποιούμε δέντρα απόφασης για να ερμηνεύσουμε τις αποφάσεις της ομαδοποίησης; Τα μικρά δέντρα αποφάσεων θεωρούνται ευρέως τυπικό παράδειγμα ενός εξηγήσιμου μοντέλου [56] [52], καθώς η απλή ιεραρχική τους δομή τα καθιστά *"προσομοιώσιμα"* (simulatable), δηλαδή ένας άνθρωπος μπορεί εσωτερικά να προσομοιώσει τη διαδικασία λήψης αποφάσεων. Όπως είναι σαφές από το σχήμα 0.2, η συμπερίληψη ενός σημείου δεδομένων x σε μία συγκεκριμένη συστάδα εξηγείται εύκολα με τον υπολογισμό των ζευγών χαρακτηριστικών-κατωφλίων από τη ρίζα του δέντρου έως το φύλλο στο οποίο αντιστοιχεί στη συστάδα αυτή. Επιπλέον, το γεγονός ότι κάθε συστάδα ορίζεται χρησιμοποιώντας το πολύ $k-1$ χαρακτηριστικά, ανεξάρτητα από τον αριθμό των διαστάσεων d οδηγεί σε σύντομες επεξηγήσεις της ομαδοποίησης που είναι χρήσιμη ιδιαίτερα όταν $k << d$. Σημειώστε ότι η ιδέα της χρήσης μη εποπτευόμενων δέντρων απόφασης για ομαδοποίηση δεν είναι νέα [47] [31] [20] [44] [15], όμως η δουλειά του Dasgupta είναι η πρώτη που παρέχει φράγματα του λόγου προσέγγισης των εξηγήσιμων αλγορίθμων.
Συνεπώς ναι, αυτό το μοντέλο φαίνεται να είναι πράγματι ερμηνεύσιμο. Ποιο είναι όμως το τίμημα που πρέπει να πληρώσουμε για να υπολογίσουμε εξηγήσιμες λύσεις; Στην περίπτωση του Σχήματος 0.2 η εξηγήσιμη ομαδοποίηση είναι μια καλή προσέγγιση της λύσης του αλγορίθμου Lloyd, η οποία δεν μπορεί να εξηγηθεί. Στην ιδανική περίπτωση, θα θέλαμε οι εξηγήσιμες λύσεις ομαδοποίησης να είναι σχεδόν εξίσου καλές με τις λύσεις ομαδοποίησης που μπορούν να βρουν οι κλασικοί (μη εξηγήσιμοι) αλγόριθμους ομαδοποίησης. Έτσι, ο στόχος της εξηγήσιμης ομαδοποίησης είναι να σχεδιάσει αλγορίθμους που επιστρέφουν λύσεις που ικανοποιούν τον "περιορισμό ερμηνευσιμότητας", ενώ ταυτόχρονα επιτυγχάνουν μια καλή προσέγγιση της βέλτιστη ομαδοποίησης *χωρίς περιορισμούς*. Ο ελάχιστος από τους λόγους προσέγγισης για όλους τους εξηγήσιμους αλγόριθμους ομαδοποίησης ονομάζεται *Τίμημα της Εξηγησιμότητας* (Price of Explainability ή PoE). Οι πίνακες 0.1 και 0.2 συνοψίζουν τις πιο σημαντικές εργασίες σχετικά με την Επεξηγήσιμη ομαδοποίηση μαζί με τα άνω και κάτω φράγματα του Τιμήματος Επεξηγησιμότητας που αποδεικνύουν. Σημειώστε ότι ο



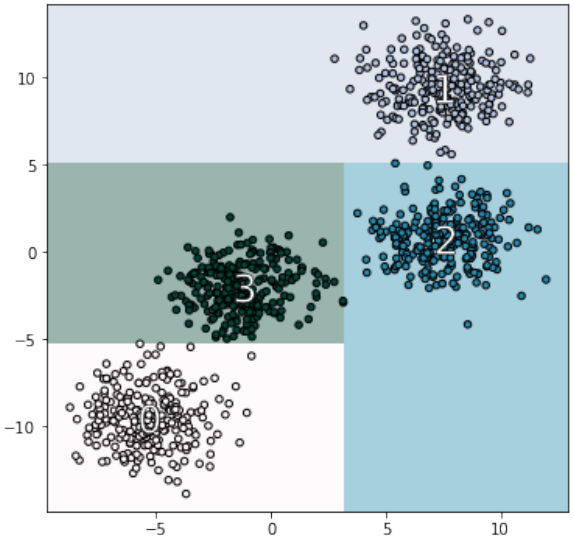
(α) Στιγμιότυπο Ομαδοποίησης



(β) Η ομαδοποίηση που επιστρέφει ο k-means++



(γ) Δέντρο Απόφασης (Κατωφλίου)



(δ) Εξηγήσιμη Ομαδοποίηση που επάγεται από το δέντρο

Σχήμα 0.2: Σύγκριση Εξηγήσιμης και μη Εξηγήσιμης Ομαδοποίησης

πρώτος αλγόριθμος των Dasgupta κ.ά επιτυγχάνει προσέγγιση $O(k)$ της βέλτιστης ομαδοποίησης (χωρίς περιορισμούς) όταν η συνάρτηση κόστους είναι η k-median, ενώ όλοι οι εξηγήσιμοι αλγόριθμοι πληρώνουν ένα αναπόφευκτο κόστος $\Omega(\log k)$ επί του βέλτιστου κόστους. Θα εμβαθύνουμε σε αυτόν τον αλγόριθμο και θα ορίσουμε επίσημα την Τίμημα της Επεξηγησιμότητας στο Κεφάλαιο 4, καθώς και θα σχολιάσουμε τις ιδέες πίσω από τους σύγχρονους εξηγήσιμους αλγόριθμους ομαδοποίησης.

k-median	k-means	ℓ_p objective	Authors
$O(k)$	$O(k^2)$		Dasgupta κ.ά [57]
$O(d \log k)$	$O(kd \log k)$		Laber and Murtinho [60]
$O(\log^2 k)$	$O(k \log^2 k)$	$O(k^{p-1} \log^2 k)$	Svensson κ.ά [59]
$O(\log k \log\log k)$	$O(k \log k \log\log k)$		Makarychev and Shan [61]
$O(\log k \log\log k)$	$O(k \log k)$		Esfandiari κ.ά [64]
$O(d \log^2 d)$			Esfandiari κ.ά [64]
	$O(k^{1-\frac{2}{d}} polylog\ k)$		Charikar and Hu [63]

Πίνακας 0.1: Άνω φράγματα για το τίμημα της Εξηγησιμότητας

k-median	k-means	ℓ_p objective	Authors
$\Omega(\log k)$	$\Omega(\log k)$		Dasgupta κ.ά [57]
	$\Omega(k)$	$\Omega(k^{p-1})$	Svensson κ.ά [59]
	$\Omega(\frac{k}{\log k})$		Makarychev and Shan [61]
$\Omega(\min(d, \log k))$	$\Omega(k)$		Esfandiari κ.ά [64]
	$\Omega(k^{1-\frac{2}{d}}/polylog\ k)$		Charikar and Hu [63]

Πίνακας 0.2: Κάτω φράγματα για το τίμημα της εξηγησιμότητας

0.4 Συνεισφορά

Ο στόχος αυτής της διπλωματικής είναι να μελετήσει το πρόβλημα της εξηγήσιμης ομαδοποίησης που όρισαν οι Dasgupta κ.ά στο [57] κάτω από παραδοχές "ομαδοποιησιμότητας" των στιγμιοτύπων εισόδου.
Το πρώτο πράγμα που αποδεικνύουμε είναι ότι ο εξηγήσιμος αλγόριθμος ομαδοποίησης IMM, τον οποίο εισάγουμε στο κεφάλαιο 4, επιτυγχάνει σταθερό λόγο προσέγγισης εάν τα στιγμιότυπα εισόδου ικανοποιούν την ιδιότητα της a-εγγύτητας, με $a \geq 2kd^{\frac{1}{p}}$, για οποιοδήποτε ℓ_p objective (συνάρτηση κόστους που είναι γενίκευση των k-means και k-median), υποδηλώνοντας ένα άνω φράγμα $O(1)$ για το Τίμημα της Εξηγησιμότητας. Στη συνέχεια, για κάθε συνάρτηση κόστους ℓ_p με $p \geq 2$, δείχνουμε ότι υπάρχει ένα στιγμιότυπο ομαδοποίησης (αυτό στο [59]) που ικανοποιεί την ιδιότητα της a-εγγύτητας με $a = \Omega\left(kd^{\frac{1}{p}}\right)$ και μας δίνει το κάτω φράγμα: $\Omega(k^{p-1})$ για το Τίμημα της Εξηγησιμότητας.
Επιπλέον, μελετάμε το πρόβλημα της εξηγήσιμης k-median ομαδοποίησης, όταν τα στιγμιότυπα εισόδου είναι a-ευσταθή σε διαταραχές. Απ'όσο γνωρίζω, δεν υπάρχει κανένα κριτήριο που να καθορίζει ένα καλό κάτω φράγμα για την ευστάθεια διαταραχών ενός στιγμιότυπου ομαδοποίησης, δεδομένου ότι ικανοποιεί ορισμένες ιδιότητες. Αποδεικνύουμε ότι εάν ένα στιγμιότυπο ομαδοποίησης ικανοποιεί την παραδοχή της a-εγγύτητας και όλες οι συστάδες στη βέλτιστη ομαδοποίηση έχουν περίπου το ίδιο κόστος, τότε το στιγμιότυπο αυτό είναι $\Omega(\sqrt{a})$-metric-perturbation stable (μια λίγο πιο ασθενής εκδοχή της ευστάθειας διαταραχών). Στην πραγματικότητα, αποδεικνύουμε ένα πιο γενικό αποτέλεσμα, όπου επιτρέπουμε στα κόστη των συστάδων στη βέλτιστη λύση να μην είναι περίπου ίσα. Χρησιμοποιούμε αυτό το γεγονός για να αποδείξουμε ότι υπάρχει ένα στιγμιότυπο k-median ομαδοποίησης (αυτό στο [57]) που είναι $\Omega(\sqrt{d})$-metric-perturbation stable, γεγονός που υπαινίσσεται το κάτω φράγμα: $\Omega(\log k)$ για το Τίμημα της Εξηγησιμότητας, όπου d είναι ο αριθμός των διαστάσεων του συνόλου δεδομένων.
Τα αποτελέσματά μας υποδηλώνουν ότι η a-ευστάθεια εγγύτητας και η a-ευστάθεια διαταραχών δεν είναι κατάλληλες παραδοχές για να μειώσουν το Τίμημα της Εξηγησιμότητας.

0.5 Οργάνωση της Εργασίας

Πρώτα απ ’όλα, στο Κεφάλαιο 2 παρέχουμε το υπόβαθρο που απαιτείται για την κατανόηση της υπόλοιπης. Περιλαμβάνει μια εισαγωγή στα προβλήματα ομαδοποίησης και στην ανισότητα του Hoeffding που θα είναι χρήσιμη για τη δημιουργία μιας σκληρής παρουσίας στο Κεφάλαιο 4.
Στο Κεφάλαιο 3 συζητάμε διάφορες έννοιες των ομαδοποιήσιμων στιγμιοτύπων και επικεντρωνόμαστε σε δύο από αυτές: a-ευστάθεια κέντρων και την a-ευστάθεια διαταραχών. Σχολιάζουμε τις ιδέες πίσω από αυτές τις έννοιες και παρουσιάζουμε τις βασικές ιδιότητές τους και μια επισκόπηση του αλγορίθμου του Αγγελιδάκη [42] που υπολογίζει σε πολυωνυμικό χρόνο τη βέλτιστη ομαδοποίηση οποιουδήποτε 2-metric perturbation stable στιγμιοτύπου.
Στο Κεφάλαιο 4, εισάγουμε το πρόβλημα της εξηγήσιμης ομαδοποίησης και ορίζουμε τυπικά το Τίμημα της Εξηγησιμότητας (PoE), το οποίο αναπαριστά το πρόσθετο κόστος που πληρώνουμε για την εξαγωγή ”ερμηνεύσιμων” ομαδοποιήσεων. Στη συνέχεια, συζητάμε τους κύριους αλγόριθμους και τα αποτελέσματα σε αυτό το πεδίο. Συγκεκριμένα, παρέχουμε μια εκτενή επισκόπηση του αλγόριθμου IMM, ο οποίος είναι ο αλγόριθμος που προτείνεται από τους Dasgupta κ.ά στο [57] και παίζει κεντρικό ρόλο στην εργασία μας. Επιπλέον, παρέχουμε μια λεπτομερή ανάλυση του IMM που είναι ουσιαστικά πανομοιότυπη με την απόδειξη των Dasgupta κ.ά αλλά παρουσιάζεται με τα δικά μου λόγια και εξηγείται σύμφωνα με τη δική μου κατανόηση. Στη συνέχεια, αναφέρουμε τους σύγχρονους αλγόριθμους για την εξηγήσιμη ομαδοποίηση μαζί με την βασική ιδέα πίσω από την ανάλυσή τους. Καταλήγοντας, περιγράφουμε ορισμένα ”δύσκολα” στιγμιότυπα για όλους τους εξηγήσιμους αλγορίθμους ομαδοποίησης που μας παρέχουν μέχρι στιγμής τα καλύτερα κάτω φράγματα του Τιμήματος της Εξηγησιμότητας.
Τέλος, στο Κεφάλαιο 5 μελετάμε την εξηγήσιμη ομαδοποίηση κάτω από παραδοχές ευστάθειας.

Chapter 1

Introduction

The motivation behind the question that we aspire to answer comes from two contemporary fields of Theoretical Computer Science. The first is Beyond the Worst-Case analysis, which aims to provide a more realistic understanding of the performance of algorithms in practical scenarios, contrary to the traditional Worst-Case analysis, which can sometimes be too pessimistic and misleading. The second field is that of Interpretable Machine Learning, whose goal is to design machine learning models that provide intelligible explanations of their decisions, which can easily be understood by humans. The goal of this project is to study the Explainable Clustering method, which is an interpretable machine learning model, from the viewpoint of algorithmic analysis that extends beyond the worst-case analysis, by measuring the performance of explainable clustering algorithms on practical input clustering instances, i.e. instances that are the most likely to arise in practice and satisfy certain "well-clusterability" assumptions.
However, before we talk about anything else, we have to offer some background on the k-clustering problem. First of all, after becoming familiar with the most important results for the k-clustering problem, it is easier to appreciate its difficulty and understand the reason to turn our attention to the analysis beyond the worst-case. Secondly, this discussion will justify the need for explainability in numerous clustering applications.

1.1 Clustering

Clustering or *Cluster Analysis* is an *unsupervised* machine learning technique that aims to organize the input data (patterns) into "sensible" groups, called *clusters*, in order to discover similarities and differences among these data and derive useful conclusions between them. This idea of grouping similar patterns is common among many fields[22], such as life sciences (biology, zoology), medical sciences (psychiatry, pathology), social sciences (sociology, archeology), earth sciences (geography, geology), and engineering. Besides, clustering is a primitive mental activity that humans have, in order to avoid processing every piece of information separately, by categorizing entities that share some key attributes into the same cluster. In that way, they can think of all of the entities that belong in the same cluster, according to these common attributes, without having to store vast amounts of information.
In the majority of the clustering problems that arise in practice, the input data are represented by a set X, which is a subset of $\mathbb{R}^d$, where $d \in \mathbb{N}^*$ is called the *dimension* of the data. Each data point $x \in X$ is a d-dimentional vector that encodes important information about a specific pattern in terms of *features*; for every $i \in [d]$, x_i is the value of the i^{th} *feature* of the pattern. Our goal is to partition X into k nonempty and disjoint sets so as to minimize a specific *cost function* (otherwise called an *objective function*), which is designed so that in low-cost solutions each cluster contains data that are "close" to each other. It is clear that to formally define this goal, we need to determine a *proximity measure*, i.e. a notion of *distance* between the input patterns that specifies what it means for them to be "similar". However, there are many different proximity measures that lead to meaningful partitions, thus their choice depends on the specific clustering application at hand.
Arguably, the most well-studied and commonly used clustering objective function is the k-means objective, probably followed by the k-median objective. If we have partitioned X into k clusters

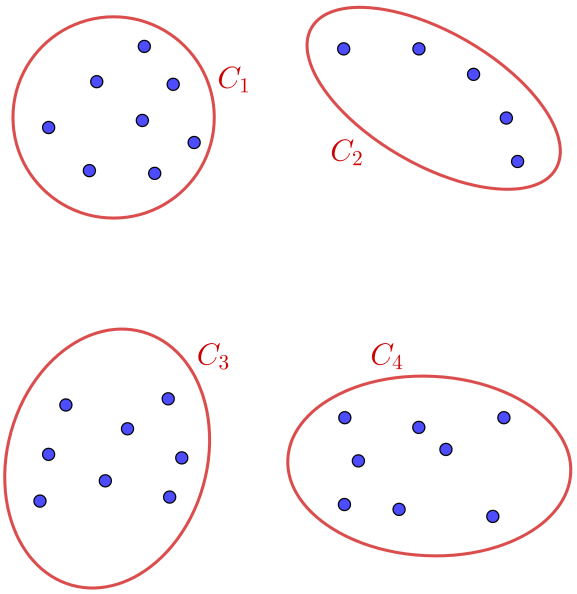


Figure 1.1: 4-clustering

$C_1, C_2, ..., C_k$ then the k-means cost of this clustering is:

$$\mathcal{H}_2(C_1, C_2, ..., C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} \delta^2(x, \mu_i)$$

where function $\delta : \mathbb{R}^d \times \mathbb{R}^d \to [0, +\infty)$ is called a *metric*, as it measures the distance between two input patterns $x, y \in \mathbb{R}^d$, while $\mu_i = \arg\min_{\mu \in \mathbb{R}^d} \sum_{x \in C_i} \delta(x, \mu)^2$ is called the *center* of cluster C_i. Usually, when we study the k-means objective in this context, we take the metric to be $\delta(x, y) = \|x - y\|_2$, i.e. the *Euclidean distance* between x and y.
Similarly, the k-median objective function is:

$$\mathcal{H}_1(C_1, C_2, ..., C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} \delta(x, \mu_i)$$

where $\mu_i = \arg\min_{\mu \in \mathbb{R}^d} \sum_{x \in C_i} \delta(x, \mu)$ and we usually choose $\delta(x, y) = \|x - y\|_1$. These two clustering cost functions belong to a wider class of objective functions called *center-based* objectives, where the clustering cost is computed by assigning a center to each cluster and the goal is to find a k-partition C of X and a set of centers M that minimize the objective function.
Through the past years, due to the popularity of the clustering settings mentioned above, a variety of algorithms have been proposed to solve them. Nevertheless, it has been shown that for most of the clustering problems, including k-means and k-median, it is NP-hard to compute the optimal partition, in the worst case [8] [21] [24] . Although it can be done in $O(n^{kd})$ time [9], since clustering is usually performed for high dimensional data sets with numerous data points, this is an unacceptable running time for practical applications and thus people have resorted to using faster clustering algorithms that do not necessarily return the optimal clustering. One such algorithm for the k-means setting is the *Lloyd's algorithm* [5], otherwise called the k-means algorithm, which has been extensively used in practice, due to its simplicity and satisfactory performance for a lot of practical applications. It is a *local search algorithm*, that is, an iterative algorithm that finds an initial clustering and at each iteration improves the current solution by finding a better one in the "neighborhood" of this solution, until all of the clusterings in its neighborhood have a higher cost than the

current solution (this is guaranteed to happen). Although it has been shown that Lloyd's algorithm may produce arbitrarily bad clustering solutions in the worst case, even for a fixed number of data points n and clusters k [19], it performs remarkably in practice, by finding satisfactory partitions after only a few steps. As we will see in the next section, this phenomenon has motivated scientists to study this algorithm as well as clustering in general, through the prism of beyond the worst-case analysis.
Apart from the above, several approximation algorithms have been developed for k-means clustering. Vassilvitskii et al. created the k-means++ algorithm[19], which is probably the most commonly used k-means algorithm in practice, by making a slight modification to Lloyd's with respect to the initial clustering solution of the local search. This algorithm achieves an $O(\log k)$ approximation of the optimal solution, i.e. solutions that cost at most $O(\log k)$ times more than the optimal clustering. A line of work on reducing the approximation ratio of k-means clustering algorithms led to *constant* approximation algorithms [49] [23] with the current best achieving a ratio of 6.357, proved by Ahmadian, Norouzi-Fard, Svensson and Ward[49].
As far as the k-median setting is concerned, a modification of the k-means++ Algorithm obtains a $O(logk)$ approximation [19] . In addition, Li and Svensson provided a $1 + \sqrt{3} + \epsilon$ approximation algorithm for the k-median objective [32], which was later improved to $2.611 + \epsilon$ by Byrka, Pensyl, Rybicki, Srinivasan and Trinh [43].
In conclusion, some interesting hardness results have been obtained for many k-clustering problems. As we mentioned before, computing the optimal clustering is NP-hard for both k-means and k-median objectives. Awasthi, Charikar, Krishnaswamy, and Sinop, [35] also showed that it is NP-hard to approximate the k-means objective within a factor of $(1 + \epsilon)$ for some positive constant ϵ. Moreover, Bhattacharya, Goyal, and Jaiswal [54] have proved that the Euclidean k-median problem cannot be approximated within a factor of $(1+\epsilon)$, assuming the Unique Games Conjecture, whereas the discrete k-median problem, where we restrict the cluster centers to belong in the input set X, is NP-hard to approximate within $(1 + \frac{2}{e})$ [11].

