Θέματα μεταπτυχιακών & διπλωματικών εργασιών για το ακ. έτος 2025- 2026

Π. Βασιλειαδης – 2025-07-08

Τα θέματα διπλωματικών και μεταπτυχιακών εργασιών συνήθως επεκτείνουν κάποια από τα υπάρχοντα εργαλεία που αναπτύσσουμε στην ομάδα μου – λίγο πιο σπάνια, ξεκινούν νέα εργαλεία.

https://github.com/DAINTINESS-Group

https://www.youtube.com/playlist?list=PL3G-N7ZzyiDfpsMLCQcm_L9KEVLDkC454

Το παρόν κείμενο περιγράφει τα προτεινόμενα θέματα και, για να διευκολύνει την κατανόηση, δίνει και μια γενική εικόνα για τα εργαλεία που αναπτύσσουμε.

Αναγκαστικά μια εργασία πρέπει να ολοκληρωθεί αυστηρά εντός ενός έτους από την ανάληψή της. Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java & σχεδίασης ΟΟ λογισμικού. Στις περισσότερες διπλωματικές χρειάζεται να μπορείτε να συνδυάσετε frameworks – e.g., Apache Spark, και ενίοτε να πρέπει να δουλέψετε με το συνδυασμό κώδικα και βάσεων δεδομένων. Επειδή θα ρωτήσετε: πουθενά δε γράφει Spring & Spring Boot. Θα δούμε όπου κολλάει να μπει web-based front-end πώς μπορεί να κολλήσει στο back-end της διπλωματικής.

Γενικά προχωράμε ως ομάδα με agile working methods, ήτοι **weekly runs & group meetings**. Κείμενο και κώδικας γράφονται εναλλάξ. Παντού δουλεύουμε github && εκτός ειδικών περιπτώσεων, Eclipse.

Επειδή πλέον οι καιροί είναι πονηροί, **δεν θα δεχτώ να εξετάσω διπλωματικές από φοιτητές, που δεν βλέπω στο εξάμηνο και μου παρουσιάζουν ξαφνικά ένα κείμενο και ένα κώδικα προς εξέταση στο τέλος**. Θα πρέπει να ασχολείστε συστηματικά με την εργασία σας στη διάρκεια του εξαμήνου.

Σημειογραφία: ό,τι (αν τυχόν) θα δείτε με …

- ~~Strikethrough~~, είναι έτοιμο.

- Υπογραμμισμένο με τελείες, είναι μέρος του συστήματος υπό ανάπτυξη/προς βελτίωση.

- **Bold** είναι για τις εκφωνήσεις εργασιών στο παρόν κείμενο.

- αχνά γκρι γράμματα είναι μέρος του σχεδίου, αλλά όχι για τώρα.

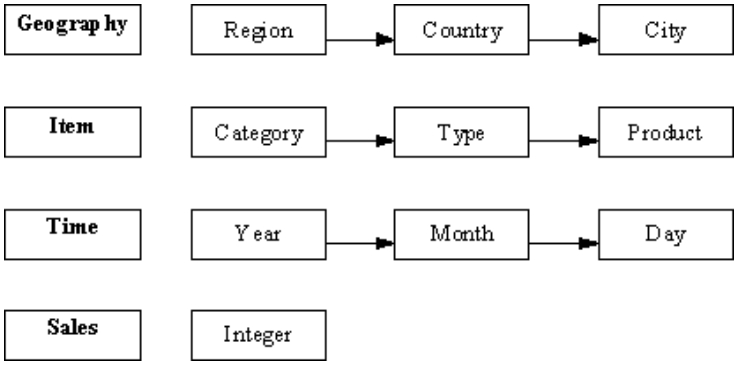
Contents

1 Business Intelligence Analytics

Οι εφαρμογές On-Line Analytical Processing (OLAP) αντιμετωπίζουν τα δεδομένα σαν *κύβους* (*data cubes*), οι οποίοι αποτελούνται από διαστάσεις και μετρήσιμες τιμές. Οι κύβοι αποτελούνται από *διαστάσεις* (χρόνος, γεωγραφική περιοχή κ.λπ.) και *μετρήσιμα μεγέθη* (π.χ. σύνολο πωλήσεων). Οι διαστάσεις οργανώνονται σε λογικές *ιεραρχίες*. Για παράδειγμα, ο χρόνος μπορεί να οργανωθεί στην ιεραρχία χρόνος, μήνας, μέρα. Για παράδειγμα, οι πωλήσεις ενός οργανισμού μπορούν να μοντελοποιηθούν σαν ένας κύβος της μορφής

Day (Year-month-day)	Product	City	Sales
1997-01-01	“Report to El Greco”	Rhodes	15
1997-01-01	“Ace of Spades”	Paris	8
1997-01-01	“Report to El Greco”	Athens	11

με διαστάσεις οργανωμένες σε *ιεραρχίες*, όπως:



Ο χρήστης μπορεί να κάνει ερωτήσεις όπως μετασχηματισμός του κύβου σε διαφορετικά επίπεδα ιεραρχίας, επιλογή κάποιων υποσυνόλων του κύβου κ.λ.π. Σε οποιοδήποτε textbook βάσεων δεδομένων, θα βρείτε ένα κεφάλαιο για αποθήκες δεδομένων και OLAP για τις βασικές έννοιες.

