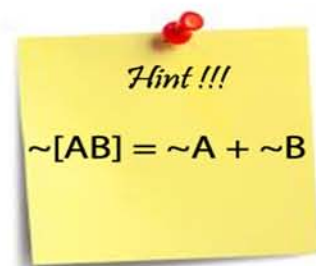




Πολυτεχνείο Κρήτης
Πολυπρακτορικά Συστήματα
ΠΛΗ-517 ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2012



Αναφορά Προγραμ.Εργασίας

06-08-2012

Δεμερτζής Ιωάννης 2008030124

Νικολακάκη Σοφία Μαρία 2008030064

Διδάσκων: Γ. Χαλκιαδάκης

1. ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΣΙΑΣ

Τα τελευταία χρόνια έχει παρατηρηθεί μία έξαρση στη βιομηχανία των διαδικτυακών διαφημίσεων. Στο χώρο αυτό έχει υπολογιστεί ότι υπάρχουν κέρδη δισεκατομμυρίων κάθε χρόνο για τους online εκδότες και πρόκειται για έναν διαδικτυακό τομέα με πολύ μεγάλη απήχηση σε όλον τον κόσμο, αφού απευθύνεται σε κάθε επιχειρηματία που τον ενδιαφέρει να διαφημίσει τα προϊόντα του μέσω του πιο δημοφιλούς επικοινωνιακού μέσο. Σε αυτό το σημείο αξίζει να αναφέρουμε ότι το Google που συντηρείται κατά κύριο λόγο από διαφημίσεις, έβγαλε το 2009 κοντά στα 24 δισεκατομμύρια ευρώ από αυτές. Εμείς στην εργασία αυτή καλούμαστε να παίξουμε το ρόλο του διαφημιστή που έχει ως στόχο να διαφημίζει τα προϊόντα των πελατών του και δεδομένου ότι υπάρχουν ανταγωνιστές, να αποκτήσει το μεγαλύτερο δυνατό κέρδος.

Πιο συγκεκριμένα, υλοποιήσαμε μια προγραμματιστική εργασία με βάση το specification του TAC Ad Auction (TAC/AA), το οποίο παρουσιάζει ένα sponsored – search σενάριο. Το σενάριο αυτό εφαρμόζει ένα αυτοματοποιημένο μηχανισμό δημοπρασίας που προσομοιώνει ένα πλήθος χρηστών που αναζητούν κάποιο προϊόν σε μια αγορά λιανικής. Τελικός σκοπός είναι η παραγωγή ιδεών και στρατηγικών ώστε να γίνουν οι κατάλληλες προσφορές επάνω σε διαφημίσεις, η υλοποίηση των στρατηγικών αυτών σε κώδικα και στη συνέχεια η ανάδειξη του πράκτορα – διαφημιστή ο οποίος μέσα σε ένα ανταγωνιστικό περιβάλλον έχει τα μεγαλύτερα κέρδη.

Περισσότερα για τις προδιαγραφές του παιχνιδιού υπάρχουν στην επίσημη ιστοσελίδα του TAC Ad Auction. Εν συντομία αναφέρονται τα πιο σημαντικά στοιχεία τα οποία διακρίνουν το συγκεκριμένο μηχανισμό δημοπρασίας.

Πράκτορες παιχνιδιού

Αρχικά υπάρχουν **τριών ειδών πράκτορες** στο παιχνίδι, οι διαφημιστές, οι εκδότες και οι χρήστες. Η αλληλεπίδραση των πρακτόρων είναι η εξής: Κάθε μέρα της προσομοίωσης ο χρήστης θέτει ένα query, ο υπεύθυνος της δημοπρασίας που είναι ο publisher τρέχει τη δημοπρασία και τα αποτελέσματα της θέσης που έχουν πάρει οι διαφημίσεις επιστρέφουν στο χρήστη σε μορφή ταξινομημένης λίστας από διαφημίσεις. Το αποτέλεσμα των διαφημίσεων που θα δει ο χρήστης και η σειρά των διαφημίσεων λέγεται **impressions**. Ο χρήστης βλέπει το impression και αποφασίζει αν θα κάνει κλικ επάνω σε κάποια διαφήμιση ή όχι. Αν ναι, τότε αυτό σημαίνει ότι ο χρήστης έχει λάβει υπόψη του τη διαφήμιση. Ο διαφημιστής, που είμαστε εμείς χρεώνεται κάποιο κόστος ανά κλικ που καθορίζεται από τον εκδότη κατά την εξέλιξη της δημοπρασίας. Όταν ο χρήστης κάνει κλικ επάνω στη διαφήμιση του διαφημιστή, τότε καθορίζεται και το αν θα αγοράσει τελικά το προϊόν από το διαφημιστή. Αν όντως αγοράσει ο χρήστης τότε έχουμε αγορά που ορίζεται ως **conversion**. Τότε, ο διαφημιστής έχει κέρδος.

Προϊόντα της αγοράς

Οι χρήστες στο συγκεκριμένο παιχνίδι ενδιαφέρονται για την αγορά προϊόντων οικιακής ψυχαγωγίας. Τα προϊόντα αυτά διακρίνονται από τον κατασκευαστή και το ρόλο τους δηλαδή από το ζεύγος (**κατασκευαστής, εξάρτημα**). Έχουμε τρεις κατασκευαστές και τρεις μορφές εξαρτημάτων. Οι κατασκευαστές είναι οι Flat, Lioneer, PG ενώ τα προϊόντα μπορούν να είναι DVD player, TV και συστήμα ήχου. Το ζευγάρι (κατασκευαστής, εξάρτημα) επομένως μπορεί να λάβει 9 διακριτές τιμές καθώς καθένας από τους κατασκευαστές παράγει καθένα από τα εξαρτήματα.

Ερωτήσεις

Οι χρήστες χωρίζονται σε τρεις ευρείς κατηγορίες, στους **non searching, searching, transacted**. Εκείνοι που είναι non searching είναι αδρανής και δεν κάνουν ερωτήσεις καθώς δεν ψάχνουν για τίποτα. Οι χρήστες που είναι transacted έχουν αγοράσει κάποιο προϊόν και δεν συνεχίζουν επομένως την αναζήτηση τους. Εκείνοι που είναι searching χωρίζονται σε δικιά τους υποκατηγορία στην οποία υπάρχουν οι **informational** και **shopping searchers**. Οι πρώτοι απλώς συλλέγουν πληροφορίες για το επιθυμητό τους προϊόν αλλά δεν τους ενδιαφέρει η αγορά, ενώ οι δεύτεροι κοιτάνε διαφημίσεις με τελικό σκοπό πιθανόν να αγοράσουν. Οι shopping searchers διαμοιράζονται περεταίρω ανάλογα με το επίπεδο του ενδιαφέροντος τους το οποίο μπορεί να είναι **low focus (F0)**, **intermediate focus(F1)** ή **high focus(F2)**. Κάθε ερώτηση μπορεί να περιέχει το πολύ δύο λέξεις, τον κατασκευαστή και το τύπο του προϊόντος που αναζητά ο χρήστης. Αν ο χρήστης διευκρινίζει και κατασκευαστή και τύπο προϊόντος δείχνει high focus. Στην περίπτωση που διευκρινίζει ή κατασκευαστή ή τύπο προϊόντος δείχνει intermediate focus, ενώ αν δεν διευκρινίζει τίποτα από τα δύο τότε δείχνει low focus. Επομένως μπορούν να υπάρξουν **9 F2 queries, 6 F1 queries** και **1 F0 query**, σύνολο 16 διακριτά queries.