1.2 Beyond Worst-Case analysis and Clustering

As we mentioned in the previous section, Lloyd's k-means algorithm performs very well in practice, despite its disappointing worst-case guarantees. Interestingly, the situation where an algorithm produces much better solutions than what we expected from analyzing its worst-case performance, is a common phenomenon that can be encountered while studying numerous problems, besides. As a result, many scientists have attempted to develop alternative analysis paradigms to the worst-case analysis. So what are the characteristics of the worst-case analysis that deem it obsolete, when studying certain problems, and what are the techniques that we can use to deal with its disadvantages?
In the *worst-case analysis*, an algorithm is judged upon its worst performance on any input instance of a given size. In other words, when we want to compare two algorithms on inputs of a given size, we measure their performance on their hardest input. Before we rush to condemn it, it is worth noting that there are reasons why the worst-case analysis is by far the most common analysis technique. Its usefulness stems from the fact that it is a convenient way to talk about the efficiency of an algorithm because if we manage to prove that it performs very well even on the hardest instances, we are certain that it will run as fast or even faster on the potentially easier instances that arise in practice. Apart from that, a wide variety of crucial problems admit algorithms, which come with very good worst-case running time guarantees, while it is also common that the hard instances of some problems are numerous and frequently arise in practice.
On the other hand, there are cases where worst-case analysis fails to explain the outstanding performance of certain algorithms, such as Lloyd's algorithm in the context of k-means clustering, deeming them useless because of their poor performance on unrealistic input instances that never occur in practice. Bear in mind, that clustering problems are solved by Lloyd's algorithm millions

of times every day and are not considered especially difficult, in spite of the fact that it is NP-hard to compute the optimal clustering solution. Another example of this phenomenon is Dantzig's simplex algorithm used to solve linear programs and despite its exponential worst-case running time, it usually outperforms the Ellipsoid algorithm [4], which has polynomial worst-case complexity [13]. Moreover, the LRU algorithm for the online paging problem displays a similar behavior, as its worst-case number of cache misses is a very pessimistic estimate of its actual performance on real-life paging instances, which exhibit *locality of reference*.
Several algorithmic analysis methods extend beyond the worst-case analysis, such as *smoothed analysis* [13] [62] [16] [26] or *average case analysis* [62] [7] [3] [1] [2] [10]. In fact, it has been proved both Lloyd's [16][26] and Dantzig's[13] algorithms have polynomial smoothed complexity, while the LRU algorithm achieves a much smaller page-fault rate for data that exhibit locality of reference[14]. However, in this project, we will focus on a different analysis technique, where the performance of algorithms is measured on a subset of the input space, which corresponds to "meaningful" input instances, such as stable instances.
To offer some motivation behind this analysis paradigm, we provide the example of clustering problems, which is also the main topic of this thesis. Note that clustering a data set aims to uncover an interesting structure that we implicitly assume there exists in the data. In particular, we make the assumption that this structure can be retrieved by partitioning the data set into coherent groups and that such a partition exists. Conversely, if such a partition does not exist, one could argue that clustering is just not the right method to extract the desired information from the data. It is, therefore, reasonable to focus on the design of efficient and accurate algorithms for instances that accept such a meaningful clustering and to not care about instances that are unlikely to arise in practice. As we will see in Chapter 3, the restriction of the input space to "well-clusterable" instances renders many clustering problems tractable and facilitates the creation of useful algorithms that perform very well in practice, but could be overlooked due to their poor results on artificially constructed, fragile hard inputs. In a sense, *Clustering is difficult only when it doesn't matter* [30]. The main goal of this project is to study whether this is also true for the *explainable clustering* setting, which we formally introduce in Chapter 4.

1.2.1 Some popular clustering stability assumptions

In order to verify that Clustering is hard only for instances that do not occur in practice, we should identify some properties that most of the practical instances have and study whether the k-clustering problem is easier, if we restrict the input space to only those instances that satisfy these properties. In other words, we want to find some "clustering stability" assumptions that meet the following requirements [37]:

1. Clustering is tractable under these assumptions, i.e. there exist polynomial-time algorithms that obtain good approximations of the optimal clustering solution, if the input is stable.

2. The assumptions are not too strict, so most of the instances that arise in practice satisfy them.

These are, of course, the minimum requirements that should be met by stability assumptions. In an ideal scenario, we would like the assumptions to satisfy the following property as well:

3. There exists an efficient algorithm that checks whether a given clustering instance satisfies the stability assumption or not.

Let us, now, try to find such stability assumptions. One property that many practical instances might satisfy is that any optimal clustering "stands out", that is, the optimal clusters are very well-separated so that each element is much closer to the center of its own cluster than to any other center in the optimal clustering. This idea gives rise to the ***a*-center-stability** assumption, or the ***a*-proximity** property, where $a \geq 1$ is the parameter that controls the amount of separation of the

optimal clustering; the higher a is, the more well-separated any optimal clustering is. Center stability is a central notion in the analysis of clustering algorithms that extends beyond the worst-case analysis because not only does it satisfy property 1, as we will see in Chapter 3, but it is also implied by other definitions of stability and thus its properties can be used to derive useful algorithms.
In order to appreciate the second stability assumption that we study in this project, we should first understand the auxiliary role of the objective function in a clustering problem. To be more specific, as we have explained in the previous section, the goal of clustering is to partition the data set into coherent groups. Therefore, the objective function is just a means to quantify numerically our desire to group the data, so that each cluster contains similar points and points that belong to different clusters are dissimilar. In some cases, the implicit structure that exists in the data indicates that only a few objective functions are suitable for the specific application. However, it is quite common that we don't really care about minimizing a specific cost function; in such cases, the cost function is just a means to an end and not the end itself. Motivated by this observation, we might expect that for practical clustering instances, the optimal solution does not strongly depend on the specifics of the proximity measure that is used to quantify the similarity of input patterns. This thought process has led to the definition of **Bilu-Linial stability** or ***a*-perturbation stability** or ***a*-perturbation resilience**. Specifically, a clustering instance with a certain proximity measure δ is a a-perturbation stable if small changes in the definition of the proximity measure, which are called a-perturbations, do not change the *unique* optimal clustering of the instance. Again, the parameter a determines the size of the perturbation; the higher a is, the more we can perturb the initial clustering instance without changing the optimal clustering. As we will see in Chapter 3, a-perturbation stability implies a-proximity.
The first to introduce perturbation stability were Bilu and Linial in [28], who gave the definition of a stable instance for discrete optimization problems and designed an exact and efficient algorithm for $O(n)$-stable Max Cut instances (this result was later improved to $O(\sqrt{n})$ by Bilu et al. [29] and was further reduced to $O(\sqrt{\log n}\log\log n)$ by Makarychev et al. [34]). Since then, many problems have been studied under perturbation stability, such as TSP [27], Minimum Multiway Cut [34] [42], Maximum Independent Set [46].
Perturbation stability in clustering was initially studied by Awasthi, Blum and Sheffet in [25], where they designed a polynomial time algorithm for 3-perturbation stable instances. Later, Balcan and Liang relaxed the stability requirement to $(1+\sqrt{2})$-stable instances [39] and finally Angelidakis et al. [42] showed that a variant of the *single linkage* clustering algorithm extracts the optimal clustering in polynomial time even for 2-perturbation stable instances, as we will see in Chapter 3. This result is essentially tight, because Ben-David and Reyzin in [33] proved that it is NP-hard to find the optimal solution for $(2-\epsilon)$-center stable k-median instances, while in the k-centers case, as Balcan et al. have proved in [36], there is no polynomial-time algorithm that can solve $(2-\epsilon)$-perturbation stable instances of the k-centers problem unless $NP = RP$, which is widely considered to be false. It is worth mentioning that, in order to prove the results above, it suffices to assume the a-proximity property rather than the stronger a-perturbation stability.
In addition, note that, as of yet, there is no efficient algorithm that checks whether a clustering instance is a-center stable or a-preturbation stable, so these assumptions do not satisfy property 3. Even more concerning is the fact that many people consider the values of the clusterability parameter a needed for the aforementioned efficiency results to be too large, so there is also doubt whether these assumptions satisfy property 2 [37].

1.3 Interpretable Machine Learning Models

To comprehend the importance of the explainable clustering problem that we study in this thesis, it is essential to understand the need for interpretable machine learning models. It is a well-known fact that in recent years, machine learning has had great success in a variety of applications, ranging from product recommendation to image recognition and sentiment analysis [17] [40] [12]. This

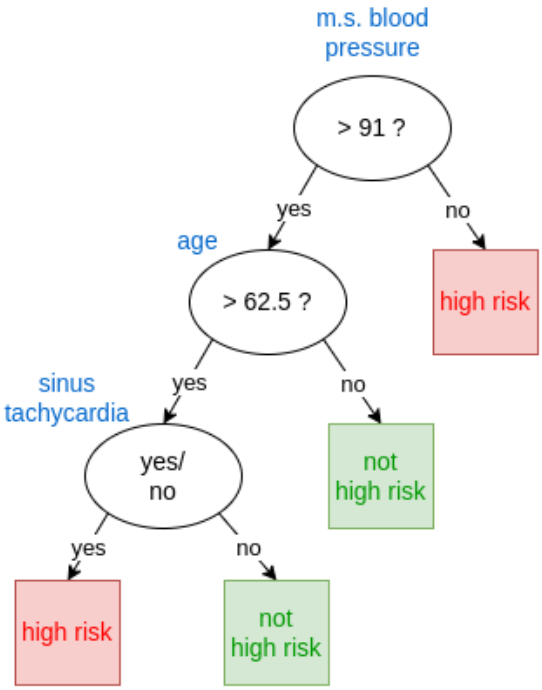


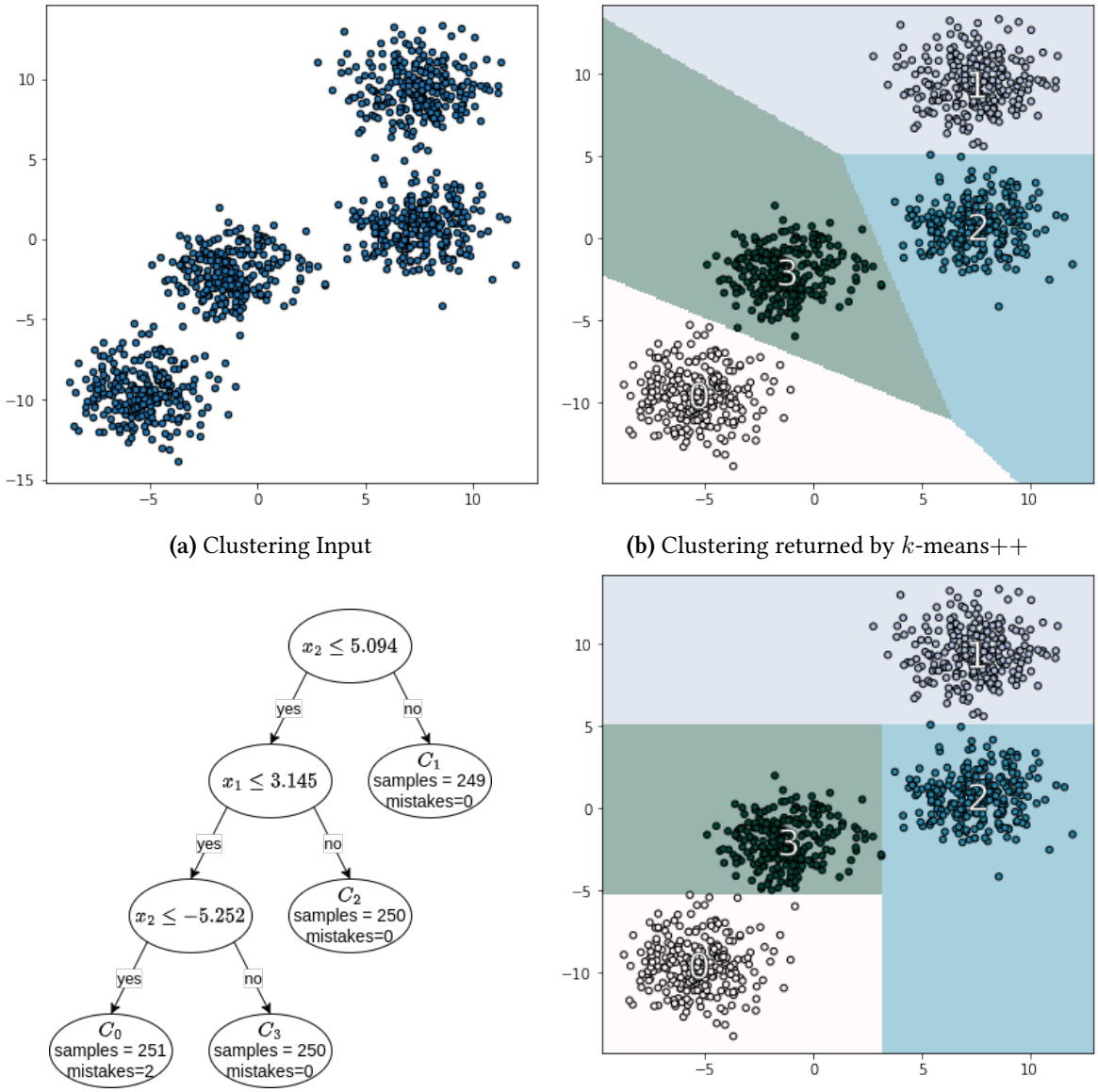
Figure 1.2: Decision Tree

success can be attributed to the ability of machine learning models to make reliable and non-trivial decisions and predictions based on the experience it has gained by training on huge datasets. Nevertheless, it is often the case, that the exact reasoning behind the model's predictions is complex and not intuitively understood by humans. As a result, the following questions arise naturally: "How can we be certain that the trained model is learning what it is supposed to learn" and "can we design models that provide explanations for their decisions, which can be checked by human experts and combined with human experience in order to reach a deeper understanding of these data"?
To illustrate the importance of interpretability in machine learning, we provide the following example from the classical work of Leo Breiman, Jerome Friedman, Charles J Stone, and R A Olshen [6]. Consider the classification problem of deciding whether a heart-attack patient has a high risk of dying within the next 30 days after being hospitalized. To solve this problem, a *decision tree* model was trained on patient data, yielding the tree appearing in Figure 1.2. This tree allows doctors to determine the risk of death by looking at three simple measurements: the patient's minimum systolic blood pressure, his/her age and whether he/she suffers from sinus tachycardia. Although these factors were known to be important in the context of heart-attacks, doctors could not come up either with the correct thresholds for each question or with the exact sequencing of these questions. Therefore, this easy-to-understand model can be combined with the domain expertise of the medical staff to make useful predictions and save lives. Now, imagine that, instead of this decision tree, a complex neural network was used for this problem and that, due to a programming error or a poor choice of layers of the network, the model ended up learning that the higher the blood pressure is, the lower the risk of death. Of course, this contrasts with the scientific data, but because of the complexity of the model, it would be impossible for doctors to discover the mistake. A similar problem is discussed in [52], where the authors mention a study [38], which aimed to design a model intended to be used in a hospital to prioritize patient care for patients suffering from pneumonia, but it eventually learned, that having asthma is associated with a lower risk of dying from pneumonia, while in reality, the opposite is true. By now it is clear, that interpretability is a crucial requirement in many applications. In [52] the authors suggest the following definition:
"Interpretable machine learning is the use of machine learning models for the extraction of relevant

information about domain relationships contained in data. Here, we view knowledge as being relevant if it provides insight for a particular audience into a chosen domain problem. These insights are often used to guide communication, actions, and discovery."
Note that the specific format in which the relevant information is presented to the audience depends on the characteristics of this audience (for instance the medical staff in the previous example) and the nature of the application.
There are two main categories of interpretation methods: *model-based* interpretability and *post hoc interpretability* [52]. The first focuses on the design of ML models with a constrained form, so that they readily provide useful information about the uncovered relationships. In other words, these methods restrict the space of potential models to those that satisfy a number of desirable properties, which can be utilized to explain their decisions. This restriction should be performed with caution because it might lead to lower predictive accuracy. Therefore, model-based methods are preferable when the underlying relationship is relatively simple. On the other hand, post hoc interpretability aims to extract information about the learned relationships of a trained (possibly unintelligible) model and it is mainly used when the underlying relationship is complicated.
There are several advantages and disadvantages to both of these types of interpretation methods. Arguably, most of the work on explainable models concerns post hoc intrepretability [50] [51] [55] [65] [45] [41] [48] [58]. However, in [53] the author claims that the benefits of designing inherently interpretable models outweigh their drawbacks and argues that "explainable black boxes" should be avoided in high-stakes decisions. The explainable clustering method proposed by Dasgupta et al. in [57] is, to my knowledge, the first model-based interpretation method in the context of clustering, that comes with theoretical guarantees on the worst-case performance of the explainable algorithm in comparison with the optimal unconstrained clustering algorithm.

1.3.1 Explainable Clustering

As we have seen previously, clustering is an important problem, with many applications and has been studied thoroughly by computer scientists, who managed to design efficient algorithms with very good approximation ratios. However, the clusterings produced by these algorithms might be hard to interpret, because it is difficult to justify the inclusion of a data point to a cluster, as the induced decision boundaries possibly depend on all of the features of the data (Figure 1.3b). This is especially problematic when we are dealing with high-dimensional data, as it is practically impossible to explain why each point was assigned to its cluster.
That's why Dasgupta et al. in [57] attempted to create an efficient clustering algorithm that aims to output a solution that minimizes some popular clustering objective, while at the same time providing a *concise explanation* of this clustering so that the inclusion of any point to its assigned cluster is easily verifiable and intuitively understood. To this end, they introduced *clustering via threshold trees*, which is a model-based interpretation method, that provides a concise tree-based characterization for each cluster, making it easy for someone to check why a specific point belongs to a specific cluster. As we can see in Figure 1.3c, each internal node of the tree contains a single feature and threshold pair (i, θ) that corresponds to the condition $(x_i \geq \theta)$, while the leaves of the tree induce the output clustering as follows: each leaf is a cluster that contains the points that agree on the conditions in the path from the root to this leaf, as shown in Figure 1.3d.
But why decision trees? Small decision trees are widely considered as a standard example of an explainable model [56] [52], as their simple hierarchical decision-making deems them *simulatable*, i.e. a human can internally simulate and explain their decisions. As it is clear from Figure 1.3, the inclusion of a data point x in a specific cluster is easily explained by computing the feature-threshold pairs from the root of the tree to the leaf that corresponds to the cluster. In addition, the fact that each cluster is defined using at most $k-1$ features, independently from the number of dimensions d leads to short explanations that are especially when $k << d$. Note that the idea of using unsupervised decision trees for clustering is not new [47] [31] [20] [44] [15], but the work of Dasgupta is the first that comes with theoretical bounds on the approximation ratio of the explainable algorithm.



(a) Clustering Input

(b) Clustering returned by k-means++

(c) Decision (Threshold) Tree

(d) Explainable Clustering induced by Tree

Figure 1.3: Explainable vs Non-Explainable Clustering

So yes, this model seems to be interpretable indeed. But what is the price we have to pay in order to compute explainable solutions? In the case of Figure 1.3 the explainable clustering is a good approximation of the solution of Lloyd's algorithm, which is non-explainable. Ideally, we would like our explainable clustering solutions to be almost as good as the *unconstrained* clustering solutions that can be obtained by classical (non-explainable) clustering algorithms. Thus, the goal of explainable clustering is to design algorithms that return solutions that satisfy the "explainability constraint", while at the same time achieving a good approximation of the *optimal unconstrained* clustering. The minimum of the approximation ratios over all explainable clustering algorithms is called the *Price of Explainability (PoE)*. Tables 1.1 and 1.2 summarize the most important papers on Explainable Clustering along with the upper and lower bounds of the PoE that they prove. Note that the first algorithm by Dasgupta et al. achieves $O(k)$ approximation of the optimal unconstrained clustering, for the k-median objective, while all explainable algorithms incur an unavoidable cost of $\Omega(\log k)$ times the optimal cost. We will delve deeper into this algorithm and formally define the Price of Explainability in Chapter 4, as well as discuss the ideas behind the state-of-the-art, near-optimal explainable clustering algorithms.

k-median	k-means	ℓ_p objective	Authors
$O(k)$	$O(k^2)$		Dasgupta et al. [57]
$O(d \log k)$	$O(kd \log k)$		Laber and Murtinho [60]
$O(\log^2 k)$	$O(k \log^2 k)$	$O(k^{p-1} \log^2 k)$	Svensson et al. [59]
$O(\log k \log\log k)$	$O(k \log k \log\log k)$		Makarychev and Shan [61]
$O(\log k \log\log k)$	$O(k \log k)$		Esfandiari et al. [64]
$O(d \log^2 d)$			Esfandiari et al. [64]
	$O(k^{1-\frac{2}{d}} polylog\ k)$		Charikar and Hu [63]

Table 1.1: Upper Bounds for the Price of Explainability

k-median	k-means	ℓ_p objective	Authors
$\Omega(\log k)$	$\Omega(\log k)$		Dasgupta et al. [57]
	$\Omega(k)$	$\Omega(k^{p-1})$	Svensson et al. [59]
	$\Omega(\frac{k}{\log k})$		Makarychev and Shan [61]
$\Omega(\min(d, \log k))$	$\Omega(k)$		Esfandiari et al. [64]
	$\Omega(k^{1-\frac{2}{d}}/polylog\ k)$		Charikar and Hu [63]

Table 1.2: Lower Bounds for the Price of Explainability

1.4 Contribution

The goal of this project is to study the explainable clustering problem defined by Dasgupta et al. in [57] under "well-clusterability" assumptions.
The first thing that we prove is that the IMM explainable clustering algorithm, which we introduce in chapter 4, achieves a constant approximation ratio if the input instance satisfies the a-proximity assumption, with $a \geq 2kd^{\frac{1}{p}}$, for any ℓ_p clustering objective, implying an $O(1)$ upper bound for the price of explainability. Next, for every ℓ_p objective with $p \geq 2$, we show that there exists a clustering instance (the one in [59]) that satisfies the a-proximity property with $a = \Omega\left(kd^{\frac{1}{p}}\right)$ and implies a lower bound of $\Omega(k^{p-1})$ for the PoE.
Afterward, we study the explainable k-median clustering problem under a-perturbation stability. As far as I am concerned, there is no criterion that determines a good lower bound for the perturbation-stability of a clustering instance, given that it satisfies some requirements. We prove that if a clustering instance has the a-proximity property and all of the clusters in the optimal clustering with centers M have roughly the same cost, i.e. for any two clusters C, C' in the optimal clustering it holds that $\frac{1}{\gamma} cost(C', M) \leq cost(C, M) \leq cost(C', M)$ for a *constant* $\gamma \geq 1$, where $cost(C, M)$ is the k-median cost of the cluster C with centers M, then the instance is $\Omega(\sqrt{a})$-metric-perturbation stable. In fact, we prove a more general result, where we allow γ to be non-constant too. We use this fact to prove that there exists a k-median clustering instance (the one in [57]) that is $\Omega(\sqrt{d})$-metric-perturbation stable and implies a lower bound of $\Omega(\log k)$ for PoE, where d is the number of dimensions of the data set.
Our results suggest that a-proximity and a-perturbation stability are not suitable "well-clusterability" assumptions for the explainable clustering problem.

1.5 Organization of the Project

First of all, in Chapter 2 we provide the background needed to understand the rest of the project. It includes an introduction to clustering problems and Hoeffding's Inequality which will be useful in the creation of a hard instance in Chapter 4.