Οι παρακάτω εργασίες εντάσσονται στο μεγάλο στόχο να αναθεωρηθεί εκ βάθρων το μοντέλο κύβων και η απάντηση κάθε ερώτησης να επαυξηθεί με αποτελέσματα μηχανικής μάθησης, εκτός από τα δεδομένα αυτά καθαυτά.

Ο χρήστης υποβάλλει ένα OLAP ερώτημα (πρακτικά ένα ερώτημα συνάθροισης που περιλαμβάνει group-by και where clause). Το ερώτημα εκτελείται σε μία back-end OLAP engine, η οποία είναι ήδη σε μια πρώτη εκδοχή. Τα αποτελέσματα πρέπει να παρουσιαστούν στον χρήστη σε μια εκδοχή “ευέλικτου Excel” που επιτρέπει την οπτικοποίηση των αποτελεσμάτων και τη διαδραστική τους επερώτηση.

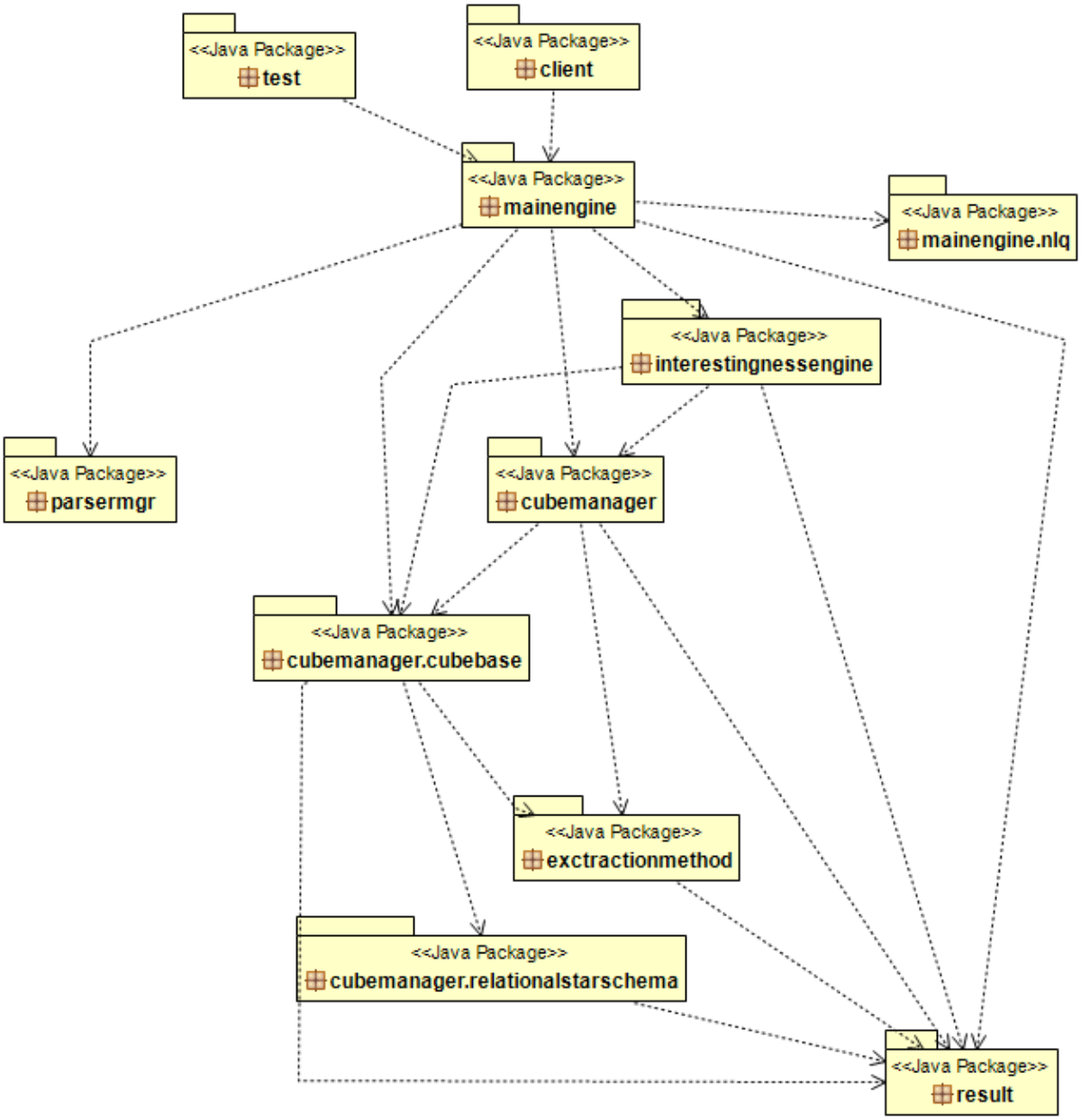
1.1 Υπόβαθρο για τις εργασίες

Στο Παν. Ιωαννίνων, έχουμε ήδη κατασκευάσει ένα σύστημα επερώτησης data cubes. Το σχετικό project έχει το δικό του repository στο Github:

https://github.com/pvassil/DelianCubeEngine

Στο σύστημα Delian Cubes, εξετάζουμε την απάντηση σε ερωτήματα μέσω *intentional queries & result mining*. Ο χρήστης υποβάλλει ένα OLAP ερώτημα (πρακτικά ένα ερώτημα συνάθροισης που

περιλαμβάνει group-by και where clause). Το σύστημα παράγει ως έξοδο (α) τα αποτελέσματα της ερώτησης, (β) μια λίστα από μοντέλα που είναι το αποτέλεσμα του τρεξίματος αλγορίθμων εξόρυξης γνώσης και στατιστικών (correlations, clusters, …) και (γ) ένα mapping μεταξύ των (α) και (β).