Διαφημιστής

Πρόκειται για τον πράκτορα που υλοποιούμε. Ο ρόλος του διαφημιστή είναι να διαχειρίζεται και να δημιουργεί προσφορές. Οι προσφορές του διαφημιστή διαμορφώνονται από το πόση αξία δίνει στα διάφορα είδη διαφημίσεων, τη θέση που θέλει να έχουν οι διαφημίσεις αυτές σε συγκεκριμένα slots καθώς και από το πώς χειρίζεται το ανταγωνιστικό περιβάλλον στο οποίο βρίσκεται. Παρόλο που κάθε διαφημιστής μπορεί να πουλήσει οποιοδήποτε προϊόν αυτός εξειδικεύεται σε συγκεκριμένο κατασκευαστή και σε συγκριμένο τύπο προϊόντος που ορίζεται στην αρχή κάθε στιγμιότυπου παιχνιδιού. Ο διαφημιστής μπορεί να αυξήσει τη τιμή πώλησης της μονάδας (Unit Sales Profit – USP) και την πιθανότητα για να υπάρξει αγορά σημαντικά αν ο χρήστης αγοράσει το συγκεκριμένο εξειδικευμένο προϊόν του διαφημιστή ή αν κάνει κλικ επάνω στη συγκεκριμένη διαφήμιση αντίστοιχα. Κάθε μέρα και για κάθε κλάση

ερωτήσεων ο πράκτορας μας επιλέγει μία διαφήμιση για να αναδείξει, ενημερώνει τις προσφορές που κάνει και αν το επιθυμεί θέτει όρια για αυτά που θα ξοδέψει.

Διαφημίσεις

Κάθε διαφημιστής επιλέγει το τύπο της διαφήμισης που θα προβάλλει για κάθε τύπο κλάσης query που αναφέραμε προηγουμένως. Μπορεί να είναι είτε **generic ad** είτε **targeted ad**. Αν ο χρήστης προτιμήσει μία targeted ad τότε οι πιθανότητες ότι θα κάνει κλικ επάνω στη διαφήμιση αυτή αυξάνονται σε σχέση με το αν η διαφήμιση είναι απλώς generic. Αντίστοιχα αν δεν τον ενδιαφέρει κάποια διαφήμιση τότε οι πιθανότητες να κάνει κλικ επάνω στη διαφήμιση ελαττώνονται. Ο διαφημιστής επιλέγει το τύπο της διαφήμισης που θα φαίνεται, το ποσό που θα κάνει προσφορά στην αρχή κάθε μέρας και αυτά για όλες τις κλάσεις ερωτήσεων . Με βάση τα δεδομένα αυτά ο εκδότης χειρίζεται τις ερωτήσεις τρέχοντας τη δημοπρασία.

Προσφορές

Κάθε μέρα ο διαφημιστής θέτει προσφορές. Τύποι τέτοιας προσφοράς είναι **(query, bid)** και **(query, ad)**. Επομένως κάθε μέρα ο εκδότης ανάλογα με αυτά που του έχει στείλει ο διαφημιστής ενημερώνει τις συγκεκριμένες προσφορές. Αν το bid είναι μηδέν, τότε θεωρείται ότι δεν γίνεται καθόλου προσφορά. Σε περίπτωση που δεν θέτει καινούρια προσφορά για το ζευγάρι (query, bid) τότε μένουν οι τιμές της προηγούμενης μέρας. Ομοίως και για το ζευγάρι (query, ad). Ακόμη μπορεί ο διαφημιστής να θέσει κάποιο όριο από το ποσό των χρημάτων που μπορεί να ξοδέψει καθημερινά. Σε περίπτωση που το κάνει αυτό τότε αν φτάσει τα όρια αυτά οι διαφημίσεις που επηρεάζονται από αυτό δεν φαίνονται στους χρήστες για το υπόλοιπο της ημέρας.

Αναφορά για κάθε Ερώτηση

Καθημερινά ο διαφημιστής δέχεται αναφορά από τον εκδότη που περιλαμβάνει στατιστικά για κάθε ερώτηση. Αυτά τα στατιστικά είναι και η φανερή πληροφορία με βάση τα οποία καθορίζει ο διαφημιστής τη τακτική που θα ακολουθήσει. Τέτοια στατιστικά είναι οι τύποι των διαφημίσεων που εμφανίζονται από τον κάθε διαφημιστή για τα queries, η θέση που βρίσκονται οι διαφημιστές στην κατάταξη για κάθε ένα από τα 16 queries, το μέσο κόστος ανά κλικ που χρεώνεται ο διαφημιστής στη δημοπρασία, ο αριθμός των χρηστών έκαναν impression επάνω σε διαφήμιση του διαφημιστή, ο αριθμός των χρηστών που κάνανε κλικ επάνω σε διαφήμιση.

2. ΚΑΘΟΡΙΣΤΙΚΟΙ ΠΑΡΑΓΟΝΤΕΣ ΓΙΑ ΤΗ ΤΑΚΤΙΚΗ ΤΟΥ ΠΡΑΚΤΟΡΑ

Παραθέτουμε παράγοντες που θεωρούμε ότι παίζουν σημαντικό ρόλο στην τακτική ενός πράκτορα καθώς και πώς τους αντιμετωπίσαμε.

Bid : Πρόκειται για τη μέγιστη τιμή που προτίθεται να πληρώσει ο πράκτορας μας για κάθε κλικ επάνω σε κάθε ερώτηση ξεχωριστά. Πρόκειται για την ουσία του παιχνιδιού καθώς με βάση την προσφορά του πράκτορα καθορίζεται η τακτική και η απόδοση του στο παιχνίδι. Ο τρόπος με τον οποίο καθορίζεται η προσφορά που κάνει ο πράκτορας μας είναι με βάση το **Focus Level** του query, την **ημέρα** που βρισκόμαστε και αν έχουμε ξεπεράσει κάποιο **spending limit**.

Ad Type : Πρόκειται για το είδος της διαφήμισης, αν αυτή δηλαδή είναι targeted (για παράδειγμα ότι πουλάμε τηλεόραση της Lioneer) ή αν είναι generic (για παράδειγμα ότι πουλάμε τηλεοράσεις). Σε περίπτωση conversion, αν πωληθεί ένα προϊόν που δεν είναι η ειδικότητα του διαφημιστή τότε αυτός λαμβάνει ένα standard unit sales profit, *USP*. Ωστόσο, αν το προϊόν αντιστοιχεί στην εξειδίκευση του διαφημιστή τότε οι απολαβές του αλλάζουν και είναι ίσες με $USP(1 + MSP)$, όπου *MSP* είναι το manufacturer specialist bonus. Επομένως, έπρεπε να λάβουμε υπόψη μας το **extra bonus** που μπορούσαμε να κερδίσουμε και να επενδύσουμε περισσότερο σε **targeted ads**.

