ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

Εργασία Προσομοίωσης Συστημάτων Λήψης Αποφάσεων σε Octave

ΣΥΣΤΗΜΑΤΑ ΛΗΨΗΣ ΑΠΟΦΑΣΕΩΝ

Νίκος Γουρνάκης, it22023
Θεόδωρος Νιάρχος, it22074
Θεοφάνης Παναγιώτης Τηληγάδας, it22103

20 Μαΐου,2024

Σημειωσεις

Τα ερωτηματα β και γ εχουν δικα τους αρχεια octave τα οποια αναφέρονται στην εκαστοτε ενοτητα. Υπαρχουν και αλλα αρχεια τα οποια περιεχουν συναρτησεις χρησιμες/απαραιτητες για την εκτελεση των αρχειων β και γ μερους.

Α’ Μέρος

Η εταιρεία που επιλέξαμε για την εργασία εξειδικεύεται στην παραγωγή bluetooth ακουστικών.

Σενάριο λήψης αποφάσεων

Στο συστημα ληψης χρησιμοποιούμε την μεθοδο SMART.
Καλούμαστε να επιλέξουμε την πιο κατάλληλη μπαταρία για τα ακουστικά, η οποία θα προσφέρει βέλτιστη χρησιμότητα βάση τα κριτήρια μας.

Κριτήρια

Το κρητιρια ειναι ως εξης:

1. Χωρητικότητα μπαταρίας
2. Βαρος μπαταριας
3. Μεγεθος μπαταριας
4. Εμβέλεια ζωης μπαταριας
5. Κοστος

Επεξηγηση κριτηριων:

1. Θα θελαμε οι μπαταριες να εχουν την μεγιστη χωρητικοτητα για να μην χρειαζεται να επαναφορτίζονται συχνα.
2. Θα θελαμε οι μπαταριες να ειναι οσο πιο ελαφριες γινεται, επειδη θα μπουν στα ακουστικα και θελουμε να μην βαρύνουν τον χρηστη.
3. Θα θελαμε το μεγεθος της μπαταριας να ειναι οσο πιο μικρο γινεται, ετσι ωστε να χωρεσουμε αλλα ηλεκτρονικα κομματια μεσα στα ακουστικα.
4. Θα θελαμε μεγιστη εμβελεια ζωης της μπαταριας, για αποφυγη αντικαταστασης και ως αποτελεσμα , κοστους.
5. Θα θελαμε ελαχιστο κοστος.

Διαθεσιμες Εναλλακτικες

	Χωρητικότητα (mAh)	Βαρος (g)	Μεγεθος μπαταριας (diameter x height)(mm)	Εμβέλεια ζωης μπαταριας (in years)	Κοστος (€)
Μπαταρια 1	30	30	9.5 x 2.7	6	0.40
Μπαταρια 2	40	50	12.5 x 2	5	0.40
Μπαταρια 3	125	50	20 x 2	3	0.60
Μπαταρια 4	1000	100	24.5 x 7.7	3	1.40

Επίπεδο διοίκησης

Για το συγκεκριμένο πρόβλημα κατάλληλο για την αναφορά και λύση του προβλήματος κρίνεται το επίπεδο του στρατηγικού σχεδιασμού μιας και θέλουμε να σχεδιάσουμε ένα νέο προϊόν.

Λήψη απόφασης

Οι **Συμμετέχοντες** (π.χ. ειδικοί ή τελικοί χρήστες) εκφράζουν την δική τους γνώμη για τα κριτήρια που έχουν ειπωθεί. Οι **Αναλυτές** (εμείς) παρέχουν διαδικασίες υποστήριξης αποφάσεων (decision support). Έπειτα ο **Μεσολαβητής** αποτελεί τον ενδιάμεσο μεταξύ αποφασίζοντα και αναλυτή και διασφαλίζει την αποτελεσματική τους επικοινωνία.Και τέλος ο **Αποφασίζων** είναι ο αρμόδιος να λάβει την τελική απόφαση.

Β’ Μέρος

Ο κώδικας εκτέλεσης του Β μέρους είναι το αρχείο part_b.m

Για την υλοποίηση του σεναρίου χρησιμοποιούμε την μέθοδο **SMART** οπότε πρέπει να υλοποιήσουμε αρχικα αυτην την εξισωση:

$$w_i = \frac{p_i}{\sum_{i=1}^{n} p_i}$$

όπου p_i είναι η εκτίμηση του ειδικού j για το κριτήριο i, και w_i είναι το βάρος του κριτηρίου i για το ειδικό j.