Το διάγραμμα πακέτων του Delian Cubes.

Η βασική κλάση που ξεκινά την απάντηση ερωτήσεων είναι η κλάση SessionQueryProcessingEngine στο πακέτο mainengine.

Αν θέλετε να καταλάβετε πώς δουλεύει το σύστημα, προσπαθήστε να καταλάβετε πώς απαντά ερωτήσεις (α) με models, και (β) με metadata, πώς δηλαδή μια εισαγόμενη συμβολοσειρά μετατρέπεται σε CubeQuery, και πώς αυτό εκτελείται και συμπληρώνεται με αποτελέσματα. (ναι, είναι ένα μονοπάτι all the way down to the result package).

Η δομή ενός ερωτήματος κύβων που υποστηρίζεται από το Delian είναι η παρακάτω:

CubeName: [το όνομα του κύβου που θα υποβάλλουμε το ερώτημα]
Name: [το ψευδώνυμο του ερωτήματος]
AggrFunc: [η συναθροιστική συνάρτηση που εφαρμόζεται στη μετρική του ερωτήματος]
Measure: [η μετρική που επιστρέφει το ερώτημα]
Gamma Expressions: [συνθήκες ομαδοποίησης του ερωτήματος]
Sigma Expressions: [συνθήκες φιλτραρίσματος του ερωτήματος]

Η παραπάνω έκφραση μετατρέπεται από το Delian στη παρακάτω ισοδύναμη SQL έκφραση, η οποία εκτελείται είτε από τη MySQL είτε από το Apache Spark.

SELECT [Gamma Expressions], [AggrFunc]([Measure])
FROM [Dimension & Fact Tables involved in WHERE & GROUP BY clause]
WHERE [JOIN Expressions] AND [Sigma Expressions]
GROUP BY [Gamma Expressions]
ORDER BY [Measure];

Στο Παν. Ιωαννίνων, έχουμε μια πρόταση για το πώς θα επερωτώνται οι κύβοι στο μέλλον. Δείτε στο

http://www.cs.uoi.gr/~pvassil/projects/olap_III/index.html

- το σχετικό άρθρο στο DOLAP 2018 [http://ceur-ws.org/Vol-2062/paper07.pdf]

- το πιο αναλυτικό άρθρο στο Information Systems 2019 [https://www.cs.uoi.gr/~pvassil/publications/2019_IS_IntentionalModel/IS2019_VassiliadisMarcelRizzi_CR_PreprintUoI.pdf]

Στόχος των παραπάνω εργασιών είναι να προστεθούν στο Delian Cubes οι υλοποιήσεις κάποιων νέων λογικών τελεστών πρόθεσης (*intentional operators*). Με τον όρο intentional operator, εννοούμε έναν αυστηρά ορισμένο τελεστή, με την δική του σύνταξη και σημασιολογία, ο οποίος θα δίνει στον χρήστη την δυνατότητα να εκφράζει μέσω ενός ερωτήματος ποια είναι η πρόθεση του για τα δεδομένα.