Per query budget : Ο προϋπολογισμός αυτός, καθορίζει το συνολικό ποσό που προτίθεται να ξοδέψει ο διαφημιστής επάνω σε ένα συγκεκριμένο query μία συγκεκριμένη ημέρα. Αυτό αρχικά μας φαινόταν καλή ιδέα καθώς θα μπορούσαμε να έχουμε ένα πολύ χαμηλό budget για τα queries με focus level 0 (ώστε να μπορούσαμε να αφιερώσουμε το μεγαλύτερο budget στα F1 και F2). Το δοκιμάσαμε στις αρχές της υλοποίησης του κώδικα αλλά προσέξαμε ότι είχε χειρότερη απόδοση σε σχέση με το over – all budget, καθώς για το δεύτερο βλέπαμε κάθε στιγμή στο Daily Revenue and Cost diagram πόσα ξόδευε σύνολο ανά πάσα στιγμή. Έτσι **εμπειρικά** εκτιμούσαμε και αλλάξαμε το **over – all budget** πράγμα που δεν ήταν το ίδιο εύκολο για το per query budget.

Over – all budget : Ο προϋπολογισμός αυτός, βάζει όρια στο συνολικό ποσό που ο διαφημιστής προτίθεται να ξοδέψει για όλα τα queries μία συγκεκριμένη μέρα. Στις αρχές αφού είχαμε υλοποιήσει τον αλγόριθμο μας προσέξαμε ότι ο πράκτορας μπορεί να ξόδευε πολύ μεγάλα ποσά σε μια μέρα πράγμα που δεν έδινε κέρδος αλλά μόνο κόστος και επομένως υπήρχε πτώση στην απόδοση του πράκτορα. Στόχος μας ήταν ο πράκτορας ποτέ να μην είχε πτώση απόδοσης (δηλαδή αρνητικό profit) και να έμενε τουλάχιστο σταθερός σε αυτό. Επομένως ήταν απαραίτητο να θέσουμε **όρια** στο **συνολικό ποσό** που θα ξόδευε κάθε μέρα ο

πράκτορας επάνω στα queries. Σημαντικό ήταν και το over – all budget στην περίπτωση που ο διαφημιστής είχε τύπο προϊόντος αντίστοιχο με αυτό που αναζητάει ο χρήστης, καθώς αυτό σήμαινε ότι υπήρχε μεγαλύτερη πιθανότητα να κάνει κλικ ο χρήστης επάνω στη διαφήμιση. Αυτό έπρεπε να ληφθεί υπόψη διότι πριν τον ορισμό ενός over – all budget οι χρήστες κάνανε **κλικ** επάνω στη διαφήμιση, **χωρίς** να κάνουνε **conversion** με αποτέλεσμα να έχουμε σημαντικό έλλειμμα και κακή επίδοση. Τέλος θεωρήσαμε ότι το over– all budget εξαρτιόταν και από το **capacity** που οριζόταν στον πράκτορα μας στην αρχή του παιχνιδιού.

Capacity limit : Πέρα από την εξειδίκευση του διαφημιστή επάνω στον κατασκευαστή και το τύπο προϊόντος λάβαμε υπόψη και όριο των αποθεμάτων που έχει (capacity limit). Πρόκειται για το άνω όριο του αριθμού των πωλήσεων που μπορούν να γίνουν για 5 μέρες, που μπορούν να γίνουν χωρίς κάποιο penalty. Το capacity που έχει ο κάθε διαφημιστής ανανεώνεται κάθε 5 μέρες, ενώ αν οι πωλήσεις υπερβούν το όριο αυτό τότε οι πιθανότητες για αγορά από επόμενους χρήστες που κάνουν κλικ επάνω στη διαφήμιση, πέφτουν. Επομένως έπρεπε να λαμβάνουμε υπόψη το γεγονός αυτό καθώς μπορεί να καταλήγαμε να **πληρώνουμε** για κλικ επάνω σε διαφημίσεις που πιθανόν να μη μας έφερναν **καθόλου κέρδος**. Σε όλα τα paper που διαβάσαμε προσέξαμε ότι το capacity ήταν ο **κύριος παράγοντας** του TAC Ad Auction με βάση το οποίο ο πράκτορας έπρεπε να σκέφτεται το πώς θα συνέδεε το πέρασμα των ημερών με τις προσφορές που ήταν σοφό να κάνει για τα queries.

3. ΑΛΓΟΡΙΘΜΟΣ EquateROI

Επιλέξαμε να αφιερώσουμε μία ενότητα στην περιγραφή του αλγορίθμου που πήραμε από paper [1] διότι η εφαρμογή του μας δυσκόλεψε ιδιαίτερα. Ο αλγόριθμος όπως αναφέραμε και προηγουμένως ήταν αρκετά κατανοητός και κατατοπιστικός ωστόσο όταν ερχόταν το σημείο να χρησιμοποιήσουμε παραμέτρους που ήδη έπρεπε να έχουμε υπολογίσει και που εφαρμόζονται στον αλγόριθμο (αλλά δεν υπήρχαν πληροφορίες για αυτές στο άρθρο), τότε ανατρέχαμε για την εύρεση τους σε άλλα άρθρα με reference στο συγκεκριμένο άρθρο και υλοποιήσεις του αλγορίθμου.

Στόχος του πράκτορα όπως αναφέραμε και προηγουμένως είναι να διαλέξει ανά query την προσφορά που μεγιστοποιεί τα κέρδη του χωρίς να υπερβαίνει το capacity. Αξίζει να

σημειώσουμε σε αυτό το σημείο ότι το κέρδος από το να κάνει ένα bid $b_q \in \mathbf{R}$ ο χρήστης επάνω σε ένα query $q \in \mathbf{Q}$ είναι $rev_q(b_q) - cost_q(b_q)$, όπου $rev_q(b_q)$ είναι η εισπραξη που κέρδισε για το query q δεδομένου ότι έκανε bid b_q και $cost_q(b_q)$ είναι το κόστος που πλήρωσε για το query q δεδομένου ότι έκανε bid b_q . Ο αντίστοιχος αριθμός των πωλήσεων κατά $sales_q(b_q)$ διαμορφώνουν το πρόβλημα ως εξής:

$$\begin{aligned} \max_{b \in \mathbb{R}_+^{|\mathbf{Q}|}} \sum_{q \in \mathbf{Q}} rev_q(b_q) - cost_q(b_q) \quad \text{s.t.} \\ \sum_{q \in \mathbf{Q}} sales_q(b_q) \leq C \end{aligned}$$

,δηλαδή θέλουμε το **μέγιστο κέρδος** που μπορεί να αποφέρει ένα bid για ένα συγκεκριμένο query, δεδομένου ότι οι πωλήσεις μας για τα παραπάνω δεν ξεπερνάνε το capacity. Το συγκεκριμένο πρόβλημα πρόκειται για στιγμιότυπο του nonlinear knap-sack problem (NLK).

Επειδή το capacity είναι περιορισμένος πόρος στο παιχνίδι μας ορίζουμε τον όρο return of investment (**ROI**) που έχει την παρακάτω τιμή:

$$ROI_q(b_q) = \frac{rev_q(b_q) - cost_q(b_q)}{sales_q(b_q)} = \frac{profit_q(b_q)}{sales_q(b_q)}$$

, δηλαδή πρόκειται για μία παράμετρο που εκτιμάει το πόσο κέρδος μας δίνει η πώληση ενός αντικειμένου επάνω σε ένα query q και δεδομένου ενός bid b_q (επιστροφή της επένδυσης που κάναμε).