In Chapter 3 we discuss several notions of well-clusterable instances and focus on two of them: a-center stability and a-perturbation stability. We offer the motivation behind these notions and present their basic properties and an overview of the algorithm of Angelidakis [42], which computes the optimal (unconstrained) clustering of any 2-metric perturbation stable input instance in polynomial time.
In Chapter 4, we introduce the explainable clustering problem and define the price of explainability (PoE), which captures the additional cost of creating clusterings that are "interpretable". Afterward, we discuss the main algorithms and results in this field. To be more specific, we provide an extensive overview of the IMM Algorithm, which is the algorithm proposed by Dasgupta et al. in [57] and plays a central role in our project. In addition, we provide a detailed analysis of IMM that is essentially identical to the proof by Dasgupta et al. but is presented in my own words and explained according to my intuition. Next, we mention the state-of-the-art algorithms for explainable clustering along with the main idea behind their analysis. Last but not least, we describe some hard instances for the explainable clustering problems that provide us with the current best lower bounds of the price of explainability (in terms of the number of clusters k).
Finally, in Chapter 5 we study explainable clustering under stability assumptions.

Chapter 2

Preliminaries

2.1 Metric Spaces

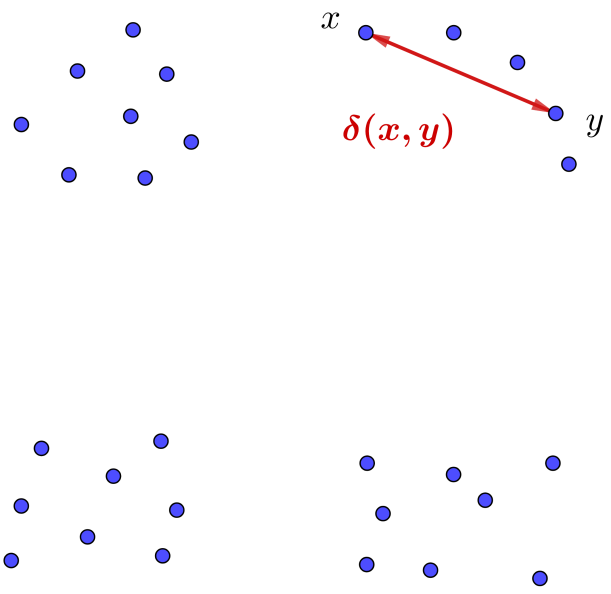


Figure 2.1: Metric Space

As we explained in the introduction, in order to formulate the clustering problem, we need to define a *proximity measure*, i.e. a notion of distance, which quantifies the similarity between any two input patterns and will be used to define the clustering cost function to be minimized. The first thing that comes to mind when we think of distance is probably the Euclidean distance in $\mathbb{R}^d$. Interestingly, this concept of distance seems to be more general and can be best captured by the notion of a metric.

Definition 2.1.1 (Metric Space). *Let X be a set and $\delta : X \times X \to [0, +\infty)$ be a function. Then (X, δ) is a called a metric space and function δ is called a metric, if for any $x, y, z \in X$ the following properties hold:*

1. $\delta(x, y) \geq 0$
2. $\delta(x, x) = 0$
3. $\delta(x, y) > 0$ *for* $x \neq y$

4. $\delta(y,x) = \delta(x,y)$ *(symmetry)*

5. $\delta(x,y) + \delta(y,z) \geq \delta(x,z)$ *(triangle inequality)*

A natural way to define a metric is in a vector space equipped with a function that specifies the distance of a point from the "origin". These vector spaces are called *normed* spaces and this function is called a *norm*. More formally:

Definition 2.1.2 (Normed Space). *A* normed space *is a vector space V over $\mathbb{R}$ (or $\mathbb{C}$) equipped with a mapping $\|\cdot\| : V \to [0, +\infty)$, which is called a* ***norm****, that satisfies the following axioms for any $x, y \in V$ and $a \in \mathbb{R}$ (or $\mathbb{C}$):*

1. $\|x\| = 0 \Rightarrow x = 0$

2. $\|ax\| = |a| \, \|x\|$

3. $\|x + y\| \geq \|x\| + \|y\|$

Remark: If $(V, \|\cdot\|)$ is a normed space, then the function $\delta : V \times V \to \mathbb{R}$ defined by the formula

$$\delta(x,y) = \|x - y\|, \ x, y \in V$$

is a metric.

Definition 2.1.3 (p-norm). *Let $d \in \mathbb{N}^*$ and $p \geq 1$. We define the p-norm or ℓ_p norm to be a function $\|\cdot\|_p : \mathbb{R}^d \to [0, +\infty)$, such that, for every $x \in \mathbb{R}^d$:*

$$\|x\|_p = \left(\sum_{i=1}^{d} |x|^p \right)^{\frac{1}{p}}$$

Then, $\ell_p^d = (\mathbb{R}^d, \|\cdot\|_p)$ is a normed space.

In this project, we will denote the metric defined by the p-norm in any normed space ℓ_p^d as δ_p.

2.2 Clustering

2.2.1 Definitions

We will now present a general definition of the k-clustering problem, which is rarely used in this form. However, since we want to study several clustering objectives and because it is helpful to think of clustering according to this definition in the context of stable clustering instances, we choose to start with it and then introduce the most common clustering objectives that we mentioned in the introduction, such as the k-means and k-median objectives, as a special case.

Definition 2.2.1 (*k*-clustering problem). *An instance of a k-clustering problem, for some positive integer k, is a tuple:*

$$\mathcal{I} = ((Y, \delta), X, \mathcal{H})$$

where (Y, δ) is a metric space, X is a finite set $X \subseteq Y$ and $\mathcal{H}$ is an objective function:

$$\mathcal{H} : \mathcal{P}_X \times \mathcal{D}_X \to [0, +\infty)$$

where $\mathcal{P}_X$ is the set of all partitions of X into k non-empty sets and $\mathcal{D}_X$ is the set of all metrics on Y. Given such an instance, our goal is to partition X into sets $C_1, C_2, ..., C_k$ so as to minimize $\mathcal{H}(C_1, C_2, ..., C_k, \delta)$.
The partition $\{C_1, C_2, ..., C_k\}$ is called a clustering of X and each C_i, $i \in [k]$ is called a cluster.

Note that in case $X = Y$, we might denote $\mathcal{I}$ as: $\mathcal{I} = ((X, \delta), \mathcal{H})$. Moreover, if δ and $\mathcal{H}$ can be inferred from the context, we might just say that X is a clustering instance.
A popular class of objectives is that of *center-based* objective functions, which define a cost for each cluster C_i by assigning a center μ^i to it, and the total cost of the clustering is the sum of the costs of the k clusters. In this project, we are going to focus on the ℓ_p objective functions, which are natural center-based objectives that capture the most commonly used cost functions for clustering problems in practice. To define these objective functions, we will need the following two definitions.

Definition 2.2.2 (ℓ_p cost of a set C with centers M). *Let a metric space (Y, δ) and two finite sets $C, M \subseteq Y$. We define the cost of C with centers M to be:*

$$cost_p(C, M, \delta) = \sum_{x \in C} \min_{\mu \in M} \delta^p(x, \mu)$$

If we want to compute the cost of C from a single center $\mu \in Y$, we may write $cost(C, \mu, \delta)$ instead of $cost(C, \{\mu\}, \delta)$. We will say that $\mu^* = \arg\min_{\mu \in Y} cost(C, \mu)$ is the optimal center of C.

Definition 2.2.3 (ℓ_p cost of a clustering $\mathcal{C}$ with centers M). *Let a metric space (Y, δ), a finite set $X \subseteq Y$ and $k \in \mathbb{N}^*$. We define the ℓ_p cost of a k-clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ of X with centers $M = \{\mu^1, \mu^2, ..., \mu^k\}$ and $p \geq 1$, where μ^i is the center of C_i for $i \in [k]$, as follows:*

$$cost_p(C_1, C_2, ..., C_k, \mu^1, \mu^2, ..., \mu^k, \delta) = \sum_{i=1}^{k} cost_p(C_i, \mu^i, \delta)$$

We will also say that the center of cluster C_i is μ^i and that each element in C_i is assigned to μ^i, for every $i \in [k]$.

Definition 2.2.4. *(clustering objective function)* *Let a metric space (Y, δ) and a finite set $X \subseteq Y$ and $k \in \mathbb{N}^*$. The ℓ_p clustering objective function, is a function $\mathcal{H}_p$ that, given a clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ of X and a metric δ, assigns the following cost to $\mathcal{C}$:*

$$\mathcal{H}_p(C_1, C_2, ..., C_k, \delta) = \min_{\{\mu^1, \mu^2, ..., \mu^k\} \subseteq Y} cost_p(C_1, C_2, ..., C_k, \mu^1, \mu^2, ..., \mu^k, \delta)$$

In other words, in the k-clustering problem with the ℓ_p objective, we try to find a clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ along with centers $M = \{\mu^1, \mu^2, ..., \mu^k\}$, so as to minimize the cost

$$\sum_{i=1}^{k} \sum_{x \in C_i} \delta^p(x, \mu^i)$$

The most commonly used ℓ_p objectives, are the ones for $p = 1$ and $p = 2$; k-clustering with the ℓ_1 objective is called ***k*-median clustering**, while k-clustering with the ℓ_2 objective is known as ***k*-means clustering**. Now, we will make some important remarks about clustering with ℓ_p objective functions.

Remark 1: Because the cost function $\mathcal{H}_p$ is to be minimized, many authors consider it redundant to specify both the clustering $\mathcal{C}$ and its centers M. This is because, in case we specify $\mathcal{C} = \{C_1, C_2, ..., C_k\}$, the set of centers that minimizes the $\mathcal{H}_p$ objective function (i.e. the optimal centers of $\mathcal{C}$) is $M = \{\mu^{*1}, \mu^{*2}, ..., \mu^{*k}\}$, where μ^{*i} is the optimal center of cluster C_i. Similarly, if we fix a set of centers $M = \{\mu^1, \mu^2, ..., \mu^k\}$, the optimal clustering with these centers is the *Voronoi partition of* X, that is, the optimal clustering is $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ where C_i is the set of points that are closer to center μ^i than to any other μ^j, for all $i \in [k]$. In this case, we will say that M *induces* clustering $\mathcal{C}$ and we will write:

$$cost_p(C_1, C_2, ..., C_k, \mu^1, \mu^2, ..., \mu^k, \delta) = cost_p(M, \delta)$$

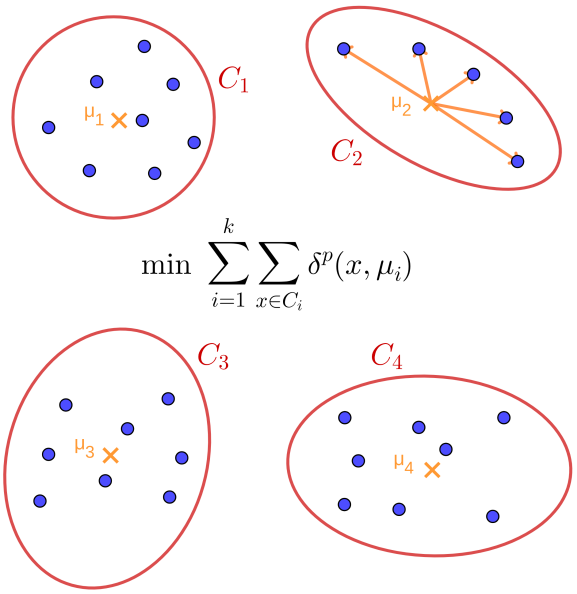


Figure 2.2: Clustering with the ℓ_p objective

Remark 2. Consider the pair $(\mathcal{C}, \mathcal{M})$ that minimizes the cost function $\mathcal{H}_p$. Then, M is an optimal set of centers for $\mathcal{C}$, and conversely, $\mathcal{C}$ is induced by M (otherwise there would exist an even cheaper clustering). Consequently, if $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ and $M = \{\mu^1, \mu^2, ..., \mu^k\}$, then for any $i, j \in [k]$ with $i \neq j$ and a point $x \in C_i$, then $\delta(x, \mu^i) \leq \delta(x, \mu^j)$, that is, *each point in the optimal clustering is closer to its own center than to any other center.* Note that the inequality becomes strict if we know that $\mathcal{C}$ is the unique optimal clustering. Nevertheless, this property does not hold in general, that is, if $\mathcal{C}$ is not the optimal clustering and M^* is the optimal set of centers for $\mathcal{C}$, then $\mathcal{C}$ is not necessarily induced by M^*.

Note, that in case p or δ can be inferred from the context, we may not include them in the notation, for example, we will write:

$$cost(M) \text{ or } cost(C_1, C_2, ..., C_k, \mu^1, \mu^2, ..., \mu^k)$$

2.3 Hoeffding's Inequality

When we analyze a randomized algorithm, we often want to know the probability that it fails to achieve a certain performance or the value of one of its parameters that allows the algorithm to work as desired, with high probability. To answer these questions, it is not sufficient to understand the *expected* behavior of our algorithm, because it might be the case that there is a high probability for the algorithm to deviate significantly from its mean performance. Therefore, a very common problem that arises in the analysis of such algorithms is to bound the probability that some random variable X deviates significantly from its mean $\mathbb{E}[X]$.
There are many ways deal with this problem, such as the *Markov's Inequality* or *Chebyshev's Inequality*, which offer us the desired bounds in terms of the first and second-order moments of the random variable X, i.e. its mean and its variance. However, in the special case when X is the sum of *independent, bounded* random variables, we can use the information contained in all of the (infinite) moments of the random variable X and thus obtain much sharper (exponentially decreasing) bounds than if we applied first or second order methods. The standard tool that we use to obtain

these sharp bounds is the *Hoeffding's Inequality*, which we present below.

Theorem 2.3.1 (Hoeffding's Inequality). *Let $X_1, X_2, ..., X_n$ be independent random variables that where for each $i \in [n]$ there exist $a_i, b_i \in R$ such that: $Pr\left(X_i \in [a_i, b_i]\right) = 1$. If $X = \sum_{i=1}^{n} X_i$, then the following two inequalities hold for any $\epsilon \geq 0$:*

1. $$Pr\left(X - \mathbb{E}[X] \geq \epsilon\right) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$$

2. $$Pr\left(|X - \mathbb{E}[X]| \geq \epsilon\right) \leq 2e^{\frac{-2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$$

Note that this inequality has numerous applications, apart from the analysis of randomized algorithms. In fact, in this project, we will make use of this inequality to prove the existence of a clustering instance via the probabilistic method.

Chapter 3

Well-Clusterable Instances

In this Chapter, we will introduce some of the most common notions of "stable" clustering instances along with some of their most important properties and explain how these can be exploited in order to develop clustering algorithms that produce almost perfect clustering solutions, in polynomial time. We focus on a-center stability, otherwise called a-proximity, and a-perturbation stability; these are the clusterability assumptions under which we study explainable clustering in Chapter 5. Last but not least, we present an efficient algorithm that returns the optimal clustering for many clustering objectives under the perturbation stability assumption, including k-means and k-median.

3.1 Center stability and its basic properties

3.1.1 Motivation and Definition

Center stability or *center proximity* is probably the most natural well-clusterability assumption. Let $\mathcal{C}$ be a unique optimal clustering of a clustering instance with optimal centers M. For any data point x it is true that x is closer to the center to which it was assigned than to any other center. Therefore, it is reasonable to assume that well-clusterable instances have optimal solutions that "stand out", which means that, each point is *much closer* to its assigned center than to any other center in any optimal clusteirng.

Definition 3.1.1 (a-center stability). *Let $\mathcal{I} = ((Y, \delta), X, \mathcal{H})$ be a clustering instance. We say that $\mathcal{I}$ is a-center stable, with $a \geq 1$, if for any optimal clustering $\mathcal{C}$ of this instance, if for any $C, C' \in \mathcal{C}$ with $C \neq C'$, for any data point $x \in C$:*

$$\delta(x, c') > a\delta(x, c)$$

where c, c' are the (optimal) centers of clusters C and C' respectively.

We will also refer to a-center stability as a-proximity.

3.1.2 Basic properties of center stability

We will now present some of the most important properties of a-center-stable instances, that are frequently used in the design of clustering algorithms achieve outstanding performance under this assumption.

Lemma 3.1.1 (Properties of a-center stability). *Let $((Y, \delta), X, \mathcal{H})$ be an a-center stable k-clustering instance. Let C_1 and C_2 be two distinct clusters in the optimal clustering, with centers c_1, c_2 respectively, radii R_1, R_2 respectively and $p, p' \in C_1$ and $q \in C_2$. Then:*

1. $\delta(p, q) > (a-1)\delta(p, c_1)$
2. $\delta(c_1, c_2) > (a-1)\delta(p, c_1)$ *(also implying $\delta(c_1, c_2) > (a-1)\max(R_1, R_2)$)*
3. $\delta(p, q) > \frac{a-1}{a+1}\delta(c_1, c_2)$ *for $a > 1$*

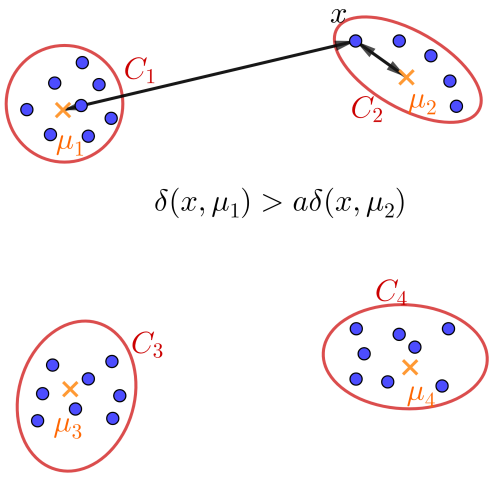


Figure 3.1: a-center-stable instance

4. $\delta(p, q) > \frac{(a-1)^2}{2a}\delta(p, p')$, *for* $a > 1$

Proof of Lemma 3.1.1.

1. For the sake of contradiction, let's assume, that $\delta(p, q) \leq (a-1)\delta(p, c_1)$. Then, by Definition 3.1.1 and the triangle inequality, we obtain:

$$a\delta(q, c_2) < \delta(q, c_1) \leq \delta(p, c_1) + \delta(p, q) \leq a\delta(p, c_1) \Rightarrow \delta(q, c_2) < \delta(p, c_1)$$

$$a\delta(p, c_1) < \delta(p, c_2) \leq \delta(q, c_2) + \delta(p, q) \leq \delta(q, c_2) + (a-1)\delta(p, c_1) \Rightarrow$$

$$\Rightarrow \delta(p, c_1) < \delta(q, c_2)$$

   which is a contradiction.

2. By triangle inequality:

$$\delta(c_1, c_2) + \delta(p, c_1) \geq \delta(p, c_2) > a\delta(p, c_1) \Rightarrow$$

$$\delta(c_1, c_2) > (a-1)\delta(p, c_1)$$

   Since we showed the above for arbitrary points $p \in C_1$, the inequality also holds for $x \in C_1$, such that $\delta(x, c_1) = R_1$. Thus:

$$\delta(c_1, c_2) > (a-1)R_1$$

   Working in the exact same way, we obtain:

$$\delta(c_1, c_2) > (a-1)R_2$$

   so:

$$\delta(c_1, c_2) > (a-1)\max(R_1, R_2)$$

3. By triangle inequality:

$$\delta(c_1, c_2) \leq \delta(c_1, p) + \delta(p, q) + \delta(q, c_2) < \left(\frac{2}{a-1} + 1\right)\delta(p, q) = \frac{a+1}{a-1}\delta(p, q)$$

where, for the strict inequality, we have used item 1.

4. Again, by triangle inequality:

$$\begin{aligned}(a-1)\delta(p, p') &\leq (a-1)\delta(p, c_1) + (a-1)\delta(p', c_1) < \delta(p, q) + \delta(c_1, c_2) \\ &< \delta(p, q) + \frac{a+1}{a-1}\delta(p, q) = \frac{2a}{a-1}\delta(p, q)\end{aligned}$$

where for the second inequality we have used items 1. and 2. and for the third, we made use of item 3.

□

An interesting consequence of item 1. of Lemma 3.1.1 is that for $a \geq 2$, each point is closer to its own center in the optimal clustering than to any other point that belongs to a different cluster. Moreover, by item 4. of Lemma 3.1.1, we notice that, for $a \geq 2+\sqrt{3}$, every point is closer to any other point in its own cluster than to any other point that belongs in a different cluster (in the optimal clustering).

3.2 Perturbation stability and its basic property

3.2.1 Motivation and Definitions

In most k-clustering problems that arise in practice, there are many proximity measures, i.e. notions of similarity between two patterns, that lead to satisfactory clustering solutions and it does not really matter which one we choose. Take the k-means clustering with the ℓ_2 metric for example, where the objective function is:

$$\mathcal{H}_2(C_1, C_2, ..., C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu^i\|_2^2$$

where μ^i are the optimal centers of clusters C_i. In many cases, the choice of the 2-norm to compute the distances is arbitrary. We would expect that for many practical instances $X \subseteq \mathbb{R}^d$, even if the distance between patterns x and y is slightly less than $\|x - y\|_2$ for every $x, y \in \mathbb{R}^d$, the optimal clustering solution should not change. This slight modification of the proximity measure is captured by the notion of *perturbation* of a metric space.

Definition 3.2.1 (a-perturbation of a metric space). *Let (Y, δ) be a metric space. A a-perturbation of this metric space is a pair (Y, δ'), where $\delta' : Y \times Y \to [0, +\infty)$ is a symmetric function such that for any $x, y \in Y$:*

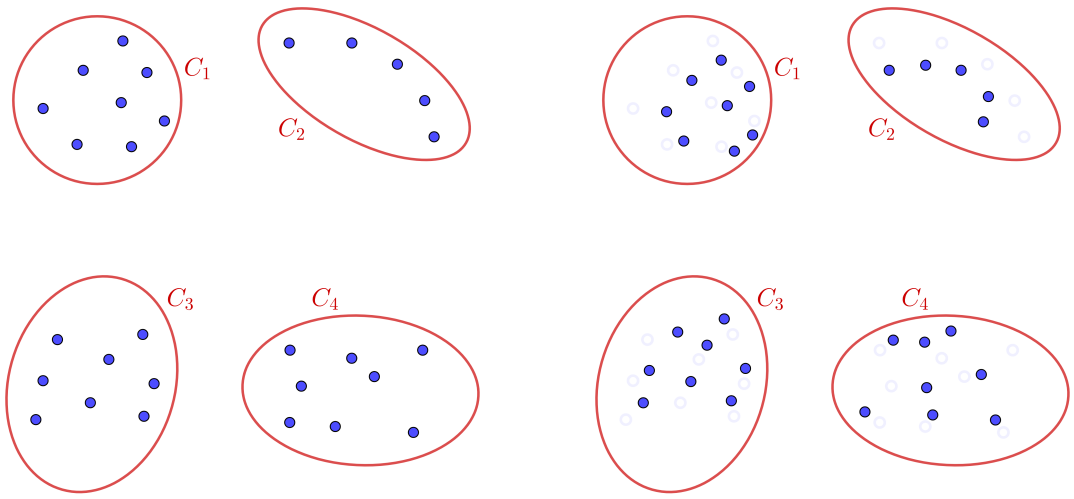
$$\delta'(x, y) \in \left[\frac{1}{a}\,\delta(x, y),\ \delta(x, y)\right]$$

In other words, all distances of an a-perturbation of a metric space are shrunk, but only by a multiplicative factor of $\frac{1}{a}$. Notice that the function δ' is not necessarily a metric. When we require that δ' is a metric, then (Y, δ') is called a *a-metric perturbation.*

This idea, that for many applications the optimal clustering should not change for any perturbation of the clustering metric space resulted in the introduction of the notion of *a-perturbation stability* or *a-perturbation resilience* or *Bilu Linial stability* [62].