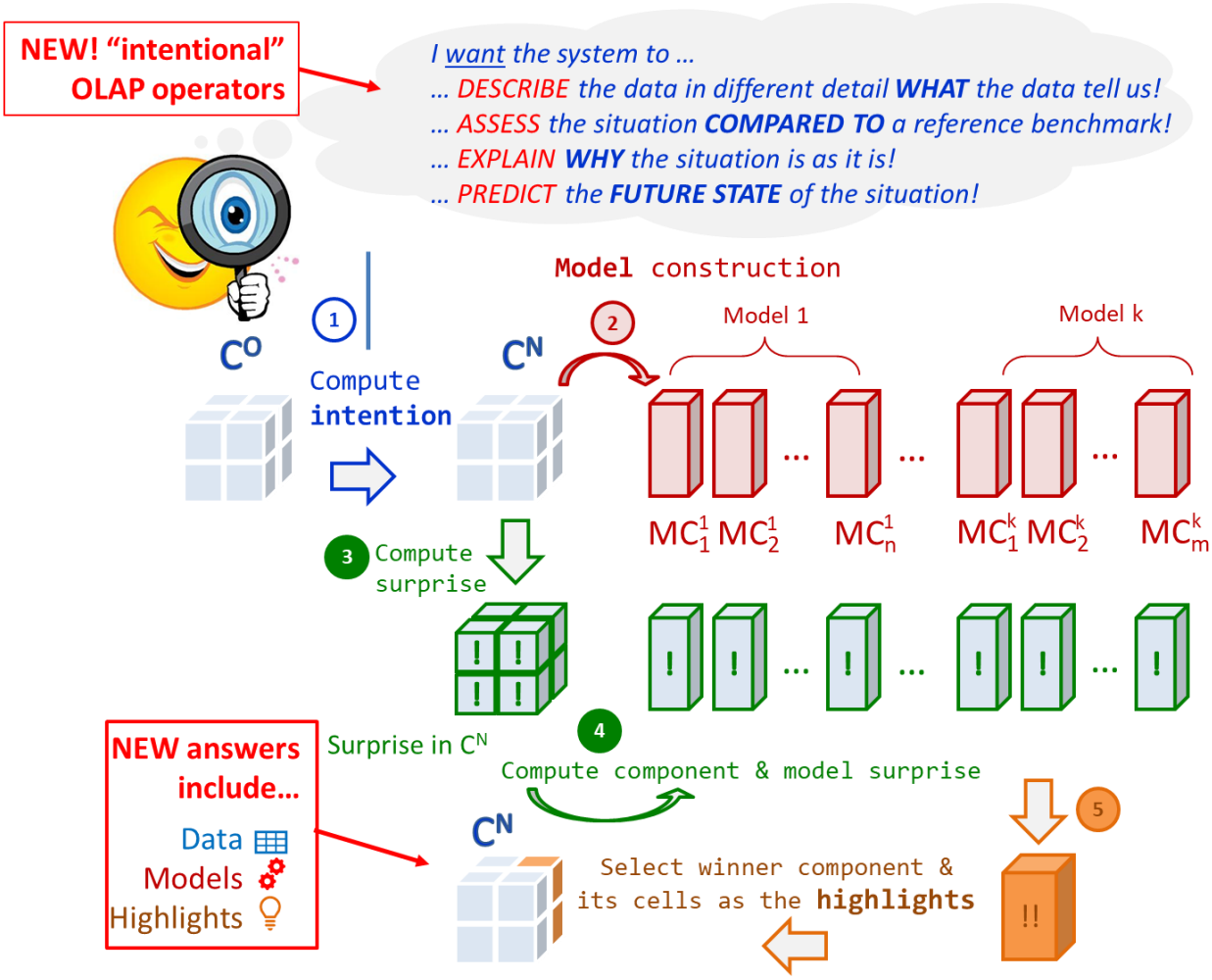
Για παράδειγμα, έστω ότι ο χρήστης θέλει να αναλύσει ένα φαινόμενο όπως η πτώση των πωλήσεων των γαλακτοκομικών τον Μάιο. Το παραπάνω κρύβει την πρόθεση της ανάλυσης. Η μέθοδος που ο χρήστης θα ακολουθούσε χωρίς την ύπαρξη ενός operator θα ήταν η υποβολή μιας σειράς ερωτημάτων τα αποτελέσματα των οποίων θα εξηγούσαν το φαινόμενο. Μια πιθανή μέθοδος θα ήταν:

- Ο έλεγχος των καθημερινών πωλήσεων των γαλακτοκομικών για τον Μάιο
- Η έλεγχος των καθημερινών πωλήσεων των γαλακτοκομικών για προηγούμενους μήνες που οι πωλήσεις ήταν «φυσιολογικές»
- Η σύγκριση των καθημερινών πωλήσεων για τον Μάιο με των προηγούμενων μηνών

Όπως γίνεται εύκολα αντιληπτό, η παραπάνω μέθοδος συνεπάγεται μια σειρά από ερωτήματα καθώς και περαιτέρω ανάλυση των αποτελεσμάτων για την εξαγωγή συμπερασμάτων. Η δημιουργία ενός operator θα αυτοματοποιούσε και θα επιτάχυνε την διαδικασία.
Παρακάτω περιγράφονται οι intentional operators που επιδιώκουμε να υλοποιηθούν και να ενταχθούν στο σύστημα Delian Cubes. Κάθε operator θα αντιστοιχεί και σε μία διπλωματική εργασία.

Η βασική ιδέα πίσω από το intentional model είναι η εξής:

- Ρίχνω μια intentional ερώτηση (predict, assess, …) η οποία κρύβει ουσιαστικά ένα group by query (μαζί με τις σχετικές επιλογές -φίλτρα, και τα αναγκαία joins)
- Το ερώτημα απαντιέται με δεδομένα, αλλά σε αυτά τα δεδομένα βάζω να τρέξουν κάποια models (για παράδειγμα: ranking of result cells, outlier detection, K-means clustering of result cells – στο πακέτο results υπάρχουν οι σχετικές υλοποιήσεις).
- Κάθε μοντέλο, έχει ένα σύνολο από vectors, όπου για κάθε κελί βάζει μια τιμή. Π.χ., στο clustering έχει από ένα vector για κάθε cluster και κάθε κελί του αποτελέσματος, αναλόγως αν είναι στο εν λόγω cluster έχει 1, αλλιώς 0 – ή το outlier detection έχει 2 components, true if the cell is indeed an outlier και false if not.
- Το άρθρο (αλλά όχι το σύστημα) προτείνει ένα μηχανισμό μέτρησης της έκπληξης και της επιλογής των πιο ενδιαφερόντων από τα highlights.



Σε όσους τελεστές ακολουθούν, δίνεται μια «προσεγγιστική» γλώσσα υποβολής ερωτημάτων σαν SQL. Αυτό που μας νοιάζει όμως, είναι να υλοποιηθεί η σχετική μέθοδος στο API του Delian με τις σωστές παραμέτρους – το συντακτικό είναι για να καταλαβαίνει κανείς χοντρικά τι ζητά η κάθε ερώτηση.