Ο EquateROI επαναληπτικά προσεγγίζει μία λύση για ένα σύστημα εξισώσεων σαν το παρακάτω (πάντα μόνο για 3 queries).

$$ROI_q(b_q) = ROI_{q'}(b_{q'})$$

$$ROI_q(b_q) = ROI_{q''}(b_{q''})$$

$$sales_q(b_q) + sales_{q'}(b_{q'}) + sales_{q''}(b_{q''}) = C$$

Διαισθητικά καταλαβαίνουμε ότι τα ROIs σε ερωτήματα θέλουμε να είναι όλα ίσα με κάποιο *targetROI* και ότι το άθροισμα των πωλήσεων θα είναι ίσο με *capacity*. Η επίλυση αυτού του συστήματος με ακρίβεια προϋποθέτει την εκτίμηση των εσόδων, του κόστους και της συνάρτησης των πωλήσεων, που δυσκολεύουν αρκετά την κατάσταση. Ωστόσο, ο αλγόριθμος του EquateROI και ο τρόπος που κάνει bid προσεγγίζει αυτούς τους υπολογισμούς προσεγγιστικά. Αυξάνει τη τιμή του *targetROI* αν οι πωλήσεις της χθεσινής μέρας υπερέβη το *capacity* και το μειώνει διαφορετικά. Στη συνέχεια οι προσφορές επιλέγονται υπολογίζοντας πρώτα το CPC (cost per click) που αντιστοιχίζεται στο τρέχον *targetROI* και στη συνέχεια αυτό το CPC αντιστοιχίζεται σε bid. Επομένως μία σχέση που συνδέει όλα τα παραπάνω :

$$ROI_q(b_q) = \frac{rev_q(b_q) - cost_q(b_q)}{sales_q(b_q)} = USP_q - \frac{CPC_q}{PrConv_q}$$

Λύνοντας ως προς CPC_q έχουμε λοιπόν :

$$CPC_q = (USP_q - ROI_q(b_q))PrConv_q$$

Άρα για ένα επιθυμητό *targetROI* θα έχουμε:

$$CPC_q = (USP_q - targetROI_q(b_q))PrConv_q$$

Ο ψευδοκώδικας του EquateROI φαίνεται παρακάτω όπου τα bids της ημέρας $d+1$ ορίζονται βάση του *targetROI* την ημέρα d και των παρατηρούμενων πωλήσεων εκείνης της μέρας. Θα εξηγήσουμε τον αλγόριθμο και θα αναφέρουμε παρακάτω ποιες αλλαγές κάναμε επάνω σε αυτόν με βάση άλλα papers, πληροφορίες στο internet και τη λογική μας.

Algorithm 1 EquateROI

Input: $sales(d)$, $targetROI(d)$, C **Output:** $\forall q, bid_q$ if $sales(d) > C$ then $targetROI(d+1) = targetROI(d) * INC_ROI$ else if $sales(d) < C$ then $targetROI(d+1) = targetROI(d) / INC_ROI$

end if

for all q do $CPC_q = (USP_q - targetROI(d+1)) * PrConv_q$ $bid_q = CPC_q + \epsilon$ end for

Ο αλγόριθμος δέχεται ως **ορίσματα** τις **πωλήσεις** της ημέρας d , το **targetROI** της ημέρας d και το **Capacity** που έχει οριστεί από την αρχή του παιχνιδιού. Στη συνέχεια για καθένα από τα 16 διακριτά queries που υπάρχουν κοιτάμε αν οι πωλήσεις τις ημέρας έχουν ξεπεράσει το Capacity. Αν ναι τότε ελατώνουμε ελάχιστα το **targetROI** της επόμενης μέρας με βάση ένα παράγοντα INC_ROI (θα αναφέρουμε για αυτόν παρακάτω). Διαφορετικά αυξάνουμε το **targetROI** της επόμενης μέρας. Τέλος υπολογίζουμε το εκτιμώμενο Cost per Click και με βάση αυτό το τελικό bid που θα κάνουμε για ένα query, το οποίο θα είναι ίσο με το Cost per Click συν μία σταθερά ϵ .

Ο παραπάνω αλγόριθμος ήταν αποτελεσματικό στο να μην έχει καθόλου αρνητικά profits αλλά έπασχε στο ότι δεν είχε ούτε σημαντικά κέρδη καθώς αυτό που κάνει είναι να υπολογίζει μία μέτρια κατάσταση και να υπολογίζει όλες τις παραμέτρους με βάση αυτή.

Στο paper δεν αναφέρεται καθόλου πώς υπολογίζουμε τις ποσότητες *Initial TargetROI*, INC_ROI , $PrConv_q$ και ϵ . Πρόκειται για τιμές μεγάλης σημασίας για τη ροή του παιχνιδιού και για αυτό δοκιμάσαμε αρκετά πράγματα ενώ παντού αναφερόταν ότι οι παράμετροι αυτοί ήταν η ουσία του αλγορίθμου. Για τις παραπάνω τιμές ανατρέξαμε σε άλλες πηγές και εξηγούμε πώς τις υπολογίσαμε.

Initial TargetROI

Δεν ορίζεται πουθενά στο αρχικό paper που διαβάσαμε, ωστόσο πρόκειται για μια σημαντική παράμετρο καθώς πρόκειται για το αρχικό TargetROI που θα δώσουμε για να συνεχίσει η ροή του παιχνιδιού. Στη συνέχεια κοιτάξαμε ένα άρθρο [6] στο οποίο αναγραφόταν ότι εκείνοι είχαν πάρει αρχικά μία πολύ μικρή τιμή για το Initial TargetROI, για την οποία όμως έχανε λεφτά ο EquateROI και για το λόγο αυτό το αύξαναν τη τιμή αυτή μέχρι να αρχίσει να κερδίζει λεφτά ο αλγόριθμος. Επομένως μέσω δικών τους πειραμάτων καταλήξαν στο ότι μία ιδανική τιμή για το Initial TargetROI ήταν το 8.2. Σε άλλη εργασία η οποία ήταν διπλωματική στο

Αριστοτέλειο Πανεπιστήμιο [3] όρισαν σαν αρχική τιμή του Initial TargetROI το 4, με βάση τις προτάσεις του Schlemazl. Εν γένει καταλάβαμε ότι η τιμή του Initial TargetROI εξαρτιόταν σε μεγάλο βαθμό από τον αλγόριθμο του πράκτορα και ήταν εμπειρικός, δηλαδή η τιμή που απέδιδε για μία ομάδα δεν ήταν απαραίτητο ότι θα αποδίδει και για κάποια άλλη. Επομένως, αρχικά δοκιμάσαμε αυτές τις εμπειρικές τιμές αλλά τρέξαμε πολλά παιχνίδια και για άλλες τιμές. Τελικά καταλήξαμε ότι καλύτερη απόδοση και περισσότερα κέρδη είχε ο πράκτορας μας όταν του ορίσαμε το Initial TargetROI ίσο με 8.2 . *Εκείνο που συμπεράναμε και είδαμε ότι είχαν συμπεράνει και άλλοι ήταν ότι όσο μεγαλύτερη απόκλιση είχε το Initial TargetROI από τη τιμή όπου μπορούσαμε να έχουμε θετικό κέρδος τόσο περισσότερο θα αργούσε για να φτάσει σε τιμή κατάλληλη ώστε να έχουμε κέρδος.* Επομένως ήταν σημαντική η σωστή εκτίμηση του στην αρχή του παιχνιδιού για να είχαμε όσο το δυνατό συντομότερο κέρδη.