Definition 3.2.2 (a-perturbation stability). *Let $\mathcal{I} = ((Y, \delta), X, \mathcal{H})$ be a clustering instance. Let $\mathcal{C}$ be the optimal k-clustering of $\mathcal{I}$. Then, $\mathcal{I}$ is a-perturbation stable, if for every instance $\mathcal{I}' = ((Y, \delta'), X, \mathcal{H})$, where (Y, δ') is a a-perturbation of (Y, δ), the unique optimal clustering of $\mathcal{I}'$ is $\mathcal{C}$.*

Again, if the above is true for every *a-metric* perturbation, we call the instance *a-metric perturbation stable.*



(a) Optimal Clustering before Perturbation **(b)** Optimal Clustering after Perturbation

Figure 3.2: Perturbation Stability: Optimal Clustering is the same for all Perturbations

3.2.2 A basic property of a-perturbation stable instances

Next, we state the most important property of a-perturbation stability and prove it. This property essentially states that a-metric perturbation stability with a center-based objective implies a-center stability (when $Y = X$). Since a-metric perturbation stability is a weaker notion than a-perturbation stability, this also means that a-perturbation stable instances are a-center stable. This is indeed a very useful property, as most of the work on the design of clustering algorithms for a-perturbation stable instances that I am familiar with makes use of the properties implied by a-center stability.

Theorem 3.2.1 (Konstantin Makarychev and Yury Makarychev, 2016). *Consider an a-metric perturbation stable k-clustering instance $\mathcal{I} = ((X, \delta), \mathcal{H}_p)$, where $\mathcal{H}_p$ is the ℓ_p objective. Then, $\mathcal{I}$ is a-center stable.*

Proof of theorem 3.2.1.
Let $C_1, C_2, ..., C_k$ be the unique optimal solution; and let $c_1, c_2, ..., c_k$ be a set of centers of $C_1, C_2, ..., C_k$. We assume, for the sake of contradiction, that there exists an $i \in [k]$, a $p \in C_i$ and a $j \in [k] \setminus \{i\}$ such that:

$$\delta(p, c_j) \leq a\delta(p, c_i) \tag{3.1}$$

In the first place, we will define a new metric δ' and prove that it is an a-metric perturbation of the metric δ. We consider the complete (undirected) graph on X and assign length $len(u, v) = \delta(u, v)$ to each edge $(u, v) \in (X \times X) \setminus \{(p, c_j)\}$ and $len(p, c_j) = r$, where we have set $r = \delta(p, c_i)$. Let δ' be the shortest path metric on the complete graph on X with edge lengths $len(u, v)$. Note that $len(u, v) \leq \delta(u, v)$, since $\delta(p, c_j) \geq \delta(p, c_i) = r$, where we have used that C_1, C_2,..., C_k is the optimal clustering for a clustering problem with a center-based objective. Observe that, δ' is a metric and for every $(u, v) \in X \times X$:

$$\delta'(u, v) \leq \delta(u, v) \tag{3.2}$$

because $\delta'(u, v) \leq len(u, v) \leq \delta(u, v)$. In addition, note that for every $(u, v) \in X \times X$, it holds that $len(u, v) \geq \frac{1}{a}\delta(u, v)$, by definition of len and (3.1). Therefore, if we consider any path $P =$

$\langle u_0, u_1, ..., u_f \rangle$ in the complete graph on X with edge lengths given by len, then:

$$\sum_{i=0}^{f-1} len(u_i, u_{i+1}) \geq \frac{1}{a}\delta(u_i, u_{i+1}) \geq \frac{1}{a}\delta(u_0, u_f)$$

Using the above and the definition of δ' as a shortest path metric, it holds that:

$$\delta'(u, v) \geq \frac{1}{a}\delta(u, v) \tag{3.3}$$

By combining (3.2) and (3.3), we reach the conclusion that, indeed, δ' is an a-metric perturbation of δ.
By definition of a-metric perturbation stability, the $((X, \delta'), \mathcal{H}_p)$ has the same unique optimal clustering as $((X, \delta), \mathcal{H}_p)$. Because $p \in C_i$, by Remark 2. in the preliminaries section, it holds that:

$$\delta'(p, c_j) > \delta'(p, c_i) = r \tag{3.4}$$

We will complete the proof by showing that δ' is equal to δ within cluster C_i.

Lemma 3.2.2. *For all $u, v \in C_i$ we have that $\delta(u, v) = \delta'(u, v)$.*

Proof of Lemma 3.2.2. Consider any two points $u, v \in X$ and the shortest path P from u to v in the complete graph on X with edge lengths given by len. There are three cases:

1. P does not contain the edge (p, c_j). Then, by definition of $len(x, y)$ for $x, y \in X$, we know that $P = \langle u, v \rangle$, i.e. it contains only one edge, (u, v). Thus:

$$\delta'(u, v) = \delta(u, v) \tag{3.5}$$

2. P contains (p, c_j) and is of the form:

$$\underbrace{u \rightsquigarrow p}_{P_1} - \underbrace{c_j \rightsquigarrow v}_{P_2}$$

Of course, P_1 and P_2 do not contain (p, c_j), because P is a path and the fact that it is the shortest path from u to v implies that: $P_1 = \langle u, p \rangle$ and $P_2 = \langle c_j, v \rangle$. Thus:

$$\delta'(u, v) = \delta(u, p) + r + \delta(c_j, v) \tag{3.6}$$

3. P contains (p, c_j) and is of the form:

$$\underbrace{u \rightsquigarrow c_j}_{P_1} - \underbrace{p \rightsquigarrow v}_{P_2}$$

Similarly, it holds that:

$$\delta'(u, v) = \delta(u, c_j) + r + \delta(p, v) \tag{3.7}$$

By combining (3.5), (3.6), (3.7), we conclude that, for any $u, v \in X$:

$$\delta'(u, v) = \min(\delta(u, v), \delta(u, p) + r + \delta(c_j, v), \delta(u, c_j) + r + \delta(p, v))$$

Hence, to prove this lemma, it suffices to show that for any $u, v \in C_i$:

$$\delta(u, v) \leq \min(\delta(u, p) + r + \delta(c_j, v), \delta(u, c_j) + r + \delta(p, v))$$

We fix some $u, v \in C_i$ and assume that $\delta'(u, v) = \delta(u, p) + r + \delta(c_j, v)$ (for the other case we work similarly). Since $v \in C_i$, we have $\delta(u, c_i) \leq \delta(u, c_j)$, therefore:

$$\delta(u, p) + r + \delta(c_j, v) \geq \delta(u, p) + \delta(p, c_i) + \delta(c_i, v) \geq \delta(u, v)$$

by triangle inequality, thus completing the proof. □

In conclusion, by (3.4) and Lemma 3.2.2:

$$r = \delta'(p, c_j) > \delta'(p, c_i) = \delta(p, c_i) = r$$

which is, of course, a contradiction. □

It is worth mentioning that Konstantin and Yury Makarychev proved a more general result than the above since the objective function is not required to be an ℓ_p objective. The only property of the objective function that is useful in the previous proof, is the fact that, in the optimal clustering, each point is closer to the cluster's optimal center, than to any other center. That's why, in [42], the authors define the class of *center-based* objective functions, which are essentially the ones that satisfy this property, and proved the above for any center-based objective function.

3.2.3 Efficient Clustering Algorithm under a-perturbation stability

Although the a-proximity and a-perturbation stability are natural assumptions that we would expect to be satisfied by many practical clustering instances, they are very powerful, as they allow us to design algorithms that find the optimal clustering solutions in polynomial time, even for very small values of a. We will now provide an example of such an algorithm, specifically the algorithm by Angelidakis et al. [42], and present the main ideas behind its analysis. The design of this algorithm is based on the two following observations:

1. For an a-metric perturbation stable clustering instance with the ℓ_p objective and $a \geq 2$, each point is closer to its own center than to any other point that belongs in a different cluster, in the optimal clustering. This fact follows from Theorem 3.2.1 and item 1. of Lemma 3.1.1.

2. Think of the input metric space as a complete weighted graph $G(V, E)$, where the weight of an edge $\{u, v\}$ is $\delta(u, v)$. Then, given a spanning tree T of the graph, we can compute in polynomial time the optimal clustering of the input, subject to the constraint that each cluster *induces a connected subgraph* of T, using easy dp.

This train of thought led Anggelidakis et al. to come up with the *signle-link++* algorithm described below.
If $\mathcal{I}$ is a-pertrubation stable with $a \geq 2$, then due to observation 1, each cluster in the optimal clus-

Algorithm 1: single-link++

Input: Clustering instance $\mathcal{I} = ((X, \delta), \mathcal{H}_p)$, number of clusters k
Output: a clustering C of X

1 Compute the minimum spanning tree T of the complete graph $G(V, E)$ that has one vertex for each $x \in X$ and edge weight $\delta(u, v)$ for every $u, v \in X$, by running the Kruskal's algorithm.
2 Among all $\binom{n-1}{k-1}$ subsets of $k-1$ edges of T and the induced clusterings that arise when we remove those edges from T (with one cluster per connected component), compute the one with the minimum ℓ_p cost.

tering of X induces a connected subgraph of the MST and therefore the Algorithm 2 will compute the optimal clustering in polynomial time (due to observation 2). To see why this is true, consider any two distinct clusters C and C' in the optimal clustering and the iteration $t \in [n-1]$ of Kruskal's Algorithm where an edge $\{u, v\}$ was added in the MST, such that $u \in C$ and $v \in C'$. Then, at iteration t, both clusters C and C' will induce a connected subgraph of the partially constructed minimum spanning tree at the end of iteration $t-1$. This is because Kruskal's algorithm attempts to add the lighter edges in the minimum spanning tree, before the heavier edges, and because of observation 1, any intra-cluster distance is smaller than any inter-cluster distance. As a result, the following is true.

Theorem 3.2.3 (Haris Angelidakis, Yury Makarychev, Konstantin Makarychev).
Let $\mathcal{I} = ((X, \delta), \mathcal{H}_p)$ be a k-clustering instance, for some $p \geq 1$ and $k \in \mathbb{N}^*$, that is a-perturbation stable with $a \geq 2$. Then, Algorithm 2 computes the optimal clustering of X (in polynomial time).

Again, Angelidakis et al. proved the above theorem for a wider class of objective functions. Note that this simple dynamic programming algorithm overcomes the NP-hardness of clustering for a very small value of a, which is essentially tight, as it is NP-hard to solve the k-centers problem exactly if we allow $(2 - \epsilon)$-perturbation stable inputs for any small $\epsilon > 0$.

Chapter 4

Explainable Clustering

It is time to give a definition of the explainable clustering problem and discuss the most important results in this field. First of all, we formally define clustering using Threshold Trees and the Price of Explainability (PoE), which is the cost due to the interpretability constraint that has to be satisfied by any solution. Next, we present some popular explainable clustering algorithms with a special focus on the Iterative Mistake Minimization (IMM) algorithm, whose performance we will analyze under various stability assumptions in Chapter 5. Last but not least, we describe some hard clustering instances that entail large PoE lower bounds. An interesting property of these instances is that they are well-separated, which will help us study their stability in Chapter 5.

4.1 Clustering using Threshold Trees

The key idea behind threshold trees is that of a *threshold cut*, which is a simple way to partition a dataset into two clusters, using only one feature (Figure 4.1). Consider a d-dimensional dataset X, a feature $i \in [d]$ and a threshold $\theta \in \mathbb{R}$. We can partition X into two clusters C_1, C_2 by placing each $x = [x_1, x_2, ..., x_d] \in X$ in C_1 if $x_i \leq \theta$ and in C_2 otherwise.

A *threshold tree* is an unsupervised variant of a (binary) decision tree (Figure 4.2). Each internal node u of the tree is associated with a feature-threshold pair (i_u, θ_u) and it induces a clustering of the input data X by iteratively applying threshold cuts. More formally, for a threshold tree T with k leaves, we assign a set Y_u for every node u of the tree, according to the following recursive

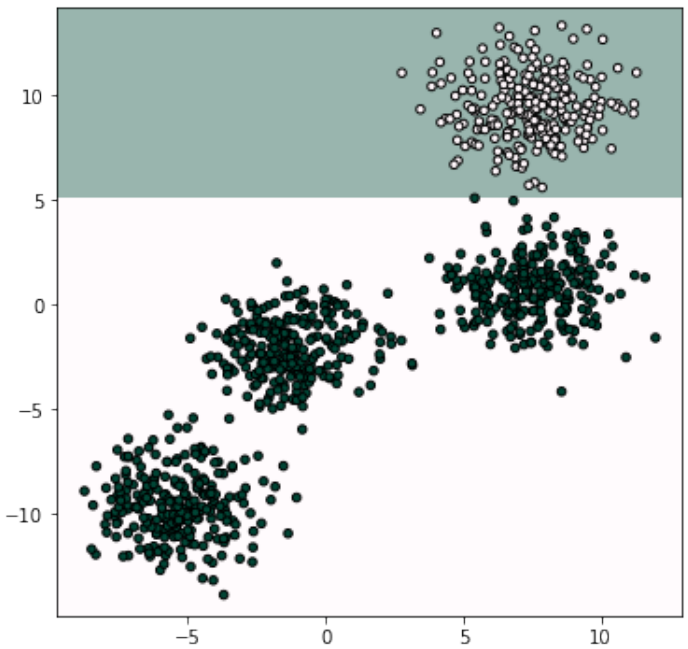


Figure 4.1: Partition induced by threshold cut $(2, 5.094)$

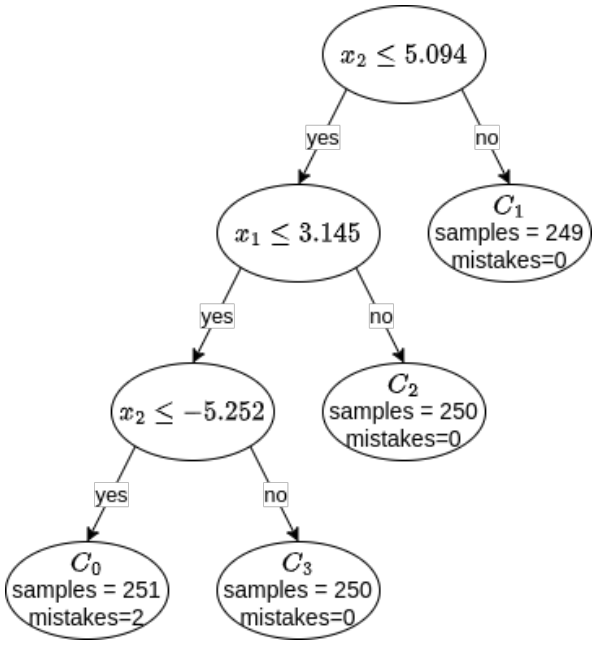


Figure 4.2: Threshold Tree

procedure:

- Let r be the root of the tree. Then $Y_r = \mathbb{R}^d$.
- For every internal node u of the tree, we set:

$$Y_{u \to left} = \{y \in Y_u : \ y_{i_u} \leq \theta_u\}$$

$$Y_{u \to right} = \{y \in Y_u : \ y_{i_u} > \theta_u\}$$

where $u \to left$ and $u \to right$ are the left and right children of u. Then the induced k-clustering is:

$$\mathcal{C} = \{X_w, \ w \in leaves(T)\}$$

where:

$$X_u = \{X \cap Y_u\} \text{ for } u \in T$$

and the cost of the explainable clustering is:

$$cost(T) = \mathcal{H}(C_1, C_2, ..., C_k, \delta)$$

that is, the cost of the clustering induced by T. In the rest of this project, we will say that "node u is split by the threshold cut (i_u, θ_u)" and that "node u contains a set $S \subseteq \mathbb{R}^d$ " when $S \subseteq Y_u$. We might also say that "(i_u, θ_u) separates two points $y_1, y_2 \in Y_u$", if $y_1 \in Y_{u \to left}$ and $y_2 \in Y_{u \to right}$ or the opposite.

4.2 Price of Explainability

As with unconstrained clustering, explainable clustering algorithms try to minimize a *cost function*, such as the k-median or the k-means objective. However, similarly to any model-based interpretation method, explainable clustering entails a decrease in performance. Restricting the possible clusterings results in solutions that potentially have a greater cost than those produced by unconstrained algorithms that make use of all features to define clusters. The inherent cost that has to be

paid by an explainable algorithm $\mathcal{A}$ due to the constrained interpretable form of valid clustering solutions is called *price of explainability* and is formally defined as follows:

$$PoE = \inf_{\substack{\text{explainable} \\ \text{algorithm } \mathcal{A}}} \sup_{\substack{X \subseteq \mathbb{R}^d, \\ |X| < \infty}} \frac{cost(\mathcal{A}(X))}{OPT(X)}$$

where $\mathcal{A}(X)$ is the threshold tree that algorithm $\mathcal{A}$ produces on input X and $OPT(X)$ is the cost of the optimal *unconstrained* clustering of X. In other words, when we want to assess the accuracy of an explainable clustering algorithm, we compare its cost to the optimal unconstrained clustering solution. In the following section, we will provide the first algorithm and the corresponding Theorem that offered an upper bound for the PoE.

4.3 The IMM Algorithm

The Iterative Mistake Minimization algorithm or IMM (Algorithm 2) was proposed in the paper [57], which introduced explainable clustering and is the first attempt to tackle the explainable k-median and k-means problems ($\delta = \delta_1$ and $\mathcal{H} = \mathcal{H}_1$ or $\delta = \delta_2$ and $\mathcal{H} = \mathcal{H}_2$). As with all of the explainable clustering algorithms (that I know of), its first step (line 1.) is to compute a *reference clustering*, i.e. a (non-explainable) clustering C with centers M of the input X, which is obtained by using one of the constant approximation algorithms for k-median (k-means).
Next, it constructs a threshold tree with k leaves in a top-down manner; starting from a single node that corresponds to the root of the tree, which contains the whole dataset X and while the tree created so far has leaves that contain more than one reference centers, it chooses some leaf u with this property and splits it into two using a threshold cut (i_u, θ_u). Hence, when the algorithm terminates, the threshold tree it returns has k *homogeneous* leaves, i.e. leaves that contain a single reference center $\mu \in M$, to which every data point in the same leaf is assigned. But how do we pick a suitable cut (i_u, θ_u), so that we are certain that the clustering produced is not arbitrarily more expensive than the optimal unconstrained clustering? As its name suggests, every time the IMM algorithm splits a node u, it chooses the threshold cut that makes the fewest mistakes. We say that the threshold cut (i_u, θ_u) that splits an internal node u of a threshold tree makes a mistake on $x \in X_u$ if it separates it from its closest reference center. The steps of the algorithm are shown in Figures 4.3 through 4.6.

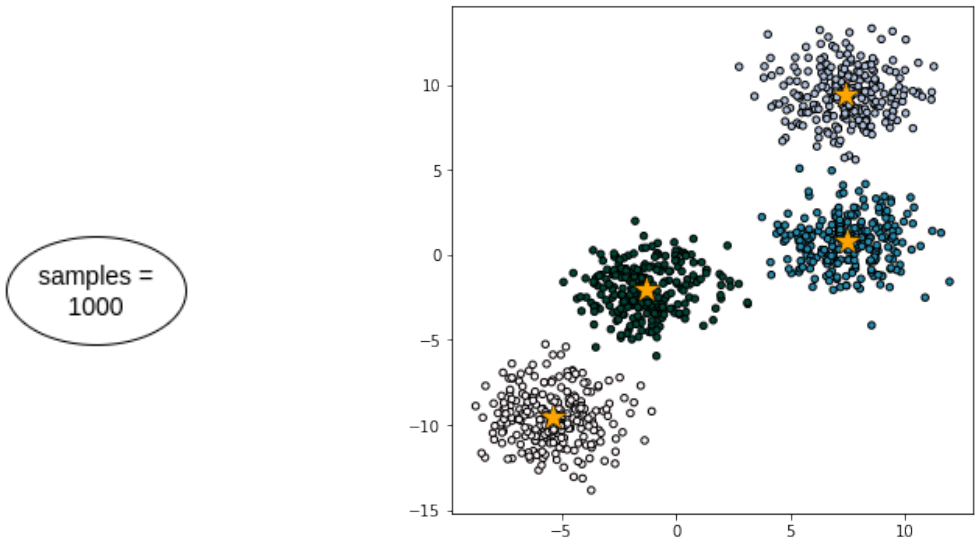


Figure 4.3: Step 0: Compute a reference clustering

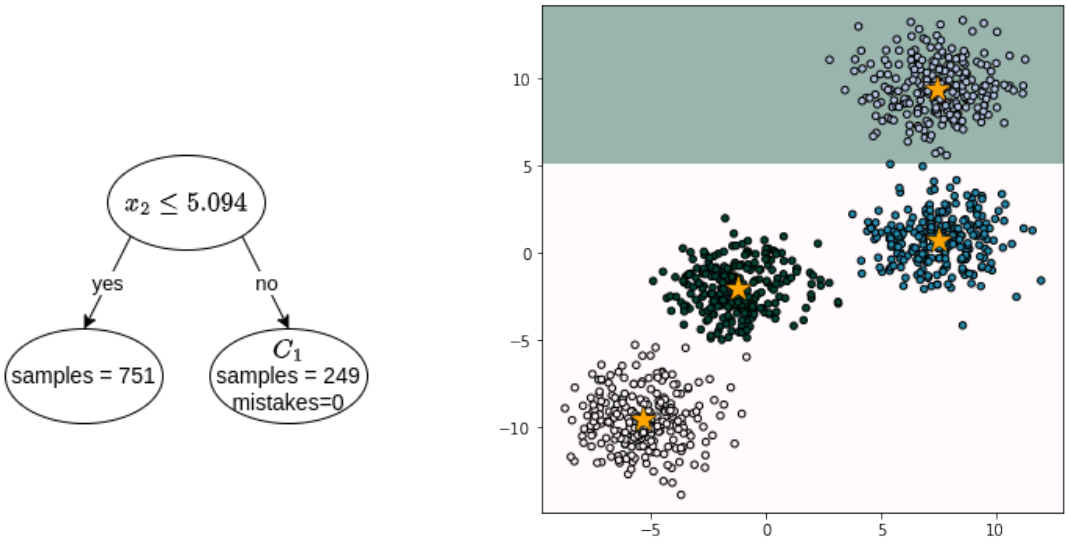


Figure 4.4: Step 1: Choose cut that makes minimum mistakes (0)

