

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Εκμάθηση Υπερπλήρους Λεξικού για Αραιές
Αναπαραστάσεις

Παναγιώτης Σ. Γεωργαντόπουλος 5732

Επιβλέπων: Εμμανουήλ Ζ. Ψαράκης

Αύγουστος 2018

Περιεχόμενα

1	Εισαγωγή	1
1.1	Η έννοια του λεξικού	1
1.2	Αναπαράσταση σήματος	1
1.3	Αραιή αναπαράσταση σήματος	2
1.4	Η χρησιμότητα της αραιότητας	4
1.5	Πρακτικές εφαρμογές αραιής αναπαράστασης	5
2	Ορισμός προβλήματος	7
2.1	Υποκαθορισμένο γραμμικό σύστημα	7
2.2	Μαθηματικό μοντέλο προβλήματος	8
3	Προσεγγίσεις	11
3.1	Προσέγγιση των δεδομένων με αραιές αναπαραστάσεις	11
3.1.1	Άπληστοι αλγόριθμοι	12
3.1.1.1	Αναζήτηση Ταιριάσματος	12
3.1.1.2	Ορθογώνια Αναζήτηση Ταιριάσματος	14
3.1.1.3	Generalized OMP	18
3.1.1.4	αOMP	18
3.1.2	Κυρτή βελτιστοποίηση	19
3.1.2.1	Least Absolute Shrinkage and Selection Operator	21
3.2	Εκμάθηση λεξικού	22
3.2.1	Πιθανοτικοί αλγόριθμοι	23
3.2.2	Παραμετρικά Λεξικά	24
3.2.3	Λεξικά ομαδοποίησης	24
3.2.3.1	K-SVD	25
3.2.3.2	Total Least Squares	27
3.2.3.3	Approximate K-SVD	28
3.2.3.4	Least Squares Coefficient Persistent	29
3.2.3.5	Recursive Least Squares	30

4	Πειραματικά αποτελέσματα	33
4.1	Υπολογισμός αραιής αναπαράστασης	33
4.2	Το λεξικό	35
4.3	Συνθετικά δεδομένα	38
4.4	Αποθρορυβοποίηση	40
4.5	Συμπλήρωση ελλειπών δεδομένων	41
4.6	K-SVD και παραλλαγές	42
5	Παράρτημα	55
5.1	Κώδικες Matlab	55

Κεφάλαιο 1

Εισαγωγή

1.1 Η έννοια του λεξικού

Ο τρόπος με τον οποίο ένα λεξικό έχει καθιερωθεί να ορίζεται, είναι αυτός του λεξικού μίας φυσικής γλώσσας. Δηλαδή, μία πεπερασμένη συλλογή λέξεων οι οποίες όταν ακολουθούν η μία την άλλη, με συντακτικά ορθό τρόπο, αποδίδουν κάποιο νόημα. Όπως είναι ήδη φανερό, οι πιθανές προτάσεις που συντάσσονται από μία μοναδική συλλογή λέξεων, είναι περισσότερες της μίας. Μπορούμε να πούμε λοιπόν, ότι μία ακολουθία λέξεων συντονίζεται από έναν συνδυασμό που εμείς επιβάλλουμε και καλούμε σύνταξη, ώστε να διατυπώσουμε και να επικοινωνήσουμε ένα νόημα.

Την παραπάνω δομή μίας γλώσσας, δεν την παρατηρούμε όμως μόνο στην φυσική γλώσσα. Ειδικά στον χώρο της πληροφορικής, χρησιμοποιούμε τον όρο της γλώσσας για να υποδηλώσουμε μία συλλογή στοιχείων τα οποία συνδυαζόμενα συναποτελούν μία πρόταση. Αντίστροφα, όλες οι πιθανές προτάσεις που αποδίδουν κάποιο νόημα – ή ευρύτερα είναι συντακτικά ορθές – ορίζουν το σύνολο της γλώσσας. Πιο συγκεκριμένα, στον χώρο της επεξεργασίας σήματος, ορίζουμε ένα λεξικό, ως μία συλλογή σημάτων, τα οποία δύνανται να συνδυαστούν γραμμικά ανά ομάδες για να αποδώσουν ένα νέο σήμα. Πριν προχωρήσουμε περαιτέρω στην δομή ενός λεξικού σημάτων πρέπει να αναφερθούμε γενικά στους τρόπους μπορούμε να αναπαραστήσουμε ένα σήμα.

1.2 Αναπαράσταση σήματος

Ένα σήμα από μόνο του δεν δίνει πολλή πληροφορία με την πρώτη ματιά. Πολλές φορές τα κομμάτια που το συνθέτουν μας δίνουν μία πιο πλούσια γνώση για αυτά. Άρα χρειαζόμαστε ένα εργαλείο με το οποίο θα μπορούμε να αναλύουμε το σήμα στα συστατικά του. Για παράδειγμα, αν έχουμε ένα σύνολο σημάτων που αποτελούν υλοποιήσεις μιας στοχαστικής διαδικασίας, τότε θέλουμε να γνωρίζουμε τα συστατικά της διαδικασίας αυτής ώστε να μπορούμε να τεμαχίσουμε την πληροφορία σε σημαντικά και πιο ασήμαντα

κομμάτια. Η πιο γνωστή ανάλυση που παραγάγει ένα τέτοιο αποτέλεσμα είναι η Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis-PCA)[8]. Σύμφωνα με αυτή, εάν το αρχικό μας σήμα είναι υλοποιήσεις, μηδενικής μέσης τιμής, μίας στοχαστικής διαδικασίας και βρίσκονται μέσα σε έναν χώρο, τότε αναζητούμε την ορθογώνια βάση που περιγράφει τις υλοποιήσεις αυτές ως ένα σύνολο σημείων με διακυμάνσεις γύρω από αυτή. Έτσι, έχουμε μία σαφή περιγραφή των δεδομένων, σχετικά με την διεύθυνση τους μέσα στο χώρο που βρίσκονται. Μία άλλη, παρόμοια και εξίσου χρήσιμη διαδικασία αποσύνθεσης ενός σήματος, είναι Ανάλυση Ανεξάρτητων Συνιστωσών (Independent Component Analysis) [10]. Σύμφωνα με αυτή, υποθέτουμε ένα πλήθος φυσικών αιτιών, ανεξάρτητων μεταξύ τους και μη Gaussian, που παράγουν τις υλοποιήσεις που παρατηρούμε, επίσης μηδενικής μέσης τιμής. Οπότε, αναζητούμε την αποσύνθεση του δεδομένων στα αρχικά σήματα που το συναποτελούν, διατηρώντας την υπόθεση της στατιστικής ανεξαρτησίας τους. Αυτό η ICA το πετυχαίνει απομακρύνοντας τις επιμέρους συναρτήσεις πυκνότητας πιθανότητας από την Gaussian.

Επεκτείνοντας την ματιά μας στον χώρο επεξεργασία σήματος, βλέπουμε και άλλες αναπαραστάσεις με ποικίλες χρησιμότητες. Ίσως ο πιο σημαντική ανάλυση σήματος με ένα γραμμικό μοντέλο, αποτελεί η ανάλυσή του σε αρμονικές συνιστώσες. Γνωστός και ως μετασχηματισμός Fourier [7], η διαδικασία αυτή διευκολύνει στην αναγνώριση του σήματος ως μία οντότητα δύο όψεων: αυτή της χρονικής/χωρικής επέκτασής του και αυτή της συχνοτικής. Ένας τέτοιος μετασχηματισμός του σήματος, δίνει την δυνατότητα να επεξεργαστούμε διαφορετικές πτυχές του σήματος, που στο πεδίο του χώρου/χρόνου θα ήταν αδύνατο.

Συναφείς μετασχηματισμοί όπως οι Wavelet, Συνημιτόνου κ.ά. ορίζουν ένα πλαίσιο στο οποίο μπορούμε να αναπαραστήσουμε σήματα, ανάλογα τις στατιστικές χωρο/χρονικές τους ιδιότητες. Εάν προσπαθήσουμε με τους παραπάνω μετασχηματισμούς να μειώσουμε τις διαστάσεις του χώρου που βρίσκεται το σήμα, τότε είναι φανερό πως χάνουμε πληροφορία.

Σε κάθε περίπτωση, όταν μετασχηματίζουμε ένα σήμα ή το αναλύουμε στις συνιστώσες του, ουσιαστικά διαχωρίζουμε την πληροφορία που περιέχει με τρόπο τέτοιο ώστε να μας είναι πιο εύκολο να το επεξεργαστούμε περαιτέρω ή να προβούμε σε κάποια συμπεράσματα. Παρ' όλα αυτά δεν μειώνουμε την πληροφορία που περιέχεται στο αρχικό σήμα, ενώ εάν το επιχειρήσουμε, η μεταβολή είναι συνήθως αισθητή.

1.3 Αραιή αναπαράσταση σήματος

Λίγο πολύ, μέχρι ώρας είναι κατανοητό ότι χρειαζόμαστε έναν τρόπο να εκμαιεύσουμε πληροφορία από το σήμα μειώνοντας τις διαστάσεις του χώρου στον οποίο βρίσκεται. Χρειαζόμαστε ένα λεξικό το οποίο θα εκμεταλλεύεται τα χαρακτηριστικά του σήματος,

προκειμένου να το αναπαραστήσει. Αυτή τη μείωση της διάστασης του σήματος της ερμηνεύουμε ως αραιότητα. Η αραιότητα όμως, όπως είδαμε παραπάνω, όταν επιβάλλεται κοστίζει χρήσιμο πληροφορία. Όλοι οι παραπάνω μετασχηματισμού και διαδικασίες προβολής σε υπόχωρους (PCA), δεν διέπονται από κάποιον περιορισμό για την αραιότητά τους.

Είναι πιο σοφό, λοιπόν, να επικεντρωθούμε στην αναζήτηση ενός λεξικού που θα περιγράφει τα δεδομένα μας με αραιό τρόπο, θέτοντας αυτό ως προϋπόθεση ενώ το λεξικό δημιουργείται.

Αμέσως, δημιουργείται η απορία για το ποιά θα είναι τα ποιοτικά χαρακτηριστικά ενός καλού λεξικού. Δεν μπορούμε να περιγράψουμε όλα τα πιθανά σήματα με μόνο ένα λεξικό. Ένα κλασσικό πείραμα στη σχετική βιβλιογραφία είναι η εύρεση όχι οποιουδήποτε λεξικού περιγράφει καλά τα δεδομένα, αλλά του λεξικού που τα παράγει. Για παράδειγμα στη φυσική γλώσσα, μπορούμε να αναλύσουμε ένα σύνολο ηχητικών δειγμάτων στις συνιστώσες που το παράγουν, αλλά αυτό δεν δεσμεύει την ανάλυσή μας να μας παράγει ένα λεξικό το οποίο θα αποτελείται από τις λέξεις αυτούσιες όπως αυτές ορίζονται από την φυσική γλώσσα.

Εάν δούμε το λεξικό από την οπτική μίας βάσης δεδομένων τότε δρούμε πολύ απλά κάνουμε απλή αποθήκευση των δεδομένων για μετέπειτα χρήση. Όμως, αυτό είναι κάτι συνήθως δύσκολο. Αναλύοντας το περιβάλλον στο οποίο ένα λεξικό γίνεται απαραίτητο, θα δούμε ότι τόσο ο αποθηκευτικός χώρος όσο και οι υπολογιστικοί πόροι, είναι μηδαμικοί μπροστά σε έναν όγκο δεδομένων οποίος διατηρείται και διακινείται σε παγκόσμια κλίμακα. Από τηλεπικοινωνίες και τοπική αποθήκευση αρχείων, έως και τον ασύλληπτα μεγάλο όγκο δεδομένων που διακινούνται στο διαδίκτυο, αλλά και σήματα τα οποία αφουγκραζόμαστε από διαστημικά σώματα, λίγο πολύ έχουμε ένα ευρύ φάσμα σεναρίων που απαιτούν μία πιο φιλοσοφημένη διαχείριση δεδομένων από απλή αποθήκευση. Έτσι, έρχεται στο προσκήνιο η έννοια της αραιότητας. Ένα λεξικό φυσικής γλώσσας, για παράδειγμα, δεν είναι αραιό. Αυτό οφείλεται στην χρήση της φυσική γλώσσας. Όταν θέλουμε να διατυπώσουμε ένα νόημα, ο τρόπος σύνταξης είναι σχεδόν ντετερμινιστικός ή τουλάχιστον υπάρχουν λίγες διαφορετικές διατυπώσεις που οδηγούν στο αρχικό νόημα. Αν και συχνά, ειδικά στην ελληνική γλώσσα συναντούμε λέξεις με περιγραφικό νόημα οι οποίες κάλλιστα αντικαθιστούν ολόκληρες προτάσεις, είναι πολύ δύσκολο να βρούμε πολλές τέτοιες περιπτώσεις μέσα σε ένα κείμενο ώστε να μπορέσουμε να εκφράσουμε το ίδιο νόημα με λιγότερες λέξεις. Οπότε καταλήγουμε στο συμπέρασμα ότι η φυσική γλώσσα δεν αποτελείται από μηδαμινής σημασίας άτομα που στο σύνολό τους συνθέτουν ένα αποτέλεσμα που εμείς εκλαμβάνουμε ως νόημα, αλλά από λέξεις, η κάθε μία με διαφορετικό νοηματικό βάρος και δυνατότητα αντικατάστασής της. Αποτελεί, θα λέγαμε ένα άκατάστατα χβαντισμένο νόημα σε μικρά υπονοήματα, με διαφορετικό μήκος και σημασία κάθε λέξης.

Από την άλλη μεριά, ένα σήμα, αποτελεί μία οντότητα που είναι πιο εύκολο να χβαντιστεί και να διαχειριστεί ως σύνθεση σημάτων. Σε παραλληλία με το νόημα μίας πρότασης φυσικής γλώσσας, μία συνήθης μετρική εγγύτητας δύο σημάτων είναι το μέσο τετραγωνικό σφάλμα. Μία τέτοια μετρική λαμβάνει υπόψιν τις στοιχείο προς στοιχείο διαφορές δύο σημάτων. Οπότε, δύο προτάσεις, θα λέγαμε ότι απέχουν, κατά μέσο όρο, όσο η μέση τετραγωνική τιμή των στοιχείο προς στοιχείο διαφορών τους. Σε αυτό το σκεπτικό, κάθε χβάντο του σήματος θεωρείται ισοδύναμο με όλα τα άλλα, στην γενική περίπτωση. Άρα σε όρους σημάτων, μας είναι πιο εύκολο να δούμε ότι ίσως είναι εφικτή μία αναπαράσταση ενός σήματος όχι ακριβώς ίδιο με το αρχικό, αλλά κοντά σε αυτό, υπό την έννοια του μέσου τετραγωνικού σφάλματος. Άρα μας δίνεται η δυνατότητα να θυσιάσουμε την ακρίβεια της αναπαράστασης του σήματος με κάποιον σκοπό. Ένας τέτοιος σκοπός μπορεί να είναι η αραιή αναπαράσταση του σήματος.

Έχοντας πει αυτά, μπορούμε πλέον να εισάγουμε την έννοια της αραιότητας. Με την χρονική ή χωρική έννοια, ένα σήμα είναι αραιό όταν έχει μη μηδενικές τιμές σε λίγα, αναλογικά με τον χωρικό/χρονικό του όγκο. Με την έννοια της αναπαράστασης, ένα σήμα μπορεί να αναπαρασταθεί ως ένας γραμμικός συνδυασμός λίγων σημάτων, αναλογικά με το συνολικό τους πλήθος.

Οι παραπάνω έννοιες είναι βασικές για πολλά πεδία της επεξεργασίας σήματος. Η συμπίεση, όπως το παράδειγμα που δόθηκε πιο πάνω, η επεξεργασία εικόνας ή σημάτων στον τομέα της ιατρικής, της βιολογίας, της σεισμολογίας, της γεωλογίας, της φυσικής και της αστρονομίας, είναι μερικοί τομείς στους οποίους με φυσικό τρόπο προκύπτει η ανάγκη για αραιές αναπαραστάσεις. Ειδικά μεγάλη συμβολή της αραιότητας εισάγεται στον τομέα της συμπιεσμένης αντίχνευσης, όπου παρακάμπτεται το βασικό θεώρημα δειγματοληψίας του Shannon, σε ειδικές περιπτώσεις.

1.4 Η χρησιμότητα της αραιότητας

Σχεδόν όλες οι φυσικές διαδικασίες που παρατηρούμε περιέχουν πληροφορία την οποία συλλέγουμε από αισθητήρες, είτε φυσικούς (βλ. αισθητήρες ορατού φωτός), είτε τεχνητούς (: οποιοδήποτε σύστημα συλλογής πληροφορίας από το περιβάλλον του). Πίσω από τα δεδομένα που συλλέγουμε, κρύβεται η φυσική διαδικασία η οποία αποτελεί το αίτιο της παρατήρησης, για την οποία θέλουμε να καταλήξουμε σε κάποια συμπεράσματα. Άρα αμέσως - αμέσως, ερχόμαστε στο εξής δίλημμα: να κρατήσουμε όση περισσότερη πληροφορία μπορούμε από τη φυσική διαδικασία ή να προσπαθήσουμε να εκμαιεύσουμε μία εκδοχή της που θα περιέχει την ίδια ή ίσως ελεγχόμενα λιγότερη πληροφορία, οδηγώντας στα ίδια συμπεράσματα για την φυσική διαδικασία. Η απάντηση δίνεται από ένα απλό παράδειγμα μιας φυσικής περιοδικής διαδικασίας. Έστω, ότι από αυτή στους αισθητήρες μας αντλούμε δείγματα το οποία ακολουθούν μία ημιτονοειδή μορφή. Είναι

προφανές ότι εάν χρειαζόμαστε όλη αυτή την χρονικά ασυμπίεστη πληροφορία για να εξάγουμε συμπεράσματα για την τυχαία διαδικασία που αφουγκραζόμαστε, δεν θα τελιώνουμε ποτέ. Εάν όμως ακολουθήσουμε τον Shannon, με μόλις μερικές περιόδους δειγμάτων του σήματος, μπορούμε να βγάλουμε συμπεράσματα για το πλάτος του, την συχνότητα και την αρχική φάση που το διέπει.

1.5 Πρακτικές εφαρμογές αραιής αναπαράστασης

Ο βασικός σκοπός της αραιής αναπαράστασης είναι η προσέγγιση σήματος, μέσω προβολών του σε κάποιον υπόχωρο. Ιδανικά, δηλαδή, θα θέλαμε να βρούμε τον μαθηματικό υπόχωρο που παράγει την διαδικασία, μέσα από τις παρατηρήσεις τις, σε έναν ευρύτερο χώρο. Προϋπόθεση για το παραπάνω είναι ο υπόχωρος που θα ορίζεται η σύνθεση των υποσημάτων να είναι μικρός σε σχέση με τον χώρο στον οποίο βρίσκεται το συλλεγόμενο σήμα. Ένα παράδειγμα πρακτικής εφαρμογής της αραιής αναπαράστασης είναι η εξαγωγή θορύβου από εικόνες. Εκπαιδεύοντας ένα λεξικό να μπορεί να ανακατασκευάζει με μηδαμινό σφάλμα εικόνες συγκεκριμένου περιεχομένου (π.χ. ιατρικού), μπορούμε να το βάζουμε σε θέση να 'μαντεύει' μέσω της ανακριβούς ανακατασκευής που θα κάνει σε μία ενθόρυβη εικόνα. Ευελπιστώντας ότι η ανακρίβεια της ανακατασκευής θα οφείλεται μόνο στον προσθετικό θόρυβο, η ανακατασκευή που θα πάρουμε θα είναι μία πιο καθαρή εκδοχή της αρχικής εικόνας πριν προστεθεί ο θόρυβος. Μία δεύτερη συνήθης εφαρμογή της αραιής αναπαράστασης είναι η κατηγοριοποίηση σημάτων ή αλλιώς ομαδοποίηση. Εδώ σκοπός είναι ο διαχωρισμός δεδομένων σε ομάδες προβάλλοντάς τα σε κάποια βάση στην οποία οι προβολές είναι αραιές. Κατά αυτόν τον τρόπο, δεδομένα τα οποία βρίσκονται εγγύτερα στον υπόχωρο αυτόν μπορούν να ομαδοποιηθούν ως αντικείμενα της ίδιας κλάσης. Οι κατηγορίες ομαδοποίησης είναι δύο: ομαδοποίηση με πρότερη γνώση των ομάδων (supervised) και με μη πρότερη γνώση των ομάδων (unsupervised).

Κεφάλαιο 2

Ορισμός προβλήματος

2.1 Υποκαθορισμένο γραμμικό σύστημα

Στην Γραμμική Άλγεβρα ορίζουμε ένα (γραμμικό) σύστημα εξισώσεων ως $Ax = b$. Εάν υποθέσουμε ότι στις στήλες του μητρώου A βρίσκουμε τις υλοποιήσεις μία τυχαίας διαδικασίας, τότε το x εκφράζει όλους τους πιθανούς γραμμικούς συνδυασμούς των στηλών του A οι οποίοι δίνουν το διάνυσμα b . Η λύση x δεν είναι πάντα υπαρκτή, και εάν υπάρχει δεν είναι πάντοτε μοναδική. Ικανή και αναγκαία συνθήκη για να υπάρχει μοναδική λύση είναι το A να είναι πλήρους τάξης τετραγωνικό μητρώο, άρα και αντιστρέψιμο. Οπότε, η μοναδική λύση θα ήταν: $x = A^{-1}b$, με A^{-1} ο αντίστροφος του A . Τώρα, για να είναι το μητρώο A πλήρους τάξης, θα πρέπει να περιέχει μόνο γραμμικά ανεξάρτητες στήλες. Το σενάριο αυτό, όμως δεν μας βολεύει ιδιαίτερα. Διότι η ανεξαρτησία δεν μας είναι κάποιο χρήσιμο ποιοτικό χαρακτηριστικό που θέλουμε να διέπει το A . Αντίθετα, όπως προείπαμε, θα θέλαμε ο A να είναι σε θέση να περιγράφει καλά τα δεδομένα b , υπό την έννοια του μέσου τετραγωνικού σφάλματος, χωρίς αναγκαστικά οι στήλες του να αποτελούν ορθογώνια βάση για τον χώρο στον οποίο βρίσκονται.

Στην δική μας περίπτωση θα θέλαμε το A να είναι ένα λεξικό και x να καθορίζει τον τρόπο με τον οποίο θα συνδυάζαμε τα άτομά του ώστε να παράξουμε το αποτέλεσμα b . Αυτό όμως δεν εξασφαλίζει ότι το x είναι αραιό. Θυμίζουμε ότι σκοπός μας είναι η αραιή αναπαράσταση των δεδομένων. Άρα θέλουμε το διάνυσμα x να περιέχει ελάχιστα, ως προς το πλήθος τους, μη μηδενικά στοιχεία. Άρα θέλουμε να αποφύγουμε να περιοριστούμε σε μία βάση του χώρου των δεδομένων και αναζητούμε ένα μη πλήρους τάξης στηλών A , ώστε να περιέχει περισσότερα γραμμικώς εξαρτώμενα διανύσματα που θα εξυπηρετούν έναν άλλον σκοπό, αυτόν του αραιού συνδυασμού.

Η λύση στο ανωτέρω πρόβλημα είναι να έχουμε ένα A με πολύ περισσότερες στήλες από ότι γραμμές. Με αυτόν τον τρόπο θα περιέχει γραμμικά εξαρτημένα άτομα, που μπορούν με κατάλληλους αραιούς συνδυασμούς να εκφράζουν τα δεδομένα b μέσω του x . Αυτό βασίζεται στο ότι όσα περισσότερα διανύσματα πάνω από την τάξη του μητρώου

A εναποτίθενται στις στήλες του (μέγιστη τάξη αποτελεί το ελάχιστο πλήθος γραμμών ή στηλών), τόσα περισσότερα διανύσματα υπερρορίζουν τον χώρο στηλών του, δίνοντας διαφορετικά εναλλακτικά x για αναπαράσταση των δεδομένων b .

Η ιδέα ενός υποκαθορισμένου συστήματος, όπως αλλιώς καλείται, υπερτερεί σε εφαρμογές όπως η ομαδοποίηση και η αραιή αναπαράσταση, διότι δεν έγκειται στην ανάλυση του χώρου στον οποίον βρίσκονται τα δεδομένα και μετά στην επεξεργασία τους (προβολή σε υπόχωρους κλπ.), αλλά στην άμεση περιγραφή των νέων δεδομένων από τα ήδη υπάρχοντα. Δηλαδή είναι μία οδηγούμενη από τα δεδομένα διαδικασία. Στην βιβλιογραφία, και από τώρα και στο εξής, θα ονομάζουμε λεξικά όπως το A , υπερπλήρη, διότι υποερορίζουν τον χώρο στον οποίο βρίσκονται. Δηλαδή, περιγράφουν τον χώρο με περισσότερα διανύσματα από,τι είναι αναγκαία.