INC ROI

Το Incremental ROI ορίζει αν θα μειωθεί ή αυξηθεί το ROI και είναι μία εμπειρική τιμή για την οποία έχουν γίνει πειράματα ώστε να εξακριβωθεί μία κατάλληλη τιμή της. Και πάλι βρήκαμε τιμές που είχανε βρεθεί όπως 1.1 και 1.05. Μάλιστα σε μία εργασία είχανε ορίσει Incremental ROI και Decremental ROI, δηλαδή ο παράγοντας ήταν διαφορετικός αν ήθελε να αυξήσει το TargetROI και διαφορετικός αν ήθελε να το μειώσει. Είδαμε ότι όσο μεγαλύτερη η αύξηση τόσο πιο γρήγορα ανεβαίναμε στη σκάλα του TargetROI με αποτέλεσμα να χάσουμε την ιδανική τιμή που θέλουμε να πετύχουμε του μέγιστου κέρδους δεδομένου συγκεκριμένου Capacity. Επομένως θέλαμε ένα αρκετά μεγάλο increment ώστε να πετύχουμε ουσιώδες κέρδος αλλά ούτε και τόσο μεγάλο ώστε να πετύχουμε αρνητικό κέρδος. *Τα έσοδα και το κόστος αυξάνονται όταν αυξάνεται το increment καθώς αυξάνεται ο αριθμός των impressions των χρηστών. Αν είναι πολύ μεγάλο το increment τότε το κόστος θα αυξάνεται πιο γρήγορα από τα έσοδα, επομένως δεν θέλουμε μεγάλο increment.* Το κατάλληλο threshold που με αρκετές ώρες παιχνιδιού θεωρήσαμε καλό για το παιχνίδι του πράκτορα μας ήταν INC_ROI = 1.2.

PrConv_g

Ο υπολογισμός της πιθανότητας για conversion ήταν ένας πολύ σημαντικός παράγοντας ο οποίος δεν ήταν απαραίτητα εμπειρικός καθώς υπήρχαν συγκεκριμένοι τύποι για αυτό. Στα specifications του παιχνιδιού βρήκαμε έναν τρόπο υπολογισμού για το συγκεκριμένο τύπο ο οποίος είναι :

$$Pr(\text{conversion}) = \begin{cases} \eta(\pi_l I_d, 1 + CSB) & \text{if user matches component specialty} \\ \pi_l I_d & \text{otherwise.} \end{cases}$$

, όπου

$$I_d = \lambda \left(\left(\sum_{i=d}^{d-(W-1)} c_i \right) - C^{cap} \right)^+$$

Επειδή δεν υπολογίσαμε τελικά έτσι την πιθανότητα για conversion απλώς θα αναφέρουμε το τύπο. Περισσότερα στο specification του παιχνιδιού [7]. Θα αναφέρουμε ωστόσο, ότι σε εμάς δεν δούλεψε αρκετά καλά αυτή η πιθανότητα. Καταλάβαμε ότι το πρόβλημα εν μέρει ήταν στο όριο του αθροίσματος που βάζαμε. Ωστόσο δεν καταφέραμε γρήγορα να εντοπίσουμε το λάθος και για το λόγο αυτό καταφύγαμε σε άλλους τρόπους υπολογισμού του probability conversion. Κοιτώντας άλλες υλοποιήσεις του αλγόριθμου EquateROI είδαμε τον κώδικα του Agent Smith [8] και προσέξαμε ότι υπολόγιζε την πιθανότητα αυτή απλά και μας έβγαζε καλά αποτελέσματα. Ο τύπος που χρησιμοποιήσαμε και που βρήκαμε στον Agent Smith είναι:

$$Pr(\text{conversion}) = \frac{\text{Total_Sales}}{\text{Total_Clicks}}$$

, δηλαδή το υπολογίσαμε ορίζοντας το ως προς τις πόσες μονάδες έχουμε πουλήσει δια τα πόσα κλικ έγιναν επάνω στη διαφήμιση. Πρόκειται για μία εκτίμηση της πιθανότητας να αγοράσει ένας χρήστης το προϊόν της διαφήμισης δεδομένου των πωλήσεων και κλικς που έχουν προηγηθεί. Ωστόσο, στην αρχή που δεν υπάρχει κάποια μέση τιμή με πολλά δείγματα ο τύπος αυτός δίνει μέτρια αποτελέσματα. Για το λόγο αυτό στις αρχές χρησιμοποιήσαμε έναν δικό μας ευριστικό αλγόριθμο για τον οποίο περισσότερα θα δούμε παρακάτω.

Ξ

Πρόκειται για μία πολύ εμπειρική σταθερά η οποία προστίθεται στο CPCs που υπολογίζεται και μας δίνει το τελικό bid που πρέπει να κάνουμε για ένα query. Για τη συγκεκριμένη σταθερά βρήκαμε ελάχιστες πληροφορίες και γι' αυτό είδαμε εμπειρικά ότι με τη τιμή 0.4 έδινε αρκετά καλά αποτελέσματα.

4. ΥΛΟΠΟΙΗΣΗ ΕΥΡΙΣΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

Σε αυτό το σημείο της αναφοράς θα παρουσιάσουμε αναλυτικά τον αλγόριθμο με τον οποίο δομείται η στρατηγική με βάση την οποία κάνουμε προσφορά και που πρόκειται ουσιαστικά για αντικείμενο του πράκτορα μας. Αξίζει να αναφέρουμε ότι ο αλγόριθμος είναι αποτέλεσμα ενός ευριστικού αλγορίθμου που σκεφτήκαμε οι ίδιοι μαζί με τον αλγόριθμο του EquateROI. Παραθέτουμε αρχικά την υλοποίηση του ευριστικού αλγορίθμου.

1. Ο ευριστικός αλγόριθμος εφαρμόζεται στην περίπτωση που είμαστε στην αρχή του παιχνιδιού ($dateIndex \leq 5$) ή έχουμε ανανέωση capacity ($dateIndex \% 5 == 0$) ή βρισκόμαστε στα μέσα για ανανέωση του capacity ($dateIndex \% 3 == 0$). Ο λόγος που λάβαμε υπόψη και το τελευταίο είναι επειδή προσέξαμε ότι ο ευριστικός αλγόριθμος απότομα reaks στα κέρδη. Ο αλγόριθμος EquateROI εξισορροπούσε την κατάσταση μετά από κάποια έξαρση στα κέρδη μας, χωρίς την ύπαρξη αρνητικού profit και γι' αυτό εφαρμόζαμε ενδιάμεσα στον ευριστικό αλγόριθμο τον EquateROI ώστε να έχουμε κάποιο reak και στη συνέχεια κάτι σταθερό. Επίσης στις πρώτες μέρες δεν ήταν καλός ο υπολογισμός του probability conversion λόγω λίγων πληροφοριών για τις πωλήσεις. Επομένως και εκεί χρησιμοποιήσαμε μόνο τον ευριστικό μας αλγόριθμο.
2. Αμέσως μετά ελέγχουμε για την ύπαρξη κατασκευαστή και αν αυτός είναι ο ίδιος με αυτόν που εξειδικευόμαστε. Δίνουμε μεγάλη σημασία στον κατασκευαστή διότι αν πετύχουμε χρήστη που ενδιαφέρεται για τον κατασκευαστή στον οποίο εξειδικευόμαστε κερδίζουμε το manufacturer bonus. Επομένως, αν έχουμε ίδιο κατασκευαστή με το χρήστη και εξειδικευόμαστε και στο ίδιο προϊόν θέλουμε να κερδίσουμε το πρώτο slot στη λίστα για το χρήστη και για αυτό κάνουμε ένα bid αρκετά μεγάλο ώστε να μας εξασφαλίσει promotion slot, αλλά και όχι τόσο μεγάλο ώστε να μας αποφέρει αρνητικά κέρδη. Το bid λοιπόν είναι ίσο με :

$$Bid_for_specialty = Bid_F2_Min + \frac{Capacity - Total_Sales_Until_Now}{Capacity}$$