Για να το πετύχουμε αυτό πρώτα πρέπει να παράγουμε τα p.

```
function ps = set_ps(rules,number_of_experts)
    ps = zeros(number_of_experts, rows(rules));

    for i = 1:number_of_experts
        for col = 1:rows(rules)
            extra = 0;

            if rules(col,2) == 1
                extra = randn();
            end

            ps(i,col) = rules(col,1) + (2 * extra);
        end
    end
end
```

όπου number_of_experts = ο αριθμός των ειδικών, και rules είναι μια δομή δεδομένων που φτιάξαμε για να κάνουμε την λύση μας πιο γενική και να εφαρμόζει και σε άλλα σενάρια:

Παράδειγμα της δομής δεδομένων:

```
# Each row is the following
# 1. The weight of the criteria
# 2. If to add randomness/expert opinion
# 3. If higher is better
rules = [ 70 1 1;
          50 1 0;
          40 1 0;
          10 0 1;
          100 0 0;];
```

Κάθε γραμμή είναι ένα κριτήριο , όπου η στήλη 1 είναι η σημαντικότητα που θέτουμε εμείς στο κριτήριο, στήλη 2 ειναι μια boolean τιμή για το άμα θα προστεθεί τυχαιότητα, και 3η στήλη είναι πάλι μια boolean τιμή όπου καθορίζεις την “κατεύθυνση” του κριτηρίου : άμα είναι 1 τότε όσο πιο μεγάλη τιμή έχει το κριτήριο τόσο πιο μεγάλη τιμή θα πάρει, και άμα 0 όσο πιο μικρή τιμή έχει το κριτήριο τόσο πιο μεγάλη τιμή θα πάρει.

Η τυχαιότητα παίζει τον ρόλο της γνώμης του ειδικού, γιατί μπορεί οι ειδικοί να συμφωνούν γενικά σε κάποια θέματα, αλλά έχουν μικρές διαφορές στην γνώμες τους.

Ύστερα για να υπολογίσουμε τα w:

```
    # Calculate the weights for each criteria, w_i
    for i = 1:number_of_experts
        for col = 1:columns(data)
            weights(i,col) = p(i,col) / sum(p(i,:));
        end
    end
```

Τώρα για να υπολογίσουμε χρησιμότητα της κάθε εναλλακτικής έχουμε το εξής

$$u_j = \sum_{i=1}^{n} w'_i s'_{ij}$$

Πρέπει να υπολογίσουμε τα S_{ij} , για να το πετύχουμε αυτό έχουμε φτιάξει την συνάρτηση generate_experts_opinions

```
function experts = generate_experts_opinions(data,rules, number_of_experts, expert_opinion_max=10, expert_opinion_min=-10)
```

data είναι ένας πίνακας ως εξής:

```
data = [ 30 30 9.5*2.7 6 0.4;
         40 50 12.5*2 5 0.4;
         125 50 20*2 3 0.6;
         1000 100 24.5*7.7 3 1.4];
```

Όπου σειρές = εναλλακτικές , και στήλες = κριτήρια.
Άρα αυτός ο πίνακας είναι ο ίδιος πίνακας από το [ερωτημα Α]

Τα rules είναι η δομή δεδομένων [που ειδαμε πριν].

number_of_experts = ο αριθμός των ειδικών.

expert_opinion_max και expert_opinion_min είναι η μέγιστη και η ελάχιστη τιμές , ανάμεσα από τις οποίες, επιτρέπεται να διακυμανθεί η γνώμη των ειδικών, με κανονική κατανομή. Δηλαδή για expert_opinion_max=10, expert_opinion_min=-10 =>

Ας ξεκινήσουμε με το περιεχόμενο της συνάρτησης:

Πρώτα φτιάχνουμε μια άλλη δομή δεδομένων, που είναι πολύ απλή

Είναι ένας πίνακας διαστάσεων (2,i) όπου i ο αριθμός κριτηρίων, όπου η πρώτη σειρά μας δείχνει τις ελάχιστες τιμές για το κάθε κριτήριο και η δεύτερη σειρά μας δείχνει τις μέγιστες τιμές για το κάθε κριτήριο.

Τρέχοντας την συνάρτηση για τον [πινακα δεδομενων] μας έχουμε το εξής:

```
ans =

   3.0000e+01   3.0000e+01   2.5000e+01   3.0000e+00   4.0000e-01
   1.0000e+03   1.0000e+02   1.8865e+02   6.0000e+00   1.4000e+00
```

Δηλαδή για το κριτήριο ένα η ελάχιστη τιμή είναι 30 και η μέγιστη είναι 1000, για το κριτήριο δύο η ελάχιστη είναι 30 και η μέγιστη 100, και ούτω καθεξής.