2.2 Μαθηματικό μοντέλο προβλήματος

Το μοντέλο, όπως ορίστηκε περιγραφικά και παραπάνω, είναι το εξής: Θεωρούμε ένα σύνολο δεδομένων $Y \in \mathbb{R}^{d \times N}$ προερχόμενα από N υλοποιήσεις της ίδιας τυχαίας διαδικασίας ¹ μέσα στον \mathbb{R}^d . Επίσης, θεωρούμε ένα σύνολο K ατόμων μέσα στον \mathbb{R}^d ως διανύσματα ενός υπερπλήρους λεξικού $D \in \mathbb{R}^{d \times K}$, τα οποία γραμμικά συνδυαζόμενα όπως υποδεικνύει το μητρώο $X \in \mathbb{R}^{K \times N}$, παράγουν τα δεδομένα του Y :

$$\mathbf{y}_i = \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \dots & \mathbf{d}_K \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iK} \end{bmatrix} = \sum_{k=1}^K x_{ik} \mathbf{d}_k, \quad \forall i = 1, \dots, N$$

ή ισοδύναμα

$$Y = DX \quad s.t. \quad \|\mathbf{x}_i\|_p \leq T_0, \quad \forall i = 1, \dots, N. \quad (2.1)$$

Ο περιορισμός που τίθεται είναι ότι το X είναι αραιό με έναν προκαθορισμένο αριθμό στοιχείων ανά στήλη, T_0 .

Το μόνο που έχουμε στην διάθεσή μας, είναι το Y . Άρα αναζητούμε τα D και X για τα οποία ισχύει η Σχέση (2.1). Είναι αδύνατο να βρούμε ταυτόχρονα και τα δύο. Αλλά θέλουμε τουλάχιστον να μπορούμε: εάν γνωρίζουμε τα Y και X , να μπορούμε να βρούμε το D και αν γνωρίζουμε τα Y και D , να μπορούμε να βρούμε το X .

Βέβαια, είναι σαφές ότι ο περιορισμός που έχουμε θέσει, θα μπορούσε να μας διευκολύνει, δίνοντας μας ένα πλήθος λύσεων στο πρόβλημα. Από την στιγμή όμως που

¹Με τον όρο τυχαία διαδικασία αναφερόμαστε στο ίδιο φυσικό αίτιο. Δεν είναι σωστό να περιέχονται δύο ειδών δεδομένα μέσα στο Y . Για παράδειγμα, δεδομένα από καρδιογράφημα δεν μπορούν να αναμειχθούν με φωνητικά δεδομένα.

υπάρχει, θα πρέπει να ξεχάσουμε κάθε μέθοδο παραγοντοποίησης μητρώου (διότι περί αυτού πρόκειται στην μην περιορισμένη περίπτωση), και να επικεντρωθούμε στην βέλτιστη λύση μπορούμε να έχουμε υπό την σκιά του περιορισμού.

Γνωρίζοντας λοιπόν, ότι δεν μπορούμε στην γενική περίπτωση να έχουμε D, X που να ικανοποιούν την εξίσωση, τουλάχιστον θέλουμε να προσεγγίζουμε το Y με το μικρότερο δυνατό σφάλμα. Δηλαδή θέλουμε D, X ώστε:

$$D^*, X^* = \arg \min_{D, X} \{\|Y - DX\|_F^2\} \quad s.t. \quad \|\mathbf{x}_i\|_p \leq T_0 \quad \forall i = 1, \dots, N, \quad (2.2)$$

όπου $\|a\|_p = \left(\sum_i |a_i|^p\right)^{1/p}$ η ℓ_p νόρμα του διανύσματος a και T_0 ένας προκαθορισμένος ακέραιος με $T_0 \ll K$.

Όπως τίθεται, το Πρόβλημα (2.2) αφήνει να την αραιότητα να ορίζεται αφαιρετικά ως προς κάποια νόρμα των στηλών του X . Αν θέλαμε να ορίσουμε αυστηρά την αραιότητα, θα λέγαμε ότι τα $N - T_0$ στοιχεία οποιουδήποτε \mathbf{x}_i θα πρέπει να είναι μηδενικά. Αυτό ισοδυναμεί με το να επαναδιατυπώσουμε το Πρόβλημα (2.2) για $p = 0$:

$$D^*, X^* = \arg \min_{D, X} \{\|Y - DX\|_F^2\} \quad s.t. \quad \|\mathbf{x}_i\|_0 \leq T_0 \quad \forall i = 1, \dots, N. \quad (2.3)$$

Αμέσως, προκύπτει ένα εμπόδιο: Το Πρόβλημα (;) είναι μη επιλύσιμο σε πολυωνυμικό χρόνο (NP-hard). Καθώς η νόρμα ℓ_0 είναι μη κυρτή συνάρτηση². Οδηγούμαστε σε ένα συνδυαστικό πρόβλημα, το οποίο είναι εξαιρετικά δύσκολο να λύσουμε.

Ένα δεύτερο πρόβλημα είναι η ταυτόχρονη ελαχιστοποίηση ως προς τα D και X . Μπορούμε σίγουρα να παραγοντοποιήσουμε το Y υπό κάποιες συνθήκες σε δύο μητρώα-παράγοντες, αλλά πουθενά δεν εμπλέκεται η έννοια της αραιότητας στο μεν και της υποεμπληρότητας στο δε. Είναι ανάγκη λοιπόν να επιλύσουμε ξεχωριστά το πρόβλημα ελαχιστοποίησης του σφάλματος και της εύρεσης αραιής ανακατασκευής.

Περαιτέρω, να πούμε ότι το μαθηματικό μοντέλο δεν υποθέτει οποιαδήποτε αλλοίωση στα δεδομένα Y . Αν και στα πειράματα που θα δούμε αργότερα υποθέτουμε διάφορα σενάρια θόρυβου, οι αλγόριθμοι που θα συζητήσουμε αφορούν δεδομένα χωρίς θόρυβο για την εκπαίδευση του λεξικού. Αυτό είναι και εν μέρει λογικό. Διότι το λεξικό είναι 'τυφλό' απέναντι στον θόρυβο και δεν κάνει διάκριση αθόρυβων ή μη σημάτων. Κάλιστα θα μάθει να τον μοντελοποιεί και αυτόν. Αυτό είναι κάτι που δεν το επιθυμούμε και για αυτό τα δεδομένα Y , ως δεδομένα εκπαίδευσης, θεωρούνται χωρίς θόρυβο.

Σε αυτό το σημείο, είναι σημαντικό να ξεκαθαρίσουμε ότι ο περιορισμός της αραιότητας, όπως ορίστηκε, δεν χρησιμοποιείται καθολικά. Πέραν της μορφής του,

²Στην πραγματικότητα, η ψευδονόρμα ℓ_0 δεν είναι εξ ορισμού νόρμα καθώς δεν πληροί την ιδιότητα της κλιμάκωσης ($\|ax\|_0 = |a| \|x\|_0, \forall a \in \mathbb{R}$)

$$\|\mathbf{x}\|_p \leq T_0$$

υπάρχουν και άλλα κριτήρια τα οποία οδηγούν σε διάφορες παραλλαγές στις υλοποιήσεις των αλγορίθμων που θα δούμε στη συνέχεια. Για παράδειγμα, ο όρος 'το πολύ' T_0 μη μηδενικά στοιχεία (στην περίπτωση $p = 0$) σημαίνει ότι και κάποια άλλη συνθήκη εμπλέκεται στον αλγόριθμο. Συνήθως, αυτή είναι το σφάλμα. Δηλαδή, ενδέχεται το σφάλμα να πέσει κάτω από ένα επιθυμητό κατώφλι με πολύ μικρότερη νόρμα $\|\mathbf{x}\|_p$ από T_0 .

Ο περιορισμός μπορεί να πάρει την ακόλουθη πιο αυστηρή μορφή:

$$\|\mathbf{x}\|_p = T_0.$$

Σε αυτή την περίπτωση, ενδέχεται η τιμή T_0 να μην είναι (συνήθως δεν είναι) κατάλληλη για όλα τα $\mathbf{x}_i, i = 1, \dots, N$.

Κεφάλαιο 3

Προσεγγίσεις

Όπως αναφέρθηκε, το πρόβλημα της αραιής αναπαράστασης¹ και της εκμάθησης του λεξικού εντάσσονται σε δύο διαφορετικές κατηγορίες εναλλάσσοντας τα δεδομένα με τα ζητούμενά τους. Χωρίζουμε την αναζήτηση της λύσης του Προβλήματος (2.3) σε δύο μεγάλες κατηγορίες:

1. Αραιή προσέγγιση των δεδομένων.
2. Εκμάθηση λεξικού.

Για κάθε κατηγορία υπάρχει αντίστοιχη βιβλιογραφία που υπερκαλύπτει τις ανάγκες μας, λόγω κυρίως της πολυχρηστικότητας του μαθηματικού μοντέλου σε ποικίλες εφαρμογές με διαφορετικές συνθήκες. Εμείς θα καλύψουμε μόνο την βασική βιβλιογραφία για κάθε μία η οποία θα βοηθήσει και στην διεξαγωγή των πειραμάτων αργότερα.

3.1 Προσέγγιση των δεδομένων με αραιές αναπαραστάσεις

Ας υποθέσουμε ότι έχουμε στην διάθεσή μας ένα λεξικό D του οποίου οι στήλες είναι κανονικοποιημένες ως προς την νόρμα l_2 , δηλαδή:

¹Θα χρησιμοποιούμε ενίοτε τον όρο 'προσέγγιση' και ενίοτε τον όρο 'αναπαράσταση', σε ότι αφορά τον υπολογισμό του αραιού μητρώου X . Η αναπαράσταση έγκειται στο γεγονός ότι αναπαριστούμε πληροφορία που βρίσκεται σε χώρο πολλών διαστάσεων, σε έναν χώρο μικρότερης διάστασης, αναπαριστώντας την ίδια ουσιαστική πληροφορία. Την έννοια προσέγγιση την οφείλουμε στο γεγονός ότι δεν είναι ποτέ εφικτό να αναπαραστήσουμε επακριβώς τα αρχικά δεδομένα. Αν και είναι συνάρτηση της εφαρμογής που το απαιτεί, η προσέγγιση είναι κάτι που στην πλειοψηφία των εφαρμογών δεν μπορούμε να αποφύγουμε, ενώ σε ορισμένες περιπτώσεις την επιζητούμε κιόλας. Για παράδειγμα σε μία εφαρμογή ανάκτησης εικόνων από βάση δεδομένων, η ακριβής αναπαράσταση δεν είναι σημαντική έως το σημείο που οι εικόνες είναι διακριτές οι μία από την άλλη. Από εκεί και πέρα οποιαδήποτε παραπάνω πληροφορία συγχρατούμε είναι περισσεια.

$$\|\mathbf{d}_k\|_2 = 1, \quad \forall k = 1, \dots, K.$$

Σκοπός μας είναι να λύσουμε το παρακάτω πρόβλημα:

$$X^* = \arg \min_X \{\|Y - DX\|_F^2\} \quad s.t. \quad \|\mathbf{x}_i^*\|_0 \leq T_0 \quad \forall i = 1, \dots, N. \quad (3.1)$$

Παίρνοντας υπόψιν μας ότι ο περιορισμός ως προς την αραιότητα επιβάλλεται σε κάθε στήλη του μητρώου X , το Πρόβλημα (3.1) είναι ισοδύναμο με τα ακόλουθα N υποπροβλήματα ελαχιστοποίησης:

$$\mathbf{x}_i^* = \arg \min_{\mathbf{x}_i} \{\|\mathbf{y}_i - D\mathbf{x}_i\|_2^2\} \quad s.t. \quad \|\mathbf{x}_i^*\|_0 \leq T_0 \quad \forall i = 1, \dots, N. \quad (3.2)$$

Όπως ήδη αναφέραμε, το Πρόβλημα (3.2), εξαιτίας του περιορισμού δεν είναι επιλύσιμο σε πολυωνυμικό χρόνο. Οι δύο βασικές στρατηγικές που ακολουθούνται, για την αποφυγή του παραπάνω εμποδίου είναι:

- Με χρήση άπληστων αλγορίθμων.
- Με χαλάρωση του μη κυρτού περιορισμού που βασίζεται στην ℓ_0 νόρμα, με χρήση άλλων κυρτών.

Στην συνέχεια θα αναλύσουμε βασικές τεχνικές που βασίζονται στις παραπάνω στρατηγικές.

3.1.1 Άπληστοι αλγόριθμοι

Όλοι οι αλγόριθμοι που ανήκουν σε αυτή την κατηγορία είναι επαναληπτικοί και έχουν ως βάση τους την Αναζήτηση Ταιριάσματος (Matching Pursuit). Αν και οι παραλλαγές του MP και ειδικά του OMP είναι πολλές, εμείς θα αναφερθούμε αναλυτικά στον OMP και περιγραφικά στους Generalized OMP και αOMP που υλοποιήσαμε.

3.1.1.1 Αναζήτηση Ταιριάσματος

Η Αναζήτηση Ταιριάσματος (Matching Pursuit-MP) [14] θέτει την βάση για την λογική αναζήτησης υπόχωρου. Επιλέγουμε ως $\mathbf{x}(k_t)$ την κανονικοποιημένη μέγιστη προβολή του σφάλματος πάνω στο λεξικό:

$$\mathbf{x}(k_t) = \max_{k_t} \left\{ \frac{|\mathbf{d}_{k_t}^T \mathbf{r}_{t-1}|}{\|\mathbf{d}_{k_t}\|_2} \right\}.$$

Η ανακατασκευή αφαιρείται από το τρέχον σφάλμα και η διαδικασία επαναλαμβάνεται για $t = 1, 2, \dots, T_0$. Αποτέλεσμα είναι το υπόλοιπο να είναι κάθετο στο τελευταίο άτομο.

Algorithm 1 Matching Pursuit*Input* : $D \in \mathbb{R}^{d \times K}$, $\mathbf{y} \in \mathbb{R}^d$, $\epsilon, T_0 \in \mathbb{R}$ *Output* : $\mathbf{x} \in \mathbb{R}^K$ $\mathbf{r}_0 \leftarrow \mathbf{y}$ $S \leftarrow \emptyset$ $t \leftarrow 1$ **while** ($\|\mathbf{r}\|_2 > \epsilon$) \wedge ($|S| \leq T_0$) **do** $k_t = \arg \max\{|\mathbf{d}_k^T \mathbf{r}_{t-1}|\}$ $S \leftarrow S \cup k_t$ $\mathbf{x}(k_t) \leftarrow \mathbf{d}_{k_t}^T \mathbf{r}_{t-1}$ $\mathbf{r}_t \leftarrow \mathbf{r}_{t-1} - D\mathbf{x}$ $t \leftarrow t + 1$ **end while****return** \mathbf{x}

Έτσι, δεν επιλέγεται ξανά το ίδιο άτομο σε και επόμενη επανάληψη ακόμη, εκτός κι αν το λεξικό περιέχει ταυτόσημα άτομα, προκαλώντας ισότιμες προβολές. Αναλυτικότερα:

Όντως, ισχύει ότι σε κάθε επανάληψη είναι:

$$\begin{aligned} \mathbf{d}_{k_t}^T \mathbf{r}_t &= \mathbf{d}_{k_t}^T (\mathbf{r}_{t-1} - D\mathbf{x}) \\ &= \mathbf{d}_{k_t}^T \mathbf{r}_{t-1} - \mathbf{d}_{k_t}^T D\mathbf{x} \end{aligned}$$

όπου $D\mathbf{x} = \mathbf{d}_{k_t} x(k_t)$ και $\mathbf{d}_{k_t}^T \mathbf{r}_{t-1} = \mathbf{x}(k_t)$. Άρα καταλήγουμε:

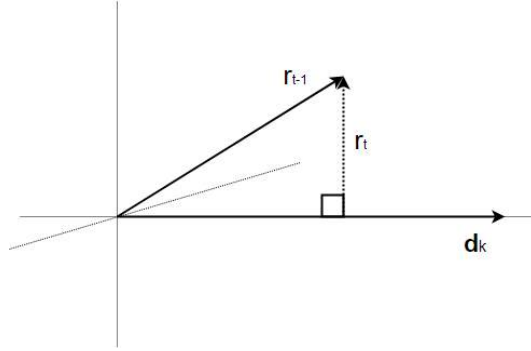
$$\begin{aligned} \mathbf{d}_{k_t}^T \mathbf{r}_t &= \mathbf{d}_{k_t}^T \mathbf{r}_{t-1} - \mathbf{d}_{k_t}^T \mathbf{d}_{k_t} \mathbf{x}(k_t) \\ &= \mathbf{x}(k_t)(1 - \|\mathbf{d}_{k_t}\|_2). \end{aligned}$$

Υποθέτοντας ότι $\|\mathbf{d}_k\|_2 = 1, \forall t$, παίρνουμε:

$$\mathbf{d}_{k_t}^T \mathbf{r}_t = 0,$$

το οποίο μπορούμε εύκολα να παρατηρήσουμε από μία γεωμετρική ματιά στην περίπτωση που $d \leq 3$.

Η καθετότητα αυτή έχει ως αποτέλεσμα $k_t \neq k_{t-1}, \forall t \in S$. Δηλαδή, πάντοτε θα διαλέγεται άτομο διαφορετικό από το αμέσως προηγούμενο, δεδομένου ότι όλα τα άτομα είναι διαφορετικά. Δεν ισχύει όμως ότι $k_t \neq k_{t'}, \forall t, t' \in S$ με $t \neq t'$. Αυτό σημαίνει ότι ένα άτομο μπορεί να επιλεγεί περισσότερες από μία φορές, εάν το σφάλμα προκύψει να είναι μη κάθετο σε αυτό σε επόμενη επανάληψη και η προβολή του σφάλματος πάνω σε αυτό είναι επίσης η μέγιστη σε σχέση με τα τρέχοντα διαθέσιμα άτομα. Αυτό είναι και το κύριο πρόβλημα που έρχεται να αντιμετωπίσει παρακάτω ο OMP.



Σχήμα 3.1: Προβολή του σφάλματος \mathbf{r}_{t-1} πάνω στο άτομο \mathbf{d}_{k_t} . Το διάνυσμα σφάλματος \mathbf{r}_t είναι κάθετο στο \mathbf{d}_{k_t} .

Σύγκλιση της Αναζήτησης Ταιριάσματος

Η παραπάνω επιλογή ατόμων οδηγεί σε μονοτονική μείωση της συνάρτησης σφάλματος. Όχι όμως στο ελάχιστο δυνατό σφάλμα (τουλάχιστον στην γενική περίπτωση.) Γενικά, όλοι οι αλγόριθμοι που θα δούμε έχουν την αδυναμία της εσφαλμένης επιλογής πρώτου ατόμου, καθώς αυτή καθορίζει την επιλογή όλων των επόμενων. Επομένως μπορεί απορρίπτοντας κάποια αρχική μέγιστη προβολή να καταλήξουμε σε κάποιο διαφορετικό S που να οδηγεί στο ελάχιστο δυνατό σφάλμα. Γενικά επειδή ισχύει:

$$\mathbf{r}_t = \mathbf{r}_{t-1} - \mathbf{d}_{k_t} \mathbf{x}(k_t)$$

και $\mathbf{d}_{k_t}^T \mathbf{r}_t = 0$, έχουμε:

$$\|\mathbf{r}_t\|_2^2 = \|\mathbf{r}_{t-1}\|_2^2 + \|\mathbf{d}_{k_t} \mathbf{x}(k_t)\|_2^2.$$

Οπότε σε κάθε επανάληψη η ελαχιστοποίηση του $\|\mathbf{r}_t\|_2^2$ μπορεί να επέλθει μεγιστοποιώντας τον όρο $\|\mathbf{d}_{k_t} \mathbf{x}(k_t)\|_2^2$, δηλαδή για:

$$k_t := \arg \max\{|\mathbf{d}_{k_t}^T \mathbf{r}_{t-1}|\}$$

όντως το σφάλμα μειώνεται σε κάθε επανάληψη. Συγκλίνοντας, τελικά, στο τοπικό ελάχιστο.

3.1.1.2 Ορθογώνια Αναζήτηση Ταιριάσματος

Κατά την Ορθογώνια Αναζήτηση Ταιριάσματος (Orthogonal Matching Pursuit-OMP) [15], αναζητούμε εκείνα τα άτομα του λεξικού που περιγράφουν καλύτερα τα δεδομένα. Η ιδέα, όπως και στον MP, για την αναγνώριση των ατόμων που εξασφαλίζουν καλή περιγραφή είναι να δούμε πως συσχετίζονται με το εκάστοτε σφάλμα. Έστω ότι έχουμε το διάνυσμα $\mathbf{y} \in \mathbb{R}^d$, το λεξικό $D \in \mathbb{R}^{N \times K}$ και ένα διάνυσμα $\mathbf{x} \in \mathbb{R}^K$ με φορέα $\text{supp}(\mathbf{x})^t := S^t = \{k : x_k \neq 0\}$. Τότε, ξεκινώντας ($t = 0$) με $S^0 = \emptyset$, έχουμε

$$\mathbf{y} = D_{S^t} \mathbf{x}_{S^t} + \mathbf{r}_t$$

όπου $\mathbf{r}_t = [r_t^1 \ r_t^2 \ \dots \ r_t^d]$ το σφάλμα προσέγγισης την χρονική στιγμή t , για το οποίο την χρονική στιγμή $t = 0$ ισχύει:

$$\mathbf{r}_0 = \mathbf{y} - D_{S^0} \mathbf{x}_{S^0} = \mathbf{y}.$$

Μπορούμε να βρούμε το άτομο του λεξικού που έχει μεγαλύτερη συσχέτιση με το σφάλμα εάν εξετάσουμε όλα τα εσωτερικά γινόμενα του σφάλματος με τις στήλες του D . Έστω ότι το άτομο k^* έχει τη μεγαλύτερη κατ' απόλυτη τιμή συσχέτιση.

$$k_t = \arg \max_{k=1, \dots, K} \{|\mathbf{d}_k^T \mathbf{r}_t|\}$$

Τότε, εντάσσουμε το k^* στον φορέα του \mathbf{x} :

$$S^t = S^{t-1} \cup k^*$$

και προβάλλουμε τα δεδομένα στον χώρο που παράγουν οι επιλεγμένες στήλες του λεξικού. Το αποτέλεσμα:

$$\mathbf{r}_t = (I - P_{S^t}) \mathbf{y},$$

αποτελεί το νέο σφάλμα προσέγγισης, όπου $(I - P_{S^t})$ το μητρώο ορθογώνιας προβολής στον χώρο στηλών του D_{S^t} ,

$$P_{S^t} = D_{S^t} (D_{S^t}^T D_{S^t})^{-1} D_{S^t}^T,$$

με $D_{S^t} = \{\mathbf{d}_k : k \in S^t\}$.

Η διαδικασία τερματίζει όταν ικανοποιηθεί το κριτήριο τερματισμού που, όπως θέσαμε, είναι ο φορέας του x , S , να έχει το πολύ T_0 μη μηδενικά στοιχεία.

Συνολικά, η παραπάνω διαδικασία βρίσκει τα άτομα του λεξικού και τα κατάλληλα βάρη τους ώστε όταν συνδυάζονται να δίνουν το ελάχιστο δυνατό σφάλμα, τοπικά: στην περιοχή έναρξης του αλγορίθμου. Αυτό αποτελεί και το μεγάλο μειονέκτημα της Αναζήτησης Βάσης. Όταν επιλέγεται ένα άτομο από το λεξικό ως το πλέον συσχετιζόμενο με το τρέχον σφάλμα, τότε η ορθογώνια προβολή των δεδομένων επάνω στον υπόχωρο που παράγει το άτομο αυτό, καθορίζει ποιά θα είναι τα υπόλοιπα άτομα προς επιλογή. Αποτέλεσμα αυτού είναι ο εγκλωβισμός της αναζήτησης σε μία περιοχή την οποία καθορίζει το πρώτο επιλεχθέν άτομο. Το ολικό ελάχιστο της συνάρτησης σφάλματος δεν είναι απαραίτητο να βρίσκεται σε αυτή την περιοχή. Εάν, ωστόσο η διαδικασία ξεκινήσει σε περιοχή γύρω από το ολικό ελάχιστο, τότε ο OMP θα το εντοπίσει δίνοντας την βέλτιστη δυνατή λύση.