1.2 Extending Cube queries and the DESCRIBE Operator

Ο Describe operator χρησιμοποιείται για να παρέχει στον χρήστη περισσότερες πληροφορίες σχετικά με κάποια δεδομένα και στην περίπτωση μας σχετικά με μία ή περισσότερες μετρήσεις ενός κύβου. Εκφράζει δηλαδή την πρόθεση του χρήστη να μάθει περισσότερα για κάποια δεδομένα. Όπως σε όλους τους operators, η κατασκευή του operator προϋποθέτει την δημιουργία ενός συντακτικού, την εκτέλεση μιας σειράς από μοντέλα, η βαθμολόγηση των μοντέλων και η επιλογή των πιο σημαντικών (*highlights*) και η παρουσίασή τους στον χρήστη.

Ο τελεστής Describe βασικά υποβάλει ένα απλό cube query στο σύστημα (βλ. παραπάνω, στη γενική περιγραφή, για τη δομή του). Για τις ερωτήσεις κύβων και τον operator Describe, υπάρχει ήδη υλοποιημένη στο σύστημα Delian Cubes μια μορφή συντακτικού για την υποβολή ερωτημάτων που επιδέχεται σημαντική βελτίωση. Η ουσία της διπλωματικής είναι να βελτιωθεί και επεκταθεί το συντακτικό και η αναπαράσταση cube queries. Συγκεκριμένα, πρέπει:

- Να **αντικατασταθεί ο απλοϊκός parser που έχουμε τώρα** για τα υπάρχοντα απλά ερωτήματα κύβων με μια πιο οργανωμένη επεξεργασία σε ένα «καθαρό» parser
- Να **προσθέσουμε τη δυνατότητα υποστήριξης σύνθετων ερωτημάτων** που να έχουν
    - Πολλαπλά είδη απλών φίλτρων (και όχι μόνο ισότητα, όπως έχουν τώρα), π.χ. Year > 2018
    - Σύνθετα φίλτρα (π.χ., μια τιμή να ανήκει σε ένα σύνολο τιμών που προκύπτει από άλλο ερώτημα)
    - Πολλαπλά measures, some of them derived (profit = sales - cost)
    - Multiple groupers per dimension (group by month, quarter, year, product)
- Να προσθέσουμε το να κάνουμε **join cube query results, on the basis of a matching function**. Π.χ. (product, sum(sales) from cube storeSales) JOIN (product, sum(sales) from cube webSales))

Για πολλά από τα παραπάνω υπάρχει ήδη κώδικας, αλλά δεν είναι ομογενοποιημένος σωστά.

Τα σύνθετα φίλτρα είναι σε μια βαριάντα της μορφής `dim.levelX IN {listOfValues}` ή `dim.levelX WITH dim.levelX.attribute > 10` το οποίο μεταφραζόμενο σε SQL θα γίνεται κάτι σαν

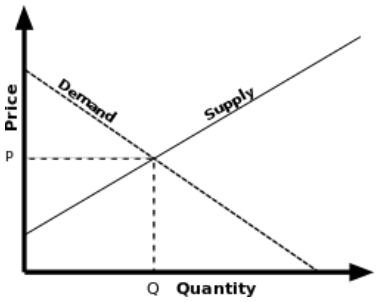
```
dim.attributeX IN (SELECT DISTINCT dim.attributeX FROM dim WHERE dim. levelX.attribute > 10)
```

Ο καθορισμός της ακριβούς σύνταξης και σημασιολογίας των επεκτάσεων θα είναι μέρος της διπλωματικής.

1.3 VERIFY Operator

Ο τελεστής VERIFY είναι ένας τελεστής που θέλει να επιβεβαιώσει μια υπόθεση εργασίας για τα δεδομένα. Ας υποθέσουμε ότι κάποιος θέλει να επιβεβαιώσει το μοντέλο του https://en.wikipedia.org/wiki/Economic_equilibrium για το μοντέλο supply-demand ενός προϊόντος

Qs = 125 + 1.5P, Qd = 189 – 2.25P



στα δεδομένα των πωλήσεων σε ένα κύβο. Πώς θα το έκανε?

Θέλουμε διάφορες εκδοχές του τελεστή:

Α. Ο τελεστής VERIFY στην πιο απλή μορφή παίρνει ως παράμετρο μια απλή φόρμουλα ενός ζυγισμένου αθροίσματος με παραμέτρους πεδία και βάρη $y = w_1A_1 + w_2A_2 + ... + w_nA_n$ και προσπαθεί να μετρήσει πόσο απέχει από την σχετική μέτρηση.

```
with SALES
FOR product.type = ’Fresh Fruit’, geo.country =’Italy’ BY product
VERIFY sum(quantity) FOLLOWS LinearRegression({geo.country.Population,0.002},
{product.price, 0.5})
```

Β. Στο σύστημα Πυθία

https://github.com/DAINTINESS-Group/Pythia/tree/master/src/main/java/gr/uoi/cs/pythia/regression

έχουμε έτοιμο τον κώδικα που δοκιμάζει να βρει **αυτόματα τη βέλτιστη γραμμική συσχέτιση**. Οπότε θα είχαμε κάτι σαν το εξής συντακτικό

```
with SALES
FOR product.type = ’Fresh Fruit’, geo.country =’Italy’ BY product
VERIFY sum(quantity) FOLLOWS AutoLinearRegression({geo.country.Population}, {product.price})
```

Γενικά θα υπάρχει μια ευελιξία στο τι παραμέτρους θα δίνουμε στην συνάρτηση regression.