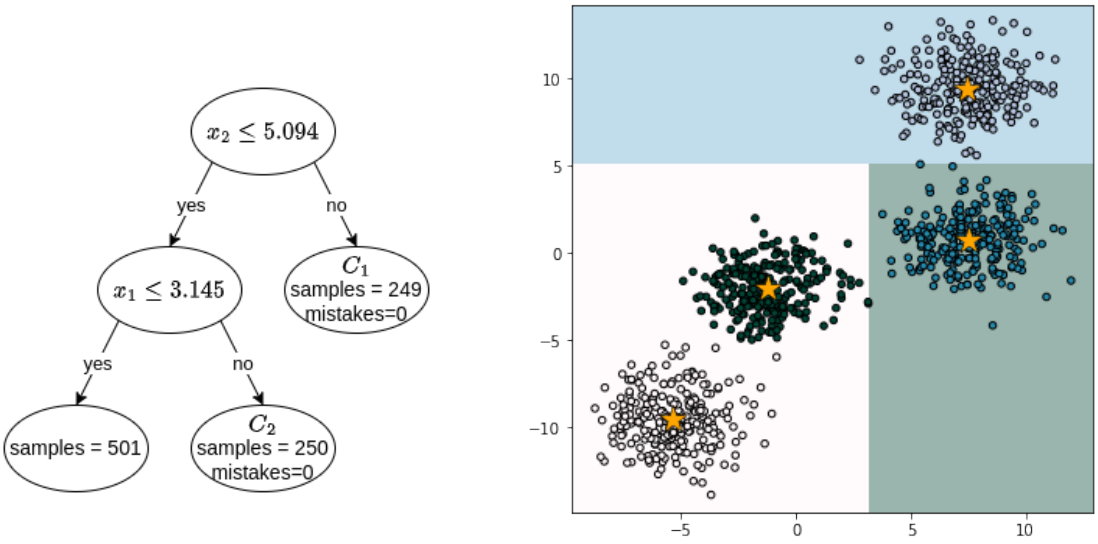


Figure 4.5: Step 2: 0 mistakes

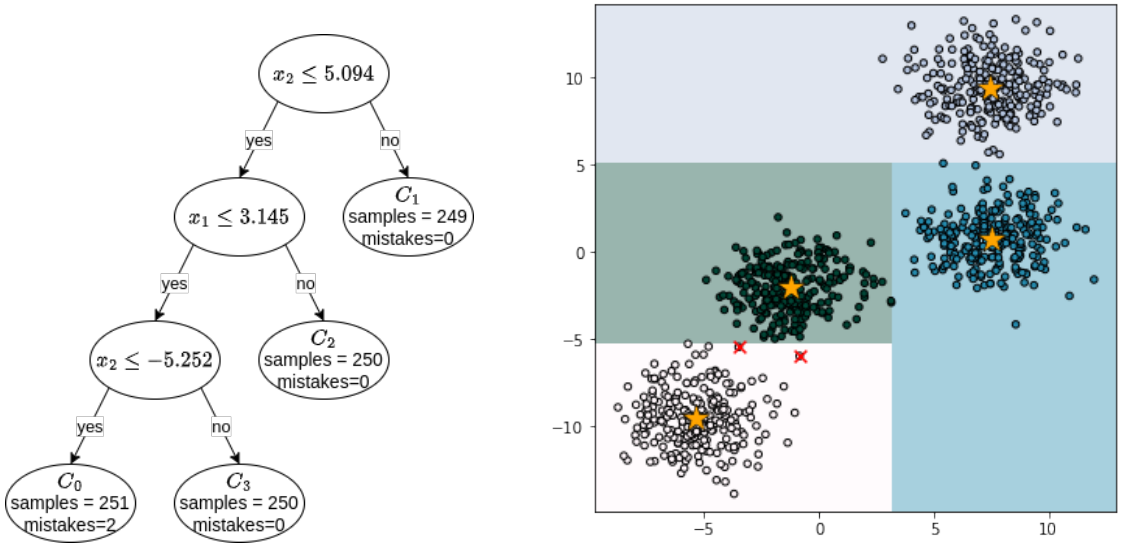


Figure 4.6: Step 3: 2 mistakes

Algorithm 2: Iterative Mistake Minimization (IMM)

Input: $\{x^1, x^2, ..., x^n\} \subseteq \mathbb{R}^d$, k
Output: root of the threshold tree

1 $\{\mu^1, \mu^2, \mu^3, ..., \mu^k\} \leftarrow k$-median$(X)$
2 **for** $j = 1$ **to** n **do**
3 $\quad y^j \leftarrow \arg\min_{1 \le l \le k} \|x^j - \mu^l\|_1$
4 **return** build_tree$(\{x^j\}_{j=1}^n, \{y^j\}_{j=1}^n, \{\mu^j\}_{j=1}^k)$
5 **build_tree** `build_tree`$(\{x^j\}_{j=1}^m, \{y^j\}_{j=1}^m, \{\mu^j\}_{j=1}^k)$:
6 $\quad$ **if** $\{y^j\}$ *is* homogeneous **then**
7 $\quad\quad$ leaf.center $\leftarrow y^1$
8 $\quad\quad$ **return** leaf
9 $\quad i, \theta \leftarrow \arg\min_{i,\theta} \sum_{j=1}^m$ mistake$(x^j, \mu^{y^j}, i, \theta)$
10 $\quad node$.condition = $x_i \le \theta$
11 $\quad F = \{j \in [m] :$ mistake$(x^j, \mu^{y^j}, i, \theta) = 1\}$
12 $\quad L = \{j \in [m] \setminus F : x_i^j \le \theta\}$
13 $\quad R = \{j \in [m] \setminus F : x_i^j > \theta\}$
14 $\quad node.left =$ built_tree$(\{x^j\}_{j \in L}, \{y^j\}_{j \in L}, \{\mu^j\}_{j=1}^k)$
15 $\quad node.right =$ built_tree$(\{x^j\}_{j \in R}, \{y^j\}_{j \in R}, \{\mu^j\}_{j=1}^k)$
16 $\quad$ **return** node
17 **mistake** `mistake`(x, μ, i, θ):
18 $\quad$ **return** $(x_i \le \theta) \ne (\mu_i \le \theta)$? 1 : 0

As we will explain, this method of choosing the threshold cuts guarantees that the explainable clustering cost will be a good approximation of the optimal clustering of the input instance. More specifically, in [57] the following theorem is proved.

Theorem 4.3.1 (Sanjoy Dasgupta, Nave Frost, Michal Moshkovitz, Cyrus Rashtchian, 2020). *Let $X \subseteq \mathbb{R}^d$ and $k \ge 2$ be the input of the IMM algorithm. Suppose that the reference centers returned in line 1. are $M = \{\mu^1, \mu^2, ..., \mu^k\}$ and that the algorithm returns a threshold tree T of depth H. Then:*

1. *The k-median cost of the explainable clustering induced by T is at most:*

$$cost_1(T) \le (2H + 1)\, cost_1(M)$$

2. *The k-means cost of the explainable clustering induced by T is at most:*

$$cost_2(T) \le (8Hk + 2)\, cost_2(M)$$

In particular, if the reference clustering is produced by a constant approximation algorithm, then IMM achieves approximation factors of $O(k)$ and $O(k^2)$ respectively (when compared to the optimal unconstrained k-median and k-means clustering).

We will now provide the analysis of Algorithm 2 for the k-median case (the k-means for the explainable clustering is similar and we refer the reader to [57] for more details). It is essentially the same proof as in [57] but presented in my own words and based on my understanding.

Proof of Theorem 4.3.1 for the k-median case.
We start with some notation. We set:

- The internal nodes of the threshold tree T

$$T_i = T \setminus leaves(T)$$

- The centers that are contained in node u:

$$M_u = M \cap Y_u$$

- A function $c : X \to M$, which maps any $x \in X$ to its assigned center in the reference clustering, i.e.:

$$c(x) = \arg\min_{\mu \in M} \|x - \mu\|_1$$

- A function $c' : X \to M$ that maps each $x \in X$ to its assigned center in the explainable clustering, i.e. for $x \in X$, if x is contained in the leaf $v \in leaves(T)$ with $\mu \in M_v$ being the only center in this leaf, then:

$$c'(x) = \mu$$

- The *minimum number of mistakes* t_u by the cut chosen to split node $u \in T_i$ of the threshold tree T.

- The *diameter* of the node u:

$$D(u) = \max_{\mu_1, \mu_2 \in M_u} \{\|\mu_1 - \mu_2\|_1\}$$

The first step of the proof is to provide an upper bound for the cost of the explainable clustering in terms of t_u in each internal node.

Lemma 4.3.2. *It holds that:*

$$cost(T) \leq cost(M) + \sum_{u \in T_i} t_u \, D(u)$$

Proof of Lemma 4.3.2.
First, we rewrite the cost of the IMM algorithm on input X as follows:

$$cost(T) = \sum_{x \in X} \|x - c'(x)\|_1 = \sum_{x \in X} \|x - c(x)\|_1 \; + \; \sum_{x \in X} (\|x - c'(x)\|_1 - \|x - c(x)\|_1) \Rightarrow$$

$$\Rightarrow cost(T) = cost(M) + \sum_{x \in X} rc(x)$$

where $rc(x) = \|x - c'(x)\|_1 - \|x - c(x)\|_1$ is the *reassignment cost* of x, that is, the extra cost we have to pay to assign x to the sub-optimal center $c'(x)$ in the explainable clustering. Now we will upper bound the reassignment cost of every $x \in X$.
Let $x \in X$ be a mistake of the algorithm at the internal node $u \in T$, which means that x was separated for the first time from its reference center $c(x)$ at node u. Notice, that x will end up in the same leaf with some reference center in M_u, when the algorithm terminates, hence $c'(x) \in M_u$. Since δ_1 is a metric, by the triangle inequality we have:

$$\|x - c(x)\|_1 \leq \|x - c'(x)\|_1 + \|c(x) - c'(x)\|_1$$

$$\Rightarrow rc(x) \leq \|c(x) - c'(x)\|_1 \leq D(u)$$

On the other hand, if the algorithm assigns x to its optimal reference center, the reassignment cost is obviously 0.
For any $u \in T_i$, we set:

$$X_u^{miss} = \{x \in X : \text{ x was separated from } c(x) \text{ at node } u\}$$

Notice that a mistake happens in exactly one node, so $X_u^{miss} \cap X_v^{miss} = \emptyset$ for $v \neq u$. Also, by the construction of the IMM algorithm, $|X_u^{miss}| = t_u$ for any $u \in T_i$. By rewriting the cost of the resulting clustering, we obtain the desired upper bound:

$$cost(T) = cost(M) + \sum_{x \in X} rc(x) = cost(M) + \sum_{u \in T_i} \sum_{x \in X_u^{miss}} rc(x) \leq$$

$$\leq cost(M) + \sum_{u \in T_i} t_u D(u) \tag{4.1}$$

□

Next, we are going to give a lower bound for the cost of the reference clustering, in terms of the minimum mistakes possible on every internal node. The next lemma essentially justifies our threshold cut choices, as it implies that if the IMM makes a lot of mistakes, which entail a high cost of the explainable clustering, it probably means that the reference clustering was very expensive to start with, so we might not lose much from the explainability constraint.

Lemma 4.3.3. *Let H be the height of the threshold tree T. Then, it holds that:*

$$cost(M) \geq \frac{1}{2H} \sum_{u \in T_i} t_u D(u)$$

Proof of Lemma 4.3.3.
To prove Lemma 4.3.3, we will lower bound the cost at each node $u \in T_i$ in terms of the minimum number of mistakes made by any threshold cut that splits u.

Lemma 4.3.4. *For every $u \in T_i$, it holds that:*

$$\sum_{x \in X_u^{cor}} \|x - c(x)\|_1 \geq \frac{t_u}{2} D(u)$$

where:

$$X_u^{cor} = \{x \in X_u : c(x) \in X_u\}$$

Proof of Lemma 4.3.4.
Fix a node $u \in T_i$. Now consider any dimension $j \in [d]$ and the projection of $X_u \cup M_u$ on this dimension. Without loss of generality, we assume that $M_u = \{\mu^1, \mu^2, ..., \mu^r\}$ for some $r \leq k$, such that:

$$\mu_j^1 \leq \mu_j^2 \leq ... \leq \mu_j^r$$

We also consider the midpoints m^i of the intervals $[\mu_j^i, \mu_j^{i+1}]$ for $i \in [k-1]$, or in other words:

$$m^i = \frac{\mu_j^i + \mu_j^{i+1}}{2}$$

Let $I_x = [x_j, c(x)_j]$, if $x_j \leq c(x)_j$, else $I_x = [c(x)_j, x_j]$, for every $x \in X_u^{cor}$. We will say that a pair $(i, i+1)$ is covered by some point $x \in X_u^{cor}$, if at least one of the following happens:

- $[\mu_j^i, m^i] \subseteq I_x$
- $[m^i, \mu_j^{i+1}] \subseteq I_x$

The key observation needed to prove this Lemma is that if t_u are the minimum number of mistakes for any threshold cut, any cut (j, m^i) with $i \in [k-1]$ results in at least t_u mistakes. This means that every pair $(i, i+1)$ is covered by at least t_u points $x \in X_u^{cor}$. We define:

$$I_x^i = I_x \cap [\mu_j^i, \mu_j^{i+1}], \text{ for } x \in X_u^{cor}, i \in [k-1]$$

$$S^x = \{i \in [k-1] : \ (i, i+1) \text{ is covered by } x\} \text{ for } x \in X_u^{cor}$$

$$S^i = \{x \in X_u^{cor} : \ x \text{ covers } (i, i+1)\} \text{ for } i \in [k-1]$$

Now, we can bound the cost of all $x \in X_u^{cor}$ along dimension j as follows:

$$\sum_{x \in X_u^{cor}} |x_j - c(x)_j| = \sum_{x \in X_u^{cor}} \sum_{i \in S^x} |I_x^i| = \sum_{i=1}^{k-1} \sum_{x \in S^i} |I_x^i|$$

$$\geq \sum_{i=1}^{k-1} t_u \frac{\mu_j^{i+1} - \mu_j^i}{2} = t_u \frac{\mu_j^r - \mu_j^1}{2} \tag{4.2}$$

where for the last inequality, we used that if x covers the pair $(i, i+1)$, then (by definition) $|I_x^i| \geq \frac{\mu_j^{i+1} - \mu_j^i}{2}$. Notice, that (4.2) holds for any dimension $j \in [d]$ and as a result, if we sum up this inequality for all $j \in [d]$, we get:

$$\sum_{x \in X_u^{cor}} \|x - c(x)\|_1 \geq \frac{t_u}{2} D(u)$$

□

Now consider a point $x \in X$ and notice that the nodes that contain it form a path from the root of the threshold tree to a leaf v corresponding to the cluster X_v in the explainable clustering. This path contains at least one and at most H nodes, where H is the height of the tree, so there are at most H nodes $u \in T_i$, such that $x \in X_u^{cor}$. Thus:

$$H \ cost(M) \geq \sum_{u \in T_i} \sum_{x \in X_u^{cor}} \|x - c(x)\|_1 \geq \sum_{u \in T_i} \frac{t_u}{2} D(u)$$

□

By combining Lemma 4.3.2 and 4.3.3, we obtain:

$$cost(T) \leq cost(M) + \sum_{u \in T_i} t_u D(u) \leq (2H+1) \, cost(M)$$

□

Theorem 4.3.1 implies that PoE is upper-bounded by $O(k)$ and $O(k^2)$ for the k-median and k-means case respectively. As we shall see in sections 4.5 and 4.6, PoE is also lower-bounded by $\Omega(\log k)$ and $\Omega(k)$ for the k-median and k-means objectives respectively. Therefore, there is still room for improvement in terms of approximation guarantees of the explainable clustering algorithms.

4.4 Improved Explainable Clustering Algorithms

After noticing this huge gap between the upper and lower bounds of PoE, it is natural to wonder whether there exist algorithms that obtain a better approximation of the optimal unconstrained clustering than IMM. A series of independent works have given a positive answer to this question by proving that the Price of Explainability can be significantly reduced, especially in the k-median case, where we are able to get an exponential improvement to the bound proven by Dasgupta et al. We will now present some of these algorithms along with the central ideas behind their inception.

4.4.1 Two randomized and oblivious explainable clustering algorithms for the k-median objective

We will now talk about two algorithms that not only imply a better upper bound for PoE but also have another interesting property: they are *oblivious* to the input data points. That is, they receive only the reference centers as input so that their running time depends only on the number of clusters k and the number of dimensions d and *not* on the number of input points n. The way that they achieve this is by replacing the IMM's greedy criterion, which chooses a threshold cut based on the minimum number of mistakes it makes, with a surprisingly simpler rule: at each step just pick a random threshold cut! In [59] Svensson et al. show that picking threshold cuts according to this rule can drastically improve the performance of our explainable clustering algorithm. To do this they make the following two crucial observations.

First of all, if we start again with a reference clustering C induced by centers M and we sample a random threshold cut, the probability that it splits a data point x from its closest center in the reference clustering, $c(x)$, is at most $\frac{\|x-c(x)\|_1}{L}$, where L is the diameter of the bounding box of the clustering instance. This also means that the expected number of points separated from their closest center is $\frac{cost_1(M)}{L}$.

In addition, notice that in (4.1) for each mistake in every node u the extra cost that we pay is at most the diameter of this node. Initially, this cost is equal to c_{max}, which is the maximum distance between any two centers. However, each child node has a smaller diameter than its predecessor, so its mistakes cost less than the mistakes of its ancestor. What the authors of [59] realized is, that if we pick threshold cuts uniformly at random, then with high probability after only a few cuts every pair of centers with distance at least $\frac{c_{max}}{2}$ are cut. As a result, every few iterations the worst-case cost of each mistake made by a threshold cut halves, thus dramatically decreasing the reassignment costs.

Motivated by these observations, Svensson et al. proposed the following algorithm:

The major difference to the IMM algorithm is the way that Algorithm 3 chooses the threshold cuts; when there exist leaves that contain two or more distinct reference centers, instead of picking the threshold cut according to a greedy rule, as IMM does, Algorithm 3 chooses a random threshold cut uniformly at random among the cuts that separate some two distinct centers contained in some leaf of the partially created tree. By making use of the observations that we presented earlier, the authors of [59] prove the following theorem.

Theorem 4.4.1 (O. Svensson, B. Gamlath, X. Jia, A. Polak). *Given reference centers $M = \{\mu_1, \mu_2, ..., \mu_k\}$ the Algorithm 3 outputs a threshold tree T, whose expected k-median cost satisfies:*

$$\mathbb{E}[cost_1(T)] \leq O\left(\log k \left(1 + log\left(\frac{c_{max}}{c_{min}}\right)\right)\right) cost_1(M)$$

where c_{max} and c_{min} are respectively the maximum and minimum distance between any two centers in M.

Nevertheless, notice that this algorithm can perform arbitrarily badly, in case c_{max} is much larger than c_{min}. This problematic ratio arises because we allow cuts that separate centers that are very close to each other compared to the diameter of the node where they are contained and one way to deal with this problem is to not allow such cuts. In that way, each threshold cut is considered a candidate cut for much fewer iterations of the algorithm, because, as we explained before, the maximum diameter of any leaf of the partially constructed tree drops rapidly, with high probability. As a result, each cut contributes less to the expected cost of the algorithm and thus reducing it significantly.

This reasoning led Svensson et al. to Algorithm 4, which is essentially the same as Algorithm 3, except for the fact that the set from which the algorithm chooses its threshold cuts uniformly at random, has changed. More specifically, if we set $c_{max}(t)$ to be the maximum distance between

Algorithm 3: Explainable k-median clustering with uniformly random cuts

Input: $\{x^1, x^2, ..., x^n\} \subseteq \mathbb{R}^d$, k
Output: root of the threshold tree

$\{\mu^1, \mu^2, \mu^3, ..., \mu^k\} \leftarrow k\text{-median}(X)$
Set $S_{qr} = \{(i, \theta) :$ threshold cut (i, θ) separates centers μ^q and $\mu^r\}$ for all $q, r \in [k]$
Initialize tree T_0 to contain only the root node, $root$.
Set $X_{root} = \{x^j\}_{j\in[n]}$ and $M_{root} = \{\mu^j\}_{j\in[k]}$
Set $t = 0$
while *partially created tree* T_t *contains leaves with more than two distinct centers* **do**
    Set $E_t = \bigcup_{u\in leaves(T_t)}\{(q, r) :\ \mu^q, \mu^r \in M_u\}$
    Set $R_t = \bigcup_{(q,r)\in E_t}\{S_{qr}\}$
    Pick (i, θ) uniformly at random from the set R_t.
    for $u \in leaves(T_t)$ **do**
        if (i, θ) *splits* u **then**
            u.condition = $x_i \leq \theta$
            $X_L = \{x \in X_u :\ x_i \leq \theta\}$
            $X_R = \{x \in X_u :\ x_i > \theta\}$
            $M_L = \{\mu \in M_u :\ \mu_i \leq \theta\}$
            $M_R = \{\mu \in M_u :\ \mu_i > \theta\}$
            $u.left =$ create_node(X_L, M_L)
            $u.right =$ create_node(X_R, M_R)
    $t = t + 1$
return root

any two centers in the same leaf of the partially constructed tree T_t at iteration t, instead of picking uniformly at random *any* threshold cut that separates some two distinct centers in a leaf of T_t, Algorithm 4 is only allowed to choose cuts that do not separate a pair of centers in the same leaf, which are at distance at most $\frac{c_{max}(t)}{k^4}$ from each other.
For this algorithm, Svensson et al. managed to prove the following theorem:

Theorem 4.4.2 (O. Svensson, B. Gamlath, X. Jia, A. Polak). *Given reference centers* $M = \{\mu_1, \mu_2, ..., \mu_k\}$, *Algorithm 4 outputs a threshold tree T, whose expected* k*-median cost satisfies:*

$$\mathbb{E}[cost_1(T)] \leq O(log^2 k)\ cost_1(M)$$

Interestingly, the ideas of Algorithm 4 can be generalized not only for the k-means cost function but also for any ℓ_p objective ($\delta = \delta_p$, $\mathcal{H} = \mathcal{H}_p$, $p \geq 1$), including k-means. The algorithm is essentially the same; the only thing that changes is the distribution from which we sample the random cuts.