Ορθότητα του OMP

Για να συγκλίνει ο αλγόριθμος OMP θα πρέπει το σφάλμα ανακατασκευής να τείνει στο 0. Αυτό προϋποθέτει ο αλγόριθμος σε κάθε επανάληψη να επιλέγει το σωστό άτομο. Βάση για να τηρείται αυτό είναι η Συνθήκη Ακριβούς Ανακατασκευής (Exact Recovery Condition) [2]. Σύμφωνα με αυτή πρέπει στο τέλος του αλγορίθμου να έχει βρεθεί η αραιή λύση που θα ανακατασκευάζει βέλτιστα το διάνυσμα \mathbf{y} . Με άλλα λόγια, εάν ο βέλτιστος φορέας του \mathbf{x} είναι ο Λ τότε πρέπει:

$$S^{t_{end}} \equiv \Lambda, \text{ για } t = t_{end}$$

Άρα πρέπει να ορίσουμε μία μετρική που να εκφράζει την δυνατότητα ακριβούς ανακατασκευής των δεδομένων. Έστω ότι για ένα \mathbf{y} έχουμε επιλέξει ένα αραιό διάνυσμα \mathbf{x} με $\text{supp}(\mathbf{x}) = S_{OPT}$. Ορίζουμε τις ποσότητες:

$$D_{OPT} := \{\mathbf{d}_k : k \in S_{OPT}\}$$

και

$$D_{comp} := \{\mathbf{d}_k : k \in S_{comp}\}$$

με $S_{comp} := \{k : k \notin S_{OPT}\}$. Προφανώς ισχύει $D = D_{OPT} \cup D_{comp}$. Θέλουμε να δούμε κατά πόσο τα άτομα που επιλέχθηκαν, από τον OMP, μπορούν να περιγράψουν τα δεδομένα, συγκριτικά με τα υπόλοιπα άτομα του λεξικού. Για αυτό ορίζουμε τον λόγο:

$$\rho(\mathbf{y}) := \frac{\|D_{comp}^T \mathbf{y}\|_{\infty}}{\|D_{OPT}^T \mathbf{y}\|_{\infty}} \quad (3.3)$$

Η παραπάνω ποσότητα θα μπορούσαμε να πούμε ότι είναι ένα μέτρο συνεκτικότητας του λεξικού. Παρατηρούμε ότι εάν ο λόγος αυτός πλησιάζει την μονάδα, τότε τα άτομα που δεν επέλεξε ο OMP θα μπορούσαν να δώσουν μία εξίσου καλή ανακατασκευή του \mathbf{y} . Κάτι τέτοιο δεν θα θέλαμε να συμβαίνει, καθώς συνεπάγεται πως το λεξικό τείνει να αναπαριστά με μικρό σφάλμα μόνο μία κατηγορία σημάτων, όμοιων του \mathbf{y} . Θα αναφερθούμε περισσότερο στα ποιοτικά χαρακτηριστικά του λεξικού στην συνέχεια. Μέχρι στιγμής, ωστόσο, θα καλυφθούμε από την Συνθήκη Ακριβούς Ανακατασκευής [2]. Για σήμα \mathbf{y} , λοιπόν, δεδομένου ενός λεξικού D και των διαμερίσεων του D_{OPT} και D_{comp} , ο OMP θα εντοπίσει ένα κατάλληλο για ανακατασκευή άτομο αν και μόνο αν:

$$\rho(\mathbf{y}) < 1$$

Έχοντας υπόψιν ότι το μητρώο ορθογωνίας προβολής στις στήλες του D_{OPT} είναι συμμετρικό, δηλαδή

$$(D_{OPT}^+)^T D_{OPT}^T = D_{OPT} D_{OPT}^+ = I$$

μπορούμε να εξάγουμε ένα άνω όριο για τον $\rho(\mathbf{y})$ ως εξής:

$$\begin{aligned} \rho(\mathbf{y}) &= \frac{\|D_{comp}^T \mathbf{y}\|_\infty}{\|D_{OPT}^T \mathbf{y}\|_\infty} \\ &= \frac{\|D_{comp}^T (D_{OPT}^+)^T D_{OPT}^T \mathbf{y}\|_\infty}{\|D_{OPT}^T \mathbf{y}\|_\infty} \\ &\leq \|D_{comp}^T (D_{OPT}^+)^T\|_{\infty, \infty} \end{aligned}$$

Εφόσον ισχύει $\|A\|_{\infty, \infty} = \max_i \{\sum_j A_{ij}\} = \max_j \{\sum_i A_{ij}^T\} = \|A^T\|_{1,1}$, έχουμε:

$$\begin{aligned} \rho(\mathbf{y}) &\leq \|D_{OPT} (D_{comp}^+)\|_{1,1} \\ &= \max_{k \in S_{comp}} \{\|D_{OPT}^+ \mathbf{d}_k\|_1\} \end{aligned}$$

Συνεπώς τώρα μπορούμε να γράψουμε την συνθήκη ως

$$\rho(\mathbf{y}) < 1 \Rightarrow \max_{k \in S_{comp}} \{\|D_{OPT}^+ \mathbf{d}_k\|_1\} < 1$$

Ορίζουμε τέλος τον Exact Reconstruction Coefficient

$$ERC(S_{OPT}; D) := 1 - \max_{k \in S_{OPT}} \|D_{S_{OPT}}^+ \mathbf{d}_k\|_1 \quad (3.4)$$

οπότε καταλήγουμε σε μία ισοδύναμη συνθήκη επιλογής σωστών ατόμων από τον OMP.

$$\max_{k \in S_{comp}} \{\|D_{OPT}^+ \mathbf{d}_k\|_1\} < 1$$

ή ισοδύναμα

$$ERC(S_{OPT}; D) > 0$$

Το πρόβλημα με όλα τα παραπάνω ως προς την σωστή λειτουργία του OMP είναι η προϋπόθεση ότι γνωρίζουμε εκ των προτέρων το S_{OPT} , κάτι το οποίο είναι αδύνατο. Άρα και η σωστή επιλογή ατόμων είναι μία μη ελέγξιμη, εκ των προτέρων, διαδικασία. Για αυτό μπορούμε να μετασχηματίσουμε τις συνθήκες σωστής επιλογής ατόμων, σε συνθήκες ανάλογες της από κοινού μη συνεκτικότητας (mutual incoherence):

$$\mu := \max_{i \neq j} | \langle D_i, D_j \rangle |$$

που ουσιαστική είναι η μέγιστη συσχέτιση μεταξύ των ατόμων του λεξικού. Τώρα έρχεται στο προσκήνιο μία άλλη ιδιότητα, η Ιδιότητα της Από Κοινού Μη Συνεκτικότητας (Mutual Incoherence Property) [3], η οποία υποχρεώνει το μ σε μικρές τιμές, διατηρώντας το λεξικό όσο πιο μη συνεκτικό γίνεται, εξασφαλίζοντας την ποικιλία λέξεων

του. Στη σχετική ερευνητική εργασία, εξάγονται διάφορες συνθήκες βασισμένες σε αυτή την ιδιότητα για διάφορα σενάρια θορύβου. Δηλαδή, θεωρούν αντί του περιορισμού που θέσαμε για την αραιότητα, τον περιορισμό του σφάλματος προσέγγισης. Οπότε και καλύπτονται διάφορα σενάρια θορύβου, το καθένα δίνοντας το δική του συνθήκη σωστής λειτουργίας του OMP.

3.1.1.3 Generalized OMP

Το πλαίσιο του gOMP [6] απαιτεί την ίδια μαθηματική περιγραφή με τον OMP, οπότε και την παρακάμπτουμε, αφιερώνοντας χώρο μόνο στην ειδοποιό διαφορά του από τον δεύτερο.

Χαλαρώνοντας, λοιπόν, την συνθήκη επιλογής ατόμων οδηγούμαστε στον Γενικευμένο OMP (gOMP). Εδώ, επιλέγονται τα M πρώτα πιο συσχετιζόμενα με το σφάλμα άτομα. Ωστόσο, ο OMP επαυξάνει τον φορέα του \mathbf{x} κατά 1 σε κάθε επανάληψη, κάτι το οποίο δεν συμβαίνει εδώ. Αντίθετα, αντί να χρησιμοποιήσει όλα τα νέα άτομα που εντόπισε, επαυξάνει το σύνολο των επιλεγμένων ατόμων. Έπειτα, ενημερώνεται το σφάλμα προσέγγισης βρίσκοντας τους συντελεστές \mathbf{x} που ελαχιστοποιούν το τετραγωνικό σφάλμα προσέγγισης και η διαδικασία επαναλαμβάνεται, έως ότου ικανοποιείται η συνθήκη τερματισμού.

Το σημείο στο οποίο υπερτερεί ο gOMP είναι ότι εκτελεί ένα υποπολλαπλάσιο του αριθμού των επαναλήψεων του OMP. Εφόσον κάθε φορά επιλέγονται M νέα άτομα, χρειάζονται T_0/M επαναλήψεις για να έχουμε ακριβώς T_0 μη μηδενικούς συντελεστές στο \mathbf{x} .

Ορθότητα του Generalized OMP

Ο gOMP βασίζεται στην Ιδιότητα Φραγμένης Ισομετρίας (Restricted Isometry Property) [4] για την ορθότητά του. Αναφέρουμε σύντομα ότι για να βρει τουλάχιστον ένα άτομο του D_{OPT} σε κάθε επανάληψή του, πρέπει να ικανοποιείται η συνθήκη:

$$\delta_{MT_0} < \frac{\sqrt{M}}{\sqrt{T_0} + 3\sqrt{M}}$$

για την οποία δ_{MT_0} είναι η Σταθερά Φραγμένης Ισομετρίας (Restricted Isometric Constant) για ένα μητρώο $A \in \mathbb{R}^{n \times m}$

$$\delta_{MT_0} := \inf \left[\delta : (1 - \delta) \|\mathbf{y}\|_2^2 \leq \|A_s \mathbf{y}\|_2^2 \leq (1 + \delta) \|\mathbf{y}\|_2^2 \right], \quad \forall |s| < MT_0, \quad \forall \mathbf{y} \in \mathbb{R}^{|s|}$$

3.1.1.4 α OMP

Όπως είδαμε και νωρίτερα ο OMP χλωαίνει από τον εγκλωβισμό του σε τοπικά ελάχιστα. Δηλαδή, όταν ξεκινά με την επιλογή ενός ατόμου ως αυτό που έχει την μέγιστη προβολή

στα δεδομένα, η επιλογή των επόμενων ατόμων είναι προκαθορισμένη και εφόσον σε μία εκτέλεση του OMP τα δεδομένα δεν μεταβάλλονται, η διαδικασία δεν περιέχει καμία αβεβαιότητα ως προς το αποτέλεσμα της.

Η ιδέα του αOMP [5] είναι να μην επιλέγουμε πάντα αυτό το πρώτο προς επιλογή άτομο, αλλά να το παρακάμπουμε με μία πιθανότητα α . Έτσι, θα διευρύνουμε την περιοχή γύρω από το τοπικό ελάχιστο στο οποίο ενδεχομένως ο OMP θα εκγλωβιστεί, ελπίζοντας ότι η νέα λύση θα βρει μία καλύτερη συνολική λύση από την τετριμμένη. Προφανώς, αυτό δεν είναι πάντοτε σωστό. Δηλαδή, αν η λύση που δίνει ο OMP είναι η βέλτιστη δυνατή (βρισκόμαστε ήδη στην περιοχή του ολικού ελαχίστου της συνάρτησης σφάλματος), τότε ο αOMP θα δώσει εσφαλμένο αποτέλεσμα.

3.1.2 Κυρτή βελτιστοποίηση

Πριν περιγράψουμε σαφώς την κυρτότητα την οποία χρησιμοποιούμε για την αραιή αναπαράσταση είναι βασικό να αναφερθούμε στην πρότερη μορφή του προβλήματος που είναι πολύ γνωστή. Παραμένουμε στο γενικό μοντέλο που θεωρήσαμε για τα δεδομένα:

$$\mathbf{y} = D\mathbf{x} + \mathbf{r}$$

Ordinary Least Squares

Το συγκεκριμένο μοντέλο είναι ίσως από τα πιο διαδεδομένα και η βάση για αυτά που ακολουθούν. Το πρόβλημα είναι ελαχιστοποίηση της παρακάτω συνάρτησης σφάλματος, όπως αναφέρθηκε και προηγουμένως:

$$\|\mathbf{r}\|_2^2 = \|\mathbf{y} - D\mathbf{x}\|_2^2$$

Ουσιαστικά, στόχος (στις 2 διαστάσεις) είναι εύρεση της κλίσης \mathbf{x} της ευθείας $\mathbf{y} - D\mathbf{x} = 0$, που ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα προσέγγισης. Όταν το λεξικό έχει γραμμικά ανεξάρτητες στήλες τότε γνωρίζουμε ότι το $D^T D$ είναι αντιστρέψιμο και η λύση δίνεται ως

$$\begin{aligned}\mathbf{y} &= D\mathbf{x} \\ D^T \mathbf{y} &= D^T D\mathbf{x} \\ \mathbf{x} &= D^+ \mathbf{y}\end{aligned}$$

όπου D^+ ο ψευδοαντίστροφος του μητρώου D .

Όμως στην περίπτωσή μας το λεξικό λόγω υπερπληρότητας του χώρου στηλών του, δίνει μία απειρία λύσεων. Για αυτό θέλουμε να μεν να τηρείται η σχέση $\mathbf{y} = D\mathbf{x}$, δίνοντας όμως παράλληλα ένα \mathbf{x} με την μικρότερη δυνατή ενέργεια.

Κανονικοποίηση Tikhonov

Το μοντέλο εδώ συμπεριλαμβάνει και περιορισμό στην συνολική ενέργειά της λύσης:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \{\|\mathbf{y} - D\mathbf{x}\|_2^2\} \quad s.t. \quad \|\mathbf{x}\|_2^2 \leq T_0$$

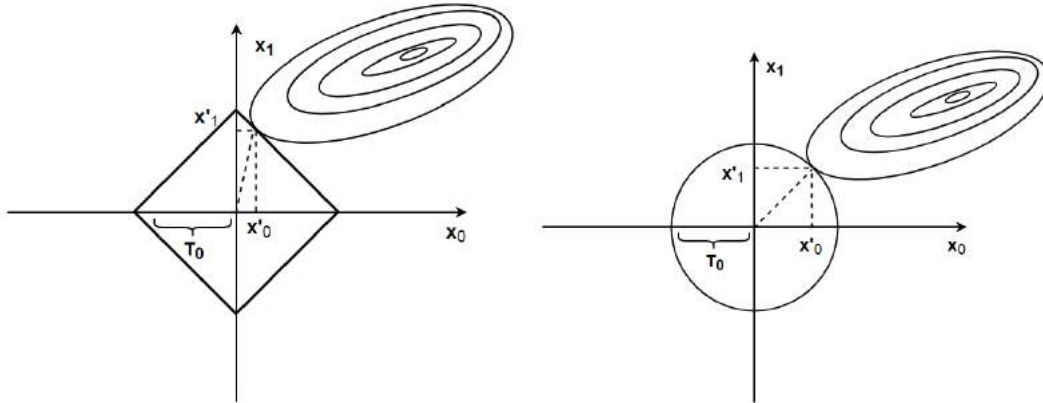
η ισοδύναμα:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \{\|\mathbf{x}\|_2^2\} \quad s.t. \quad \|\mathbf{y} - D\mathbf{x}\|_2 \leq \mathbf{r}$$

διατυπώνοντας στην μορφή Lagrange το πρόβλημα, έχουμε:

$$\mathcal{L}(\mathbf{x}, \lambda) = \|\mathbf{x}\|_2^2 + \lambda^T (\mathbf{y} - D\mathbf{x})$$

όπου πλέον το λ είναι συνάρτηση του T_0 γεφυρώνουν τις δύο εναλλακτικές διατυπώσεις του προβλήματος και ορίζουν το πόσο 'πληρώνουμε' τον περιορισμό στην ενέργεια του \mathbf{x} ή το σφάλμα προσέγγισης, αντίστοιχα. Μία γεωμετρική αναπαράσταση του μοντέλου αυτού φαίνεται στο Σχήμα (3.2).



Σχήμα 3.2: Νόρμες l_1 και l_2 στο ίδιο καρτεσιανό επίπεδο με τις ισοϋψείς καμπύλες της συνάρτησης κόστους. Οι αιχμές που προτάσει η l_1 , οδηγούν με μεγάλη πιθανότητα σε μεγαλύτερη αντίθεση τιμών x'_0, x'_1 , ωθώντας την μία κοντά στο T_0 και την άλλη κοντά στο 0.

Συνεχίζοντας την ελαχιστοποίηση της $\mathcal{L}(\mathbf{x}, \lambda)$, έχουμε:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda) &= 0 \\ \frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{x} + \lambda^T (\mathbf{y} - D\mathbf{x})) &= 0 \\ 2\mathbf{x} - D^T \lambda &= 0 \\ \mathbf{x} &= \frac{1}{2} D^T \lambda \end{aligned} \quad (3.5)$$

$$\begin{aligned}\frac{\partial}{\partial \lambda} \mathcal{L}(\mathbf{x}, \lambda) &= 0 \\ \frac{\partial}{\partial \lambda} (\mathbf{x}^T \mathbf{x} + \lambda^T (\mathbf{y} - D\mathbf{x})) &= 0 \\ \mathbf{y} &= D\mathbf{x}\end{aligned}$$

Αντικαθιστώντας, καταλήγουμε:

$$\begin{aligned}\mathbf{y} &= \frac{1}{2} DD^T \lambda \\ \mathbf{y} &= \frac{1}{2} DD^T \lambda \\ \lambda &= 2(DD^T)^{-1} \mathbf{y}\end{aligned}\tag{3.6}$$

Άρα, αντικαθιστώντας στην Σχέση (3.6) την Σχέση (3.5), έχουμε:

$$\mathbf{x}^* = D^T (DD^T)^{-1} \mathbf{y}$$

Ωστόσο, η νόρμα ℓ_2 παρά την ελαχιστοποίηση της ενέργειας του \mathbf{x} , δεν εξασφαλίζει την αραιότητά του. Για αυτό, και προχωρούμε στην παρακάτω μοντελοποίηση.

3.1.2.1 Least Absolute Shrinkage and Selection Operator

Βασική ιδέα του LASSO είναι η χαλάρωση του περιορισμού αραιότητας από ℓ_0 σε ℓ_1 νόρμα και ελαχιστοποιώντας το άθροισμα της ενέργειας του \mathbf{x} και το μέσο τετραγωνικό σφάλμα προσέγγισης.

$$\mathbf{x}_i^* = \arg \min_{\mathbf{x}_i} \left\{ \frac{1}{N} \sum_{i=1}^N \|\mathbf{y}_i - D\mathbf{x}_i\|_2^2 \right\} \quad s.t. \quad \|\mathbf{x}_i\|_1 \leq T_0 \quad \forall i = 1, \dots, N.$$

Εφόσον η ελαχιστοποίηση του αθροίσματος τετραγώνων έγκειται στην ελαχιστοποίηση κάθε αθροιστέας ποσότητας ξεχωριστά, μπορούμε να διατυπώσουμε το πρόβλημα στη Lagrange μορφή του ως:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - D\mathbf{x}\|_2^2 \right\}\tag{3.7}$$

για κάθε διαφορετικό διάνυσμα δεδομένων \mathbf{y} .

Τώρα, ορίζουμε το διάνυσμα

$$\boldsymbol{\gamma} := \text{sign}(\hat{\mathbf{x}}) = [\text{sign}(\hat{x}_1), \dots, \text{sign}(\hat{x}_K)]^T, \quad \text{sign}(\hat{x}_k) = \begin{cases} \text{sign}(\hat{x}_k) & , \hat{x}_k \neq 0 \\ l \in [-1, 1] & , \hat{x}_k = 0 \end{cases}$$

που καλούμε υποπαράγωγο του $\|\cdot\|_1$, καθώς η κανονική παράγωγός της δεν ορίζεται στο 0. Οι συνθήκες Karush-Kush-Tucker, ορίζουν επιλυσιμότητα του LASSO, όταν η υποπαράγωγος μηδενίζεται. Τέλος, ορίζουμε το σύνολο S ως:

$$S := \{k : \frac{|\mathbf{d}_k^T(\mathbf{y} - D\hat{\mathbf{x}})|}{2} = \lambda\} = \{k : |\gamma_k| = 1\}$$

και το διάνυσμα γ_S ως το υποσύνολο του γ που ορίζει το σύνολο S :

$$\gamma_S := \text{sign}(D_S^T(\mathbf{y} - D\hat{\mathbf{x}}))$$

Οπότε η λύση $\hat{\mathbf{x}}$ δίνεται ως:

$$\hat{\mathbf{x}} = \begin{cases} 0 & , k \notin S \\ \hat{\mathbf{x}}_k : \frac{\mathbf{d}_k^T(\mathbf{y} - D\hat{\mathbf{x}}_k)}{2} = \lambda\gamma_k & , k \in S \end{cases}$$

Άλλες μέθοδοι

Μέχρι τώρα είδαμε άπληστους αλγορίθμους εύρεσης αραιής αναπαράστασης και αλγορίθμους κυρτής βελτιστοποίησης. Ωστόσο, υπάρχουν και άλλες προσεγγίσεις όπως η εκμάθηση κατά Bayes και ο FOCUSS όπου εισάγει την έννοια της σταθμισμένης νόρμας. Δηλαδή, η ενέργεια του x να μην περιορίζεται, αλλά με όχι ομοιόμορφο τρόπο, προσαρμόζοντας της σημαντικότητα του περιορισμού ανά στοιχείο του x .

3.2 Εκμάθηση λεξικού

Μέχρι ώρα θεωρήσαμε το λεξικό ως μία ντετερμινιστική ποσότητα, κάτι το οποίο ήταν εκ των προτέρων γνωστό. Θεωρήσαμε, σε κάθε περίπτωση, ότι είναι υπερπλήρες και ότι έχει γραμμικά ανεξάρτητες γραμμές, ώστε να υπάρχει ο αριστερός αντίστροφός του. Ωστόσο, το λεξικό, όπως προαναφέραμε, είναι μία συλλογή σημάτων τα οποία δεν εκφράζουν μη εσφαλμένα με αραιό τρόπο τα δεδομένα πάντα, ή τουλάχιστον εάν τα εκφράζουν με μικρό σφάλμα, η αραιότητα δεν είναι εγγυημένη. Σε κάθε περίπτωση, δηλαδή, έχουμε να ζυγίσουμε τις έννοιες του σφάλματος και της αραιότητας των αναπαραστάσεων. Όπως είπαμε, λεξικό που να περιγράφει τα πάντα, δεν υπάρχει. Η διαδικασία περιγραφής είναι μία διαδικασία μάθησης. Το λεξικό είτε μαθαίνει και ύστερα εκφράζει, είτε μαθαίνει και καθώς εκφράζει συνεχίζει να μαθαίνει. Σε κάθε περίπτωση, η εκμάθησή του, είναι μια διαδικασία οδηγούμενη από τα δεδομένα η οποία είναι και αυτή που διαμορφώνει το λεξικό. Οι περιορισμοί και τα ειδικά χαρακτηριστικά που θα διέπουν το λεξικό είναι λίγο-πολύ κατά περίπτωση επιλεγόμενα. Πολύ διαδεδομένη ιδιότητα, όπως είδαμε και στον LASSO, είναι το λεξικό να είναι τάξης στηλών όσες και οι γραμμές του. Η ορθογωνιότητα, μπορεί να είναι χρήσιμο εργαλείο για πολλά και

ποικίλα δεδομένα, αλλά υπονομεύει την αραιότητα, ενώ η πολύ μικρές γωνίες μεταξύ των λέξεων, είναι αχρείαστες, καθώς εκφράζουν ξανά και ξανά την ίδια ομάδα δεδομένων.

Ιδιότητες όπως η Από Κοινού Μη Συνεκτικότητα, που συναντήσαμε στον OMP, υποθέτουν μία δομή στο λεξικό, κάτι το οποίο η διαδικασία μάθησης πρέπει να σέβεται. Για αυτό η ενημέρωση του λεξικού και επανεξέταση αραιών λύσεων, αν και μελετώνται ξεχωριστά, είναι δύο συνυφασμένες έννοιες.

Οι βιβλιογραφία χωρίζεται σε τρεις κύριες ενότητες αναφορικά με την εκμάθηση λεξικού: Πιθανοτικές προσεγγίσεις, Μεθόδους Ομαδοποίησης Δεδομένων και Μεθόδους Παραμετρικών Λεξικών. Οι μέθοδοι ομαδοποίησης δεδομένων είναι η ομάδα που εξετάστηκε πειραματικά και θα αναλύσουμε σε βάθος. Για τις άλλες δύο κατηγορίες θα δούμε περιγραφικά τις κύριες ιδέες στις οποίες βασίζονται.

3.2.1 Πιθανοτικοί αλγόριθμοι

Στους πιθανοτικούς αλγορίθμους εισάγεται, με μεγαλύτερο βάρος από ό,τι πριν, η έννοια της πληροφορίας. Όπως έχουμε δει ήδη στα Νευρωνικά Δίκτυα, η χαρτογράφηση της συμπεριφοράς τους γίνεται καλύτερα με όρους πληροφορίας.