Γ. Σε μια εναλλακτική μορφή του, **ο τελεστής συγκρίνει τα αποτελέσματα δύο ερωτήσεων** και ανάγει τα επί μέρους αποτελέσματα με μια συνάρτηση αναγωγής

```
with SALES
FOR product.type = ’Fresh Fruit’, geo.country =’Italy’ BY product
VERIFY sum(quantity) FOLLOWS AutoLinearRegression({geo.country.Population}, {product.price})
COMPARE continent=’EU’ AutoLinearRegression({geo.contintent.Population}, {product.price})
```

Είναι απαραίτητο, τα αποτελέσματα του τελεστή να οργανωθούν ως highlights ανάλογα με τις τιμές σύγκρισης και σφάλματος που δίνει ο τελεστής.

Δ. *[Αν τελικά έχουμε χρόνο -- λογικά θα έχουμε]* Σε μια εναλλακτική μορφή του ο τελεστής παίρνει για είσοδο **ένα δέντρο απόφασης**, και αποτιμά κατά πόσο οι τιμές ενός κύβου συμμορφώνονται με τις τιμές-φύλλα που το δέντρο ορίζει (ενδεχομένως μέσω μιας συνάρτησης labeling).

1.4 Για όλα τα παραπάνω

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς / MSc*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java, Apache Spark

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία βρίσκεται κυρίως στη σωστή σχεδίαση του λογισμικού, την κατανόηση του υπάρχοντος κώδικα και το κούμπωμα του νέου κώδικα με απλό τρόπο. Τα οφέλη για ένα φοιτητή είναι: (α) εμπλοκή στο χώρο του OLAP && Business Intelligence, και (β) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού.

Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε προγραμματιστικά θέματα και σε θέματα διαχείρισης δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java και βάσεων δεδομένων.

ΤΡΟΠΟΣ ΕΡΓΑΣΙΑΣ: Το σύστημα Delian Cubes βρίσκεται στο δικό του repository στο Github,το οποίο επιβλέπει ο Υποψήφιος Διδάκτορας Μάριος Ιακωβίδης, από τον οποίο θα έχετε και τη σχετική υποστήριξη στα αντίστοιχα θέματα.

Στο σύστημα, στη παρούσα φάση είναι αρκετά εύκολο να προστεθεί ένας ή παραπάνω operators ταυτόχρονα. Ουσιαστικά, θα έχουμε μια νέα service class, που θα βοηθά τη SessionQueryProcessingEngine, για κάθε operator. Το πλάνο είναι να φτιαχτεί ένα νέο branch για κάθε έναν Operator που υλοποιείται και σε συχνά διαστήματα να κάνουμε merge στο Master branch ώστε όλοι να είμαστε up to date χωρίς να επηρεάζουμε την δουλειά των υπολοίπων. Επομένως, αν και σχεδόν πάντα το merge θα είναι διαδικασία που θα μας αφορά όλους και θα γίνεται όταν όλοι μπορούν, η γνώση του git και των εντολών του είναι σημαντική.

Όπως ίσως γνωρίζετε, ο τρόπος λειτουργίας της ομάδας εδώ είναι με μικρά increments δουλειάς και περιλαμβάνει εβδομαδιαία MS Teams meetings. Η πράξη έχει αποδείξει καταφανώς ότι η συμμετοχή είναι πολύ χρήσιμη, στο να βοηθά όλους να μαθαίνουν για παρόμοια προβλήματα και θέματα, αλλά και στο να καλλιεργεί ένα ομαδικό πνεύμα. Στο βαθμό που οι εργασίες είναι συναφείς, αυτό είναι ακόμα πιο έντονο.

2 Timeseries Management

Μια χρονοσειρά είναι μια λίστα από τιμές (που ευλόγως μπορούμε να υποθέσουμε ότι είναι πραγματικοί αριθμοί). Κάθε μία τιμή έχει λοιπόν την θέση της στη λίστα και ενδεχομένως και άλλες ιδιότητες, όπως π.χ., κάποιο χρονόσημο. Δείτε για παράδειγμα μια χρονοσειρά με δεδομένα εξαγωγών.

Ήδη είναι σε φάση ολοκλήρωσης μια διπλωματική που χρησιμοποιεί την SPMF https://www.philippe-fournier-viger.com/spmf/index.php?link=algorithms.php (section "Time series mining", mining algos) για την αναπαράσταση χρονοσειρών και θα έχει ολοκληρωθεί μέχρι Οκτώβρη.

Στα θέματα των διπλωματικών με χρονοσειρές θα έχετε και υποστήριξη από τον Υποψήφιο Διδάκτορα Βασίλειο Σταματόπουλο.