, δηλαδή το ελάχιστο Bid που μπορούμε να βάλουμε για query με υψηλό focus level συν μία εκτίμηση του πόσο ακόμα μπορούμε και θέλουμε να διαθέσουμε για bid δεδομένου ότι υπολογίζουμε πόσα προϊόντα έχουμε διαθέσιμα ($Capacity - Total_Sales_Until_Now$) προς το συνολικό Capacity που έχουμε διαθέσιμο και ορίζεται στην αρχή του παιχνιδιού. Το κλάσμα αυτό παίρνει τιμές μεταξύ 0 και 1 και δεν μας αφήνει να ξεφύγουμε στην προσφορά ώστε να υπερβαίνουμε αυτό που θέλουμε και να έχουμε αρνητικό κέρδος. Για παράδειγμα αν δεν έχουμε πουλήσει ακόμα κανένα προϊόν και έχουμε matching manufacturer και component specialty, τότε θέλουμε

μεγάλα bids για να αρχίσουμε να έχουμε conversions. Το Bid_F2_Min είναι ορισμένο στην κλάση Parameters ίσο με 1.8.

3. Στην περίπτωση που δεν έχουμε component αλλά έχουμε ίδιο manufacture specialty με το χρήστη πάλι κερδίζουμε MSB (Manufacturer Specialty Bonus). Επομένως ομοίως με πριν θα ορίσουμε :

$$Bid_for_specialty = Bid_F1_Min + \frac{Capacity - Total_Sales_Until_Now}{Capacity}$$

,ώστε να πετύχουμε conversion με το χρήστη. Η μόνη διαφορά είναι ότι τώρα θα έχουμε Bid_F1_Min που αντιστοιχεί σε queries με μεσαίο level focus. Το Bid_F1_Min έχει οριστεί στην κλάση Parameters ίσο με 1.7.

4. Τέλος στην περίπτωση που είμαστε στις δύο πρώτες μέρες ($dateIndex \% W == 0 \ || \ dateIndex \% W == 1$) έχουμε πολύ αυξημένο Capacity και πιθανόν οι τακτικές των υπόλοιπων πρακτόρων να μην έχουν αρχίσει ακόμα να επηρεάζουν τον πράκτορα μας. Επομένως εκεί κάναμε ένα ακόμα μεγαλύτερο bid, καθώς προσέξαμε ότι δε χάναμε από αυτό και το οποίο είναι ίσο με :

$$bid = 1.3 * bid;$$

5. ΥΛΟΠΟΙΗΣΗ EquateROI ΑΛΓΟΡΙΘΜΟΥ

Όπως αναφέραμε και προηγουμένως, χρησιμοποιήσαμε τον αλγόριθμο EquateROI γιατί αυτός μετά από ένα peak κέρδους φρόντιζε να εξισορροπεί τα conversions και τα impressions έτσι ώστε να μην έχουμε αρνητικό κέρδος. Αναλυτική περιγραφή της εφαρμογής του αλγορίθμου έγινε στην προηγούμενη υποενότητα. Επομένως στο σημείο αυτό παραθέτουμε την υλοποίηση που έγινε με βάση το paper και χρησιμοποιείται στις μέρες που ακολουθούνε μετά από κάποιο αυξημένο κέρδος, δηλαδή μετά από την εφαρμογή του ευριστικού αλγορίθμου.

1. Η πιθανότητα του probability conversion υπολογίζεται όπως αναφέραμε στην υποενότητα περιγραφής του αλγορίθμου EquateROI.
2. Για τον υπολογισμό του ROI κάνουμε αυτό που αναφέραμε προηγουμένως. Ωστόσο, αξίζει να αναφέρουμε ότι στην αρχή αντιμετωπίσαμε δυσκολίες στον υπολογισμό του totalSales καθώς αρχικά η τιμή του είναι μηδέν μιας και δεν έχουν γίνει πωλήσεις τις πρώτες μέρες. Επίσης, να θυμίσουμε ότι το ROI αυτό πρέπει να υπολογιστεί για κάθε query. Αυτό όμως δεν ήταν δυνατό για τον παραπάνω λόγο (μηδενικά sales αρχικά για ορισμένα queries). Επομένως, έπρεπε να βρούμε κάποιο τρόπο ευριστικό να το υπολογίσουμε. Σε papers που είδαμε υπήρχαν estimators για το λόγο αυτό. Εμείς, λόγω του περιορισμένου χρόνου που είχαμε δεν υλοποιήσαμε κάτι αντίστοιχο και υπολογίσαμε τελικά ένα ROI για όλα τα queries και όχι για το καθένα ξεχωριστά ούτως ώστε το totalSales να είναι σίγουρα διάφορο του 0 (θα γίνει σίγουρα κάποια αγορά για κάποιο query).
3. Πρόκειται για τον υπολογισμό του targetROI το οποίο αν τα totalSales έχουν ξεπεράσει είναι ίσα με το Capacity μειώνεται, διαφορετικά αυξάνεται. Στη συνέχεια υπολογίζεται και το estimated Cost Per Click όπως αναφέραμε και σε προηγούμενη υποενότητα ώστε να υπολογιστεί τελικά και το κατάλληλο bid. Αξίζει να αναφέρουμε και εδώ ότι το targetROI δεν υπολογίζεται ξεχωριστά για το κάθε query αλλά υπολογίζουμε ένα targetROI για όλα τα queries.
4. Υπολογίζουμε το bid που θα κάνουμε με βάση το CPC που έχουμε υπολογίσει παραπάνω. Όπου 0.4 είναι το ϵ που αναφέραμε σε προηγούμενη ενότητα. Σε περίπτωση που βρισκόμαστε είτε στην πρώτη είτε στη δεύτερη μέρα του παραθύρου και το capacity μας είναι αυξημένο τότε κάνουμε μία μεγαλύτερη προσφορά καθώς έχουμε τα περιθώρια. Διαφορετικά κάνουμε την προσφορά που μας λείει και ο αλγόριθμος EquateROI δηλαδή $\epsilon + CPC$. Ο λόγος που κάναμε αυτή τη διαφοροποίηση στον υπολογισμό του bid ήταν επειδή είδαμε ότι μας απέδιδε καλύτερα αποτελέσματα η προσθήκη επιπλέον bid όταν είμαστε στην αρχή του παραθύρου. Ωστόσο, καταλάβαμε μετέπειτα ότι δεν μπαίνει ποτέ σε αυτή τη συνθήκη καθώς αυτή την περίπτωση την ελέγχουμε σε άλλη συνθήκη.