Η βασική ιδέα αντικατοπτρίζεται στην δουλειά των Olshausen - Field [1] όπου εισήγαγαν την έννοια της μέγιστης πιθανοφάνειας. Διατύπωσαν το πρόβλημα ως αυτό της εύρεσης του λεξικού που μεγιστοποιεί την δεσμευμένη πιθανότητα κατασκευής των δεδομένων από το συγκεκριμένο λεξικό:

$$D^* = \arg \max_D \{\log \Pr(\mathbf{y}|D)\} \quad (3.8)$$

$$= \arg \max_D \{\log \int_{\mathbf{x}} \Pr(\mathbf{y}|\mathbf{x}, D) \Pr(\mathbf{x}) d\mathbf{x}\} \quad (3.9)$$

Ισοδύναμα και κάτω από κάποιους περιορισμούς ως προς το μοντέλο αναπαράστασης, μετατρέπουν την συνάρτηση ενδιαφέροντος σε μία προς ελαχιστοποίηση συνάρτηση έκφρασης ενέργειας:

$$D^* = \arg \min_{D, \mathbf{x}} \{E(\mathbf{y}, \mathbf{x}|D)\} \quad (3.10)$$

$$= \arg \min_{D, \mathbf{x}} \{-\log(\Pr(\mathbf{y}|\mathbf{x}, D) \Pr(\mathbf{x}))\} \quad (3.11)$$

$$= \arg \min_{D, \mathbf{x}} \{\|\mathbf{y} - D\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1\} \quad (3.12)$$

Παρατηρούμε ότι καταλήγουμε πάλι στο LASSO, οπότε ξέρουμε ήδη την λύση για το \mathbf{x} . Τώρα μπορούμε να θεωρήσουμε σταθερό το D και να υπολογίσουμε ένα αραιό \mathbf{x} ή το αντίστροφο. Ωστόσο, μπορούμε (και έχει αναφερθεί σε αρκετές εργασίες) να αντικαταστήσουμε την εύρεση αραιής αναπαράστασης από κάποιον αλγόριθμο κυρτής

βελτιστοποίησης ή αναζήτησης ταιριάσματος και να διατηρήσουμε μόνο το βήμα ανανέωσης του λεξικού. Στην Μέθοδο των Βέλτιστων Κατευθύνσεων (Method of Optimal Directions), χρησιμοποιείται ο OMP για την εύρεση του x , και έπειτα ελαχιστοποιείται η συνάρτηση ενδιαφέροντος της Μέγιστης Πιθανοφάνειας. Ωστόσο, όπως προαναφερθήκαμε, ο OMP επιφυλάσσει τον κίνδυνο του εγκλωβισμού σε τοπικά ακρότατα.

Μία άλλη ιδέα είναι η αντικατάσταση του περιορισμού αραιότητα ℓ_1 από την απόκλιση Kullback-Leibler [12].

3.2.2 Παραμετρικά Λεξικά

Είδαμε ότι το τέλειο λεξικό δεν υπάρχει, και ανάλογα την εφαρμογή θέλουμε να έχει κάποια χαρακτηριστικά που να το διέπουν. Εδώ υπεισέρχεται και η έννοια της δομής του λεξικού, η οποία αποφασίζεται εκ των προτέρων, κατά περίπτωση. Μέχρι στιγμής και όχι απαραίτητα στον αραιό χώρο, γνωρίζουμε λεξικά όπως Fourier και DCT ή Haar, τα οποία διέπονται από μία συγκεκριμένη δομή. Για παράδειγμα το λεξικό Fourier περιέχει όλες τις αρμονικές συναρτήσεις βάσης, η κάθε μία εκπροσωπώντας μία συχνότητα. Άλλο παράδειγμα είναι τα λεξικά Gammatone που περιέχει αρμονικές συναρτήσεις βάσης πολλαπλασιασμένες με μία παραμετροποιημένη εκδοχή της κατανομής Γάμμα. Και στις δύο περιπτώσεις, είναι προφανές πως κοντινά άτομα, παραπλήσιας συχνότητας, συσχετίζονται κατά πολύ. Αυτό συμβαίνει κατά όλο το μήκος του υπερπλήρους λεξικού, το οποίο συνεπάγεται υψηλή συνοχή στο λεξικό, πράγμα που προσπαθήσαμε να αποφύγουμε σε κάθε προσέγγιση μέχρι τώρα, χάριν αραιότητας.

3.2.3 Λεξικά ομαδοποίησης

Ίσως τα πιο ευρέως διαδεδομένα λεξικά, κυρίως λόγω ποικίλων εφαρμογών, είναι αυτά που βασίζονται στην κβάντιση της πληροφορίας. Όπως είδαμε τα δεδομένα είναι τοποθετημένα μέσα στον \mathbb{R}^d , με άστατο τρόπο. Σκοπός του λεξικού είναι να βρει ποιές είναι οι κύριες ομάδες λέξεων, οι οποίες μπορούν να αντιπροσωπευθούν από μόλις μερικές λέξεις, ελαχιστοποιώντας το σφάλμα προσέγγισης. Ουσιαστικά προσπαθούμε να αντιστοιχίσουμε τα δεδομένα σε αντιπροσωπευτικά δεδομένα, ή αλλιώς να εφαρμόσουμε κβάντιση των διανυσμάτων.

Στην πραγματικότητα γνωρίζουμε ήδη μία διαδικασία που κάνει ακριβώς αυτό: τον αλγόριθμο K-means. Η βασική ιδέα πίσω από αυτόν είναι η ομαδοποίηση των δεδομένων σε $K < d$ ομάδες όπου κάθε μία αντιπροσωπεύεται από το μέσο διάνυσμα κάθε ομάδας. Αν απλουστεύσουμε τον OMP, ορίζοντας $T_0 = 1$, τότε η όλη διαδικασία μετασχηματίζεται στον K-means. Αν δούμε πιο αναλυτικά την διαδικασία, βλέπουμε ποιό άτομο του λεξικού ευθύνεται περισσότερο για το σφάλμα προσέγγισης (αρχικά $\epsilon = \mathbf{y}$), δηλαδή είναι πιο κοντά στα δεδομένα, και έπειτα κρατείται μόνο αυτό ως το αντιπροσωπευτικό

του δεδομένου \mathbf{y} . Όσο εξετάζουμε τα δεδομένα όλο και περισσότερα \mathbf{y} αντιστοιχίζονται στο άτομο αυτό, δημιουργώντας εν τέλει μία ομάδα.

3.2.3.1 K-SVD

Ο K-SVD [9] είναι ο πιο διαδεδομένος αλγόριθμος και αποτελεί σημείο αναφοράς για τους αλγόριθμους εκμάθησης λεξικού. Βασική του ιδέα είναι η γενίκευση του K-means, χρησιμοποιώντας την SVD για την διατήρηση της αραιότητας της αραιής αναπαράστασης ή και την μείωσή της.

Ας ξαναγράψουμε, για ευκολίας μας, το πρόβλημα βελτιστοποίησης το οποίο θέλουμε να επιλύσουμε:

$$D^*, X^* = \arg \min_{D, X} \{\|Y - DX\|_F^2\}, \quad s.t. \|\mathbf{x}_i\|_0 \leq T_0 \quad \forall i = 1, \dots, N$$

Το υποπρόβλημα ελαχιστοποίησης ως προς X , διατηρώντας το D σταθερό, το έχουμε αναλύσει διεξοδικά νωρίτερα, οπότε επικεντρωνόμαστε στο δεύτερο υποπρόβλημα ελαχιστοποίησης ως προς D . Αναλύοντας την νόρμα Frobenius γνωρίζουμε ότι:

$$\|Y - DX\|_F^2 = \sum_{i=1}^N \|\mathbf{y}_i - D\mathbf{x}_i\|_2^2$$

διότι για μητρώο $A \in \mathbb{R}^{M \times N}$ ισχύει:

$$\|A\|_F^2 = \sqrt{\text{Tr}(AA^T)}^2 = \text{tr}(A^T A) = \sum_{i=1}^N \|\mathbf{a}_i\|_2^2.$$

Οπότε μπορούμε να αποσυνθέσουμε το πρόβλημα ελαχιστοποίησης ως ελαχιστοποίηση του αθροίσματος θετικών ποσοτήτων, ή καλύτερα την ελαχιστοποίηση κάθε αθροιστέας ποσότητας ξεχωριστά. Άρα, εφόσον κοιτάμε το σφάλμα ως συνάρτηση κάθε διανύσματος δεδομένων ξεχωριστά μπορούμε να εξετάσουμε ένα ένα τα άτομα του λεξικού και να τα ενημερώσουμε ανάλογα. Άρα είναι αναγκαίο να εκφράσουμε την συνάρτηση ενδιαφέροντος, ως συνάρτηση του προς ενημέρωση ατόμου k :

$$\begin{aligned} \|Y - DX\|_F^2 &= \left\| Y - \sum_{j=1}^K \mathbf{d}_j \mathbf{x}_T^j \right\|_F^2 \\ &= \left\| Y - \sum_{j \neq k}^K \mathbf{d}_j \mathbf{x}_T^j - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 \\ &= \left\| E_k - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 \end{aligned} \quad (3.13)$$

Πλέον το σφάλμα διασπάται στο κομμάτι το οποίο οφείλεται στο k -οστό άτομο το εναπομείναν σφάλμα τάξης 1. Η έξυπνη ιδέα του K-SVD είναι να ενημερώσουμε το μητρώο τάξης 1, $\mathbf{d}_k \mathbf{x}_T^k$, ως την πιο καλή προσέγγισης τάξης 1 που μπορούμε να κάνουμε για το E_k . Γνωρίζουμε ήδη ότι το E_k , όπως και κάθε μητρώο τάξης r μπορεί να αναλυθεί σε άθροισμα r μητρώων τάξης 1, με το πρώτο να είναι αυτό με την μεγαλύτερη ιδιάζουσα τιμή, το δεύτερο με την αμέσως μικρότερη κ.ο.κ. Αυτή η ανάλυση είναι η SVD και οι προαναφερθέντες ιδιάζουσες τιμές αφορούν τον E_k . Δηλαδή:

$$E_k = \sum_{r=1}^{\text{rank}(E_k)} \sigma_r \mathbf{u}_r \mathbf{v}_r^T.$$

Ωστόσο, η απευθείας εφαρμογή ανάλυσης ιδιάζουσων τιμών, θα αναλύσει το E_k σε μητρώα τάξης 1 τα οποία δεν είναι αραιά, όσον αφορά τα \mathbf{v}_r^T . Οπότε μία λύση που προτείνεται είναι η εξής: Να επιλέξουμε μόνο τα δεδομένα που αφορούν το προς ενημέρωση άτομο και να εξάγουμε ένα νέο, μειωμένων διαστάσεων μητρώο σφάλματος, το οποίο θα αναλύσουμε σε ιδιάζουσες συνιστώσες. Τότε, μπορούμε να ενημερώσουμε το x_T^k μόνο στις θέσεις όπου ούτως ή άλλως είχε μη μηδενικά στοιχεία. Έτσι η αραιότητα του είτε θα παραμείνει ίδια, είτε στην καλύτερη περίπτωση θα μειωθεί κι άλλο. Δηλαδή, ορίζουμε τα υπομητρώα:

$$x_R^k = \{x_T^k(i) : x_T^k(i) \neq 0\}, \quad E_k^R = \{\mathbf{e}_i : x_T^k(i) \neq 0\}$$

Οπότε τώρα μπορούμε να γράψουμε αναλύσουμε το E_k^R στις ιδιάζουσες συνιστώσες του. Η διαφορά τώρα είναι ότι όταν επιλέξουμε \mathbf{x}_R^k ως την πρώτη ιδιάζουσα συνιστώσα του χώρου γραμμών, αυτή δεν θα είναι αραιή. Αλλά εάν τα στοιχεία του επαναδιαμοιραστούν στις αρχικές επιλεγμένες θέσεις του \mathbf{x}_T^k , τότε το νέο $\hat{\mathbf{x}}_T^k$ θα είναι αραιό. Οπότε τα ανανεωμένα $\hat{\mathbf{d}}_k$ και $\hat{\mathbf{x}}_R^k$ είναι:

$$\hat{\mathbf{d}}_k = \mathbf{u}_1 \tag{3.14}$$

$$\hat{\mathbf{x}}_R^k = \sigma_1 \mathbf{v}_1^T \tag{3.15}$$

έτσι ώστε $\mathbf{u}_1 \sigma_1 \mathbf{v}_1^T$ να είναι το πλησιέστερο μητρώο τάξης 1 στο E_k^R . Άρα για το $\hat{\mathbf{x}}_T^k$ έχουμε:

$$\hat{\mathbf{x}}_T^k = \{\hat{\mathbf{x}}_R^k(i) : x_T^k(i) \neq 0\}$$

Με τον τρόπο αυτό ικανοποιούμε την συνθήκη:

$$\text{supp}(\hat{\mathbf{x}}_T^k) \equiv \text{supp}(\mathbf{x}_T^k)$$

Προϋποθέσεις για τον K-SVD

Όπως και σε κάθε άλλο αλγόριθμο που έχουμε αναφερθεί μέχρι τώρα τα άτομα του λεξικού θεωρούνται κανονικοποιημένα ως προς την ℓ_2 νόρμα. Ας μην ξεχνάμε ότι η ποιότητα του λεξικού έγκειται στην Ιδιότητα Αμοιβαίας Μη Συνεχτικότητας που εγγυάται την σωστή επιλογή αραιής αναπαράστασης. Άρα τα εσωτερικά γινόμενα των ατόμων παίζουν τον καθοριστικό ρόλο στις αραιές αναπαραστάσεις, Για να είναι δίκαιες λοιπόν οι συγκρίσεις μεταξύ ατόμων πρέπει το μέτρο τους να είναι το ίδιο (χάριν ευκολίας να είναι ίσο με 1). Άρα όταν ενημερώνουμε τα άτομα του λεξικού, είναι σημαντικό να διατηρούμε το χαρακτηριστικό αυτό. Η SVD παράγει ορθοκανονικά διανύσματα βάσης του χώρου στηλών και του χώρου γραμμών, οπότε οι και τα άτομα όντως διατηρούν το μήκος τους στη ροή του αλγορίθμου. Επίσης, ο K-SVD μειονεκτεί στο ότι μπορεί να παγιδευτεί σε τοπικά ακρότατα λόγω κακής αρχικοποίησης του λεξικού. Ο OMP είναι αυτός που αναλαμβάνει αυτό το κομμάτι. Ωστόσο στην πράξη φαίνεται να δουλεύει με πολύ καλά αποτελέσματα και ανάλογα την εφαρμογή που τον χρησιμοποιεί. Άρα μία καλή επιλογή αρχικοποίησης λεξικού αποτελεί προϋπόθεση για την σίγουρη σύγκλιση του αλγορίθμου στο ολικό ελάχιστο της συνάρτησης κόστους.

3.2.3.2 Total Least Squares

Ο TLS [13] ακολουθεί την ίδια αρχή με τον K-SVD, ότι δηλαδή η καλύτερη προσέγγιση ενθόρυβων δεδομένων είναι αυτή που διατηρεί τον ίδιο χώρο γραμμών και στηλών, μηδενίζοντας τις τελευταίες $N - K$ ιδιάζουσες τιμές, όπου N είναι το πλήθος των υλοποιήσεων Y (Eckart-Young [1936]). Πιο αναλυτικά, για το κλασσικό μοντέλο $Y = DX$ θεωρούμε ότι έχουμε μία ενθόρυβη εκδοχή του που θέλουμε να βελτιστοποιήσουμε ως προς το D . Δηλαδή:

$$Y + Y_\epsilon = (D + D_\epsilon)X, \quad Y, Y_\epsilon \in \mathbb{R}^{d \times N}, \quad D, D_\epsilon \in \mathbb{R}^{d \times K}, \quad X \in \mathbb{R}^{K \times N}$$

ή ισοδύναμα:

$$\begin{bmatrix} X^T & Y^T \end{bmatrix} \begin{bmatrix} D^T \\ -I \end{bmatrix} = 0$$

Τώρα μπορούμε να αναλύσουμε το $\begin{bmatrix} X^T & Y^T \end{bmatrix}$ ως

$$\begin{bmatrix} X^T & Y^T \end{bmatrix} = \begin{bmatrix} U_X & U_Y \end{bmatrix} \begin{bmatrix} S_X & 0 \\ 0 & S_Y \end{bmatrix} \begin{bmatrix} V_{XX} & V_{XY} \\ V_{YX} & V_{YY} \end{bmatrix}^T \quad (3.16)$$

Για το οποίο θέλουμε να ισχύει:

$$\begin{bmatrix} X^T + X_\epsilon^T & Y^T + Y_\epsilon^T \end{bmatrix} = \begin{bmatrix} U_X & U_Y \end{bmatrix} \begin{bmatrix} S_X & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{XX} & V_{XY} \\ V_{YX} & V_{YY} \end{bmatrix}^T \quad (3.17)$$

ή ισοδύναμα:

$$\begin{aligned} \begin{bmatrix} X_\epsilon^T & Y_\epsilon^T \end{bmatrix} &= \begin{bmatrix} U_X & U_Y \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -S_Y \end{bmatrix} \begin{bmatrix} V_{XX} & V_{XY} \\ V_{YX} & V_{YY} \end{bmatrix}^T \\ &= -U_Y S_Y \begin{bmatrix} V_{YX} \\ V_{YY} \end{bmatrix}^T \\ &= -\begin{bmatrix} X & Y \end{bmatrix} \begin{bmatrix} V_{YX} \\ V_{YY} \end{bmatrix} \begin{bmatrix} V_{YX} \\ V_{YY} \end{bmatrix}^T \end{aligned}$$

Οπότε κατασκευάσαμε X_ϵ^T και Y_ϵ^T ώστε:

$$\begin{bmatrix} X^T + X_\epsilon^T & Y^T + Y_\epsilon^T \end{bmatrix} \begin{bmatrix} V_{YX} \\ V_{YY} \end{bmatrix} = 0 \quad (3.18)$$

Εάν τώρα το V_{YY} είναι αντιστρέψιμο, μπορούμε να πολλαπλασιάσουμε από δεξιά με $-V_{YY}^{-1}$, παίρνοντας:

$$\begin{bmatrix} X^T + X_\epsilon^T & Y^T + Y_\epsilon^T \end{bmatrix} \begin{bmatrix} -V_{YX}V_{YY}^{-1} \\ -V_{YY} - V_{YY}^{-1} \end{bmatrix} = 0 \quad (3.19)$$

$$\begin{bmatrix} X^T + X_\epsilon^T & Y^T + Y_\epsilon^T \end{bmatrix} \begin{bmatrix} -V_{YX}V_{YY}^{-1} \\ -I \end{bmatrix} = 0 \quad (3.20)$$

Άρα αρκεί να διαλέξουμε:

$$D^T = -V_{XY}V_{YY}^{-1}$$

ώστε να ελαχιστοποιείται η νόρμα Frobenius του επαυξημένου μητρώου $\begin{bmatrix} X^T & Y^T \end{bmatrix}$.

3.2.3.3 Approximate K-SVD

Πέραν του μεγαλύτερου προβλήματος της K-SVD, που είναι να παγιδευτεί σε τοπικά ελάχιστα, έχει ένα περαιτέρω και μικρότερης σημασίας πρόβλημα υπολογιστικής πολυπλοκότητας, λόγω ότι σε κάθε επανάληψή του χρειάζεται να εκτελεί K φορές ανάλυση ιδιάζουσων τιμών. Αν και ο OMP είναι αυτός που κατέχει το κύριο μέρος του χρόνου εκτέλεσης, όπως θα δούμε και παρακάτω πειραματικά, μία πιο οικονομική λύση στο στάδιο ενημέρωσης του λεξικού μπορεί να προσφέρει σημαντικά μικρότερους συνολικούς χρόνους εκτέλεσης.

Εδώ εισάγουμε την προσεγγιστική λύση του Approximate K-SVD [11]. Ουσιαστικά, γνωρίζουμε ότι οι καλύτερη προσέγγιση των δεδομένων του σφάλματος από μητρώο τάξης 1 είναι το πρώτο ιδιάζων μητρώο που προσφέρει η SVD. Οπότε, σίγουρα για την

ίδια συνάρτηση σφάλματος δεν μπορούμε να κάνουμε κάτι καλύτερο. Μπορούμε ωστόσο να γλιτώσουμε χρόνο. Αυτό μπορεί να επιτευχθεί εάν προσπαθήσουμε να εφαρμόσουμε την ελαχίστων τετραγώνων λύση για κάθε ένα άτομο του λεξικού. Δηλαδή, εφόσον έχουμε στην διάθεσή μας ένα αραιό \mathbf{x} για κάθε δεδομένο \mathbf{y} , μπορούμε να παραγωγίσουμε την συνάρτηση κόστους ως προς κάθε ένα \mathbf{d} , θεωρώντας όλο το υπόλοιπο λεξικό ντετερμινιστικό. Έπειτα, εφόσον ενημερώσουμε το λεξικό, μπορούμε να ενημερώσουμε και την αραιή αναπαράσταση με την ίδια λογική. Ωστόσο, για την ενημέρωση του λεξικού διατηρούμε την ιδέα του υπολογισμού του σφάλματος μόνο για τα δεδομένα τα οποία χρησιμοποιούν το τρέχον άτομο του λεξικού προς ενημέρωση. Δηλαδή, λύνουμε κατευθείαν το πρόβλημα:

$$\begin{aligned} \left\| Y_R - \mathbf{d}_k \mathbf{x}_R^k \right\|_F^2 &= \text{Tr}((E_k^R - \mathbf{d}_k \mathbf{x}_R^k)^T (E_k^R - \mathbf{d}_k \mathbf{x}_R^k)) \\ &= \text{Tr}(E_k^{RT} E_k^R - E_k^{RT} \mathbf{d}_k \mathbf{x}_R^k - \mathbf{x}_R^{kT} \mathbf{d}_k^T E_k^R + \mathbf{x}_R^{kT} \mathbf{d}_k^T \mathbf{d}_k \mathbf{x}_R^k) \\ &= \text{Tr}(E_k^{RT} E_k^R) - 2\text{Tr}(\mathbf{x}_R^k E_k^{RT} \mathbf{d}_k) + \text{Tr}(\left\| \mathbf{x}_R^k \right\|_2^2 \left\| \mathbf{d}_k \right\|_2^2) \end{aligned}$$

Οπότε παραγωγίζοντας ως προς \mathbf{d}_k και εξισώνοντας με το 0, παίρνουμε:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{d}_k} [\text{Tr}(E_k^{RT} E_k^R) - 2\text{Tr}(\mathbf{x}_R^k E_k^{RT} \mathbf{d}_k) + \text{Tr}(\left\| \mathbf{x}_R^k \right\|_2^2 \left\| \mathbf{d}_k \right\|_2^2)] &= 0 \\ 0 - 2E_k^{RT} \mathbf{x}_R^{kT} + 2 \left\| \mathbf{x}_R^k \right\|_2^2 \mathbf{d}_k &= 0 \\ \mathbf{d}_k &= \frac{E_k^{RT} \mathbf{x}_R^{kT}}{\left\| \mathbf{x}_R^k \right\|_2^2} \end{aligned}$$

Συνεχίζοντας με την ελαχιστοποίηση ως προς \mathbf{x}_R^k :

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_R^k} [\text{Tr}(E_k^{RT} E_k^R) - 2\text{Tr}(\mathbf{x}_R^k E_k^{RT} \mathbf{d}_k) + \text{Tr}(\left\| \mathbf{x}_R^k \right\|_2^2 \left\| \mathbf{d}_k \right\|_2^2)] &= 0 \\ 0 - 2\mathbf{d}_k^T E_k^R + 2 \left\| \mathbf{x}_R^k \right\|_2^2 \mathbf{d}_k^T &= 0 \\ \mathbf{x}_R^k &= \frac{\mathbf{d}_k^T E_k^R}{\left\| \mathbf{d}_k \right\|_2^2} \\ \mathbf{x}_R^k &= \mathbf{d}_k^T E_k^R \end{aligned}$$

Οπότε καταλήξαμε σε μία πιο υπολογιστικά φθηνή ενημέρωση τόσο των ατόμων του λεξικού όσων και της αραιής αναπαράστασης.

3.2.3.4 Least Squares Coefficient Persistent

Μία ακόμη πιο φθηνή προσέγγιση του K-SVD και υποπερίπτωση της Approximate K-SVD μπορούμε προτείνουμε για μείωση του υπολογιστικού κόστους. Μετά από τον

Αναζήτηση Ταιριάσματος, μπορούμε απευθείας να ελαχιστοποιήσουμε την συνάρτηση σφάλματος, χωρίς να διατηρήσουμε τις θέσεις των δεδομένων που χρησιμοποιεί κάθε άτομο. Αυτό βέβαια θα αλλοιώσει το βήμα της ενημέρωσης της αραιής αναπαράστασης αλλάζοντας τα ήδη μηδενικά στοιχεία. Ωστόσο, μπορούμε να επιμείνουμε στην αραιή αναπαράσταση που έδωσε ο OMP και απλώς να ενημερώσουμε τα άτομα του λεξικού. Δηλαδή:

$$\begin{aligned} \|Y - \mathbf{d}_k \mathbf{x}_T^k\|_F^2 &= \text{Tr}((E_k - \mathbf{d}_k \mathbf{x}_T^k)^T (E_k - \mathbf{d}_k \mathbf{x}_T^k)) \\ &= \text{Tr}(E_k^T E_k - E_k^T \mathbf{d}_k \mathbf{x}_T^k - \mathbf{x}_T^{kT} \mathbf{d}_k^T E_k + \mathbf{x}_T^{kT} \mathbf{d}_k^T \mathbf{d}_k \mathbf{x}_T^k) \\ &= \text{Tr}(E_k^T E_k) - 2\text{Tr}(\mathbf{x}_T^k E_k^T \mathbf{d}_k) + \text{Tr}(\|\mathbf{x}_T^k\|_2^2 \|\mathbf{d}_k\|_2^2) \end{aligned}$$

και ομοίως με πριν παραγωγίζουμε και εξισώνουμε με το 0:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{d}_k} [\text{Tr}(E_k^T E_k) - 2\text{Tr}(\mathbf{x}_T^k E_k^T \mathbf{d}_k) + \text{Tr}(\|\mathbf{x}_T^k\|_2^2 \|\mathbf{d}_k\|_2^2)] &= 0 \\ -2E_k^T \mathbf{x}_T^{kT} + 2\|\mathbf{x}_T^k\|_2^2 \mathbf{d}_k &= 0 \end{aligned}$$

Θα διαπιστώσουμε και παρακάτω πειραματικά ότι μία τέτοια εκμάθηση λεξικού οδηγεί μεν σε μεγαλύτερο σφάλμα σύγκλισης, αλλά συγκλίνει πολύ πιο γρήγορα.