Οπότε για κάθε cell του [πινακα] , κάθε ειδικός βάζει μια τιμή. Για κάθε cell ελέγχουμε πρώτα τον [πινακα rules], άμα η "κατεύθυνση" του κριτηρίου είναι "μεγαλύτερο = καλύτερο" ή "μικρότερο = καλύτερο". Ελέγχουμε άμα το cell είναι η μέγιστη τιμή ή ελάχιστη τιμή του κριτηρίου για να του αναθέσουμε 10 ή 100 ανάλογα. Άμα δεν είναι τότε βάζουμε , αναλόγως την "κατεύθυνση", την τιμή του cell αναλόγως την απόσταση του απο τα άκρα των τιμών. Δηλαδή άμα έχουμε μέγιστη τιμή κριτήριο 100 και ελάχιστη τιμή κριτηρίου 0 και έχουμε μια εναλλακτική όπου στο κριτήριο έχει τιμή 50 τότε το S_{ij} του θα είναι 55 (γιατί (10 + 100) / 2 = 55). Τέλος, εισάγουμε την γνώμη του ειδικού, τυχαία [βαση την κατανομη].

Και έτσι έχουμε τα S_{ij} για όλους του ειδικούς.

Άρα υπολογίζουμε τα W'_i

```
# Calculate the average weight for each criteria, w_i'
weight_avgs = zeros(1, columns(data));
for col = 1:columns(data)
    weight_avgs(col) = sum(weights(:,col)) / number_of_experts;
end
```

Και τα S'_{ij}

```
# Calculate the average expert opinion for each criteria, s_ij'
experts_avgs = zeros(rows(data), columns(data));
for row = 1:rows(data)
    for col = 1:columns(data)
        experts_avgs(row,col) = sum(experts(row,col,:)) / number_of_experts;
    end
end
```

Και εφόσον τα έχουμε όλα μπορούμε να υπολογίσουμε τα u_j

```
# Calculate the utility for each alternative, u_j
u = zeros(rows(data), 1);
for row = 1:rows(data)
    u(row) = sum(experts_avgs(row,:) .* weight_avgs);
end
```

Όσο για το σενάριο όπου ένας ειδικός δεν θα μας δώσει αξιολόγηση για ένα κριτήριο ή στοιχείο απλά θα βάζαμε την τιμή βάση το πινακα rules.

Γ' μερος

Ο κώδικας εκτέλεσης του Γ μέρους είναι το αρχείο sensitivity_analysis.m

Για να βάλουμε τα perturbations στο utilities αποφασίσαμε να επιλέξουμε ένα κριτήριο και να προσθέσουμε το perturbation στο w αυτό το κριτηρίου για όλους τους ειδικούς.

Δηλαδή:

```
weights =

   0.259754   0.186785   0.148434   0.036821   0.368207
   0.259073   0.187250   0.143498   0.037289   0.372891
   0.261032   0.183128   0.145298   0.037322   0.373220
   0.268641   0.183691   0.139315   0.037123   0.371230
   0.246265   0.189424   0.149659   0.037696   0.376957
   0.256648   0.189757   0.142790   0.037346   0.373460
   0.259652   0.188277   0.138354   0.037611   0.376106
   0.263095   0.182959   0.141495   0.037496   0.374956
   0.248913   0.190330   0.148623   0.037467   0.374668
   0.255591   0.181730   0.145110   0.037961   0.379608
   0.257541   0.183486   0.156632   0.036576   0.365765
   0.262823   0.190459   0.143576   0.036649   0.366493
   0.250186   0.189478   0.149912   0.037311   0.373113
   0.260677   0.167962   0.159210   0.037468   0.374683
   0.259504   0.176080   0.161444   0.036634   0.366337
```

Αυτά είναι τα **w πριν την προσθήκη των perturbations**, όπου κάθε σειρά είναι ένας ειδικός και κάθε στήλη είναι ένα κριτήριο.

Διαλέξαμε το κριτήριο κόστος (5η στήλη) καθώς για το σενάριο μας έχουμε κρίνει οτι το κόστος είναι το πιο σημαντικό κριτήριο βάση αυτο.