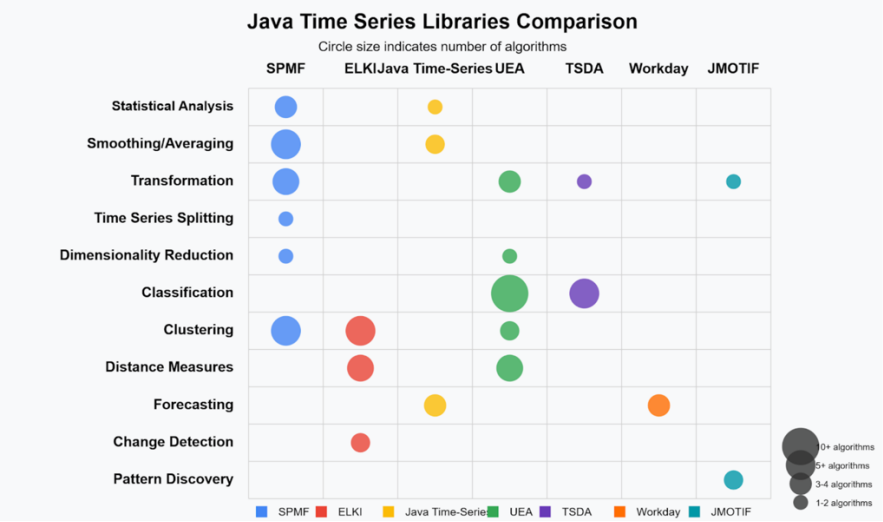
2.1 Επέκταση αλγορίθμων διαχείρισης Χρονοσειρών

ΠΕΡΙΛΗΨΗ: Επέκταση και βελτίωση αλγορίθμων διαχείρισης χρονοσειρών σε υπάρχουσα βιβλιοθήκη, με στόχο την ολοκληρωτική υποστήριξη σύγχρονων αναγκών στην ανάλυση, εξόρυξη και οπτικοποίηση χρονοσειρών.

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Υπάρχει ήδη μια εκτενής βιβλιοθήκη αλγορίθμων διαχείρισης και εξόρυξης χρονοσειρών, η SPMF. Η βιβλιοθήκη περιλαμβάνει μια σειρά αλγορίθμων για στατιστική ανάλυση, smoothing και μετασχηματισμών, μείωσης διαστάσεων κ.α. Αλλά παραμένουν κενά ή δυνατότητες για επέκταση, με τεχνικές που εμφανίστηκαν πιο πρόσφατα στη βιβλιογραφία ή που προκύπτουν από σύγχρονες πρακτικές ανάγκες.

ΣΤΟΧΟΙ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ: Η **εργασία θα επικεντρωθεί στη βελτίωση και αξιολόγηση νέων αλγορίθμων για την ανάλυση χρονοσειρών, τόσο στη διαχείριση όσο και στην εξόρυξη προτύπων (pattern mining), clustering, segmentation, ανίχνευση αλλαγών (change detection), motif discovery κ.α.** Η τελική υλοποίηση θα πρέπει να είναι πλήρως ενσωματωμένη με την υπάρχουσα βιβλιοθήκη, τεκμηριωμένη και να συνοδεύεται με συγκριτική αξιολόγηση με γειτονικές βιβλιοθήκες υλοποίησης μεθόδων ανάλυσης σε χρονοσειρές.

Στην παρακάτω εικόνα δείχνουμε πως συγκρίνονται Java βιβλιοθήκες για ανάλυση χρονοσειρών.



Μέχρι στιγμής, έχουν προστεθεί στην SPMF τεχνικές για μέτρηση αποστάσεων, π.χ. Manhattan, cosine, DTW, Euclidean κ.α.

Ενδεικτικές προτάσεις για επέκταση:

- Προσθήκη του αλγορίθμου Matrix Profile / STUMPY για motif mining. (https://www.cs.ucr.edu/~eamonn/MatrixProfile.html)
- Προσθήκη αλγορίθμων για shapelet extraction (https://www.cs.ucr.edu/~eamonn/shaplet.pdf)
- Υποστήριξη σύγχρονων μεθόδων για dimensionality reduction όπως το UMAP και το t-SNE.
- Ενσωμάτωση εργαλείων για forecasting χρονοσειρών με advanced μοντέλα (ARIMA, Prophet, state space models).