5. Ο ρόλος του ελέγχου αυτού είναι για να μην ξεπερνάει η προσφορά μας κάποιο threshold που έχουμε ορίσει. Επομένως αν η προσφορά πάει να ξεπεράσει τα 4.0 δολλάρια τότε τη ρίχνουμε στο μισό. Διαφορετικά βγαίνουμε από τη συνθήκη.

6. ΔΥΣΚΟΛΙΕΣ ΠΟΥ ΑΝΤΙΜΕΤΩΠΙΣΑΜΕ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΚΩΔΙΚΑ

Γενικά αυτό που αντιληφθήκαμε κατά την υλοποίηση του πράκτορα ήταν ότι το μεγαλύτερο μέρος των αλγορίθμων που έχουν ήδη εφαρμοστεί είναι βασισμένο ευριστικούς αλγορίθμους βασισμένους σε τακτικές multiagent. Ακόμη, ήταν αλγόριθμοι παικτών που συμμετέχουν ή συμμετείχαν στο διαγωνισμό και επομένως πολλές πληροφορίες παραμένουν ελλιπείς. Τις δυσκολίες που αντιμετωπίσαμε τις αναφέραμε και παραπάνω μαζί με τον τρόπο που επιλύθηκαν. Τις παραθέτουμε συνοπτικά και παρακάτω:

- Λίγο υλικό στο internet επάνω στον αλγόριθμο που θέλαμε να υλοποιήσουμε.
- Δυσκολία στη μεταφορά του Probability Conversion από τα specifications στο κώδικα και εύρεση εναλλακτικού τύπου που πάλι ήθελε προσαρμογή.
- Πολλές σταθερές που έχουν ευριστικές τιμές βασισμένες στις τακτικές και το παιχνίδι των πρακτόρων έπρεπε να υπολογιστούν για να δίνουν τη μέγιστη απόδοση στο δικό μας παιχνίδι. Επομένως μας πήρε μέρες για να τρέξουμε πολλά παιχνίδια και να αντιληφθούμε αν όχι τις ιδανικές τουλάχιστο κάποιες κατάλληλες τιμές για τις παραμέτρους αυτές. (π.χ Initial TargetROI, INC_ROI, €)
- Δυσκολία στον υπολογισμό του totalSales για κάθε query τις πρώτες μέρες του υπολογισμού, διότι στα περισσότερα queries δεν είχαν γίνει πωλήσεις. Θα αναφέρουμε και πάλι ότι αυτό το λύσαμε ορίζοντας σαν totalSales τις συνολικές πωλήσεις για όλα τα queries. Αυτό όμως θεωρούμε ότι είναι πολύ γενικό και ότι πρέπει να υπολογιστεί με χρήση estimator.
- Παρόλο που κατεβάσαμε το server από το site για να βλέπουμε το παιχνίδι χρειαζόμασταν ένα plugin της java που πιθανόν να μην είχανε τα ubuntu στο firefox, για να βλέπουμε τα γραφικά. Επομένως συνδεόμασταν στο server από άλλο pc και όχι locally, πράγμα που μας δυσκόλεψε. Ρωτήσαμε και άλλα παιδιά που κάνανε την εργασία και είχανε το ίδιο πρόβλημα.
- Γενικά τα εκτελέσιμα που υπήρχαν στο site δεν μπορούσαμε να τα τρέξουμε με ευκολία. Για παράδειγμα ο TacTex όταν πηγαίναμε να κάνουμε ./run-agent μας έβγαζε errors και το ίδιο και οι άλλοι. Μπορούσαμε να τρέξουμε μόνο τον EPFL agent και τον Ashton. Λόγω του περιορισμένου χρόνου κατορθώσαμε ο πράκτορας μας να κερδίζει μόνο τον EPFL agent και προσαρμόσαμε την τακτική του πράκτορα μας επάνω σε αυτόν.
- Δυσκολευτήκαμε να εξοικειωθούμε με τη λογική του παιχνιδιού και τις παραμέτρους του σκελετού του agent που μας δόθηκε. Αυτό μας πήρε πολύ χρόνο αλλά μας βοήθησε να καταλάβουμε τη λογική του παιχνιδιού.

7. ΣΧΟΛΙΑΣΜΟΣ ΑΠΟΔΟΣΗΣ ΤΟΥ ΠΡΑΚΤΟΡΑ ΜΑΣ ΣΤΟΝ ΕΣΩΤΕΡΙΚΟ ΔΙΑΓΩΝΙΣΜΟ

Εξαιτίας του μικρού αριθμού συμμετοχών στον εσωτερικό διαγωνισμό δεν μπορέσαμε να εκτιμήσουμε με ακρίβεια την επίδοση του πράκτορα μας σε ανταγωνιστικό περιβάλλον. Θεωρούμε όμως ότι γενικά η επίδοση του δεν ήταν κακή και γνωρίζουμε τα μεγαλύτερα μειονεκτήματα της τακτικής του πράκτορα μας καθώς και σφάλματα του αλγορίθμου μας. Πολλά από τα μειονεκτήματα αυτά τα γνωρίζαμε και πριν τον εσωτερικό διαγωνισμό ωστόσο οποιαδήποτε αλλαγή γινόταν στον πράκτορα έπρεπε να δοκιμαστεί για αρκετά παιχνίδια ώστε να είμαστε σίγουροι ότι δε θα απέδιδε αυτός χειρότερα και λόγω του περιορισμένου χρόνου προλάβαμε να κάνουμε συγκεκριμένες αλλαγές. Επομένως παρακάτω θα κάνουμε θετικά και αρνητικά σχόλια για την επίδοση του πράκτορα μας. Στα αρνητικά σχόλια αναφέρουμε και τα μειονεκτήματα του κώδικά μας.

Θετικά σχόλια

1. Στη γραφική παράσταση Advertiser Profits είχαμε πάντα μια σχετική γραμμικότητα και ποτέ δεν είχαμε αρνητικά profits που ήταν κύριος στόχος μας βλέποντας άλλους πράκτορες.
2. Το capacity που χρησιμοποιούσαμε ήταν για capacity 600 σταθερά γύρω στο 175, για capacity 450 σταθερά γύρω στο 150 και για 300 σταθερά γύρω στο 125. Αυτό ήταν και ο αρχικός μας στόχος καθώς δε θέλαμε το capacity να παίρνει ακραίες τιμές (είτε να ξεοδεύεται όλο αμέσως, είτε να μη χρησιμοποιείται καθόλου)
3. Παρατηρήσαμε ότι πάντα στα προϊόντα που ήταν targeted είχαμε promotion slot που σημαίνει ότι κάναμε σωστές προσφορές, ενώ σε προϊόντα που δεν επιθυμούσαμε έστω και με μικρή προσφορά καταφέρναμε να μπαίνουμε στη λίστα που βλέπει ο χρήστης προς το τέλος ώστε να γινόμαστε τουλάχιστο ανταγωνιστικοί.
4. Όσον αφορά την κατάταξη μας στον εσωτερικό διαγωνισμό θεωρούμε ότι πήγαμε καλά διότι είχαμε αρκετά περισσότερες απολαβές από πράκτορα που και στον προκριματικό γύρο είχε μέγιστο 30.000 απολαβές, ενώ στον τελικό εσωτερικό διαγωνισμό είχε ακόμα χαμηλότερες απολαβές. Ο πράκτορας πάνω από εμάς μας πέρασε με διαφορά περίπου 4.000 δολάρια, ωστόσο δεδομένου ότι ήταν και κώδικας που έχει διαγωνιστεί πολλές φορές στο παγκόσμιο διαγωνισμό θεωρούμε ότι αυτό ήταν αρκετά καλό.