Theorem 4.4.3. *Given a* k*-clustering instance* $((\mathbb{R}^d, \delta_p), X, \mathcal{H}_p)$ *and reference centers* $M = \{\mu_1, \mu_2, ...-, \mu_k\}$, *for any* $p \geq 1$ *there exists a randomized algorithm that outputs a threshold tree T, whose expected* ℓ_p *cost, for* $p \geq 1$ *satisfies:*

$$\mathbb{E}[cost_p(T)] \leq O(k^{p-1} \log^2 k)\ cost_p(M)$$

4.4.2 State-of-the-art explainable clustering algorithms for the k-median case

Independently from Svensson et al., other scientists have proposed algorithms that achieve even better approximation guarantees for the explainable clustering problem under the k-median objective. More precisely, these state-of-the-art algorithms are essentially identical to algorithms 3 and

Algorithm 4: Explainable k-median clustering with uniformly random cuts and forbidden cuts

Input: $\{x^1, x^2, ..., x^n\} \subseteq \mathbb{R}^d$, k
Output: root of the threshold tree

$\{\mu^1, \mu^2, \mu^3, ..., \mu^k\} \leftarrow k$-median$(X)$
Set $S_{qr} = \{(i, \theta) :$ threshold cut (i, θ) separates centers μ^q and $\mu^r\}$ for all $q, r \in [k]$
Initialize tree T_0 to contain only the root node, $root$.
Set $X_{root} = \{x^j\}_{j\in[n]}$ and $M_{root} = \{\mu^j\}_{j\in[k]}$
Set $t = 0$
while *partially created tree* T_t *contains leaves with more than two distinct centers* **do**
    Set $E_t = \bigcup_{u\in leaves(T_t)}\{(q, r) : \mu^q, \mu^r \in M_u\}$
    Set $c_{max}(t) = \max_{(q,r)\in E_t}\|\mu^q - \mu^r\|_1$
    Set

$$A_t = \bigcup_{(q,r)\in E_t} S_{qr} \text{ and } B_t = \bigcup_{\substack{(q,r)\in E_t: \\ \|\mu^q-\mu^r\|_1 \leq \frac{c_{max}(t)}{k^4}}} S_{qr}$$

    Set $R_t = A_t \setminus B_t$.
    Pick (i, θ) uniformly at random from the set R_t.
    for $u \in leaves(T_t)$ **do**
        if (i, θ) *splits* u **then**
            u.condition = $x_i \leq \theta$
            $X_L = \{x \in X_u : x_i \leq \theta\}$
            $X_R = \{x \in X_u : x_i > \theta\}$
            $M_L = \{\mu \in M_u : \mu_i \leq \theta\}$
            $M_R = \{\mu \in M_u : \mu_i > \theta\}$
            $u.left =$ create_node(X_L, M_L)
            $u.right =$ create_node(X_R, M_R)
    $t = t + 1$
return root

4, but the cost analysis provided is tighter.
First of all, Makarychev et al. in [61] study Algorithm 4 with a slight modification and prove that it offers better guarantees than those proved by Svensson et al. The only difference is the definition of B_t, i.e. the set of forbidden threshold cuts that separate centers that are very close to each other at iteration t. More specifically, for every $u \in leaves(T_t)$ instead of not allowing the separation of two centers $\mu, \mu' \in M_u$ such that $\|\mu - \mu'\|_1 \leq \frac{c_{max}(t)}{k^4}$, we replace k^4 with k^3. The authors of [61] initially provide a similar analysis to that of [59], achieving the same result. However, after more careful analysis, they manage to prove the following theorem.

Theorem 4.4.4 (Konstantin Makarychev, Liren Shan). *Given reference centers $M = \{\mu_1, \mu_2, ..., \mu_k\}$ the Algorithm 4 with the modification mentioned above outputs a threshold tree T, whose expected k-median cost satisfies:*

$$\mathbb{E}[cost_1(T)] \leq O(\log k \ \log\log k)\ cost_1(M)$$

The insight that led Makarychev and Shan to prove Theorem 4.4.5 was that the diameter of a leaf in the partially constructed tree is too pessimistic an upper bound for the reassignment cost of a data point x that is contained in this leaf. To see why this is the case, consider a point x, whose reference center is μ and a node u of the resulting threshold tree, where x is separated from μ, for which it holds that $D(u) >> \|x - \mu\|_1$. If there exists a center μ' that is very close to μ and it

so happens that after the cut x and μ' remain in the same leaf, then it is too pessimistic to upper bound the reassignment cost with $D(u)$, as it is much more preferable to assign x to μ'. Therefore, in the improved analysis, an additional bound for the reassignment cost of a point is used, namely its distance from the closest center *after separation.*
A completely different approach was followed by Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan, who study Algorithm 3 from a different perspective. Instead of bounding the expected cost of each cut chosen by the algorithm, they bound the reassignment cost of each point $x \in X$ separately and prove the following theorem by using the union bound.

Theorem 4.4.5 (Hossein Esfandiari, Vahab Mirrokni, Shyam Narayanan). *Given reference centers $M = \{\mu_1, \mu_2, ..., \mu_k\}$ the Algorithm 3 outputs a threshold tree T, whose expected k-median cost satisfies:*

$$\mathbb{E}[cost_1(T)] \leq O(\log k \ \log\log k)\ cost_1(M)$$

4.5 Lower Bound for the k-median case

In the previous sections, we have seen some popular explainable clustering algorithms, which achieve good approximation ratios. It is now the time to wonder about the optimality of these algorithms. Does the explainability constraint entail an unavoidable cost, compared to the unconstrained problem? In this section, we explain why this is indeed the case and discuss the results of [57], who give a lower bound for the price of explainability in the k-median case, by proving the following theorem:

Theorem 4.5.1. *For any $k \geq 2$ there exists a data set X, such that any explainable k-median algorithm $\mathcal{A}$ that produces a threshold tree T on input X has cost:*

$$cost_1(T) \geq \Omega(logk)\ OPT_1(X)$$

where $OPT_1(X)$ is the cost of the optimal unconstrained k-median clustering of X.

Now we are going to define this instance and its basic properties and explain the idea behind them and how they are going to be used in the proof. To create the instance we follow the steps below:

- We pick k points $M = \{\mu^1, \mu^2, ..., \mu^k\} \subseteq \{\pm 1\}^d$ according to Claim 4.5.1, so that they satisfy several desirable properties. These points are essentially the centers of the optimal clustering of the instance we are going to define.
- For each $i \in [k]$ we define the set:

  $$X^{\mu^i} := \{\mu^i - \mu^i_j e^j,\ j \in [d]\}$$

  where e^j is the unit vector along the j^{th} dimension. Notice that each point in C_i differs from μ^i at exactly one dimension, where it is equal to 0.
- The dataset is:

  $$X = \bigcup_{i \in [k]} X^{\mu^i}$$

Now, let's define the properties that the set M should satisfy to ensure a high explainable clustering cost. First of all, we define *an assignment* to a set of features $I \subseteq [d]$ as a function $\sigma : I \to \{\pm 1\}$. We will say that a point $x \in \mathbb{R}^d$ agrees with σ if $x_i = \sigma(i),\ \forall i \in I$.

Claim 4.5.1. *For any $k \in \mathbb{N}$ with $k \geq 3$, there exist k points $M \subseteq \{\pm 1\}^d$ that have the following properties, for any $\epsilon \geq \frac{\ln k}{\sqrt{k}}$:*

1. $d = k^3$
2. *for every* $\mu, \mu' \in \mu$ *with* $\mu \neq \mu'$*, it holds that:* $|\{\mu_i \neq \mu'_i\}| \geq \frac{d}{4}$
3. *for every set of features* $I \subseteq [d]$ *with size* $l \leq \frac{\ln k}{50}$ *and every assignment* σ *to* I*, the number of points in* M *that agrees with the assignment* σ *is at least* $k\left(\frac{1}{2^l} - \epsilon\right)$.

First of all, notice that $\mathcal{C} = \{X^{\mu^i}, i \in [k]\}$ is the optimal clustering of X with optimal centers μ and a cost of at most dk. Due to item 2. of claim 4.5.1, the optimal clustering is very well separated (as we will see in chapter 4. it is $\Omega(d)$-center stable and $\Omega(\sqrt{d})$-metric perturbation stable). As a result, if two points from different X^{μ^i} end up in the same cluster, due to the triangle inequality, they will contribute a cost of $\Omega(d)$ in the clustering cost, instead of only 2, if they where assigned to their closest μ^i. In other words, we would like to avoid separating μ^i from any of the points in X^{μ^i}. However, due to item 3. this is impossible. To see why this is the case, consider a node u in depth $l \leq \frac{\ln(k)}{50}$ of a threshold tree T created by some explainable clustering algorithm $\mathcal{A}$. Let $(i_1, \theta_1), (i_2, \theta_2), ..., (i_l, \theta_l)$ be the threshold cuts chosen by the algorithm in the path from the root of T to u, $I = \{i_1, i_2, ..., i_l\}$ and $\mu_u^+ = \{v \in Y_u : \ \mu_i = 1\}$, $\mu_u^- = \{v \in Y_u : \ \mu_i = -1\}$. Observe that since the projection of any $v \in V$ on any dimension can take two possible values (either +1 or -1), then all the points in μ_u^+ agree with some assignment σ to the set I (the assignment induced by the threshold cuts). The same holds for μ_u^-. Nevertheless, from item 3., we know that both $|\mu_u^+|$ and $|\mu_u^-|$ contain at least $k(2^{-l} - \epsilon)$ elements, thus, by the construction of the data set, the threshold cut (i_l, θ_l) will separate at least $k(2^{-l} - \epsilon)$ data points from their closets points in M. *In other words, the data set is constructed in such a way so as to ensure that any choice of threshold cuts up to a depth of* $O(\log k)$ *results in* $\Omega(k2^{-l})$ *mistakes, incurring the cost of* $\Omega(dk2^{-l})$. Now, since at each level of the tree there are 2^l nodes (we can prove that until the depth of $\frac{\ln k}{50}$ we can assume that the tree is complete, without loss of generality) and $\epsilon << 2^{-l}$ we get that the total cost is $\Omega(dk\log(k))$, which is $\Omega(\log(k))$ times more than the optimal clustering cost which is dk.
I will not provide all of the details of the formal proof of Theorem 4.5.1, but I will show Claim 4.5.1, which explains the existence of the hard instance. We will revisit this instance in Chapter 5 to study its center stability and perturbation stability.

Proof of Claim 4.5.1. We are going to prove the existence of those points via the *probabilistic method.* More specifically, we are going to show that if we choose a subset M of the vertices of the hypercube $\{\pm 1\}^d$ *uniformly at random*, with $|M| = k$ and $d = k^3$, the probability that M satisfies the properties above is *strictly* positive, which will lead us to the conclusion that indeed there exists such a set.
In the first place, let's prove item 2. Let $M = \{\mu^1, \mu^2, ..., \mu^k\}$ be the random subset of the hypercube's vertices and X_{ij} be random variables for every $i, j \in [k]$ and $i < j$ that are equal to 1 if $|\{r : \mu_r^i \neq \mu_r^j\}|$ and 0 otherwise. Also, consider d additional random variables Y_{ijr} for $i, j \in [k]$ with $i < j$ and $r \in [d]$ that are equal to 1 if $\mu_r^i = \mu_r^j$ and 0 otherwise. Fix some $i, j \in [k]$ with $i < j$ and notice that:

$$X_{ij} = \sum_{r=1}^{d} Y_{ijr}$$

Moreover, due to the linearity of expectation:

$$\mathbb{E}[X_{ij}] = \sum_{r=1}^{d} \mathbb{E}[Y_{ijr}] = \frac{d}{2}$$

Because $\{Y_{ijr}\}_{r \in [d]}$ are independent random variables, we can apply Hoeffding's inequality:

$$Pr\left(X_{ij} - \mathbb{E}[X_{ij}] \geq \frac{d}{4}\right) \leq \exp\left(-\frac{2\left(\frac{d}{4}\right)^2}{d}\right) \Rightarrow$$

$$\Rightarrow Pr\left(X_{ij} \geq \frac{3d}{4}\right) \leq \exp\left(-\frac{d}{8}\right)$$

By using the union bound, we have:

$$Pr\left(\exists i,j \in [k] : i < j \wedge X_{ij} \geq \frac{3d}{4}\right) \leq \binom{k}{2} \exp\left(-\frac{d}{8}\right) =$$

$$= \binom{k}{2} \exp\left(-\frac{k^3}{8}\right) < 1 - e^{-1}, \text{ for } k \geq 3$$

As for item 3, we first fix some $I \subseteq [d]$ with $|I| = l$ and some assignment $\sigma \in \sigma(I)$, where $\sigma(I)$ is the set of all assignments to I. We define the random variables $Z(i,\sigma)$ for all $i \in [k]$ and set them equal to 1 if μ^i agrees with σ and 0 otherwise. We also define the random variable $Z(\sigma) = \sum_{i\in[k]} Z(i,\sigma)$. Notice that for any $i \in [k]$, $Pr\left(Z(i,\sigma) = 1\right) = 2^{-l}$ and as a result $\mathbb{E}[Z(\sigma)] = k2^{-l}$. Bear in mind that $Z(i,\sigma)$ are independent random variables, so we can apply Hoeffding's inequality:

$$Pr\left(|Z(\sigma) - k2^{-l}| \geq k\epsilon\right) \leq 2\exp\left(-2k\epsilon^2\right) \Rightarrow$$

$$\Rightarrow Pr\left(Z(\sigma) \leq k\left(2^{-l} - \epsilon\right)\right) \leq 2\exp\left(-2k\epsilon^2\right)$$

Since the number of distinct subsets of $[d]$ of size l is $\binom{d}{l}$ and for each of these subsets, the number of assignments to these subsets is 2^l, by using the union bound, we obtain:

$$Pr\left(\exists I \subseteq [d]\ \exists \sigma \in \sigma(I) :\ Z(\sigma) \leq k\left(2^{-l} - \epsilon\right)\right) \leq \binom{d}{l} 2^{l+1} \exp\left(-2k\epsilon^2\right) \leq$$

$$\leq \exp\left(3l\ln(k) + 2l + 1 - 2\epsilon^2 k\right) < e^{-1}, \text{ for } \epsilon \geq \frac{\ln k}{\sqrt{k}} \text{ and } l \leq \frac{\ln(k)}{50}$$

where for the last inequality we used that $\binom{d}{l} \leq \left(\frac{ed}{l}\right)$. Since $e^{-1} < 1$, we have completed the proof. □

4.6 Lower Bounds for all ℓ_p objectives

We have described an explainable clustering instance with a very high k-median cost, but what about other center-based objectives? In this section, we present a theorem that offers us the current best lower bounds of the price of explainability for every l_p objective for $p \geq 2$, including the k-means objective.

Theorem 4.6.1 (O. Svensson, B. Gamlath, X. Jia, A. Polak). *For any $p \geq 1$ there exist some $k \geq 2$ and $d \geq 1$ and a k-clustering instance $((\mathbb{R}^d, \delta_p), X, \mathcal{H}_p)$, such that any explainable clustering algorithm $\mathcal{A}$ that outputs a clustering induced by a threshold tree T has a cost:*

$$cost_p(T) \geq \Omega(k^{p-1})\, OPT_p(X)$$

where $OPT_p(X)$ is the cost of the optimal unconstrained k-clustering with the ℓ_p objective and δ_p metric.

Notice that for k-means explainable clustering, the theorem above implies an $\Omega(k)$ lower bound for the price of explainability.
To create this instance, we follow the steps below:

- We pick a prime number m and set the number of dimensions $d = m(m-1)$ and the number of clusters $k = m$.

- We pick k centers $\mu^1, \mu^2, ..., \mu^k$ whose coordinates are given by d functions f_i, $i \in [d]$. More specifically, the function f_i determines the projection of any center to dimension i like this: $\mu_i^j = f_i(j)$. The set $\{f_i, i \in [k]\}$ is the set of all functions over $\mathbb{Z}_m$ with non-zero slope, thus:

$$f_i(x) = (a_i\, x + b_i) \text{ where:}$$

$$a_i = \left(1 + \left\lfloor \frac{i}{m} \right\rfloor\right)\ mod\, m \text{ and } b_i = i\ mod\, m$$

- Similarly to the instance in the previous section, for each $j \in [k]$ we define $B_j = \{\mu^j + c\, e^j :\ c \in \{\pm 1\},\ i \in [d]\}$. Then, the clustering instance is going to be:

$$X = \bigcup_{j \in k} B_j$$

This instance has two crucial properties that are sufficient to prove Theorem 4.6.1:

1. Any two centers are at distance $\Delta = \Theta(d^{\frac{1}{p}} k)$
2. Any threshold cut that separates two centers $\mu^1, ..., \mu^k$ separates also some two points from the same B_j.

Notice that the distance of each point in B_j from μ^j is exactly 1, thus $OPT_p(X)$ is at most $2kd$. Since at least one threshold cut has to be non-trivial, we conclude that there will exist a leaf of the threshold tree that will contain two points from different sets B_j. However, due to the triangle inequality, the distance between these two points will be at least $\Delta - 2$, which means that $cost_p(T)$ will be at least $\Omega(\Delta^p) = \Omega(k^p d) = \Omega(k^{p-1}) OPT_p(X)$.

Chapter 5

Well-clusterability vs. price of explainability

Motivated by the improved algorithms for (unconstrained) clustering under stability assumptions, we want to study whether there exist explainable clustering algorithms that perform better if their input space is restricted to only well-clusterable instances. More specifically, we want to know if the Price of Explainability is reduced, if the input clustering instances are either a-center stable or a-perturbation stable, and extract the relationship between the parameter a and PoE.
In order to state and analyze our results, it is useful to define the notion of a-separation of a clustering.

Definition 5.0.1. *Let $((Y, \delta), X, \mathcal{H})$ be a k-clustering instance and centers $M = \{\mu^1, \mu^2, ..., \mu^k\} \subseteq Y$ that induce the clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$, where μ^i is the center of C_i, $i \in [k]$, and a real number $a \geq 1$. The clustering $\mathcal{C}$ with centers M, denoted by the pair $(\mathcal{C}, M)$, is a-separated, if for every $i \in [k]$, $j \in [k] \setminus \{i\}$ and $x \in C_i$, it holds that:*

$$\delta(x, \mu^j) > a\delta(x, \mu^i)$$

Note that a clustering instance is a-center stable if and only if any optimal clustering is a-separated.

5.1 Explainable clustering under a-center stability

5.1.1 For sufficiently well-separated instances PoE becomes constant

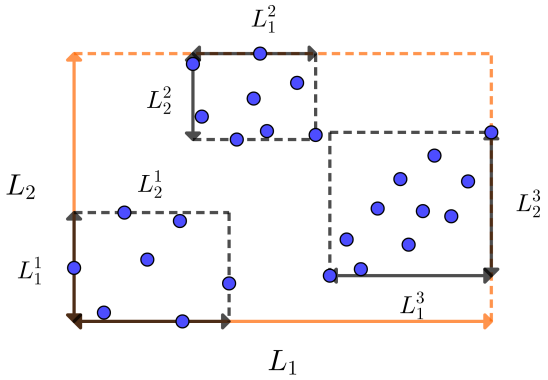
In this section, we prove that indeed there exists a large enough a such that the price of explainability is reduced if the input instances are a-center stable. In fact, we find a sufficiently large a such that the IMM algorithm that we described in Chapter 4 outputs the reference clustering unaltered, thus achieving a clustering cost that is at most a constant factor of the optimal cost.

Theorem 5.1.1. *Let the input of the IMM algorithm be a k-clustering instance $((\mathbb{R}^d, \delta_p), X, \mathcal{H}_p)$, for some $k, d \in \mathbb{N}^*$, $k \geq 2, p \geq 1$ and suppose that the reference centers $M = \{\mu^1, \mu^2, ..., \mu^k\}$ returned in line 1 of the algorithm induce a clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$, such that the pair $(\mathcal{C}, M)$ satisfies the a-separation property with $a \geq 2kd^{\frac{1}{p}}$. Then the IMM algorithm will output a threshold tree T that induces the reference clustering.*

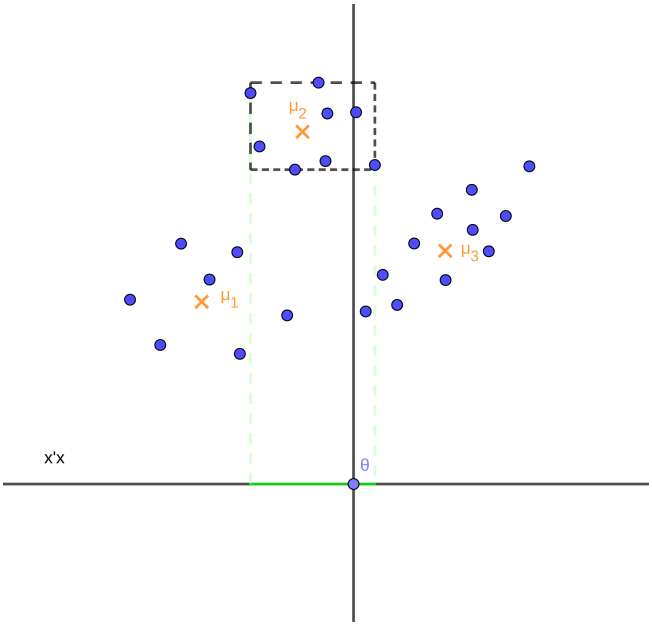
First of all, we are going to show the following lemma that directly implies Theorem 5.1.1.