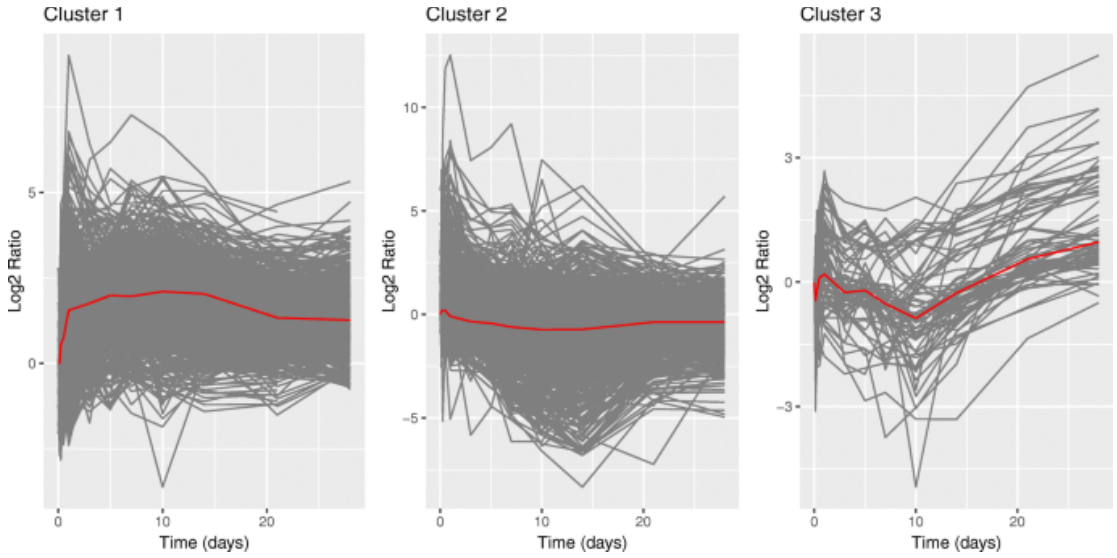
ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java, επέκταση στην υπάρχουσα βιβλιοθήκη (SPMF)

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η κύρια πρόκληση είναι η εις βάθος κατανόηση των αλγορίθμων διαχείρισης χρονοσειρών και η σωστή υλοποίηση/ένταξή τους σε ένα ευμέγεθες λογισμικό. Τα οφέλη για τον φοιτητή είναι: (α) εξοικείωση με την έρευνα σε ένα αναδυόμενο τεχνικό πεδίο (time series data mining), (β) hands-on εμπειρία στην ανάπτυξη και αξιολόγηση αλγορίθμων, και (γ) συνεισφορά σε ένα ανοικτό λογισμικό που χρησιμοποιείται και από άλλους ερευνητές. Απαιτούμενα προσόντα: Πολύ καλή γνώση Java, ικανότητα κατανόησης αλγορίθμων και κώδικα, καλή γνώση αγγλικών (για μελέτη/τεκμηρίωση).

2.2 Timeseries Clustering, Visualization and Storytelling

Έχουμε πολλά δεδομένα τύπου χρονοσειρών και θέλουμε να μπορέσουμε να βρούμε σε αυτά μια σειρά από χρονοσειρές οι οποίες τα περιγράφουν. Θέλουμε δηλαδή να τα οργανώσουμε σε συστάδες (ιδεατά και σε δέντρα αποφάσεων) και να μπορούμε για κάθε μια ομάδα να δώσουμε μια χρονοσειρά που επαρκώς περιγράφει τις χρονοσειρές της. **Θα χρειαστούμε δηλαδή ένα εργαλείο που θα μας επιτρέψει (α) να κάνουμε μαζικά register μεγάλους αριθμούς από timeseries (possibly in a different file each) (β) να κάνουμε τις σχετικές εργασίες συσταδοποίησης και σύγκρισης, (γ) να οπτικοποιήσουμε το αποτέλεσμα και (δ) να βγάλουμε αναφορές που, με αυτόματα παραγόμενο κείμενο και εικόνες, να περιγράφουν τα ευρήματα**.



Η μέθοδος της κλασσικής συσταδοποίησης αποτελεί μια λύση του προβλήματος, αλλά θα θέλαμε σε κάθε περίπτωση και να συνοδεύεται από ένα δείγμα αντιπροσώπων των συστάδων. Μια πρόσφατη σχετική δουλειά (https://www.vldb.org/pvldb/vol18/p915-bao.pdf), δίνει εκπροσώπους. Θα χρησιμοποιήσουμε όσο μπορούμε έτοιμους αλγορίθμους και λογισμικό, και φυσικά η πρόκληση είναι να «κουμπώσουμε» τα έτοιμα πακέτα σε μία ενιαία (και ως συνήθως, συντηρήσιμη) αρχιτεκτονική.

Time Series visualizations, and reports

- https://github.com/jtablesaw/tablesaw
  https://jtablesaw.github.io/tablesaw/userguide/TimeSeries
- https://github.com/binjr/binjr

Τα δεδομένα που θα χρησιμοποιήσουμε είναι από μια μεγάλη μελέτη που έχουμε κάνει στο χώρο του schema evolution, αλλά θα πρέπει να μπορεί το σύστημα να δουλέψει με χρονοσειρές οποιουδήποτε τύπου. Για MSc, θα πρέπει να αποτιμήσουμε και τα ερευνητικά ευρήματα από την εφαρμογή της συσταδοποίησης.

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς / MSc*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java/ Swing / … any other Java viz. library

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία της διπλωματικής έγκειται στο πρόβλημα του να καταλάβει κανείς εντελώς διαφορετικές τεχνολογίες και να κουμπώσουν όλα σε ένα ενιαίο εργαλείο. Δε θα

πρέπει να υποτιμηθεί η δυσκολία στο κομμάτι του data registration και final reporting. Τα οφέλη για ένα φοιτητή είναι: (α) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού, (β) τεχνογνωσία σε θέματα εξόρυξης δεδομένων και ενσωμάτωσης αλγορίθμων εξόρυξης δεδομένων σε κώδικα, και (γ) εμπλοκή σε ένα τελείως νέο χώρο, αυτόν του data storytelling, που φαίνεται να έχει ιδιαίτερες προοπτικές στο μέλλον.