Άρα προσθέτοντας το s στα παραπάνω βάρη έχουμε:

```
weights =

   0.241343   0.168374   0.130023   0.018410   0.441849
   0.240428   0.168605   0.124853   0.018645   0.447469
   0.242371   0.164467   0.126637   0.018661   0.447864
   0.250080   0.165129   0.120754   0.018561   0.445475
   0.227417   0.170576   0.130811   0.018848   0.452348
   0.237975   0.171084   0.124117   0.018673   0.448152
   0.240847   0.169472   0.119548   0.018805   0.451327
   0.244347   0.164211   0.122747   0.018748   0.449948
   0.230180   0.171597   0.129889   0.018733   0.449601
   0.236610   0.162749   0.126130   0.018980   0.455530
   0.239252   0.165198   0.138344   0.018288   0.438918
   0.244499   0.172134   0.125251   0.018325   0.439792
   0.231531   0.170822   0.131256   0.018656   0.447735
   0.241943   0.149228   0.140476   0.018734   0.449620
   0.241188   0.157763   0.143127   0.018317   0.439605
```

Για να εξηγήσουμε τι γίνεται μεταξύ πριν και μετά ας πάρουμε παράδειγμα τα βάρη για τον ειδικό ένα.

Πριν s

```
0.259754   0.186785   0.148434   0.036821   0.368207
```

Μετά s

```
0.241343   0.168374   0.130023   0.018410   0.441849
```

Τι κάνουμε είναι ως εξής:

```
petrubation = s * weights(i,col_of_interest);
```

το perturbation ειναι s * w_5 (5 η στήλη που κρίναμε πριν οτι θα χρησιμοποιήσουμε για την ανάλυση ευαισθησίας)

και προσθέτουμε αυτό το perturbation στο w_5

```
weights(i,col_of_interest) = weights(i,col_of_interest) + petrubation;
```

Τώρα για να ισούται το άθροισμα τον w με 1 πρέπει να αφαιρέσουμε αυτο το perturbation / νούμερο κριτηρίων - 1 (επειδή πρέπει να το αφαιρέσουμε ίσα από όλες τις στήλες μείον την 5 που είπαμε ότι κάνουμε την ανάλυση ευαισθησίας)

```
for col = 1:columns(data)
    if col ≠ col_of_interest
        weights(i,col) = weights(i,col) - petrubation / (columns(data) - 1)
    end
end
```

Για τον ειδικό ένα έχουμε οτι:

```
Petrubation 1: 0.073641
Pertrubation / (columns(data) - 1) [4]: 0.018410
```

Άρα:

```
0.259754    0.186785    0.148434    0.036821    0.368207
```

0.368207 + 0.073641 = 0.441849 που είναι ακριβώς ο αριθμός στην στήλη 5 από πριν

```
0.241343    0.168374    0.130023    0.018410    0.441849
```

και μετά πρέπει να αφαιρέσουμε 0.073641 / 4 = 0.018410 από όλες τις άλλες στήλες, κάνοντας το στην πρώτη:
0.259754 - 0.018410 = 0.241343 που είναι ακριβώς ο αριθμός στην στήλη 1

Κάνοντας αυτήν την διαδικασία για όλους τους ειδικούς, για τα perturbations strengths που μας δώσατε, 10^4 φορές έχουμε την ανάλυση ευαισθησίας. Τώρα, για να υπολογίσουμε την πιθανότητα αναστροφής κατάταξης έχουμε το εξής:

πρώτα βρίσκουμε τα utilities χωρίς τα perturbations και βλέπουμε την φθίνουσα σειρά των εναλλακτικών βάση utility, μετά τρέχουμε την προσομοίωση monte carlo και κάθε επανάληψη υπολογίζουμε νεα utilities + perturbation όπως είπαμε πριν. Άμα η σειρά αυτών των utilities είναι διαφορετική από την αρχική τότε προσθέτουμε ένα στο counter τις πιθανότητας αναστροφής κατάταξης. Στο τέλος τις προσομοίωσης , για κάθε perturbation strength διαιρούμε με το 10^4 (αριθμο επαναλήψεων) και έτσι έχουμε την πιθανότητα αναστροφής κατάταξης για perturbation s_i

Τέλος το εμφανίζουμε σε ένα γράφημα για να δούμε την σχέση μεταξύ s και πιθανότητας αναστροφής κατάταξης:

```
plot(Prrs,0.2:0.1:0.6);
ylabel("s");
xlabel("Prr");
title("Prr vs s");
input("Press ENTER to exit");
```

```
Prrs =

   0   0   0   0   0
```

Στην δικιά μας περίπτωση, η πιθανότητα αναστροφής κατάστασης είναι 0 που σημαίνει ότι τα utilities των εναλλακτικών είναι ανθεκτικά στον θόρυβο.