3.2.3.5 Recursive Least Squares

Μέχρι τώρα έχουμε δει αλγόριθμους μάθησης για τους οποίους θεωρούμε δεδομένο έναν όγκο πληροφορίας Y το οποίο διαχειριζόμαστε ολικά με σκοπό την εκπαίδευση του λεξικού. Ωστόσο, δεν είναι πάντα τέτοιες οι περιστάσεις. Πολλές φορές, είναι αναγκαίο να προσαρμόζουμε ένα ένα τα δεδομένα στην διαδικασία μάθησης, είτε διότι θέλουμε μία on the fly διαδικασία, λόγω υπολογιστικών πόρων, είτε διότι θέλουμε το λεξικό να προσαρμόζεται συνεχώς σε νέα δεδομένα, αγνοώντας τα κατά πολύ παλαιότερα. Ένας κατάλληλος αλγόριθμος για την επίλυση αυτής της μορφής προβλημάτων είναι ο RLS.

$$\begin{aligned} DX &= Y \\ \Rightarrow DXX^T &= YX^T \\ \Rightarrow D &= (YX^T)(XX^T)^{-1} \\ \Rightarrow D &= BC \end{aligned}$$

ώστε $B = YX^T, C = (XX^T)^{-1}$. Τώρα μπορούμε να υπολογίσουμε για ένα νέο δεδομένο \mathbf{y} την αλλαγή που θα επιφέρει στα μητρώα B, C και επομένως στο λεξικό D .

$$B_{i+1} = B_i + \mathbf{y}\mathbf{x}^T$$

$$C_{i+1} = (C_i^{-1} + \mathbf{x}\mathbf{x}^T)^{-1} = C_i - \frac{C_i\mathbf{x}\mathbf{x}^T C_i}{\mathbf{x}^T C_i \mathbf{x}}$$

από την ταυτότητα Sherman-Morrison. Επίσης, λόγω της συμμετρίας του C_i , μπορούμε να ορίσουμε $\mathbf{u} = C\mathbf{x}^T$ και $a = \frac{1}{\mathbf{x}^T C \mathbf{x} + 1} = \frac{1}{\mathbf{x}^T \mathbf{u} + 1}$. Οπότε, το D_{i+1} μπορεί να γραφεί εύκολα ως:

$$\begin{aligned} D_{i+1} &= B_{i+1}C_{i+1} \\ &= (B_i + \mathbf{y}\mathbf{x}^T)(C_i - a\mathbf{u}\mathbf{u}^T) \\ &= B_i C_i - B_i a \mathbf{u}\mathbf{u}^T + \mathbf{y}\mathbf{x}^T C_i - \mathbf{y}\mathbf{x}^T a \mathbf{u}\mathbf{u}^T \\ &= D_i + a r \mathbf{u}^T \end{aligned}$$

Άρα με πολύ φθηνούς υπολογισμούς μπορούμε να ενημερώσουμε το λεξικό για κάθε νέο δεδομένο \mathbf{y} που αναπαριστάται από το τρέχον λεξικό, D_i , ως $\hat{\mathbf{y}}$, με σφάλμα $r = \mathbf{y} - \hat{\mathbf{y}}$.

Μία επέκταση της λογικής αυτής είναι να λαμβάνεται υπόψιν όχι μόνο το τελευταίο σφάλμα προσέγγισης, αλλά τα τελευταία $k > 1$ σφάλματα τα οποία θα σταθμίζονται εκθετικά όσο πιο παλιά είναι. Δηλαδή, εάν ένα δεδομένο προσεγγίζεται με μεγάλο σφάλμα από το τρέχον λεξικό, αλλά είναι παλιό, τότε δεν θα παίζει τόσο μεγάλο ρόλο στην ενημέρωση του λεξικού. Εάν ωστόσο τα τρέχοντα δεδομένα δεν εκφράζονται καλώς, τότε το λεξικό θα μεταβάλλεται προς όφελός τους.

Κεφάλαιο 4

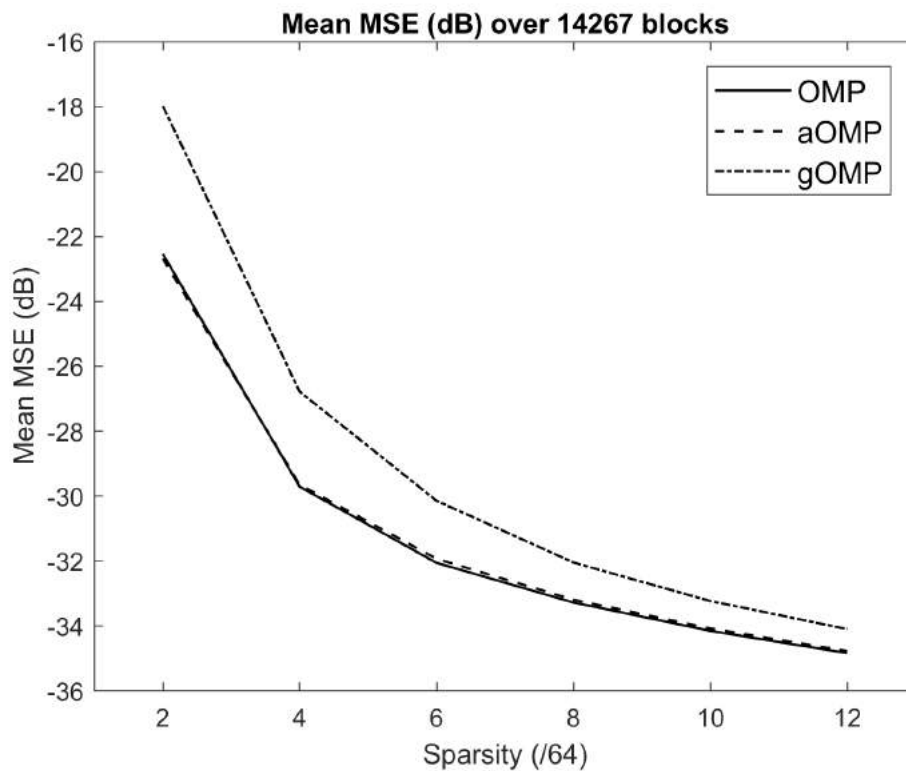
Πειραματικά αποτελέσματα

4.1 Υπολογισμός αραιής αναπαράστασης

Στο πλαίσιο την αραιής αναπαράστασης, όπως περιγράψαμε στην προσεγγίσεις, θα εξετάσουμε τον OMP ως state of the art αλγόριθμο, καθώς και τις δύο πιο σημαντικές παραλλαγές του: τους αOMP και gOMP. Στο Σχήμα (4.1) βλέπουμε την σύγκριση του τετραγωνικού σφάλματος και των τριών αλγόριθμων, τόσο συναρτήσεως του χρόνου όσο και του μέσου τετραγωνικού σφάλματος θεωρώντας αραιότητα από 2 έως 12 μη μηδενικούς συντελεστές.

Παρατηρούμε, όπως είναι αναμενόμενο, ότι ο OMP διατηρεί σταθερά το μικρότερο σφάλμα ανακατασκευής, ενώ ακολουθεί ο αOMP και τρίτος ο gOMP. Ωστόσο, και οι τρεις φαίνεται να συγκλίνουν στο ίδιο μέσο τετραγωνικό σφάλμα όσο η αραιότητα μειώνεται. Η διαφορά στον αOMP δίνει την δυνατότητα αναζήτησης ολικού ελάχιστου γύρω από την περιοχή έναρξης του αλγορίθμου, η οποία εξαρτάται από την αρχικοποίηση του λεξικού. Αυτό γίνεται απορρίπτοντας με κάποια πιθανότητα το πρώτο άτομο με την μεγαλύτερη προβολή στο σφάλμα. Η πιθανότητα αυτή, τέθηκε στο $1/2$ για το ανωτέρω πείραμα. Οπότε, τώρα είναι σαφέστερος ο λόγος για τον οποίο σε αραιότητα 2, το σφάλμα είναι μεγαλύτερο από του OMP: διότι εάν το πρώτο άτομο απορριφθεί (στο πέρας των επαναλήψεων απορρίπτεται τις μισές φορές) είναι το 1 από τα 2 η πιθανότητα η επιλογή να μην είναι σωστή είναι $1/2$, επιβαρύνοντας το σφάλμα, το οποίο καλείται να διορθώσει το δεύτερο και τελευταίο προς επιλογή άτομο. Ωστόσο, όσα περισσότερα μη μηδενικά στοιχεία επιβάλλουμε στο X , τόσο πιο πολλά άτομα έχουν την ευκαιρία να διορθώσουν την 'κακή' επιλογή απόρριψης του πρώτου ατόμου.

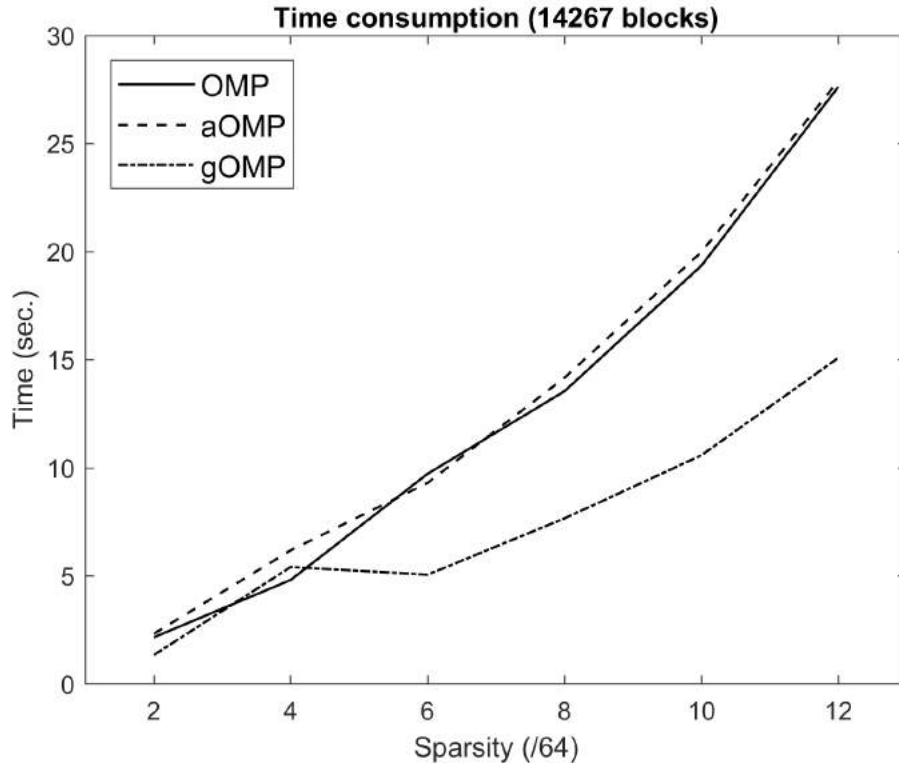
Όσον αφορά τον gOMP, η ιδέα είναι η επιλογή των N ατόμων με τη μεγαλύτερη προβολή στο σφάλμα. Εδώ, το σφάλμα είναι ακόμη μεγαλύτερο, κατά μέσο όρο. Αυτό, διότι όταν κατά την εκτέλεση του OMP επιλέγουμε τα άτομα ένα-ένα (πάντα το πρώτο με τη μεγαλύτερη προβολή), η τελική συλλογή των ατόμων είναι και η καλύτερη (εφόσον βρισκόμαστε ήδη στην περιοχή του ολικού ελάχιστου). Όταν λοιπόν το σφάλμα



Σχήμα 4.1: Σύγκριση των OMP, aOMP και gOMP, σε λογαριθμική κλίμακα, για το μέσο τετραγωνικό σφάλμα ανακατασκευής για 14267 blocks, από πραγματικές ομοειδής εικόνες. Καταγράφεται συναρτήσει της αραιότητας που επιβλήθηκε· από 2 έως 12 μη μηδενικούς συντελεστές, σε σύνολο 64 ατόμων.

ανανεώνεται μετά από κάθε ανανέωση του αραιού διανύσματος, το λεξικό διέπεται από διαφορετική κατανομή προβολών των ατόμων του πάνω στο νέο σφάλμα. Οπότε για να είναι σωστή η επιλογή των πρώτων 2, ως πούμε ατόμων αντί μόνον του πρώτου, δεν συνεπάγεται ότι το δεύτερο άτομο θα είναι αυτό που θα επιλεγόταν στην 2^η επανάληψη του OMP. Ωστόσο, μπορούμε να περιορίσουμε την συνέπεια αυτή ως εξής: μπορούμε να επιλέξουμε να διαλέγονται π.χ. 2 πρώτα άτομα με τις μέγιστες προβολές, φέρνοντας στο ήμισυ τον χρόνο εκτέλεσης του αλγορίθμου, αλλά περιορίζοντας τα λάθος άτομα προς επιλογή. Έτσι ναι μεν μπορεί το 2ο άτομο να μην ταυτίζεται με αυτό που θα επέλεγε ο OMP, αλλά τουλάχιστον το λάθος θα περιοριστεί εκεί. Ενώ στην ακραία περίπτωση που διαλέγαμε απευθείας τα T_0 άτομα που χρειαζόμασταν, η απόφαση ως προς το \mathbf{x} είναι με μεγάλη πιθανότητα πολύ πιο μακριά από την απόφαση που θα έπαιρνε ο OMP. Παρ' όλα αυτά, ισχύει ό,τι αναφέραμε και για τον aOMP. Σε μικρή αραιότητα (π.χ. $T_0 = 2$), η ταυτόχρονη επιλογή δύο ατόμων, αντί της ακολουθιακής, υποβαθμίζει το αποτέλεσμα. Ενώ σε μεγαλύτερη αραιότητα, η απόφαση έχει περισσότερες ευκαιρίες να συμπεριλάβει τα σωστά άτομα στην αραιή συλλογή.

Τέλος, ως προς τον χρόνο εκτέλεσης, είναι φανερό ότι ο gOMP χρησιμοποιεί όλο και λιγότερο χρόνο από τους άλλους δύο όσο το T_0 αυξάνεται. Στο Σχήμα (4.2) βλέπουμε τις συγκρίσεις.

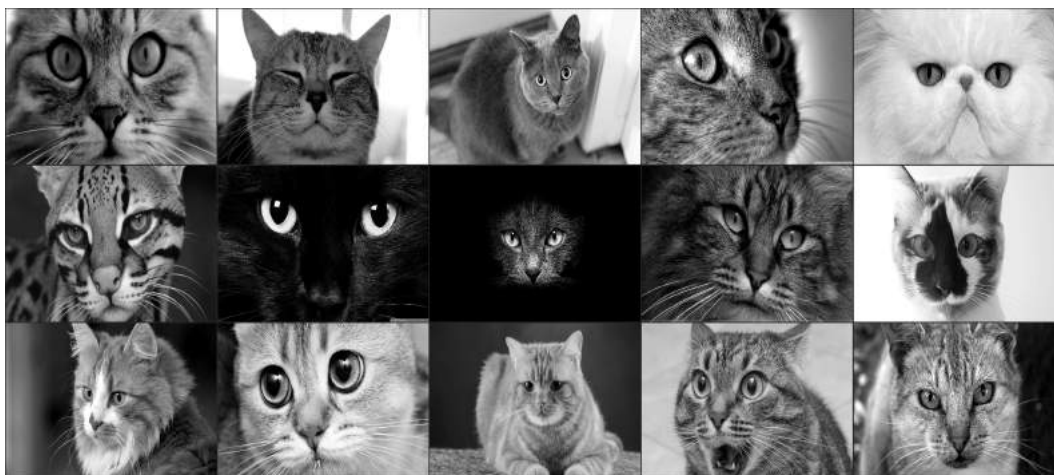


Σχήμα 4.2: Σύγκριση μέσου χρόνου εκτέλεσης για τους OMP, aOMP, gOMP, για 14267 blocks συναρτήσει της αραιότητας.

4.2 Το λεξικό

Το λεξικό που χρησιμοποιήθηκε, που φαίνεται στο Σχήμα (4.4), στα πειράματα των Σχημάτων (4.1) και (4.2), εκπαιδεύτηκε με χρήση του K-SVD, ώστε να έχουμε συνεπή αποτελέσματα στα προβλήματα που θα εξετάσουμε στην συνέχεια. Αξίζει να σημειώσουμε κάποιες παρατηρήσεις πριν δούμε αναλυτικά αποτελέσματα. Η διαδικασία εκπαίδευσης ήταν η εξής: Από 15 εικόνες κλίμακας του γκρι, με ίδιο περιεχόμενο (εικόνες από γάτες), μεγέθους 1920×1080 , επιλέχθηκαν 5000 blocks μεγέθους 8×8 έκαστο. Κάθε βλοκx διανυσματοποιήθηκε και δημιουργήθηκε ένα μητρώο Y μεγέθους 64×75000 . Για το μέγεθος αυτό των δεδομένων εκπαίδευσης, θεωρήθηκε κατάλληλο μέγεθος λεξικού 100 ατόμων. Είναι ένα σενάριο που ευνοεί την σύγκλιση του λεξικού σε μία μορφή που θα κάνει εμφανή τα χαρακτηριστικά εκείνα που αναδεικνύουν την ποιότητα ενός λεξικού. Επίσης, όπως αναφέραμε και νωρίτερα, τα άτομα του λεξικού θεωρούνται μοναδιαία και κατασκευάζονται ως τέτοια. Ενώ η μέση τιμή τους είναι 0.

Επίσης, η μέση τιμή κάθε block είναι 0 και είναι και αυτά κανονικοποιημένα. Αργότερα θα δούμε ότι η μόνη χρήσιμη πληροφορία που περιέχεται στο λεξικό είναι διακυμάνσεις, η οποίες συμπληρώνουν έναν σταθερό όρο. Δηλαδή, η μέση τιμή δεν περιέχει κάποια χρήσιμη πληροφορία και για αυτό προς το παρόν και πριν δούμε ειδικότερες εφαρμογές, αγνοείται. Τέλος, επιβλήθηκε σταθερή αραιότητα για όλα τα διανύσματα, 5 μη μηδενικών συντελεστών.



Σχήμα 4.3: Σύνολο εικόνων που χρησιμοποιήθηκαν για εκπαίδευση, διάστασης 1920×1080 η κάθε μία.

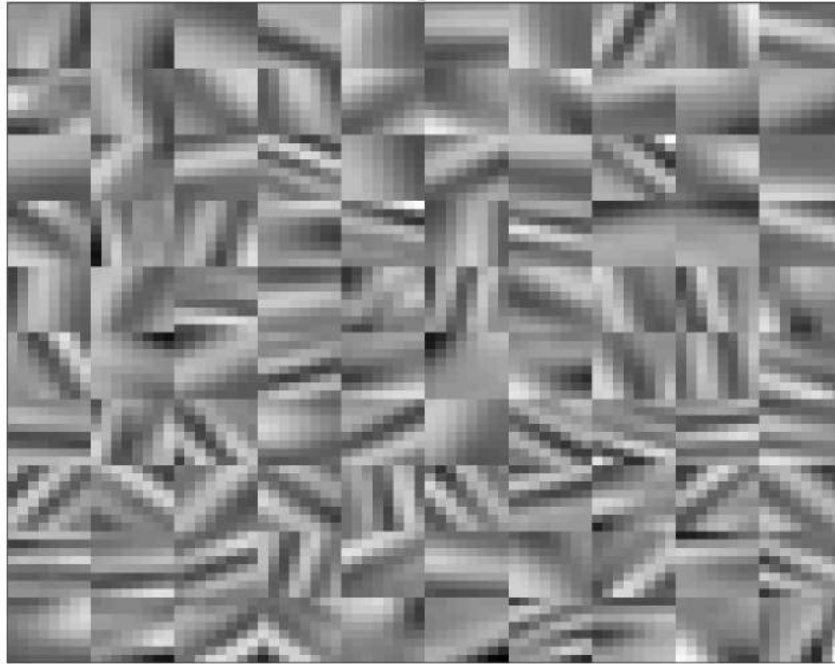
Στο Σχήμα (4.5) βλέπουμε το λεξικό στο οποίο αναφερόμαστε και μία δεύτερη προσπάθεια, με ένα λεξικό μεγαλύτερων block. Θα δούμε πως η επιλογή των 8×8 δεν είναι τυχαία.

Παρατηρούμε ότι στο λεξικό μικρότερης διάστασης, που τελικά χρησιμοποιήσαμε και στα πειράματα, περιέχει περισσότερα 'φυσικά' χαρακτηριστικά από ό,τι αυτό της μεγαλύτερης διάστασης. Μιλώντας για διακυμάνσεις, τα άτομα του πρώτου, φαίνονται να απεικονίζουν ακριβώς ότι είναι πιο χρήσιμο σε μία εικόνα προκειμένου να είναι αναγνωρίσιμο ένα αντικείμενο από το μάτι: ακμές. Θυμίζει κατά πολύ μάλιστα φίλτρα Gabor καθώς βλέπουμε πολλές ακμές περιστραμμένες προς όλες τις πιθανές κατευθύνσεις.

Επίσης, όπως φαίνεται στο μητρώο αυτοσυσχετίσεων, στο Σχήμα (4.6), ικανοποιείται η Ιδιότητα Αμοιβαίας Μη Συνεκτικότητας καθώς παρατηρούνται μεγάλες κατ' απόλυτη τιμή συσχετίσεις ανάμεσα σε ελάχιστα άτομα.

Όπως στο ιστόγραμμα του Σχήματος (4.7) είναι φανερό ότι το μητρώο αυτοσυσχετίσεων απέχει ελάχιστα από το διαγώνιο. Παρακάτω, φαίνονται με λευκό μόνο οι θέσεις αυτοσυσχετίσης μεγαλύτερης του 0.8.

Στο Σχήμα (4.8) παρατηρούμε ακριβώς ό,τι βλέπουμε και στο ιστόγραμμα: οι πιο πολλές συσχετίσεις βρίσκονται κοντά στο 0 ενώ οι υπόλοιπες κατανέμονται σχεδόν ομοιόμορφα στο εναπομένον διάστημα. Παρακάτω φαίνονται τα blocks με την χειρότερη

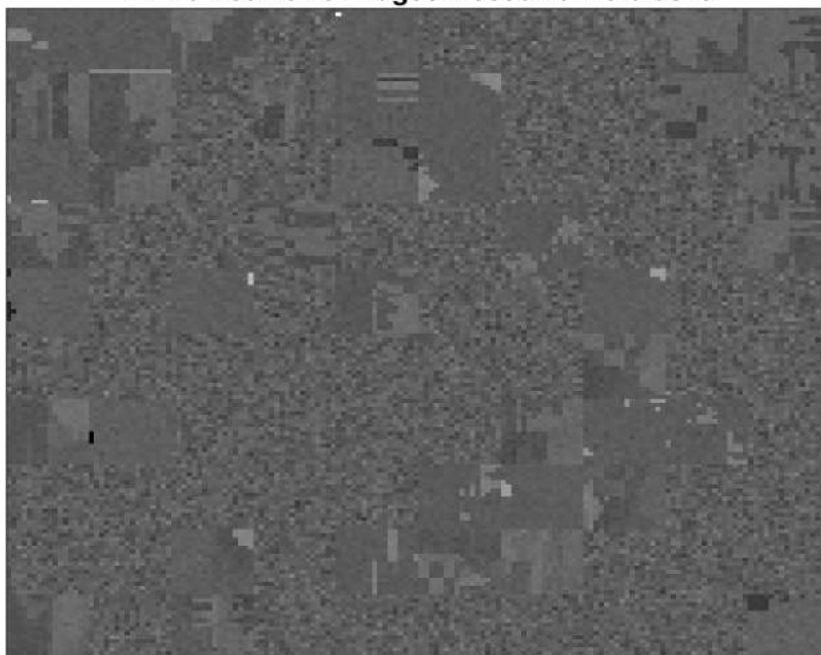
D: Trained w/ 15 images x 5000 8x8 blocks

Σχήμα 4.4: Λεξικό 100 8×8 blocks-ατόμων, εκπαιδευμένο με 75000 blocks από εικόνες με γάτες.

και καλύτερη ανακατασκευή.