Lemma 5.1.2. *If the requirements of Theorem 5.1.1 are met, then there exists a threshold cut (i, θ) that makes no mistakes, i.e. it does not separate any point in X from its assigned center in M.*

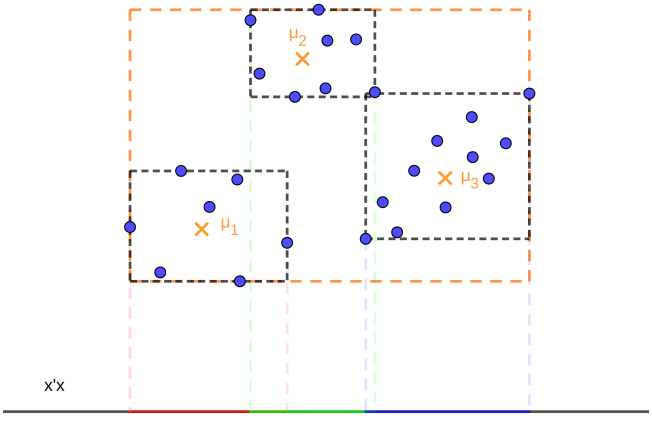
Proof of Lemma 5.1.2.
We consider the bounding boxes B_i of each cluster C_i of the clustering $\mathcal{C}$, as well as the bounding box of the whole instance B. In addition, let I_j^i be the projection of B_i to the j-th dimension for each $i \in [k]$ and $j \in [d]$ and $L_j^i := |I_j^i|$ (similarly we define I_j and L_j for the projections of B).
For the sake of contradiction, we assume that for every $j \in [d]$ and $\theta \in I_j$ the threshold cut (j, θ)



(a) Reference Clustering and Bounding Boxes



(b) Threshold cut (i, θ) that makes some mistake



(c) If all cuts make some mistake, the projection of the bounding box is covered

Figure 5.1: Proof Idea

makes some mistake. We define j^* as follows: $j^* = \arg\max_{j \in [d]} \{L_j\}$. From the assumption above we get that:

$$\forall \theta \in I_{j^*} \; \exists i \in [k] : \; \theta \in I^i_{j^*} \Rightarrow I_{j^*} \subseteq \bigcup_{i \in k} I^i_{j^*} \Rightarrow$$

$$\Rightarrow L_{j^*} \leq \sum_{i=1}^{k} L^i_{j^*}$$

Therefore, there exists $i^* \in [k]$ such that:

$$L^{i^*}_{j^*} \geq \frac{L_{j^*}}{k} \tag{5.1}$$

Next, we set $L = \left(\sum_{i=1}^{d} (L_j)^p \right)^{\frac{1}{p}}$, i.e. the diameter of the bounding box B, which is the maximum distance of any two points inside B and it has the following property:

$$L = \left(\sum_{i=1}^{d} (L_j)^p \right)^{\frac{1}{p}} \leq \left(\sum_{i=1}^{d} (L_{j^*})^p \right)^{\frac{1}{p}} \leq d^{\frac{1}{p}} L_{j^*} \tag{5.2}$$

We need one more inequality to complete the proof. For any $i \in [k]$, if D_i is the diameter of the cluster $C_i \in C$, i.e.:

$$D_i := \max_{\boldsymbol{x}, \boldsymbol{y} \in C_i} \{ \|\boldsymbol{x} - \boldsymbol{y}\|_p \}$$

then:

$$\forall j \in [d] : \; D_i \geq L^i_j \tag{5.3}$$

To see why this is true, we consider the bounding box B_i and for some $j \in [d]$ we choose $\boldsymbol{x}, \boldsymbol{y} \in C_i$ such that $|x_j - y_j| = L^i_j$. We know that those points exist by definition of the bounding box of the cluster. Then, it is apparent that:

$$D_i \geq \|\boldsymbol{x} - \boldsymbol{y}\|_p \geq |x_j - y_j| = L^i_j$$

Now, let a point $\boldsymbol{x} \in C_{i^*}$ such that $\|\boldsymbol{x} - \boldsymbol{\mu^{i^*}}\|_p = R_{i^*}$ and a center $\boldsymbol{\mu'} \in M \setminus \{\boldsymbol{\mu^{i^*}}\}$. By combining (5.1), (5.2), (5.3) and using the a-separation property and the fact that $a \geq 2kd^{\frac{1}{p}}$, we conclude that:

$$\|\boldsymbol{x} - \boldsymbol{\mu'}\|_p > aR_{i^*} \geq a\frac{D_{i^*}}{2} \geq a\frac{L^{i^*}_{j^*}}{2} \geq$$

$$\geq a\frac{L_{j^*}}{2k} \geq \frac{a}{2kd^{\frac{1}{p}}} L \Rightarrow$$

$$\Rightarrow \|\boldsymbol{x} - \boldsymbol{\mu'}\|_p > L$$

and we have reached a contradiction, as both $\boldsymbol{x}$ and $\boldsymbol{\mu'}$ are inside the bounding box B and L is the greatest distance between any two points in B. □

Now that we have shown Lemma 5.1.2, we can easily prove Theorem 5.1.1, as follows:

Proof of Theorem 5.1.1.
We will prove the theorem by induction in the number of clusters k.
Base Case $(k = 2)$
According to the Lemma 5.1.2 there exists a threshold cut (i, θ) that makes no mistakes. The IMM algorithm chooses at each iteration a cut that makes the fewest mistakes, thus it will opt for a cut that makes 0 mistakes as well. Therefore the threshold tree induces the reference clustering.

Inductive step
Let $((\mathbb{R}^d, \delta_p), X, \mathcal{H}_p)$ be a k-clustering instance with $k > 2$. By Lemma 5.1.2 we know that the IMM algorithm will choose a cut (i, θ) that makes no mistakes to split the root node. We set:

$$X_L = \{\boldsymbol{x} \in X : \ x_i \leq \theta\}, \ X_R = X \setminus X_L$$

$$M_L = \{\boldsymbol{\mu} \in \mu : \ \mu_i \leq \theta\}, \ \mu_R = M \setminus \{\mu_L\}$$

Next, the IMM algorithm will produce a threshold tree for each of the two children of the root node and attach these trees to the root; the root node of the left subtree will contain X_L along with their correct centers M_L and the root of the right subtree will contain X_R and M_R. Therefore, the left subproblem that the algorithm solves recursively is the k_L-clustering instance $((\mathbb{R}^d, \delta_p), X_L, \mathcal{H}_p)$, where $k_L = |M_L|$ with reference centers M_L. Note that $k_L < k$, because the IMM algorithm chooses only non-trivial threshold cuts, i.e. cuts that separate some two reference centers. Hence, if C_L is the induced clustering of X_L by the centers M_L, then (C_L, M_L) satisfies the a-separation property with $a \geq 2kd^{\frac{1}{p}} \geq 2k_L d^{\frac{1}{p}}$ (since the pair (C, M) satisfied this property). By induction, the algorithm will solve the left subproblem without making any mistakes. Similarly, we can show that it will also solve the right subproblem without making any mistakes. Therefore, the tree returned by the algorithm induces the reference clustering. □

The theorem 5.1.1 directly implies the following corollary:

Corollary 5.1.2.1. *Let a k-clustering instance $((\mathbb{R}^d, \delta_p), X, \mathcal{H}_p)$, with $k, d \in \mathbb{N}^*$, $k \geq 2, p \geq 1$, that satisfies the a-proximity property with $a \geq 12kd^{\frac{1}{p}}$. There exists a polynomial-time explainable clustering algorithm that returns a threshold tree T, which induces a constant-approximation clustering (PoE = $O(1)$).*

Proof of Corollary 5.1.2.1.
In line 1. of IMM, we can use an algorithm that computes the optimal *discrete unconstrained* clustering in polynomial time, given an a-center stable clustering instance, to obtain the reference centers $M = \{\mu^1, \mu^2, ..., \mu^k\}$ by using algorithm 2 (in the discrete clustering problem, we minimize the same objective, but the cluster centers can only be a subset of X).
The clustering C induced by M is such, that (C, M) satisfies the $\frac{(a-1)^2}{2a}$-separation property (due to item 4. of Lemma 3.1.1), so the conditions of theorem 5.1.1 are met, if $a \geq 12kd^{\frac{1}{p}}$ (we have used that $\frac{(a-1)^2}{2a} \geq \frac{a}{6}$, for $a \geq 1$), and the explainable clustering will be the same as the reference clustering. Since the optimal discrete clustering is at most a constant factor of the optimal continuous clustering (if we assume that p is constant), the resulting clustering is an $O(1)$-approximation of the optimal unconstrained solution, thus $PoE = O(1)$. □

5.1.2 a-center stability of hard instances

In the previous section, we saw that indeed there exists a large value of a such that PoE drops dramatically for a-center stable input instances. The problem is, however, that this value is so large $\left(\Omega\left(kd^{\frac{1}{p}}\right)\right)$, that it makes the a-center stability assumption impractical. That's why we either want to significantly improve this upper bound or find a matching lower bound for the value of a needed to increase the performance of explainable clustering algorithms and give up on this idea. Unfortunately, we manage to prove the second case, by showing that the hard instances created by Dasgupta et al. and Svensson et al. were already a- center stable for very large values of a.
As far as the hard instance of Svensson et al. is concerned, the optimal k-clustering of this instance

(for any $p \geq 1$) is $\mathcal{C} = \{B_j : j \in [k]\}$ (see Section 4.6) for definitions). By property 1. of the instance and the triangle inequality, we have that the distance between each point $x \in B_j$ and some $\mu^{j'}$ with $j' \neq j$ is:

$$\|x - \mu^{j'}\|_p \geq \Delta - 1$$

since $\|x - \mu^j\|_p = 1$. Consequently, because $\Delta = \Omega\left(kd^{\frac{1}{p}}\right)$, the instance is $\Omega\left(kd^{\frac{1}{p}}\right)$-center stable, for any fixed p. Note that this value of a matches (in terms of order of magnitude) our result for the value needed for IMM to make no mistakes. Also, remember that for $p \geq 2$, $\Omega(k^{p-1})$ is the best lower bound of PoE that has been discovered yet and the upper bound given by Theorem 4.4.3 is almost tight ($O(k^{p-1}\log^2 k)$). *In other words, if we want the lower bound of PoE to drop significantly for a-center stable input instances compared to the general case, an impractically large value of a is necessary.*

On the other hand, for $p = 1$, Svensson's lower bound for the price of explainability is $\Omega(1)$, which is not the current best lower bound for the PoE of explainable k-median clustering. Therefore, we study the a-proximity property of Dasgupta's lower bound in [57].

Consider the k-median clustering instance described in section 4.5. We can easily see that the unique optimal clustering is $\mathcal{C} = \{X^{\mu^i},\ i \in [k]\}$ with optimal centers $M = \{\mu^i,\ i \in [k]\}$. Moreover, consider some $x \in X^\mu$ and $\mu' \in M \setminus \{\mu\}$. Notice that x is at distance 1 from its center μ and that for each $j \in [d]$ where $\mu_j \neq \mu'_j$ is holds that $|\mu_j - \mu'_j| = 2$, as $\mu_j, \mu'_j \in \{\pm 1\}$. From item 2. of Claim 4.5.1 we know that:

$$|j \in [d] :\ \mu_j \neq \mu'_j| \geq \frac{d}{4}$$

hence:

$$\|\mu - \mu'\|_1 \geq \frac{d}{2}$$

As a result:

$$\|x - \mu'\|_1 \geq \left(\frac{d}{2} - 1\right)\|x - \mu\|_1 > \frac{d}{3}\|x - \mu\|_1$$

which means that this instance is $\frac{d}{3}$-center stable, which is $\Omega(d)$-center stable (the same holds for the hard instance by Esfandiari et al. in [64]). Although this does not match our result, it is still dependent on the number of dimensions of the instance, which deems the stability assumption impractical.

5.2 Explainable k-median clustering under a-metric perturbation stability

We have seen that explainable clustering under a-center stability is not easier, but we shouldn't give up just yet, as there could be other reasonable stability assumptions, independent from the above or stronger, which imply a low price of explainability. It is time to study one of the most commonly used stability assumptions, the Bilu-Linial stability, in the context of explainable clustering. To this end, we develop a tool that helps us analyze the perturbation stability of a given instance, assuming that it satisfies certain conditions.

By Theorem 3.2.1 we already know that (when $X = Y$), a-metric perturbation stability implies a-center stability. Nevertheless, what we really need is the opposite direction: does a-center stability imply a'-metric perturbation stability for some $a' \leq a$? If this were true, then we would be able to determine a lower bound for the perturbation stability of an instance by checking its center stability, which is easier to identify (at least for the hard instances of explainable clustering that we consider in this project). Unfortunately, this is clearly not the case, because for any $a > 1$ and $1 < a' < a$ there exists an a-center stable k-clustering instance, such that we can find some a'-metric perturbation with a different optimal clustering than that of the original instance.

It is, therefore, necessary to assume additional conditions, that in combination with center stability

imply metric perturbation stability. Note that the k-median clustering instance of Theorem 4.5.1 is not only a-center stable for a large value of a, but all the clusters in the optimal clustering have exactly the same cost, as they are essentially the same cluster, moved to different positions. Intuitively, this has to be a perturbation stable instance and that's what we will try to prove. More specifically, we prove the following theorem.

Theorem 5.2.1. *Let $\mathcal{I} = ((Y, \delta), X, \mathcal{H}_1)$ be a k-median clustering instance with $k \in \mathbb{N}^*$. Suppose there exist centers $M = \{\mu^1, \mu^2, ..., \mu^k\}$ that induce a clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ of $\mathcal{I}$ and real numbers $a \geq 1$, $\beta \geq 1$ and $H > 0$, such that:*

1. *The pair $(\mathcal{C}, M)$ is $a-$ separated, with $a \geq 2\beta + 9$.*
2. *For every $i \in [k]$: $\frac{H}{\beta} \leq \sum_{x \in C_i} \delta(x, \mu^i) \leq H$.*

Then $\mathcal{I}$ is γ-metric perturbation stable*, where:*

$$\gamma = \frac{-9 + \sqrt{3^4 + 8a\beta}}{4\beta}$$

The above theorem directly implies the following:

Corollary 5.2.1.1. *Let (X, δ) be a k-median clustering instance with $X \subseteq \mathbb{R}^d$ and a metric $\delta : \mathbb{R}^d \to \mathbb{R}^d$ with $d, k \in \mathbb{N}^*$. Suppose there exists a k-clustering $\mathcal{C} = \{C_i, i \in [k]\}$ of (X, δ) with centers $\boldsymbol{\mathcal{M}}$, that satisfies the following properties for some **constant** $\epsilon \geq 1$ and $H > 0$:*

1. *a-separation for some $a \geq 1$*
2. *for every $i \in [k]$: $\frac{H}{\epsilon} \leq \sum_{x \in C_i} \delta(x, \mu_i) \leq H$*

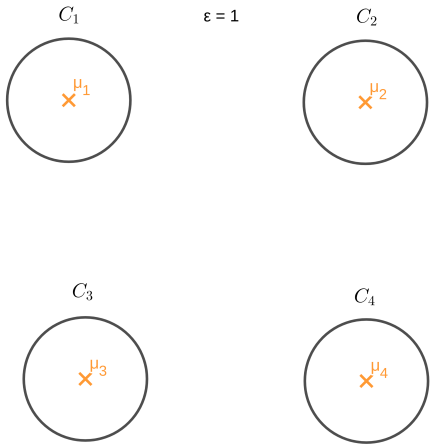
where $\mu_i \in \boldsymbol{\mathcal{M}}$ is the center of C_i for $i \in [k]$. Then (X, δ) is $\Omega(\sqrt{a})$-perturbation stable.

Let's assume for the moment that Corollary 5.2.1.1 is indeed true and apply it to the hard instance of [57]. As we have seen in the previous section, this instance satisfies the a-proximity property with $a = \Omega(d)$, hence the optimal clustering $C = \{X^{\mu^i} : i \in [k]\}$ with centers $M = \{\mu^i : i \in [k]\}$ is $\Omega(d)$-separated. Furthermore, since every point in X^{μ^i} for any $i \in [k]$ is at distance 1 from μ^i and there are exactly d points in each cluster, we can see that all clusters in the optimal clustering have exactly the same cost, so it satisfies the second condition of the Corollary 5.2.1.1 as well, with $\epsilon = 1$. As a result, this instance is at least $\Omega(\sqrt{d})$-metric perturbation stable.
Before we get into the proof of Theorem 5.2.1, we provide a proof sketch that showcases the most important ideas of the proof.

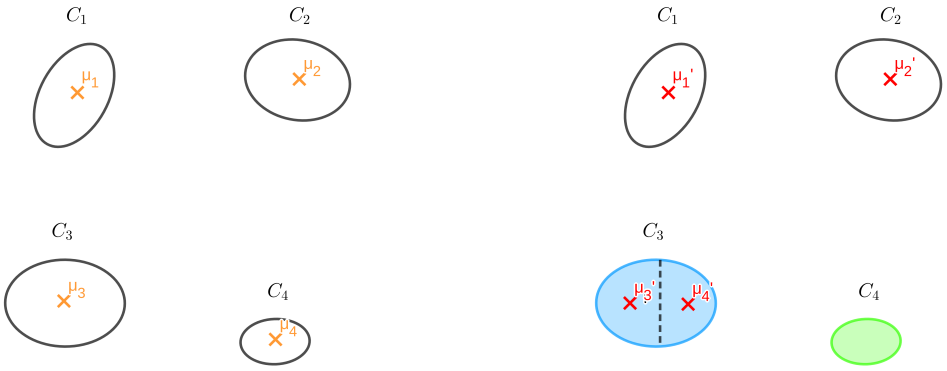
Proof sketch of Theorem 5.2.1 and Lemma 5.2.2.
To prove that a k-clustering instance is γ-metric perturbation stable, we want to show that any γ-metric perturbation of the starting instance will have the same optimal clustering as this instance. To do this, we will make use of two important properties of the resulting instance after the perturbation:

1. The clustering C with centers M continues to be well-separated after the γ-metric perturbation. More specifically, if it was α separated before, it is $\frac{\alpha}{\gamma}$ separated after.

2. The cost of each cluster C_i in C with its center μ^i in M does not change much after a γ-metric perturbation. To be more precise, if the clusters had exactly the same cost before the perturbation, their costs are γ-close after the perturbation, i.e. no cluster can cost more than γ times more than any other cluster.



(a) Optimal Cluster centers M before the Perturbation



(b) Clustering with centers M after the perturbation

(c) Optimal Clustering after the Perturbation

Figure 5.2: Proof Idea

As a result, after the γ-metric perturbation, the clustering C remains well-separated and all its clusters have roughly the same cost. To complete the proof, we will show Lemma 5.2.2, which is the heart of our proof. It essentially states that if there exists a clustering C with centers M that is very well separated and all of its clusters have roughly the same cost, then C is the unique optimal clustering. More formally, we prove that if a clustering is a' separated and the costs of its clusters are γ'-close, then it is the induced clustering is optimal, if (roughly) $a' \geq 2\gamma'$, and since, in our case, $a' = \frac{a}{\gamma}$ and $\gamma' = \gamma$, we get $\gamma \leq \sqrt{\frac{a}{2}}$.
Suppose, for the sake of contradiction, that this is not the case. Then there exists a set of optimal M' that induce the optimal clustering $C' \neq C$, like in Figure 5.2c. We identify three types of clusters of C:

1. Clusters that have only one optimal center in their neighborhood, like C_1 and C_2. These clusters with centers from M' (red) contribute roughly the same cost as with centers from M (orange).

2. Clusters that have two or more centers in their neighborhood, like C_3. These clusters may

cost less if the clustering centers are M'.

3. Clusters that have no centers in their neighborhood, like C_4. These clusters cost much more in the red clustering, than in the orange clustering, because all of their points are assigned to a center that does not belong in their neighborhood.

To reach a contradiction we need to prove that the optimal clustering after the perturbation (red) is more expensive than the optimal clustering before the perturbation (orange). The only way that red is cheaper than orange is if the cost that we avoid due to type 2 clusters (C_3) in the red clustering is more than the extra cost we have to pay for type 3 clusters (C_4). This could be possible only if the cost of C_4 in the orange clustering is negligible compared to C_3. However, since we know that all clusters in the orange clustering have roughly the same cost and using the fact that type 3 clusters are as much or even more than type 2 clusters, we reach a contradiction.

□

Now that we have seen the basic ideas of the proof, it is easier to follow the formal proof, which we give below.
In the first place, let's prove Lemma 5.2.2, which is the heart of the proof of Theorem 5.2.1.

Lemma 5.2.2. *Let $\mathcal{I} = ((Y, \delta), X, \mathcal{H}_1)$ be a k-median clustering instance with $k \in \mathbb{N}^*$. Suppose there exist centers $M = \{\mu^1, \mu^2, ..., \mu^k\}$ that induce a clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ of $\mathcal{I}$ and real numbers $a' \geq 1$, $\beta' \geq 1$ and $H > 0$, such that:*

1. *The pair $(\mathcal{C}, M)$ is $a'-$ separated, with $a' \geq 2\beta' + 9$.*
2. *For every $i \in [k]$: $\frac{H}{\beta'} \leq \sum_{x \in C_i} \delta(x, \mu^i) \leq H$.*

*Then, clustering $\mathcal{C}$ is the **unique** optimal clustering of $\mathcal{I}$.*

Proof of Lemma 5.2.2.
We start with some notation. Let $S = \{\sigma^1, \sigma^2, ..., \sigma^k\}$ be a set of centers where σ^i is the optimal center for the cluster C_i of the clustering $\mathcal{C}$ (remember μ^i might not be the optimal center for C_i). Note that S induces $\mathcal{C}$, as we will explain later. For the sake of contradiction, we assume that there exist centers $M' = \{\mu'^1, \mu'^2, ..., \mu'^k\}$ that induce a clustering $\mathcal{C}' \neq \mathcal{C}$ with $cost(M') \leq cost(M)$. Let $h : X \to M'$ be the function that assigns each $x \in X$ to its optimal (closest) center in M'.
As in the proof sketch, we will consider the neighborhood of each cluster. To describe this neighborhood, we use the (closed) balls $A_1, A_2, ..., A_k$ and $B_1, B_2, ..., B_k$ that are defined as follows:

$$A_i := B(\mu^i, (\beta' + 4)R_i)$$

$$B_i := B(\mu^i, (\beta' + 2)R_i)$$

where R_i is the radius of cluster C_i, that is: $R_i = \max_{x \in C_i} \delta(x, \mu^i)$.
Now consider 3 types of sets (Figure 5.3):

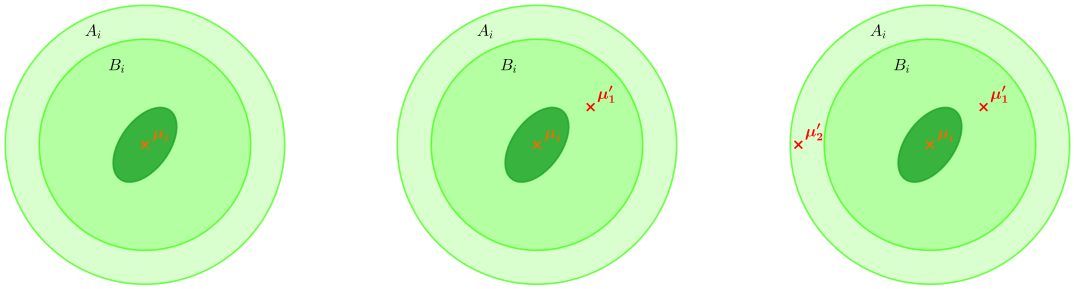
$$I_0 = \{i \in [k] : \ B_i \cap M' = \emptyset\}$$

$$I_1 = \{i \in [k] : \ |B_i \cap M'| = 1 \ \wedge \ |A_i \cap M'| = 1\}$$

$$I_2 = [k] \setminus (I_0 \cup I_1)$$

The first thing we notice is that I_0, I_1, and I_2 form a partition of $[k]$. In addition, for any $i \in I_2$ it holds that $|A_i \cap M'| \geq 2$, because otherwise $i \in I_0 \cup I_1$. By rewriting the optimal cost, we have:

$$cost(M') = \sum_{i \in I_0} cost(C_i, M') + \sum_{i \in I_1} cost(C_i, M') + \sum_{i \in I_2} cost(C_i, M') \Rightarrow$$



(a) I_0: Clusters with no optimal centers in their neighborhood

(b) I_1: Clusters with only 1 optimal in their neighborhood

(c) I_2: Clusters with at least 2 optimal in their neighborhood

Figure 5.3: The 3 types of clusters in the Proof of 5.2.2

$$\Rightarrow cost(M') \geq \sum_{i \in I_0} cost(C_i, M') + \sum_{i \in I_1} cost(C_i, M') \tag{5.4}$$

Next, we bound the 2 terms in (5.4).
For the first term, if $i \in I_0$, by triangle inequality we know that for every $x \in C_i$:

$$\delta(x, h(x)) \geq \delta(\mu^i, h(x)) - \delta(\mu^i, x) > (\beta' + 2)R_i - R_i =$$

$$= (\beta' + 1)R_i \geq (\beta' + 1)\delta(x, \mu^i)$$

Let's assume that $|I_0| > 0$. Then:

$$\sum_{i \in I_0} cost(C_i, M') = \sum_{i \in I_0} \sum_{x \in C_i} \delta(x, h(x)) > (\beta' + 1) \sum_{i \in I_0} \sum_{x \in C_i} \delta(x, \mu^i) \Rightarrow$$

$$\Rightarrow \sum_{i \in I_0} cost(C_i, M') > (\beta' + 1) \sum_{i \in I_0} cost(C_i, M) \tag{5.5}$$

To bound the second term, we notice that all $x \in C_i$ for any $i \in I_1$ are assigned to the same center in the optimal clustering. To see why this is the case, we consider the unique $\mu' \in M' \cap B_i$ and any other center $\mu'' \in M' \setminus A_i$. Then, for any $x \in C_i$:

$$\delta(x, \mu') \leq \delta(x, \mu^i) + \delta(\mu^i, \mu') \leq R_i + (\beta' + 2)R_i \leq (\beta' + 3)R_i$$

$$\delta(x, \mu'') \geq \delta(\mu^i, \mu'') - \delta(\mu^i, x) > (\beta' + 4)R_i - R_i = (\beta' + 3)R_i$$

Thus:

$$\sum_{i \in I_1} cost(C_i, M') = \sum_{i \in I_1} \sum_{x \in C_i} \delta(x, h(x)) \geq \sum_{i \in I_1} \sum_{x \in C_i} \delta(x, \sigma_i) \Rightarrow$$

$$\Rightarrow \sum_{i \in I_1} cost(C_i, M') \geq \sum_{i \in I_1} cost(C_i, S) \tag{5.6}$$

where we have used the fact that S is the set of optimal centers for $\mathcal{C}$ in the clustering setting.
To complete the proof, we will need the following auxiliary lemma.