Αρνητικά σχόλια

1. Παρατηρήσαμε ότι ενώ γενικά δεν χάναμε profit ωστόσο, δεν κάναμε και μεγάλα reaks κέρδους. Επομένως, δεν είχαμε μεγάλες απολαβές ώστε να έχουμε μεγαλύτερα κέρδη όπως έκανε ο πράκτορας των παιδιών που βγήκε πρώτος στο διαγωνισμό. Αυτός θεωρούμε ότι ήταν και ο κύριος λόγος που δεν πήγαμε καλύτερα.
2. Το διαγωνισμό τον τρέξαμε δύο φορές διότι την πρώτη φορά για κάποιο λόγο κόλλησε το παιχνίδι. Ωστόσο, στα παιχνίδια εκείνα που είχαμε παίξει στην αρχή ο πράκτορας που βγήκε δεύτερος είχε πάρει πολύ χαμηλά κέρδη. Θεωρούμε ότι αν πάρει κάποιος πολύ χαμηλά κέρδη (περίπου 3000 δολάρια) μετά είναι πολύ δύσκολο για αυτόν να έχει έναν πολύ ψηλό μέσο όρο όπως έγινε στο δεύτερο γύρο παιχνιδιών.
3. Όταν οι αντίπαλοι μας είχανε reaks κέρδους τότε ο αλγόριθμος μας δεν άλλαζε στρατηγική και έπαιζε με τον ίδιο τρόπο. Επομένως αυτό έριχνε την απόδοση του καθώς δεν μπορούσε να προσαρμοστεί στο περιβάλλον αλλά συνέχιζε να παίζει το ίδιο παιχνίδι ανεξάρτητα από τους αντιπάλους του. Αυτό ήταν και το μεγαλύτερο ελάττωμα του.

8. ΕΝΕΡΓΕΙΕΣ ΒΕΛΤΙΩΣΗΣ ΤΟΥ ΠΡΑΚΤΟΡΑ ΜΑΣ

Με βάση την απόδοση του πράκτορα στον εσωτερικό διαγωνισμό, αλλά και τις αλλαγές που θέλαμε να πραγματοποιήσουμε και δεν προλάβουμε αναφέρουμε τις κύριες ενέργειες που μπορούν να γίνουν ώστε να βελτιωθεί και η απόδοση του πράκτορα και να γίνει περισσότερο ανταγωνιστικός:

1. Πρέπει να γίνουν ορισμένες αλλαγές στον ευριστικό κυρίως αλγόριθμο ώστε να πετυχαίνουμε μεγαλύτερες αυξήσεις του κέρδους όταν έχουμε ανανέωση του capacity.
2. Πολλές από τις παραμέτρους στον αλγόριθμο EquateROI που έχουμε υπολογίσει ευριστικά παίζοντας αρκετά παιχνίδια πρέπει να τις αλλάξουμε (ειδικά τη σταθερά ϵ) ώστε να μας δίνει καλύτερα αποτελέσματα.
3. Η προσφορά που κάνουμε σε queries που δεν είναι targeted και που είναι μικρή δεν πρέπει να γίνεται τυχαία όπως έχουμε κάνει αλλά είτε να υπολογίζεται βάση κάποιου τύπου είτε να υπολογιστεί μία ευριστική καλή τιμή για αυτήν.
4. Να εφαρμοστεί ο αλγόριθμος EquateROI επακριβώς και να υπολογίζεται το targetROI για κάθε query ώστε να υπάρχει μία πιο ακριβής τιμή. Για να γίνει αυτό όμως θα πρέπει να ετοιμαστούν estimators που να εκτιμούν τη τιμή που θα έχει το total_sales στον παρανομαστή του κλάσματος υπολογισμού του ROI.
5. Κατά την προετοιμασία του πράκτορα εμείς τον δοκιμάσαμε εναντίον του πράκτορα του EPFL με αποτέλεσμα να τον προσαρμόσουμε έτσι ώστε να κερδίζει το συγκεκριμένο πράκτορα. Ο πράκτορας όμως πρέπει να παίξει και εναντίον κι άλλους παίκτες ώστε να προσαρμοστεί πιο γενικά.
6. Να μελετήσουμε την τακτική του πράκτορα μας ώστε να μπορεί να προσαρμόζει καλύτερα την τακτική του στο ανταγωνιστικό περιβάλλον και ειδικά όταν οι αντίπαλοι έχουν υψηλά κέρδη.

9. ΑΝΑΦΟΡΕΣ (REFERENCES)

1. Jordan Berg, Amy Greenwald, Victor Naroditskiy, and Eric Sodomka. *A First Approach to Autonomous Bidding in Ad Auctions*.
<http://www.cs.uwaterloo.ca/conferences/tada2010/Brown.pdf>
2. Jordan Berg, Amy Greenwald, Victor Naroditskiy, and Eric Sodomka. *A Knapsack-Based Approach to Bidding in Ad Auctions*.
<http://cs.brown.edu/people/sodomka/resources/papers/2010-ecai.pdf>
3. Σκάντζα Αναστασία και Τζαμτζή Βασιλεία. *Ανάπτυξη Μηχανισμού Πλειοδοσίας Σε Συστήματα Δημοπρασιών Ηλεκτρονικής Διαφήμισης*.
http://vivliothmyy.ee.auth.gr/1072/1/Diploma_Thesis_Skantza_Tzamtzi.pdf
4. Hang Chen and Amy Greenwald. *DREU Final Report*.
http://www.cra.org/Activities/craw_archive/dmp/awards/2009/Chen/report.pdf
5. Tyler Odean. *Marginal Bidding : An Application of the Equimarginal Principle to Bidding in TAC SCM*.
<http://www.cs.brown.edu/research/pubs/theses/masters/2009/odean.pdf>
6. Jonas Vautherin and Ludek Cigler. *Semester project – An analysis of bidding strategies for the Trading Agent Competition*.
http://lia.epfl.ch/uploads/project_reports/report_253.pdf
7. Patrick R. Jordan, Akshat Kaul and Michael P. Wellman. *Trading Agent Competition Ad Auctions The Ad Auctions Game*.
http://aa.tradingagents.org/wp-content/uploads/2010/03/tac_aa_spec_2010-v10.1.0.01.pdf
8. *Agent – Smith*.
<http://code.google.com/p/agent-smith/source/browse/trunk/agent-smith/src/sampleAgent/sampleOptimizer/EquateROI.java?r=81>
9. *Ενημερωμένο από το Χαρίλαο*.
<http://www.intelligence.tuc.gr/~gehalk/COMP517/TUC-ExampleAAgent.zip>

** Από τα παραπάνω references λήφθηκαν οι πληροφορίες για την υλοποίηση του κώδικα μας. Δεν περιλαμβάνονται όλα τα papers τα οποία μελέτησαμε. **