Εδώ είναι αναγκαίο να κάνουμε μία αξιολογική αναφορά. Το 'ελάχιστο' σφάλμα που απεικονίζεται στο Σχήμα (4.10) είναι στην πραγματικότητα το ελάχιστο μη μηδενικό. Μέσα στα δεδομένα εκπαίδευσης προφανώς υπάρχουν και blocks αποτελούμενα μόνο από μηδενικά. Η προβολή, όμως, των ατόμων του λεξικού πάνω σε αυτά είναι μηδενική. Έτσι, δεν υπάρχει άλλη επιλογή από το να μην επιλεγεί κανένα άτομο προς ανακατασκευή τους. Δηλαδή καταλήγουμε σε ένα αντίστοιχο αραιό διάνυσμα επίσης γεμάτο με μηδενικούς συντελεστές. Αυτό μας δίνει ένα πολύ σημαντικό στοιχείο για τον ποιοτικό χαρακτηρισμό του λεξικού. Ας πάρουμε τα πράγματα από την αρχή. Τα blocks αποτελούμενα μόνον από μηδενικά προέρχονται από οποιοδήποτε αρχικό block ήταν πλήρως ομοιογενής περιοχή κάποιας εικόνας. Αυτό οδηγεί σε ένα πολύ σημαντικό συμπέρασμα. Τα DC blocks, όπως αλλιώς ονομάζονται, δεν προσφέρουν τίποτα στην εκπαίδευση του λεξικού. Ουσιαστικά ο αλγόριθμος εκπαίδευσης τα αγνοεί. Έτσι εξάγεται το συμπέρασμα ότι μόνο οι περιοχές με διακυμάνσεις (= ακμές και υφές) προσφέρουν χρήσιμη πληροφορία προς εκπαίδευση.

Βέβαια αυτό δεν μας απαλλάσσει από την ευθύνη για την ποιότητα των εικόνων εκπαίδευσης. Πρέπει να φροντίσουμε, ειδικά σε δείγματα περιορισμένου πλήθους, να περιέχουν πληροφορία σχετική μόνο με το είδος φυσικού αιτίου που προσπαθούμε να εκφράσουμε μέσω του λεξικού. Για παράδειγμα, και ένα ποδήλατο στο προσκήνιο θα

D: Trained w/ 15 images x 5000 16x16 blocks

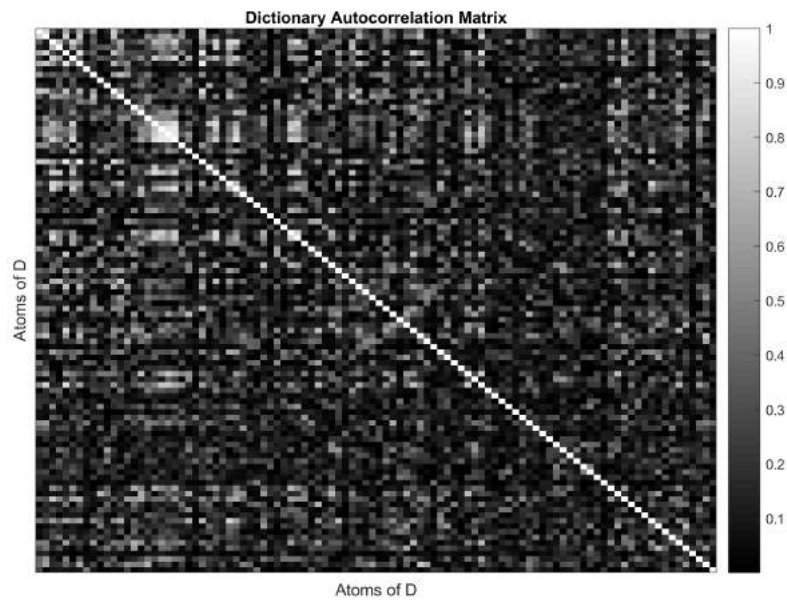
Σχήμα 4.5: Λεξικό 500 16×16 blocks-ατόμων, εκπαιδευμένο με 75000 blocks από εικόνες με γάτες.

αποτελούσε πηγή ακμών προς ανακατασκευή, αλλά εάν συναντούμε συχνά ποδήλατα στις εικόνες εκμάθησης, το λεξικό δεν έχει παρά να εισάγει και αυτή την πληροφορία στα άτομά του, αποκλίνοντας από τον κύριο ρόλο του, που είναι να αναπαριστά γάτες.

4.3 Συνθετικά δεδομένα

Το Υ είπαμε ότι αποτελείται από διανύσματα τα οποία είναι υλοποιήσεις μιας τυχαίας διαδικασίας. Δηλαδή, οι παρατηρήσεις αυτές παράγονται από κάποιο φυσικό αίτιο. Ευελπιστούμε, λοιπόν, ότι από το φυσικό αυτό αίτιο, η πιο χρήσιμη πληροφορία θα αποθηκευτεί μέσα στο λεξικό που θα εκπαιδύσουμε. Άρα, ένα έξυπνο πείραμα είναι να ορίσουμε εξ αρχής το λεξικό ως το φυσικό αίτιο. Έτσι, εάν προσπαθήσουμε να εκπαιδύσουμε ένα νέο λεξικό με δεδομένα που παράγονται με την βοήθεια του πρώτου, τότε το νέο λεξικό θα πρέπει να προσεγγίζει το αρχικό.

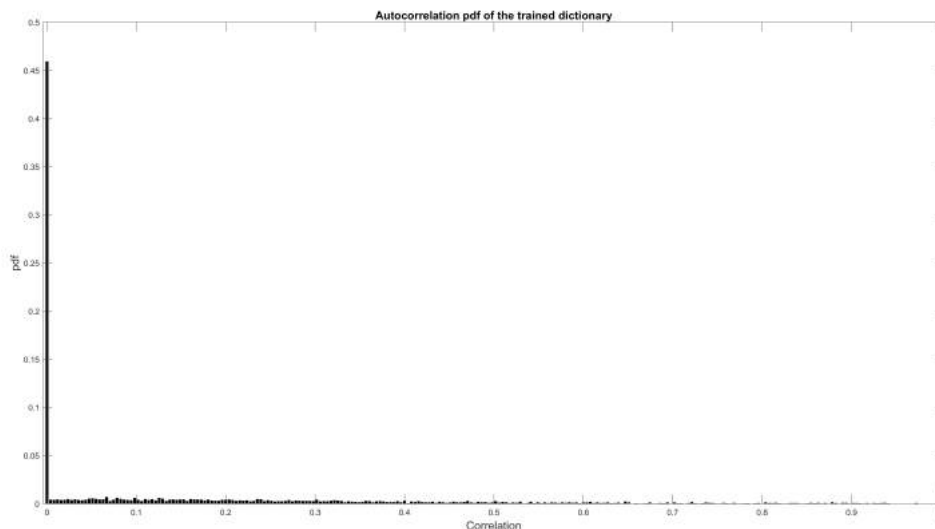
Όταν αναφερόμαστε στον όρο 'φυσικό αίτιο' μπορούμε να δώσουμε πολλές νοηματικές αναλύσεις. Αλλά αυτή που, φαινομενικά τουλάχιστον από τα αποτελέσματα, είναι η πιο ακριβής είναι ότι το φυσικό αίτιο, για μια εικόνα είναι τα χαρακτηριστικά που ανήκουν στο ίδιο είδος εικόνας. Ένα βήμα πιο πίσω είναι αρκετό για να ισχυριστούμε ότι το αντικείμενο που απεικονίζεται είναι ουσιαστικά το φυσικό αίτιο. Αλλά αυτό δυστυχώς δεν είναι πάντα αληθές. Το γιατί θα το διαπιστώσουμε στη συνέχεια στην παράγραφο 4.4.



Σχήμα 4.6: Απόλυτες συσχετίσεις μεταξύ των ατόμων του λεξικού.

Παρατηρούμε, στο Σχήμα (4.11), ότι σε συνθήκες ίσης ισχύος θορύβου με σήματος, οι επιτυχίες είναι πολύ χαμηλές αλλά με πολύ μεγάλη διακύμανση ανά τις προσπάθειες. Αυτό όμως παύει να συμβαίνει όταν αυξήσουμε το πλήθος των δεδομένων εκπαίδευσης σε 7500, όπως φαίνεται παρακάτω.

Επίσης, στο Σχήμα (4.13) μπορούμε να δούμε το τρόπο με τον οποίο εντοπίζονται τα άτομα του προκύπτοντος λεξικού και του αρχικού. Όπως είναι ξεκάθαρο, τις περισσότερες φορές (μέσο επίπεδο), σε οποιοδήποτε περιβάλλον θορύβου (για 7500 blocks εκπαίδευσης), τα περισσότερα άτομα του αρχικού λεξικού, εντοπίζονται σε διαφορετικές θέσεις ($\text{Times used} = 1$) του εκπαιδευμένου λεξικού. Ενώ τα υπόλοιπα είτε δεν εντοπίζονται καθόλου, είτε εντοπίζονται ως διπλότυπα. Από την άλλη, τα άτομα του εκπαιδευμένου λεξικού, εντοπίζονται στο αρχικό λεξικό κατά την πλειοψηφία, ενώ τα υπόλοιπα δεν εντοπίζονται καθόλου. Το δεύτερο αποτέλεσμα εκφράζει κατά πολύ το αποτέλεσμα καταγράφεται στις καμπύλες SNR - Matched atoms. Δηλαδή, επιβεβαιώνεται το γεγονός ότι τα άτομα του εκπαιδευμένου λεξικού δεν εντοπίζονται ως διπλότυπα στο αρχικό λεξικό. Κάτι τέτοιο θα σήμαινε πως το αρχικό λεξικό θα περιείχε αρκετά διπλότυπα άτομα. Αυτό όμως θα ερχόταν σε αντίθεση με την παρατήρησή μας, ότι το μητρώο αυτοσυσχετίσεων του αρχικού λεξικού είναι σχεδόν διαγώνιο.



Σχήμα 4.7: Ιστόγραμμα απόλυτων συσχετίσεων των ατόμων του λεξικού.

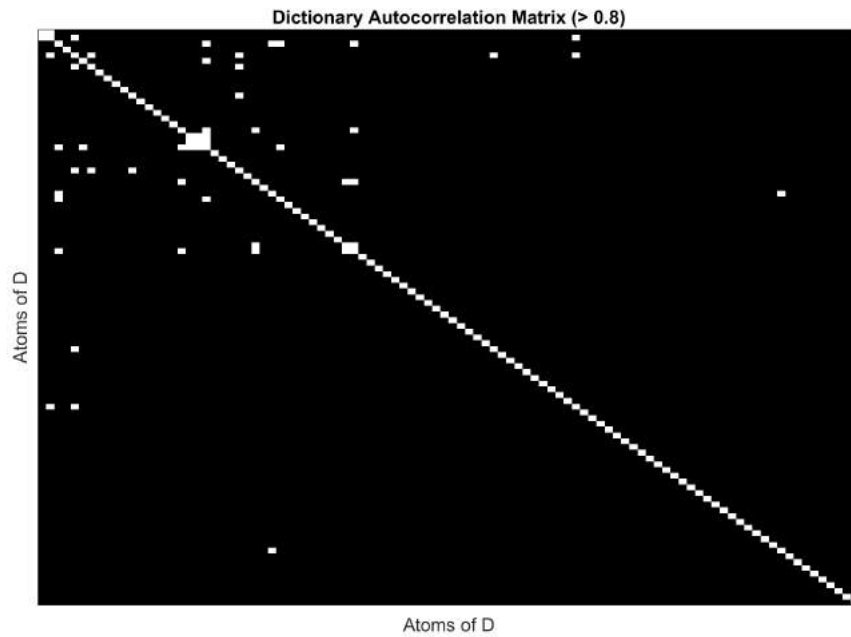
4.4 Αποθορυβοποίηση

Επιστρέφουμε πίσω στα πραγματικά δεδομένα και το λεξικό με τις γάτες. Σε αυτή την παράγραφο θα εξετάσουμε το κατά πόσο είναι δυνατό το λεξικό να ανακατασκευάσει γνωστές και άγνωστες εικόνες από ποικίλα φυσικά αίτια. Οι δοκιμαστικές εικόνες λοιπόν, είναι αυτές που φαίνονται στο Σχήμα (4.14).

Θεωρούμε σενάρια λευκού Gaussian θορύβου, για τυπικές αποκλίσεις 0,10 και 50%. Θεωρήθηκε επίσης ότι το λεξικό αποτελείται από το εκπαιδευμένο κομμάτι των 99 από 100 ατόμων, ενώ ένα άτομο αποτελεί τον DC όρο. Για την ακρίβεια το λεξικό πριν εκπαιδευθεί αρχικοποιήθηκε ως τυχαίο πλην ενός ατόμου που είναι σταθερό. Και το εκπαιδευμένο λεξικό φαίνεται να διατηρεί το άτομο αυτό ανέπαφο. Έτσι έχουμε το λεξικό του Σχήματος (4.15).

Στο λεξικό του Σχήματος (4.15) τα άτομα έχουν μηδενική μέση τιμή διότι σε κάθε γραμμικό συνδυασμό που συμβάλλουν διαθέτουν το σταθερό block για οποιοδήποτε offset χρειάζεται στο intensity. Η ανακατασκευές φαίνονται στο Σχήμα (;;) μαζί με τις συγκρίσεις των ιστογραμμάτων των αρχικών και ανακατασκευασμένων εικόνων:

Στο Σχήμα (4.17) παρατηρούμε και στις τρεις περιπτώσεις θορύβου τις ίδιες σχέσεις σφαλμάτων. Ωστόσο, αξιοσημείωτο είναι το γεγονός ότι η εικόνα που ανήκει στο σύνολο εικόνων εκπαίδευσης έχει παρόμοιο σφάλμα ανακατασκευής με την εικόνα του παπαγάλου και του ηθοποιού. Αυτό μας βάζει σε σκέψεις, όπως είπαμε νωρίτερα, για το ποιό είναι το φυσικό αίτιο που τείνει να περιγράφει το εκπαιδευμένο λεξικό. Προφανώς, αυτό που συμβαίνει είναι ότι το περιεχόμενο της εικόνας δεν παίζει τόσο μεγάλο ρόλο όσο τα χαρακτηριστικά των ακμών και της υψής. Ωστόσο, μία επιλογή εικόνας κατά πολύ διαφορετική τόσο στο περιεχόμενο, όσο και στις ακμές και τις υψές της, όπως το



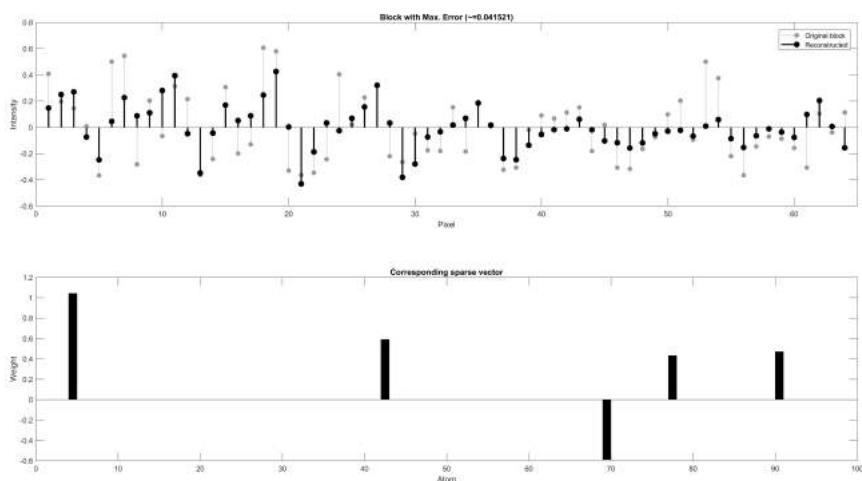
Σχήμα 4.8: Απόλυτες συσχετίσεις μεταξύ των ατόμων του λεξικού με τιμή άνω του 0.8.

αστικό τοπίο ή τα σπίτια, δίνουν πολύ μεγαλύτερο σφάλμα.

4.5 Συμπλήρωση ελλιπών δεδομένων

Ένα ακόμη σημαντικό πρόβλημα στη βιβλιογραφία είναι η συμπλήρωση δεδομένων που λείπουν από το σήμα, ή όπως λέγεται στις εικόνες 'imprinting'. Στο πλαίσιο αυτό, τα δεδομένα δοκιμής του λεξικού είναι ελλιπή, δίνοντας την ευκαιρία στο λεξικό, με την ανακατασκευή που θα προσπαθήσουμε να γεμίσει τα κενά. Στο Σχήμα (4.18) φαίνεται το σενάριο ελλιπούς πληροφορίας των μισών εικονοστοιχείων από κάθε block όλης της εικόνας, για κάθε εικόνα του συνόλου δοκιμών.

Η συμπλήρωση κενής πληροφορίας είναι μία πιο ευέλικτη διαδικασία ως προς το σφάλμα, ειδικά όταν μιλάμε για εικόνες και ειδικά όταν η έλλειψη πληροφορίας έγκειται χωρικά σε ομοιόμορφα διεσπαρμένα εικονοστοιχεία. Η διαφορά από έναν παρατηρητή δεν είναι εύκολο να εντοπισθεί, τουλάχιστον στο σενάριο με την έλλειψη μόλις του 20% των εικονοστοιχείων. Ωστόσο, τα αποτελέσματα φαίνονται καλύτερα όταν στρέφουμε το βλέμμα μας στα μέσα τετραγωνικά σφάλματα και ειδικά σε λογαριθμική κλίμακα. Στο Σχήμα (4.19) βλέπουμε ότι τα σφάλματα ανακατασκευής είναι κατά πολύ μικρότερα απ' ό,τι στο σενάριο ενθόρυβων εικόνων.

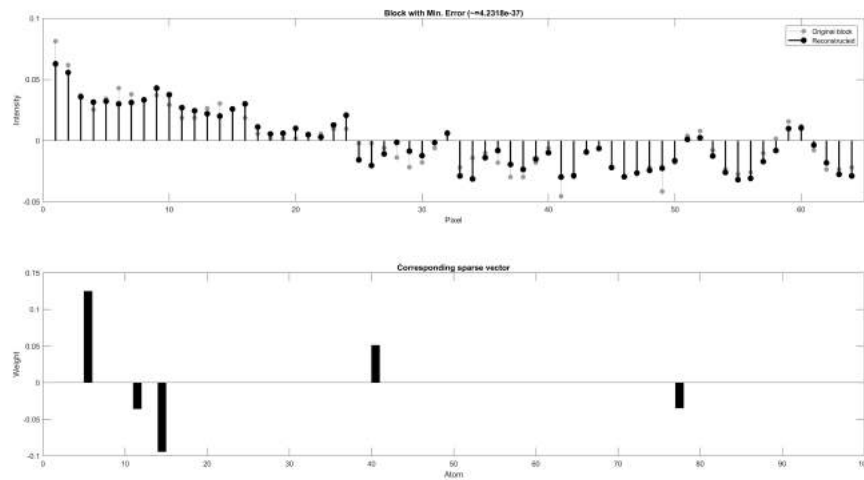


Σχήμα 4.9: Block με το μεγαλύτερο σφάλμα ανακατασκευής και το αντίστοιχο αραιό διάνυσμα.

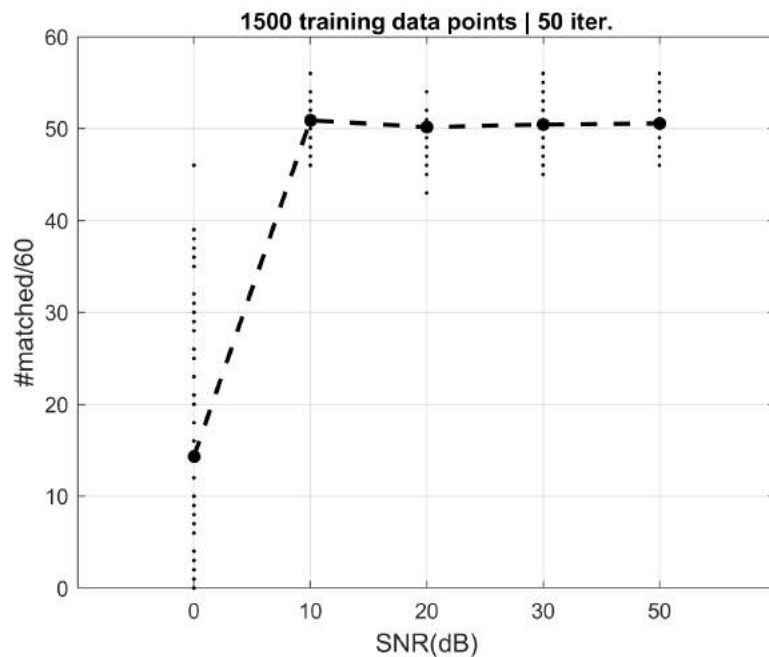
4.6 K-SVD και παραλλαγές

Τέλος, για το κομμάτι εκμάθησης του λεξικού δοκιμάστηκαν ο βασικός αλγόριθμος αναφοράς που εκπαίδευσε και το λεξικό στα προηγούμενα πειράματα, καθώς και μία παραλλαγή του. Στο Σχήμα (4.20) βλέπουμε την σχετική απόδοση στην κλίμακα του χρόνου, των K-SVD και AppK-SVD.

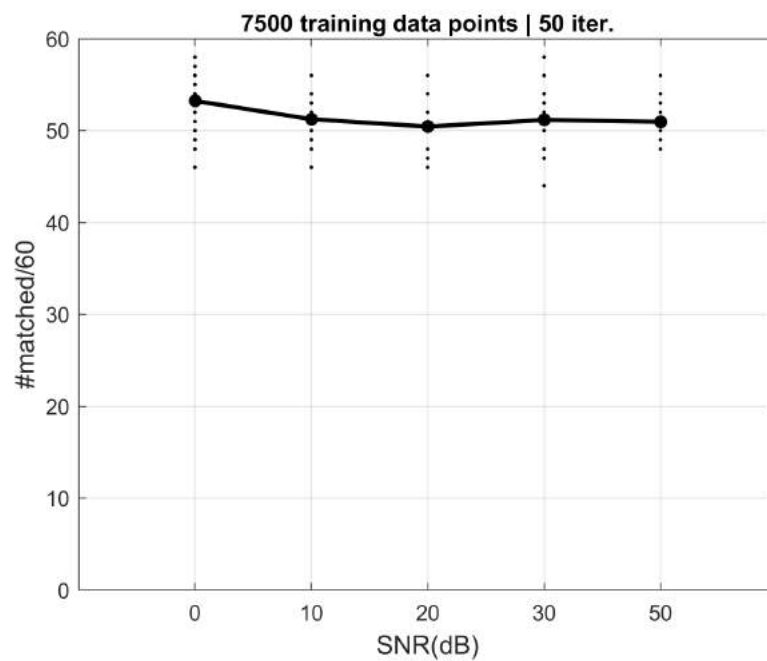
Σαφέστατα, ο AppK-SVD είναι γρηγορότερος κατά πολύ. Δίνοντας την ευκαιρία στο σφάλμα να παίρνει μικρότερες τιμές σε συντομότερο χρόνο. Αυτό δεν πρέπει όμως να μας ξεγελά. Ο K-SVD διατηρεί την ιδιότητα της βέλτιστης επιλογής \mathbf{x}_i και \mathbf{d}_i σε κάθε επανάληψη. Απλώς, η SVD καταναλώνει πολύ χρόνο σε σχέση με την επίλυση μίας απλής γραμμικής εξίσωσης. Περαιτέρω δουλειά μπορεί να γίνει ως προς την παραλλαγή του K-SVD πάντα στο επίπεδο της χρονικής κατανάλωσης. Αλλά εξαρτάται κυρίως από την εφαρμογή στην οποία θα αξιοποιηθεί το λεξικό.