Lemma 5.2.3. *It is true that* $|I_0| \geq |I_2|$.

Proof of Lemma 5.2.3. Since I_0, I_1, I_2 form a partition of $[k]$ we get that:

$$|I_0| + |I_1| + |I_2| = k \tag{5.7}$$

In addition, by item 2. of Lemma 3.1.1, we get that for $i, j \in [k]$ with $i \neq j$:

$$\delta(\mu^i, \mu^j) > (a-1)\max\{R_i, R_j\} \geq \frac{a-1}{2}(R_i + R_j) \geq (\beta' + 4)(R_i + R_j)$$

since $a \geq 2\beta' + 9$, so $A_i \cap A_j = \emptyset$. This also means that $(M' \cap A_i) \cap (M' \cap A_i) = \emptyset$ for $i \neq j$. Hence:

$$k \geq \left| \bigcup_{i \in I_1 \cup I_2} (M' \cap A_i) \right| = \sum_{i \in I_1} |M' \cap A_i| + \sum_{i \in I_2} |M' \cap A_i| \geq$$
$$\geq |I_1| + 2|I_2| \geq k - |I_0| + |I_2|$$

where for the last inequality we used (5.7). Consequently:

$$|I_0| \geq |I_2|$$

□

In conclusion, by combining (5.4), (5.5), (5.6) and Lemma 5.2.3 we obtain:

$$cost(M') > \sum_{i \in I_0} cost(C_i, M) + \sum_{i \in I_1} cost(C_i, S) + \beta' \sum_{i \in I_0} cost(C_i, M) \geq$$
$$\geq \sum_{i \in I_0} cost(C_i, S) + \sum_{i \in I_1} cost(C_i, S) + |I_0|H \geq$$
$$\geq \sum_{i \in I_0} cost(C_i, S) + \sum_{i \in I_1} cost(C_i, S) + |I_2|H \geq$$
$$\geq \sum_{i \in I_0} cost(C_i, S) + \sum_{i \in I_1} cost(C_i, S) + \sum_{i \in I_2} cost(C_i, M) \geq$$
$$\geq \sum_{i \in I_0} cost(C_i, S) + \sum_{i \in I_1} cost(C_i, S) + \sum_{i \in I_2} cost(C_i, S)$$
$$= cost(S)$$

where once again every time we substitute M with S we make use of the optimality of S.
We have reached a contradiction. Therefore, it has to be $|I_0| = |I_2| = 0$ for any optimal clustering. As a result, $I_1 = [k]$ and thus M' induces $\mathcal{C}$, so $\mathcal{C} = \mathcal{C}'$, which is again a contradiction. This is why $\mathcal{C}$ is the unique optimal clustering of $\mathcal{I}$. □

Now that we have proved 5.2.2, we can finally complete the proof of 5.2.1.

Proof of Theorem 5.2.1.
For some $\gamma \geq 1$ we consider a γ-perturbation $\mathcal{I}' = ((Y, \delta'), , \mathcal{H}_1)$ of $\mathcal{I}$. We observe that, for any $i, j \in [k]$ with $i \neq j$ and a data point $x \in C_i$, it holds that:

$$\delta'(x, \mu^j) \geq \frac{1}{\gamma}\delta(x, \mu^j) > \frac{a}{\gamma}\delta(x, \mu^i) \geq \frac{a}{\gamma}\delta'(x, \mu^i)$$

where we have used the a-separation property of $(\mathcal{C}, M)$ and the definition of a metric perturbation. Therefore, if we set $a' = \frac{a}{\gamma}$, we get that the clustering $(\mathcal{C}, M)$ is a'-separated after the perturbation. Specifically, we set:

$$\gamma = \frac{\sqrt{3^4 + 8a\beta} - 9}{4\beta}$$

We observe that, if we also set $\beta' = \beta\gamma$, then:

1. $\beta' \geq 1$ for $a \geq 2\beta + 9$

2. $a' = 2\beta' + 9 \geq 1$.

3. for any $i \in [k]$:

$$\frac{H}{\beta'} = \frac{H}{\beta\gamma} \leq \sum_{x \in C_i} \frac{1}{\gamma}\delta(x, \mu^i) \leq \sum_{x \in C_i} \delta'(x, \mu^i) \leq \sum_{x \in C_i} \delta(x, \mu^i) \leq H$$

As a result, the centers M satisfy the conditions of Lemma 5.2.2, therefore $\mathcal{C}$ is the *unique* optimal k-clustering of the perturbation. In conclusion, we have shown that the optimal k-clustering $\mathcal{C}$ of the initial k-median instance is also the *unique* optimal k-clustering of any γ-metric perturbation of this instance, for $\gamma = \frac{\sqrt{3^4+8a\beta}-9}{4\beta}$, therefore $\mathcal{I}$ is $\left(\frac{\sqrt{3^4+8a\beta}-9}{4\beta}\right)$*-metric perturbation stable* for $a \geq 2\beta + 9$. □

5.3 Discussion of the Results

As it turns out, explainable clustering algorithms do not improve under stability assumptions, at least those studied above.

As we have explained in the introduction, perturbation stability is considered by many to be a restrictive assumption and it is not clear if practical instances are a-perturbation stable, even for very small values of a such as 2. In addition, notice that for a as small as 2 there exist algorithms that solve k-clustering problems optimally in polynomial time. Therefore, it is safe to say that a center-stability or perturbation-stability assumption is considered practical if a does not depend either on the number of clusters k or the number of dimensions d of the data set. Besides, in most clustering applications, d is quite large.

Nevertheless, Theorem 5.1.1 requires $a = \Omega\left(kd^{\frac{1}{p}}\right)$, so that the IMM algorithm achieves a constant approximation ratio, under a-center stability. Unfortunately, for every ℓ_p objective with $p \geq 1$, this dependence on the number of dimensions d and the number of clusters k is necessary, as we have shown that the hard instances of [59] are $\Omega\left(kd^{\frac{1}{p}}\right)$-center stable and imply a lower bound of $\Omega(k^{p-1})$ for $p > 1$, while in the k-median case, there is an $\Omega(d)$-center stable instance with lower bound $\Omega(\log k)$ for the Price Of Explainability. In addition, there are almost optimal algorithms that achieve $O(k^{p-1}) \log^2 k$ approximation ratios for $p > 1$ and $O(\log k \log\log k)$ for the $p = 1$. As a result, for the Price of Explainability to reduce, we need to assume that the input instances are impractically center stable. Moreover, even for such values of a there exist almost optimal explainable clustering algorithms that do not assume the a-center stability of their inputs.

As far as perturbation stability is concerned, the situation is similar. Although this assumption is more strict than center stability, we prove that the k-median instance of [57] is $\Omega(\sqrt{d})$-metric perturbation stable, so again the stability parameter is dependent on the number of dimensions. However, in my opinion, the proof of this statement comes with an interesting consequence: Corollary 5.2.1.1 is a statement that can be used to determine a lower bound for the perturbation stability of a given instance, which to my knowledge, did not exist before. Unfortunately, the requirements of this Corollary are too strict and give us useful information about the perturbation stability of an instance for large values of a. On the other hand, it would be interesting to study the requirements needed for similar theorems that offer a more useful characterization of the perturbation stability of a given instance. In Figure 5.4 we can see the (approximate) relationship of the stability parameters with the Price of Explainability.

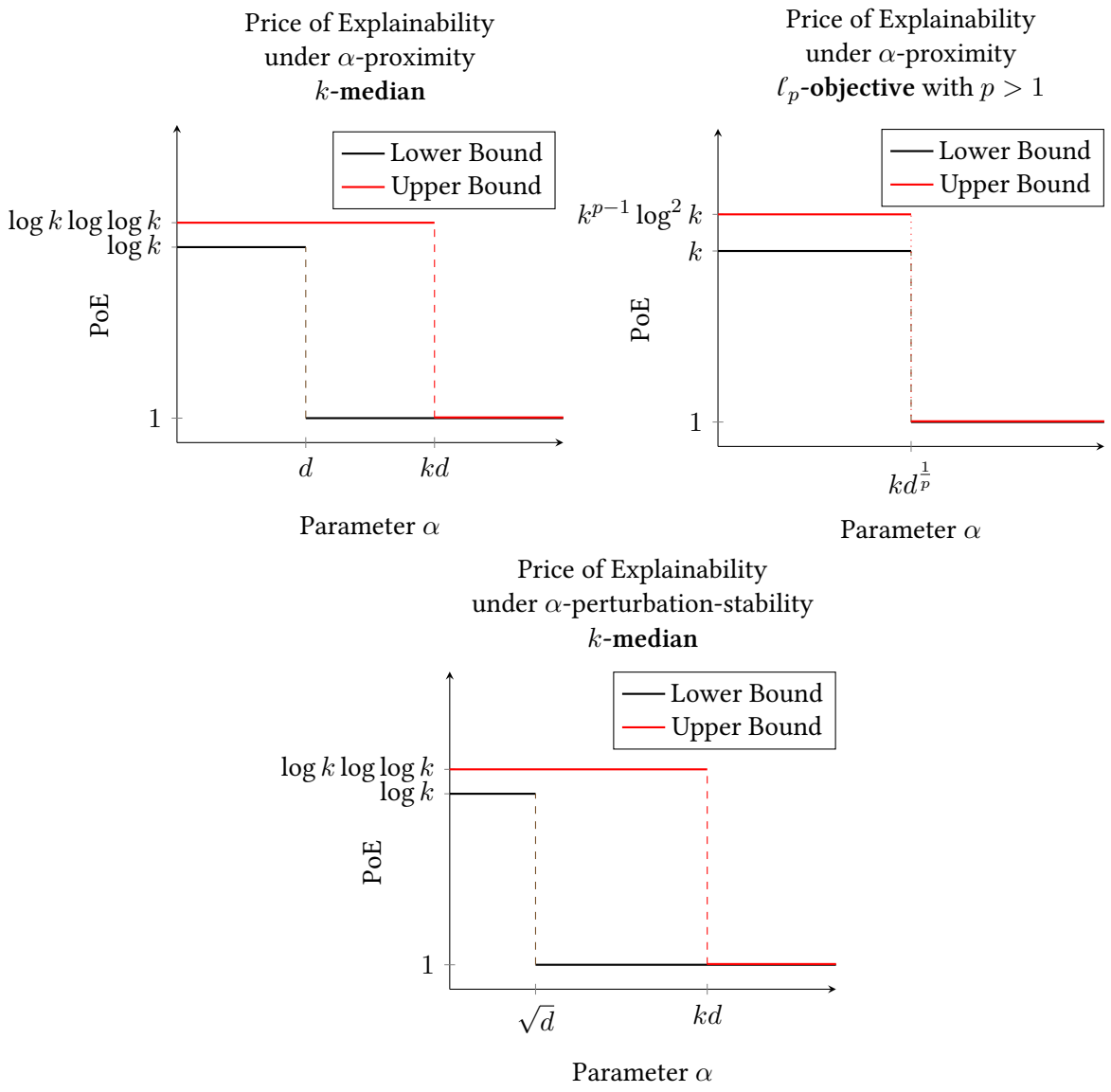


Figure 5.4: Relationship between the stability parameter and PoE

We end this thesis with a high-level interpretation of our results:
Although explainable clustering algorithms might make fewer mistakes on well-separated instances, when they do make a mistake the reassignment cost is enormous, due to this separation. In fact, this is exactly the property that is exploited in most of the hard instances we studied in this thesis.

Bibliography

[1] Jon Louis Bentley and Michael Ian Shamos. *Divide and Conquer for Linear Expected Time*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1977.

[2] Richard M Karp. "Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane." In: *Mathematics of operations research* 2.3 (1977), pp. 209–224.

[3] Greg N Frederickson. "Probabilistic analysis for simple one-and two-dimensional bin packing algorithms." In: *Information Processing Letters* 11.4-5 (1980), pp. 156–161.

[4] Leonid G Khachiyan. "Polynomial algorithms in linear programming." In: *USSR Computational Mathematics and Mathematical Physics* 20.1 (1980), pp. 53–72.

[5] Stuart Lloyd. "Least squares quantization in PCM." In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[6] Leo Breiman et al. *Classification and Regression Trees*. Philadelphia, PA: Chapman & Hall/CRC, Jan. 1984.

[7] David S Johnson. "The NP-completeness column: an ongoing guide." In: *Journal of Algorithms* 5.2 (1984), pp. 284–299.

[8] Nimrod Megiddo and Kenneth J Supowit. "On the complexity of some common geometric location problems." In: *SIAM journal on computing* 13.1 (1984), pp. 182–196.

[9] Mary Inaba, Naoki Katoh, and Hiroshi Imai. "Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering." In: *Proceedings of the tenth annual symposium on Computational geometry*. 1994, pp. 332–339.

[10] Donald E Knuth. *The art of computer programming: Volume 3: Sorting and Searching*. Addison-Wesley Professional, 1998.

[11] Sudipto Guha and Samir Khuller. "Greedy strikes back: Improved facility location algorithms." In: *Journal of algorithms* 31.1 (1999), pp. 228–248.

[12] Richard O Duda, Peter E Hart, and David G Stork. *Pattern Classification*. en. 2nd ed. A Wiley-Interscience publication. Nashville, TN: John Wiley & Sons, Oct. 2000.

[13] Daniel A. Spielman and Shang-Hua Teng. *Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time*. 2001. arXiv: cs/0111050.

[14] Susanne Albers, Lene M Favrholdt, and Oliver Giel. "On paging with locality of reference." In: *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. 2002, pp. 258–267.

[15] Bing Liu, Yiyuan Xia, and Philip S Yu. "Clustering via decision tree construction." In: *Studies in Fuzziness and Soft Computing* 180 (2005), p. 99.

[16] David Arthur and Sergei Vassilvitskii. "Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method." In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE. 2006, pp. 153–164.

[17] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[18] Sanjoy Dasgupta Christos Papadimitriou and Umesh Vazirani. *Algorithms*. 1st ed. McGraw-Hill Education, 2006. ISBN: 9780073523408.

[19] David Arthur and Sergei Vassilvitskii. "K-means++ the advantages of careful seeding." In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. 2007, pp. 1027–1035.

[20] Pierre Geurts et al. "Inferring biological networks with output kernel trees." In: *BMC bioinformatics* 8.2 (2007), pp. 1–12.

[21] Sanjoy Dasgupta. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California ..., 2008.

[22] Konstantinos Koutroumbas and Sergios Theodoridis. *Pattern recognition*. Academic Press, 2008.

[23] Ankit Aggarwal, Amit Deshpande, and Ravi Kannan. "Adaptive sampling for k-means clustering." In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*. Springer. 2009, pp. 15–28.

[24] Daniel Aloise et al. "NP-hardness of Euclidean sum-of-squares clustering." In: *Machine learning* 75 (2009), pp. 245–248.

[25] Pranjal Awasthi, Avrim Blum, and Or Sheffet. *Center-based Clustering under Perturbation Stability*. 2010. eprint: arXiv:1009.3594.

[26] David Arthur, Bodo Manthey, and Heiko Röglin. "Smoothed analysis of the k-means method." In: *Journal of the ACM (JACM)* 58.5 (2011), pp. 1–31.

[27] Matúš Mihalák et al. "On the complexity of the metric tsp under stability considerations." In: *SOFSEM 2011: Theory and Practice of Computer Science: 37th Conference on Current Trends in Theory and Practice of Computer Science, Novỳ Smokovec, Slovakia, January 22-28, 2011. Proceedings 37*. Springer. 2011, pp. 382–393.

[28] Yonatan Bilu and Nathan Linial. "Are stable instances easy?" In: *Combinatorics, Probability and Computing* 21.5 (2012), pp. 643–660.

[29] Yonatan Bilu et al. "On the practically interesting instances of MAXCUT." In: *arXiv preprint arXiv:1205.4893* (2012).

[30] Amit Daniely, Nati Linial, and Michael Saks. "Clustering is difficult only when it does not matter." In: *arXiv preprint arXiv:1205.4891* (2012).

[31] Ricardo Fraiman, Badih Ghattas, and Marcela Svarc. "Interpretable clustering using unsupervised binary trees." In: *Advances in Data Analysis and Classification* 7 (2013), pp. 125–145.

[32] Shi Li and Ola Svensson. "Approximating k-median via pseudo-approximation." In: *proceedings of the forty-fifth annual ACM symposium on theory of computing*. 2013, pp. 901–910.

[33] Shalev Ben-David and Lev Reyzin. "Data stability in clustering: A closer look." In: *Theoretical Computer Science* 558 (2014), pp. 51–61.

[34] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. "Bilu–Linial stable instances of max cut and minimum multiway cut." In: *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM. 2014, pp. 890–906.

[35] Pranjal Awasthi et al. "The hardness of approximation of euclidean k-means." In: *arXiv preprint arXiv:1502.03316* (2015).

[36] Maria-Florina Balcan, Nika Haghtalab, and Colin White. "k-center Clustering under Perturbation Resilience." In: *arXiv preprint arXiv:1505.03924* (2015).

[37] Shai Ben-David. "Computational feasibility of clustering under clusterability assumptions." In: *arXiv preprint arXiv:1501.00437* (2015).

[38] Rich Caruana et al. “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission.” In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1721–1730.

[39] Maria Florina Balcan and Yingyu Liang. “Clustering under perturbation resilience.” In: *SIAM Journal on Computing* 45.1 (2016), pp. 102–155.

[40] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[41] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “” Why should i trust you?” Explaining the predictions of any classifier.” In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[42] Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. “Algorithms for stable and perturbation-resilient problems.” In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 2017, pp. 438–451.

[43] Jarosław Byrka et al. “An improved approximation for k-median and positive correlation in budgeted optimization.” In: *ACM Transactions on Algorithms (TALG)* 13.2 (2017), pp. 1–31.

[44] Badih Ghattas, Pierre Michel, and Laurent Boyer. “Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods.” In: *Pattern Recognition* 67 (2017), pp. 177–185.

[45] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions.” In: *Advances in neural information processing systems* 30 (2017).

[46] Haris Angelidakis et al. “Bilu-linial stability, certified algorithms and the independent set problem.” In: *arXiv preprint arXiv:1810.08414* (2018).

[47] Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. “Interpretable clustering via optimal trees.” In: *arXiv preprint arXiv:1812.00539* (2018).

[48] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Anchors: High-precision model-agnostic explanations.” In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[49] Sara Ahmadian et al. “Better guarantees for k-means and euclidean k-median by primal-dual algorithms.” In: *SIAM Journal on Computing* 49.4 (2019), FOCS17–97.

[50] David Alvarez-Melis et al. “Weight of evidence as a basis for human-oriented explanations.” In: *arXiv preprint arXiv:1910.13503* (2019).

[51] Daniel Deutch and Nave Frost. “Constraints-based explanations of classifications.” In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. 2019, pp. 530–541.

[52] W James Murdoch et al. “Interpretable machine learning: definitions, methods, and applications.” In: *arXiv preprint arXiv:1901.04592* (2019).

[53] Cynthia Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.” In: *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[54] Anup Bhattacharya, Dishant Goyal, and Ragesh Jaiswal. “Hardness of Approximation of Euclidean k-Median.” In: *arXiv preprint arXiv:2011.04221* (2020).

[55] Damien Garreau and Ulrike Luxburg. “Explaining the explainer: A first theoretical analysis of LIME.” In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1287–1296.

[56] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[57] Michal Moshkovitz et al. “Explainable k-means and k-medians clustering.” In: *International conference on machine learning*. PMLR. 2020, pp. 7055–7065.

[58] Kacper Sokol and Peter Flach. “LIMEtree: Interactively customisable explanations based on local surrogate multi-output regression trees.” In: *arXiv preprint arXiv:2005.01427* (2020).

[59] Buddhima Gamlath et al. “Nearly-tight and oblivious algorithms for explainable clustering.” In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 28929–28939.

[60] Eduardo S Laber and Lucas Murtinho. “On the price of explainability for some clustering problems.” In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5915–5925.

[61] Konstantin Makarychev and Liren Shan. “Near-optimal algorithms for explainable k-medians and k-means.” In: *International Conference on Machine Learning*. PMLR. 2021, pp. 7358–7367.

[62] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021.

[63] Moses Charikar and Lunjia Hu. “Near-optimal explainable k-means for all dimensions.” In: *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2022, pp. 2580–2606.

[64] Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. “Almost tight approximation algorithms for explainable clustering.” In: *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2022, pp. 2641–2663.

[65] Jacob Kauffmann et al. “From clustering to cluster explanations via neural networks.” In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).