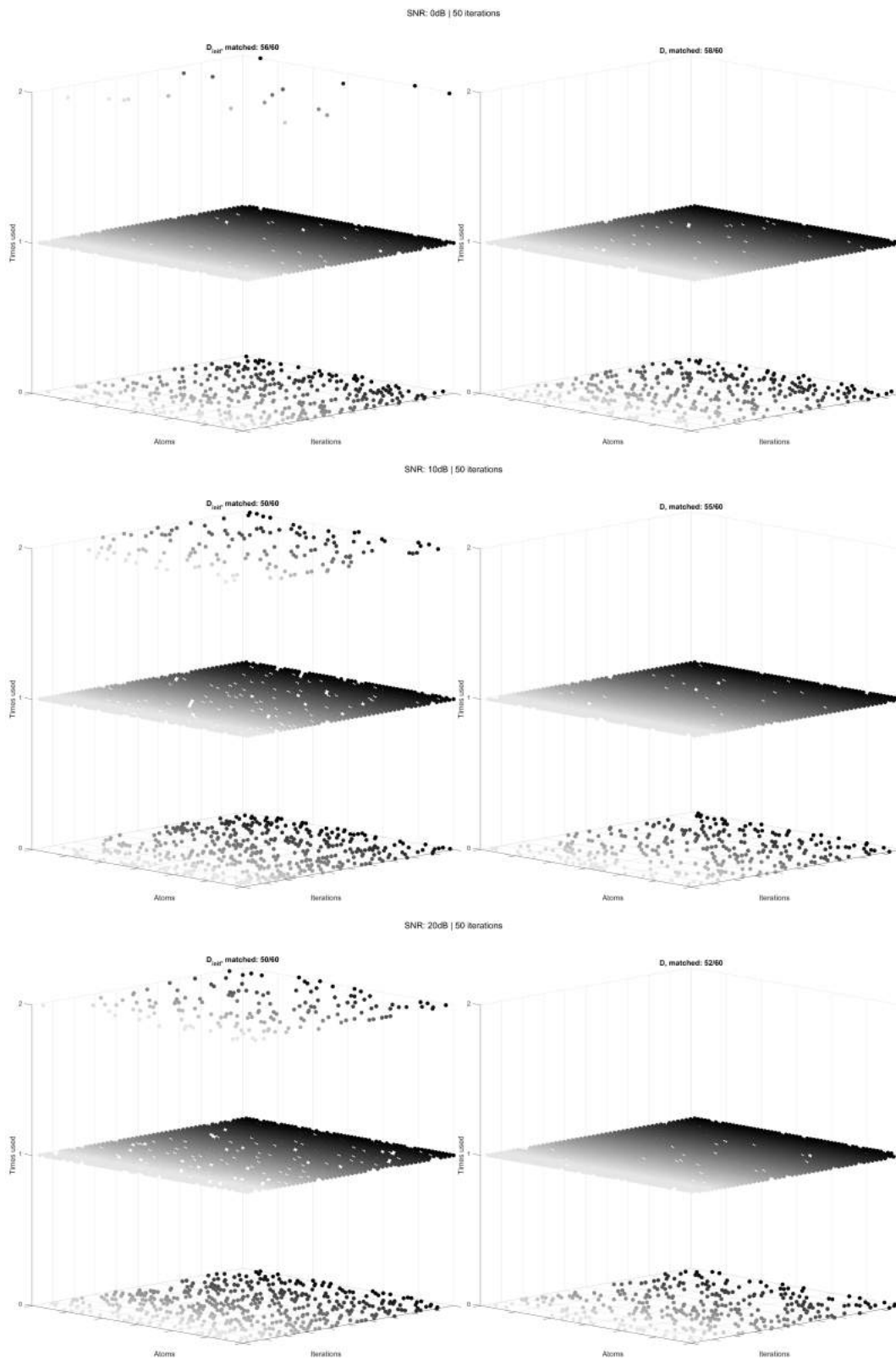
Σχήμα 4.10: Block με το μικρότερο μη μηδενικό σφάλμα ανακατασκευής και το αντίστοιχο αραιό διάνυσμα.

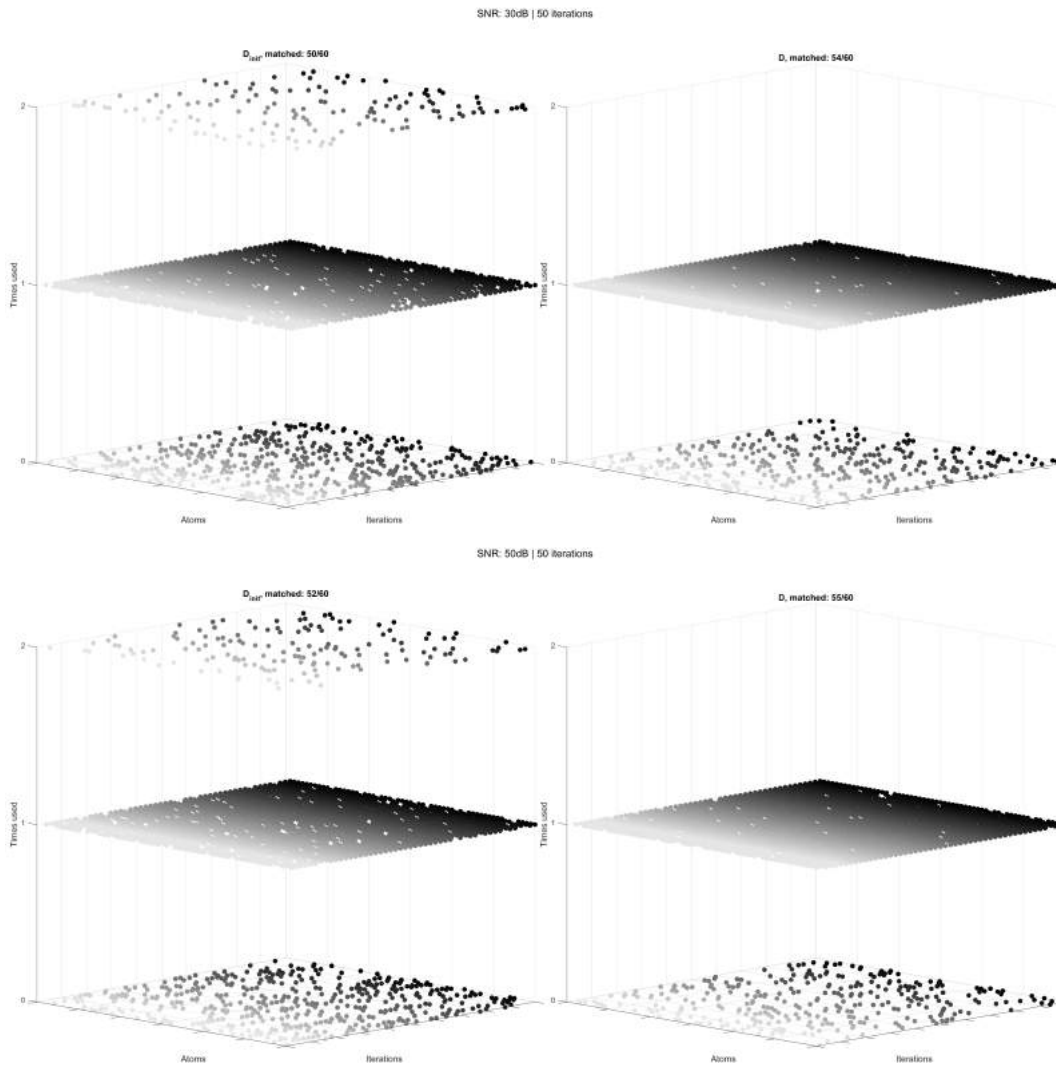


Σχήμα 4.11: Μέσο πλήθος εντοπισμένων ατόμων του λεξικού που 'κρύβεται' πίσω από τα δεδομένα, συναρτήσει του SNR. Το πλήθος των δεδομένων εκπαίδευση είναι 1500, ενώ τόσο το λεξικό, όσο και τα δεδομένα είναι τυχαία ακολουθώντας της κανονική κατανομή $\mathcal{N}(0, 1)$. Μέσος όρος από 50 επαναλήψεις του πειράματος. Το κριτήριο εντοπισμού ενός ατόμου είναι η απόλυτη προβολή του να ξεπερνά το 0.99.



Σχήμα 4.12: Μέσο πλήθος εντοπισμένων ατόμων του λεξικού που 'κρύβεται' πίσω από τα δεδομένα, συναρτήσει του SNR. Το πλήθος των δεδομένων εκπαίδευση είναι 7500, ενώ τόσο το λεξικό, όσο και τα δεδομένα είναι τυχαία ακολουθώντας της κανονική κατανομή $\mathcal{N}(0, 1)$. Μέσος όρος από 50 επαναλήψεις του πειράματος. Το κριτήριο εντοπισμού ενός ατόμου είναι η απόλυτη προβολή του να ξεπερνά το 0.99.



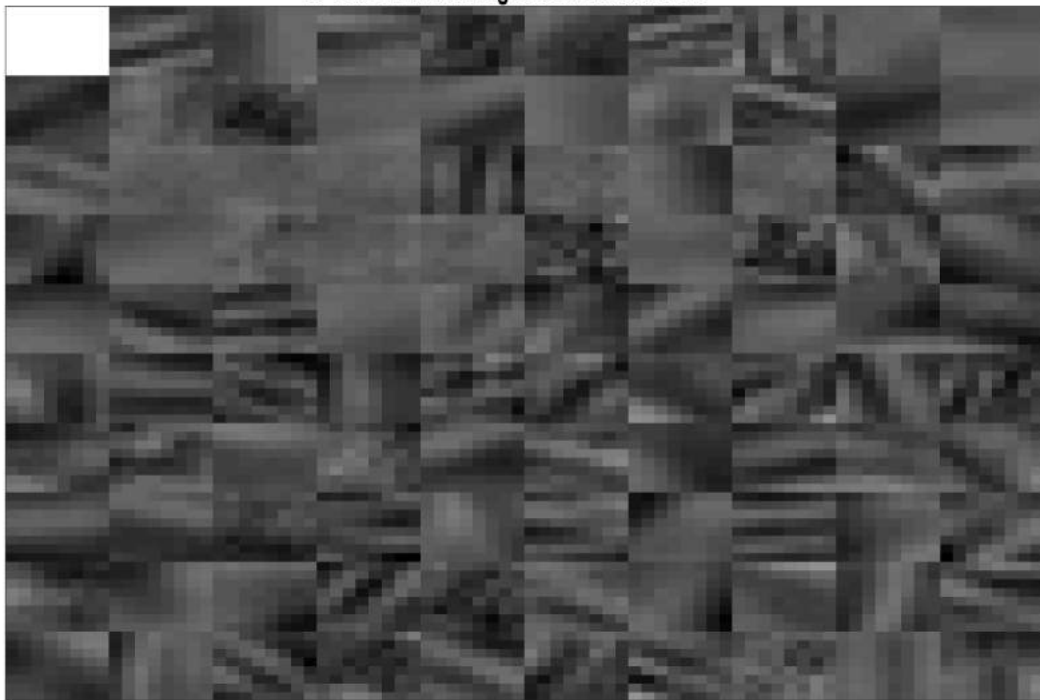


Σχήμα 4.13: Αριστερά: Αντιστοιχίσεις ατόμων του αρχικού λεξικού σε άτομα του εκπαιδευμένου, Δεξιά: Αντιστοιχίσεις ατόμων του εκπαιδευμένου λεξικού σε άτομα του αρχικού. Καταγράφονται κατά μήκος των 50 προσπάθειες (περισσότερη σκίαση προς την τελευταία προσπάθεια).



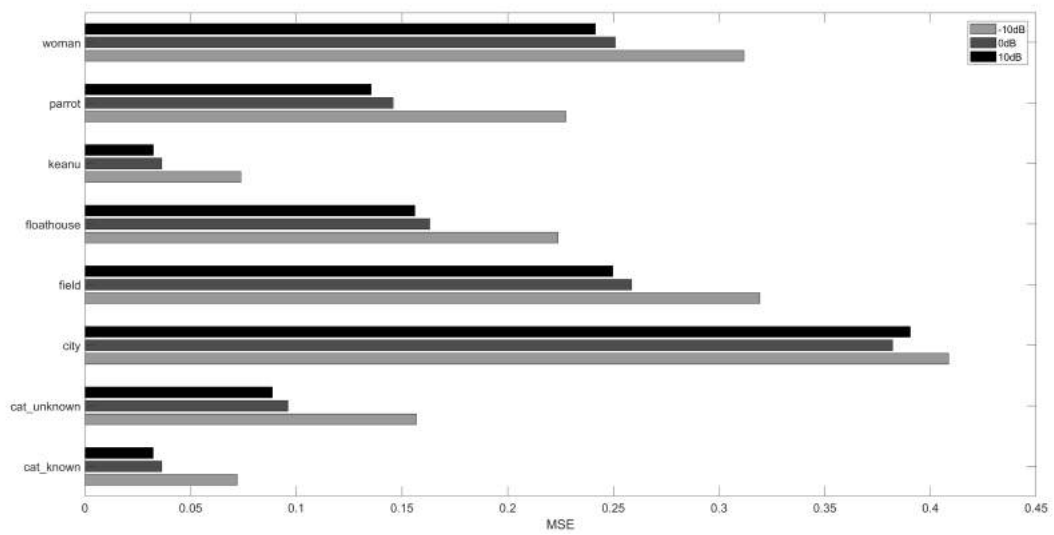
Σχήμα 4.14: Σύνολο εικόνων για δοκιμή ανακατασκευής τους από ενθόρυβες εκδοχές τους. Η πρώτη εικόνα με γάτα είναι γνωστή στο λεξικό, αφού ανήκε στο σύνολο εκπαίδευσης.

D: Trained w/ 15 images x 5000 8x8 blocks



Σχήμα 4.15: Λεξικό εκπαιδευμένο με το ίδιο σύνολο εικόνων, με την διαφορά ενός επιπρόσθετου DC block για την κωδικοποίηση της μέσης τιμής κάθε block.

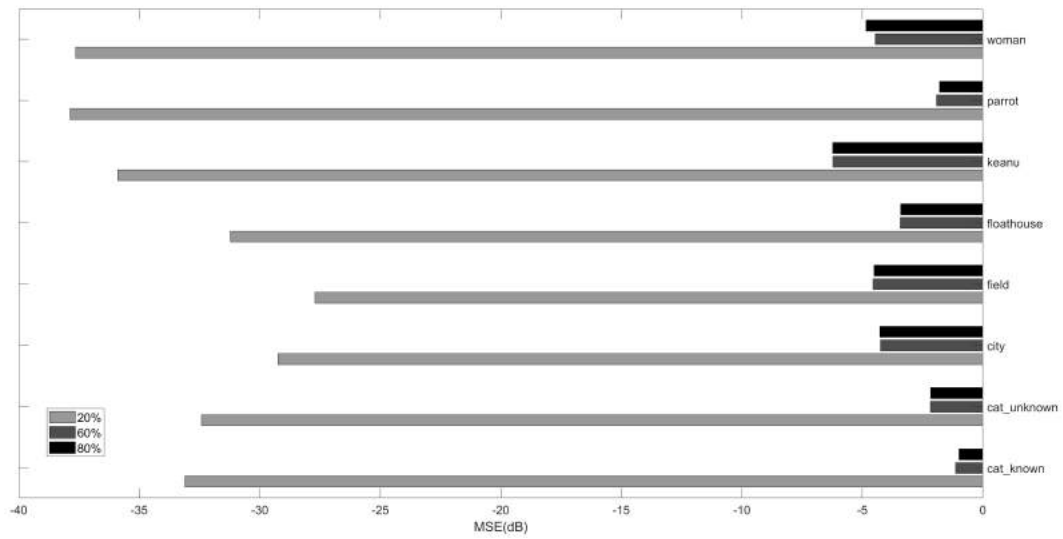




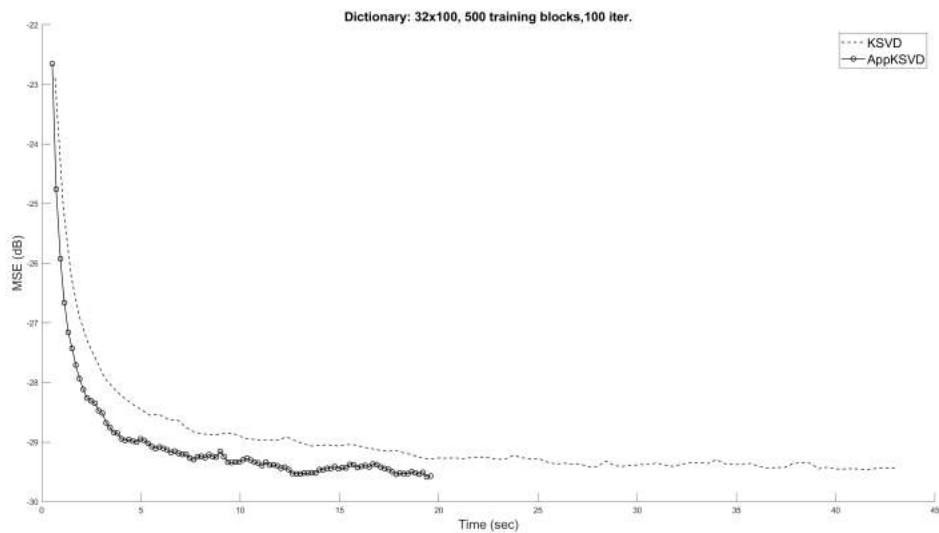
Σχήμα 4.17: Σφάλματα ανακατασκευής για όλες τις εικόνες δοκιμών, σε λόγους ισχύος σήματος προς θόρυβο $-10dB$, $0dB$ και $10dB$.



Σχήμα 4.18: Για κάθε τετράδα εικόνων ίδιου αντικειμένου: αρχική (πάνω αριστερά), ανακατασκευασμένη για 20% απόντων εικονοστοιχείων (πάνω δεξιά), για 50% (κάτω αριστερά), για 80% (κάτω δεξιά).



Σχήμα 4.19: Σύγκριση μέσω τετραγωνικών σφαλμάτων σε όλες τις δοκιμαστικές ει-
κόνες, για τα διαφορετικά ποσοστά απώσας πληροφορίας.



Σχήμα 4.20: Σύγκριση μέσω τετραγωνικών σφαλμάτων για 100 επαναλήψεις K-SVD
και AppK-SVD.

Bibliography

- [1] D.J. Fiel B.A. Olshausen. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1. 1996.
- [2] J. Idier R. Gribonval C. Herzet, C. Soussen. Exact Recovery Conditions for Sparse Representations with Partial Support Information. 2013.
- [3] X. Huo D.L. Donoho. Uncertainty Principles and Ideal Atomic Decomposition. 2001.
- [4] T. Tao E.J. Candes. Decoding by Linear Programming. 2005.
- [5] Carlos Fisch Brito Gustavo Malkomes, Joao Paulo Pordeus. An improvement of the K-SVD algorithm with applications on face recognition. 2014.
- [6] B. Shim J. Wang, S. Kwon. Generalized Orthogonal Matching Pursuit. 2014.
- [7] J.B.J.Fourier. Théorie analytique de la chaleur. 1878.
- [8] K.Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. 1901.
- [9] Alfred Bruckstein Michal Aharon, Michael Elad. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. 2006.
- [10] P.Comon. Pierre Comon (1994) Independent component analysis, a new concept? 1994.
- [11] Michael Elad Ron Rubinstein, Michael Zibulevsky. Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit. 2008.
- [12] R.A. Leibler S. Kullback. On information and sufficiency. 1951.
- [13] B.D. Rao S.F. Cotter. Application of Total Least Squares to the Design of Sparse Signal Representation Dictionaries. 2002.
- [14] Zhifeng Zhang Stephane G. Mallat. Matching Pursuit With Time-Frequency Dictionaries. 1993.

- [15] Lie Wang T.Tony Cai. Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise. 2011.

Κεφάλαιο 5

Παράρτημα

5.1 Κώδικες Matlab

```
1 function x = GenOMP(varargin)
2
3 % x = GenOMP(D,y,arg);
4 % x = GenOMP(D,y,arg,N,p);
5
6 % N is the max. number of projections to keep in each iteration
7 % For N = 1 GenOMP implements OMP. For N > 1 GenOMP implements
8 % the Generalized OMP.
9 % For arg > 1 : arg = nocoeffs
10 % For arg < 1 : arg = stopping error
11 % For p > 0 aOMP is implemented along with any other option selected
    above
12
13 narginchk(3,5);
14
15 D = varargin{1};
16 y = varargin{2};
17 arg = varargin{3};
18 if nargin == 5
19     N = varargin{4};
20     p = varargin{5};
21 else
22     N = 1;
23     p = 0;
24 end
25
26 if arg < 0 || arg > size(D,2) || N < 1 || arg > size(D,2) || p < 0 || p >
    1
27     error('Invalid_input_arguments');
28 end
29
```

```

30 err = y;
31 x = zeros(size(D,2),1);
32
33 % aOMP part %
34 if p > rand(1)
35     proj = D'*err;
36     [~,mi] = max(abs(proj)./sum(D.^2)');
37     D(:,mi) = 0;
38 end
39
40 % Option 1: Stop for a Number of Coefficients or untill the error is
    negligible %
41 if arg > 1
42     i = 1;
43     while (i <= arg/N && norm(err) > eps)
44         proj = D'*err;
45         [~,mi] = maxk(abs(proj)./sum(D.^2)',N);
46         x(mi) = proj(mi);
47         x(x~=0) = pinv(D(:,(x~=0)))*y;
48         err = y - D*x;
49         i = i + 1;
50     end
51 % Option 2: Stop for a given error %
52 else
53     while norm(err) > arg
54         proj = D'*err;
55         [~,mi] = maxk(abs(proj)./sum(D.^2)',N);
56         x(mi) = proj(mi);
57         x(x~=0) = pinv(D(:,(x~=0)))*y;
58         err = y - D*x;
59     end
60 end
61
62 end

```

```

1 % Comparison of OMP aOMP and GenOMP time consumption as a function of
    sparsity %
2
3 %% Loading the images and training the dictionary using K-SVD %
4 dicttrain
5
6 Y = Y(:,sum(abs(Y))~=0);
7
8 %% Sparse Decomposition run time collection
9 N = length(Y);
10 K = 100;

```

```

11 nocoeffs = 2:2:12;
12 err1 = zeros(length(nocoeffs),1);
13 err2 = zeros(length(nocoeffs),1);
14 err3 = zeros(length(nocoeffs),1);
15 omptime1 = zeros(length(nocoeffs),1);
16 omptime2 = zeros(length(nocoeffs),1);
17 omptime3 = zeros(length(nocoeffs),1);
18
19 for t = 1:length(nocoeffs)
20     disp(['nocoeffs:_' num2str(nocoeffs(t))])
21     noc = nocoeffs(t);
22     % OMP
23     X1 = zeros(K,N);
24     omptic1 = tic;
25     for i = 1:N
26         X1(:,i) = GenOMP(D,Y(:,i),noc);
27     end
28     omptime1(t) = toc(omptic1);
29     err1(t) = mean(mean((Y - D*X1).^2));
30     % aOMP
31     X2 = zeros(K,N);
32     omptic2 = tic;
33     for i = 1:N
34         X2(:,i) = GenOMP(D,Y(:,i),noc,1,0.5);
35     end
36     omptime2(t) = toc(omptic2);
37     err2(t) = mean(mean((Y - D*X2).^2));
38     % gOMP
39     X3 = zeros(K,N);
40     omptic3 = tic;
41     for i = 1:N
42         X3(:,i) = GenOMP(D,Y(:,i),noc,2,0);
43     end
44     omptime3(t) = toc(omptic3);
45     err3(t) = mean(mean((Y - D*X3).^2));
46 end
47 err = [err1 err2 err3];
48 omptime = [omptime1 omptime2 omptime3];
49
50 %% Plots
51
52 figure;
53 h = plot(nocoeffs,10*log10(err));
54 title(['Mean_MSE_(dB)_over_' num2str(length(Y)) '_blocks']);
55 set(h,{'Color'},{'k','k','k'},{'LineStyle'},{'-','--','-.'},{'LineWidth',1});
56 legend({'OMP','aOMP','gOMP'},'FontSize',12);

```

```

57 xlabel(['Sparsity_/' num2str(size(Y,1)) '']);
58 ylabel('Mean_MSE_(dB)');
59 xticks(nocoeffs);
60 xlim([1,13]);
61
62 figure;
63 h = plot(nocoeffs,omptime);
64 title(['Time_consumption_(' num2str(length(Y)) '_blocks)']);
65 set(h,{'Color'},{'k';'k';'k'},{'LineStyle'},{'-';'--';'-.'},'LineWidth'
    ,1);
66 legend({'OMP','aOMP','gOMP'},'FontSize',12,'Location','NW');
67 xlabel(['Sparsity_/' num2str(size(Y,1)) '']);
68 ylabel('Time_(sec.)');
69 xticks(nocoeffs);
70 xlim([1,13]);

```

```

1 % K-SVD (as is)
2 % Training the dictionary using an online batch of small images. https://
    www.cs.toronto.edu/~kriz/cifar-10-matlab.tar.gz %
3
4
5 %% PARAMETRIZATION
6 ImageSize = 32; % number of pixels to select per block %
7 Ns = 500; % number of images in Y %
8 K = 100; % number of atoms in D %
9 J = 100; % number of algorithm iterations %
10 iterations = 1; % TOTAL ITERATIONS to average out for RESULTS %
11 nocoeffs = 5; % number of coefficients to keep per block %
12 rand_update = false; % randomly update atoms of D when true %
13
14 % data_batch_1 contains 10000 32x32 vectorized images (row-wise) %
15 load('online_batches/data_batch_1.mat');
16
17 %% TRAINING
18
19 keepMSE = zeros(length(Ns),J);
20 keepTime = zeros(length(Ns),J);
21 for n = 1:length(Ns)
22     N = Ns(n);
23
24     Y = double(data(1:N,1:ImageSize)');
25     Y = Y - repmat(mean(Y),ImageSize,1);
26     Y = normc(Y);
27
28     MSE = zeros(J,N,iterations);
29     time = zeros(J,iterations);

```

```

30
31 for it = 1:iterations
32     % D is zero mean Gaussian and column-wise normalized %
33     D = randn(ImageSize,K);
34     D = D - repmat(mean(D),ImageSize,1);
35     D = normc(D);
36     D = [ones(ImageSize,1) D(:,2:end)];
37     X = zeros(K,N);
38     for j = 1:J
39         disp(['KSVD,_N:_ ' num2str(Ns(n)) ' ,_try:_ ' num2str(it) ' ,_iteration:_ '
              num2str(j)]);
40         tic1 = tic;
41         % ----- K-SVD (as is)----- %
42         parfor i = 1:N
43             X(:,i) = GenOMP(D,Y(:,i),nocoeffs);
44         end
45         for k = pick
46             ind = find(X(k,:) ~= 0);
47             E = Y - D*X + D(:,k)*X(k,:);
48             E_reduced = E(:,ind);
49             if ~isempty(ind)
50                 [u1,s1,v1] = svds(E_reduced,1,'largest');
51                 D(:,k) = u1;
52                 X(k,:) = 0;
53                 X(k,ind) = s1*v1;
54             end
55         end
56         MSE(j,:,it) = sum((Y-D*X).^2)/ImageSize;
57         % ----- %
58         time(j+1,it) = time(j,it) + toc(tic1);
59     end
60 end
61 keepMSE(n,:) = mean(mean(MSE,2),3);
62 keepTime(n,:) = mean(time(2:end,:),2);
63 end
64
65 save(['RESU\KSVD_' num2str(iterations) 'try_' num2str(J) 'iter_' num2str(
      nocoeffs) 'coeffs.mat']);

```

```

1 % AppK-SVD
2 % Training the dictionary using an online batch of small images. https://
  www.cs.toronto.edu/~kriz/cifar-10-matlab.tar.gz %
3
4 %% PARAMETRIZATION
5 ImageSize = 32; % number of pixels to select per block %
6 Ns = 500; % number of images in Y %

```

```

7 K = 100;      % number of atoms in D %
8 J = 100;      % number of algorithm iterations %
9 iterations = 1; % TOTAL ITERATIONS to average out for RESULTS %
10 nocoeffs = 5; % number of coefficients to keep per block %
11 rand_update = false; % randomly update atoms of D when true %
12
13 % data_batch_1 contains 10000 32x32 vectorized images (row-wise) %
14 load('online_batches/data_batch_1.mat');
15
16 %% TRAINING
17 keepMSE1 = zeros(length(Ns), J);
18
19 keepTime1 = zeros(length(Ns), J);
20
21 for n = 1:length(Ns)
22     N = Ns(n);
23
24     Y = double(data(1:N, 1:ImageSize)');
25     meanY = mean(Y);
26     Y = Y - repmat(meanY, ImageSize, 1);
27     Y = normc(Y);
28
29     MSE1 = zeros(J, N, iterations);
30     time1 = zeros(J, iterations);
31
32     for it = 1:iterations
33         % D is zero mean Gaussian and column-wise normalized with a DC block at
34         % the beginning %
35         D1 = randn(ImageSize, K);
36         D1 = D1 - repmat(mean(D1), ImageSize, 1);
37         D1 = normc(D1);
38         D1 = [ones(ImageSize, 1) D1(:, 2:end)];
39         X1 = zeros(K, N);
40         for j = 1:N
41             disp(['AppKSVD, N:_' num2str(Ns(n)) ', try:_' num2str(it) ', iteration
42                 :_' num2str(j)]);
43             tic2 = tic;
44             % ----- approximate K-SVD ----- %
45             for i = 1:N
46                 X1(:, i) = GenOMP(D1, Y(:, i), nocoeffs);
47             end
48             for k = pick
49                 ind = find(X1(k, :) ~= 0);
50                 Ek = Y - D1*X1 + D1(:, k)*X1(k, :);
51                 E1 = Ek(:, ind);
52                 if norm(X1(k, :)) ~= 0
53                     D1(:, k) = (E1*X1(k, ind)')/norm(E1*X1(k, ind)');

```

```

52     end
53     X1(k,ind) = E1'*D1(:,k);
54     end
55     MSE1(j, :, it) = sum((Y-D1*X1).^2)/ImageSize;
56     % ----- %
57     time1(j+1,it) = time1(j,it) + toc(tic2);
58     end
59     end
60     keepMSE1(n, :) = mean(mean(MSE1,2),3);
61     keepTime1(n, :) = mean(time1(2:end,:),2);
62     end
63
64     save(['AppKSVD_' num2str(iterations) 'try_' num2str(J) 'iter_' num2str(
        nocoeffs) 'coeffs.mat']);

```

```

1  % LS-COPER (An AppK-SVD derivation)
2  % Training the dictionary using an online batch of small images. https://
    www.cs.toronto.edu/~kriz/cifar-10-matlab.tar.gz %
3
4  clear
5  close all
6  clc
7
8  %% PARAMETRIZATION
9  ImageSize = 32;    % number of pixels to select per block %
10 Ns = 500;        % number of images in Y %
11 K = 100;        % number of atoms in D %
12 J = 100;        % number of algorithm iterations %
13 iterations = 1;  % TOTAL ITERATIONS to average out for RESULTS %
14 nocoeffs = 5;   % number of coefficients to keep per block %
15 rand_update = false; % randomly update atoms of D when true %
16
17 % data_batch_1 contains 10000 32x32 vectorized images (row-wise) %
18 load('online_batches/data_batch_1.mat');
19
20 %% TRAINING
21 keepMSE2 = zeros(length(Ns),J);
22
23 keepTime2 = zeros(length(Ns),J);
24
25 for n = 1:length(Ns)
26     N = Ns(n);
27
28     Y = double(data(1:N,1:ImageSize)');
29     Y = Y - repmat(mean(Y),ImageSize,1);
30     Y = normc(Y);

```



```

31
32 MSE2 = zeros(J,N,iterations);
33 time2 = zeros(J,iterations);
34
35 for it = 1:iterations
36
37     % D is zero mean Gaussian and column-wise normalized with an additional
        DC block at the begining%
38     D2 = randn(ImageSize,K-1);
39     D2 = D2 - repmat(mean(D2),ImageSize,1);
40     D2 = normc(D2);
41     D2 = [ones(ImageSize,1) D2];
42     X2 = zeros(K,N);
43     for j = 1:J
44         disp(['COPER_','N:_' num2str(Ns(n)) ' ','try:_' num2str(it) ' ','iteration:_'
            ' num2str(j)]);
45         tic3 = tic;
46         % ----- LS-COPER ----- %
47         for i = 1:N
48             X2(:,i) = GenOMP(D2,Y(:,i),nocoeffs);
49         end
50         for k = pick
51             E2 = Y - D2*X2 + D2(:,k)*X2(k,:);
52             if norm(X2(k,:)) ~= 0
53                 D2(:,k) = (X2(k,:)*E2')/norm(X2(k,:))^2;
54             end
55             nd = norm(D2(:,k));
56             D2(:,k) = D2(:,k)/nd;
57             X2(k,:) = X2(k,)*nd;
58         end
59         MSE2(j,:,it) = sum((Y-D2*X2).^2)/ImageSize;
60         % ----- %
61         time2(j+1,it) = time2(j,it) + toc(tic3);
62     end
63 end
64 keepMSE2(n,:) = mean(mean(MSE2,2),3);
65 keepTime2(n,:) = mean(time2(2:end,:),2);
66 end
67
68 save(['RESU\COPER_' num2str(iterations) 'try_' num2str(J) 'iter_' num2str
    (nocoeffs) 'coeffs.mat']);

```

```

1 % Conducting the "Impainting" experiment of filling in missing pixels
    using an already trained dictionary %

```

```

2

```

```

3 % tight_subplot function being used can be found here: https://www.

```

```

    mathworks.com/matlabcentral/fileexchange/27991-tight_subplot-nh-nw-gap
    -marg_h-marg_w %
4
5 % Training the dictionary %
6 %-----%
7 dicttrain
8 % -- OR -- %
9 load('Dcats_wDC');
10 %-----%
11
12 % PARAMETRIZATION %
13 K = 100; % Number of atoms in the dictionary %
14 nocoeffs = 5; % Numver of coefficients per block %
15 r = [0.2 0.6 0.8]; % Percentage of missing pixels %
16 bsize = 8; % Block size %
17
18 files = dir('newcats\test\*.jpg');
19 MSE = cell(length(r),1);
20 na = cell(length(files),1);
21
22 for k = 1:length(files)
23 % Loading test image %
24 im = rgb2gray(im2double(imread(['newcats\test\' files(k).name])));
25 [Y_test_init,~] = im2vect(im,bsize);
26 figure(k);
27 tsh{k} = tight_subplot(2,2,[0 0],[0 0],[0 0]);
28 axes(tsh{k}(1));
29 imagesc(im); colormap('gray');
30 xticks(''); yticks('');
31 for rc = 1:length(r)
32
33 % Distrorting test image %
34 randind = randperm(bsize^2);
35 randind = randind(1:floor(r(rc)*bsize^2));
36 invind = setdiff(1:bsize^2,randind);
37 Y_dist = Y_test_init;
38 Y_dist(randind,:) = 300;
39 Y_test = Y_test_init(invind,:);
40 im_dist = vect2im(Y_dist,0,size(im));
41 im_dist(im_dist > 255) = 0;
42
43 % Recovering test image %
44 Di = D(invind,:);
45 parfor i = 1:size(Y_test,2)
46 X(:,i) = GenOMP(Di,Y_test(:,i),nocoeffs);
47 end
48 Y_rec = D*X;

```

```

49  rec_im = vect2im(Y_rec,0,size(im));
50
51  imwrite(im,'tmpres\im.jpg');
52  im = double(imread('tmpres\im.jpg'));
53  imwrite(rec_im,'tmpres\rec_im.jpg');
54  rec_im = double(imread('tmpres\rec_im.jpg'));
55
56  % Calculating Mean Squared Reconstruction Error %
57  MSE{rc}(k) = mean((im(:)/255 - rec_im(:)/255).^2);
58
59  axes(tsh{k}(rc+1));
60  imagesc(rec_im); colormap('gray');
61  xticks(''); yticks('');
62  drawnow
63  end
64  set(gcf, 'Position', [246 277 1175 689]);
65  na{k} = strtok(files(k).name, '.');
66
67  end
68
69  figure;
70  colormap([0 0 0;0.3 0.3 0.3;0.6 0.6 0.6]);
71  hb = barh(10*log10(cell2mat(MSE)'), 'grouped');
72  hb(3).FaceColor = [0 0 0];
73  hb(2).FaceColor = [0.3 0.3 0.3];
74  hb(1).FaceColor = [0.6 0.6 0.6];
75  lg = legend('20%', '60%', '80%');
76  lg.FontSize = 14;
77  yticklabels(na);
78  xlabel('MSE (dB)', 'FontSize', 14);
79  set(gca, 'TickLabelInterpreter', 'none');
80  set(gcf, 'Position', [20 229 1173 737]);

```

```

1  % Conducting the experiment of "Synthtic data". Given a Dictionary (
    initialized using data from Y) and a random sparse X, we synthesize
    data Y and try to retrieve D. %
2
3  % PARAMETRIZATION %
4  iterations = 50; % Total iterations to average out for results %
5  J = 30; % Iterations of the training algorithm %
6  K = 60; % Number of atoms in the dictioary %
7  nocoeffs = 3; % Number of coefficients %
8  bsize = 5; % Block size %
9  SNRs = [0 10 20 30 50]; % SNR scenarios %
10 lengthY = 7500; % Number of vectors in Y %
11

```

```

12 matchedD_init = zeros(length(SNRs),iterations);
13 matchedD = zeros(length(SNRs),iterations);
14 similarity = cell(length(SNRs),iterations);
15
16
17 % Rand data dictionary %
18 D_init = randn(bsize^2,K);
19 D_init = D_init - repmat(mean(D_init),bsize^2,1);
20 D_init = normc(D_init);
21
22 for t = 1:length(SNRs)
23   for it = 1:iterations
24     % Initializing X, Y %
25     X = zeros(K,lengthY);
26     parfor i = 1:lengthY
27       x = [ones(nocoeffs,1); zeros(K-nocoeffs,1)];
28       randind = randperm(K);
29       X(:,i) = x(randind);
30     end
31     Y = D_init*X;
32     Ps = mean(Y(:).^2);
33     Pn = Ps*10^(-SNRs(t)/10);
34     Y = Y + Pn*randn(size(Y));
35     Y = Y - repmat(mean(Y),bsize^2,1);
36     Y = normc(Y);
37
38     % Dictionary initialization with synthetic data %
39     D = Y(:,1:K);
40
41     % Sparse decomposition of Y %
42     for j = 1:J
43       disp(['iteration_', num2str(j)]);
44       parfor i = 1:length(Y)
45         X(:,i) = GenOMP(D,Y(:,i),nocoeffs);
46       end
47       for k = 1:K
48         ind = find(X(k,:) ~= 0);
49         E = Y - D*X + D(:,k)*X(k,:);
50         E_r = E(:,ind);
51         if ~isempty(ind)
52           [u1,s1,v1] = svds(E_r,1,'largest');
53           D(:,k) = u1;
54           X(k,:) = 0;
55           X(k,ind) = s1*v1;
56         end
57       end
58     end

```

```

59 similarity{t,it} = abs(D_init'*D);
60 err_map = 1 - similarity{t,it};
61 errthres = 0.01;
62 err_map(err_map > errthres) = 0;
63 err_map(err_map > 0) = 1;
64 distD_init = sum(err_map,2);
65 matchedD_init(t,it) = numel(find(distD_init > 0));
66 distD = sum(err_map);
67 matchedD(t,it) = numel(find(distD > 0));
68 end
69 end
70
71 save('syntheticRand7500.mat');
72
73 % Plots %
74 figure;
75 load('syntheticRand.mat');
76 plot(matchedD_init,'.k','LineWidth',0.0001); hold
77 plot(mean(matchedD_init,2),'--*k','LineWidth',2); hold
78 xlabel('SNR (dB)','FontSize',12'); ylabel(['#matched/' num2str(K)],'
      FontSize',12);
79 xlim([0 length(SNRs)+1]);
80 ylim([0 K]);
81 xticks(1:5);
82 xticklabels({'0','10','20','30','50'});
83 grid
84 title('1500_training_data_points_|_50_iter. ');
85 print(gcf,'res\synthetic1500','-dpng','-r300');
86 figure;
87 load('syntheticRand7500.mat');
88 plot(matchedD_init,'.k','LineWidth',0.0001); hold
89 plot(mean(matchedD_init,2),'--*k','LineWidth',2); hold
90 xlabel('SNR (dB)','FontSize',12'); ylabel(['#matched/' num2str(K)],'
      FontSize',12);
91 xlim([0 length(SNRs)+1]);
92 ylim([0 K]);
93 xticks(1:5);
94 xticklabels({'0','10','20','30','50'});
95 grid
96 title('7500_training_data_points_|_50_iter. ');
97
98 %%
99 for t = 1:size(similarity,1)
100 distD_init = cell(size(similarity,2),1);
101 distD = cell(size(similarity,2),1);
102 figure;
103 tsh = tight_subplot(1,2,[0 0.03],[0.04 0.04],[0.03 0]);

```

```

104  subplot(['SNR:_' num2str(SNRs(t)) 'dB_|_50_iterations']);
105  for it = 1:size(similarity,2)
106    err_map = 1 - similarity{t,it};
107    errthres = 0.01;
108    err_map(err_map > errthres) = 0;
109    err_map(err_map > 0) = 1;
110    distD_init{it} = sum(err_map,2);
111    matchedD_init(t,it) = numel(find(distD_init{it} > 0));
112    distD{it} = sum(err_map);
113    matchedD(t,it) = numel(find(distD{it} > 0));
114  end
115
116  p(1:5) = 1-5/50;
117  p(6:size(similarity,2)) = 1 - (6:size(similarity,2))/50;
118  axes(tsh(1));
119  hold on;
120  for it = 1:size(similarity,2)
121    stem3(1:K,it*ones(1,K),distD_init{it},'filled','Color',p(it)*[1 1 1],'
      LineStyle','none');
122    title(['D_{init},_matched:_' num2str(matchedD_init(t,it)) '/' num2str(K
      )], 'FontSize',12);
123    xlabel('Times_used','FontSize',12); xlabel('Atoms','FontSize',12);
      ylabel('Iterations','FontSize',12); zlim([0 K]); zlim([0 2]);
124    zticks(0:2);
125  end
126  grid on;
127  view(45, 10);
128  hold off;
129  axes(tsh(2));
130  hold on;
131  for it = 1:size(similarity,2)
132    stem3(1:K,it*ones(1,K),distD{it},'filled','Color',p(it)*[1 1 1],'
      LineStyle','none');
133    title(['D,_matched:_' num2str(matchedD(t,it)) '/' num2str(K)], 'FontSize
      ',12);
134    xlabel('Times_used','FontSize',12); xlabel('Atoms','FontSize',12);
      ylabel('Iterations','FontSize',12); zlim([0 K]); zlim([0 2]);
135    zticks(0:2);
136  end
137  grid on;
138  view(45, 10);hold off;
139  set(gcf,'Position',[1 31 1920 973]);
140  print(gcf,['res\syntheticMatchesSNR' num2str(SNRs(t))'],'-dpng','-r300');
141  end

```

```

2
3 % Parameters %
4 noblocks = 500; % Number of blocks to take per image %
5 bsize = 8; % Block size %
6 K = 100; % Number of atoms in the dictionary %
7 J = 30; % Training algorithm iterations%
8 nocoeffs = 5; % Number of non zero coefficients to keep per block %
9
10 % Loading and Vectorizing images %
11 files = dir('newcats\train\*.jpg');
12 Y = zeros(bsize^2,length(files)*noblocks);
13 tsh = tight_subplot(3,5,[0 0],[0 0],[0 0]);
14 for i = 1:length(files)
15     im = rgb2gray(im2double(imread(['newcats\train\' files(i).name]]));
16     axes(tsh(i));imagesc(im); colormap('gray'); xticks(''); yticks('');
17
18     [Y_all,~] = im2vect(im,bsize);
19     ind = randperm(size(Y_all,2));
20     Y(:,(i-1)*noblocks+1:i*noblocks) = Y_all(:,ind(1:noblocks));
21
22 end
23
24 % Training the dictionary %
25
26 X = zeros(K,length(files)*noblocks);
27 D = randn(bsize^2,K);
28 D = D - repmat(mean(D),bsize^2,1);
29 D = normc(D);
30 D = [ones(bsize^2,1) D(:,2:end)];
31 figure(1);
32 imagesc(vect2im(D,0,[80,80])); colormap('gray'); xticks(''); yticks('');
33 figure(2);
34 imagesc(D'*D > 0.9); colormap('gray'); xticks(''); yticks('');
35 drawnow;
36
37 for j = 1:J
38     disp(['iteration_' num2str(j)]);
39     parfor i = 1:length(files)*noblocks
40         X(:,i) = GenOMP(D,Y(:,i),nocoeffs);
41     end
42     for k = 1:K
43         ind = find(X(k,:) ~= 0);
44         E = Y - D*X + D(:,k)*X(k,:);
45         E_r = E(:,ind);
46         if ~isempty(ind)
47             [u1,s1,v1] = svds(E_r,1,'largest');
48             D(:,k) = u1;

```

```

49     X(k,:) = 0;
50     X(k,ind) = s1*v1;
51     end
52     end
53     figure(1);
54     imagesc(vect2im(D,0,[80,80])); colormap('gray'); xticks(''); yticks('');
55     figure(2);
56     imagesc(D'*D > 0.9); colormap('gray'); xticks(''); yticks('');
57     drawnow;
58     end
59
60     save('D','Dcats_wDC');
61
62     % Plots %
63
64     figure;
65     imagesc(vect2im(D,0,[80 80])); colormap('gray');
66     title('D:_Trained_w/_15_images_x_5000_8x8_blocks');
67     xticks(''); yticks('');
68     print(gcf,'res\Dcats8','-dpng','-r300');
69     figure;
70     for i = 1:length(D)
71         imagesc(reshape(D(:,i),8,8)); title(num2str(i)); colormap('gray');
72         drawnow
73         pause
74     end
75
76     figure;
77     A = D'*D;
78     [h,c] = imhist(A);
79     bar(c,h/numel(A),'FaceColor','k');
80     xlabel('Correlation','FontSize',14); ylabel('pdf','FontSize',14);
81     title('Autocorrelation_pdf_of_the_trained_dictionary','FontSize',14);
82
83     figure;
84     imagesc(abs(D'*D)); colormap('gray');
85     title('Dictionary_Autocorrelation_Matrix');
86     xlabel('Atoms_of_D'); ylabel('Atoms_of_D');
87     xticks('');yticks('');
88
89     figure;
90     imagesc(abs(D'*D)>0.8); colormap('gray');
91     title('Dictionary_Autocorrelation_Matrix_(>0.8)');
92     xlabel('Atoms_of_D'); ylabel('Atoms_of_D');
93     xticks('');yticks('');
94
95     figure;

```



```

96 subplot(2,2,[1,2]);
97 mse = mean((Y-D*X).^2);
98 [max_mse,max_ind] = max(mse);
99 stem(Y(:,max_ind),'filled','Color',[0.6 0.6 0.6]); hold;
100 stem(D*X(:,max_ind),'filled','k','LineWidth',2); hold;
101 legend('Original_block','Reconstructed');
102 xlim([0 bsize^2+1]);
103 title(['Block_with_Max._Error_(~= num2str(max_mse) ')']);
104 xlabel('Pixel'); ylabel('Intensity');
105 subplot(2,2,[3,4]);
106 hb = bar(X(:,max_ind),'histc');
107 set(hb,'FaceColor','k');
108 title('Corresponding_sparse_vector');
109 xlabel('Atom'); ylabel('Weight')
110 xlim([0 K]);
111
112 figure;
113 subplot(2,2,[1,2]);
114 [min_mse,min_ind] = min(mse(mse>0));
115 stem(Y(:,min_ind),'filled','Color',[0.6 0.6 0.6]); hold;
116 stem(D*X(:,min_ind),'filled','k','LineWidth',2); hold;
117 legend('Original_block','Reconstructed');
118 xlim([0 bsize^2+1]);
119 title(['Block_with_Min._Error_(~= num2str(min_mse) ')']);
120 xlabel('Pixel'); ylabel('Intensity');
121 subplot(2,2,[3,4]);
122 hb = bar(X(:,min_ind),'histc');
123 set(hb,'FaceColor','k');
124 title('Corresponding_sparse_vector');
125 xlabel('Atom'); ylabel('Weight')
126 xlim([0 K]);

```

```

1 % Conducting the experiment of reconstructing noisy images using an
   already trained dictionary. %
2
3 % tight_subplot function being used can be found here: https://www.
   mathworks.com/matlabcentral/fileexchange/27991-tight_subplot-nh-nw-gap
   -marg_h-marg_w %
4
5 % Training the dictionary %
6 %-----%
7 dicttrain
8 % -- OR -- %
9 load('Dcats_wDC');
10 %-----%
11

```

```

12 K = 100; % Number of atoms in the dictionary %
13 K = 100; % Number of atoms in the dictionary %
14 ncoeffs = 5; % Numver of coefficients per block %
15 bsize = 8; % Block size %
16
17 %% Reconstruction testing
18
19 files = dir('newcats\test\*.jpg');
20 SNR = [-10 0 10]; % SNR scenarios (in dB)%
21 mse = cell(length(SNR),1);
22 R = zeros(1,length(files));
23 na = cell(length(files),1);
24
25 for k = 1:length(files)
26 im = rgb2gray(im2double(imread(['newcats\test\' files(k).name])));
27 imwrite(im,'tmpres\im.jpg');
28 imp = double(imread('tmpres\im.jpg'));
29 figure(k);
30 tsh{k} = tight_subplot(2,2,[0 0],[0 0],[0 0]);
31 axes(tsh{k}(1));
32 imagesc(imp); colormap('gray');
33 xticks(''); yticks('');
34 na{k} = strtok(files(k).name, '.');
35 for n = 1:length(SNR)
36 [Y_test,~] = im2vect(im,bsize);
37 Y_test = Y_test + sqrt(var(Y_test(:))/(10^(SNR(n)/10)))*randn(size(
    Y_test));
38 disp(['SNR:' num2str(SNR(n)) 'dB, _Image:_' files(k).name]);
39
40 % Sparse coding %
41 X_test = zeros(K,size(Y_test,2));
42 parfor i = 1:size(Y_test,2)
43 X_test(:,i) = GenOMP(D,Y_test(:,i),ncoeffs);
44 end
45 rec_Y = D*X_test;
46
47 rec_im = vect2im(rec_Y,0,size(im));
48
49 imwrite(rec_im,'tmpres\rec_im.jpg');
50 rec_im = double(imread('tmpres\rec_im.jpg'));
51
52 mse{n}(k) = mean((im(:)/255 - rec_im(:)/255).^2);
53
54 figure(k);
55
56 axes(tsh{k}(n+1));
57 imagesc(rec_im); colormap('gray');

```

```
58  xticks(''); yticks('');
59  drawnow
60  end
61  set(gcf, 'Position', [20 229 1173 737]);
62  end
63
64  figure;
65  colormap([0 0 0;0.3 0.3 0.3;0.6 0.6 0.6]);
66  hb = barh(cell2mat(mse)', 'grouped');
67  hb(3).FaceColor = [0 0 0];
68  hb(2).FaceColor = [0.3 0.3 0.3];
69  hb(1).FaceColor = [0.6 0.6 0.6];
70  lg = legend('-10dB', '0dB', '10dB');
71  lg.FontSize = 12;
72  yticklabels(na);
73  xlabel('MSE', 'FontSize', 16);
74  set(gca, 'TickLabelInterpreter', 'none');
75  set(gcf, 'Position', [20 229 1173 737]);
```
