

Ad Exchange Workshop

2016-17

PineApple – Design Specification

8/1/2017

מגשים:

305135956	אורי בר אל
305210189	אייל וסרמן
203317565	יוסי דדוש
201507845	שני לויים



תוכן עניינים

3.....	הקדמה
4.....	קביעת ה-Campaign Opportunity Bid
4.....	שערוך הרווחיות של קמפיין
4.....	קביעת ה-Campaign Opportunity Budget בפועל: B_i
5.....	קביעת ה-UCS Bid
5.....	קביעת רמת ה-UCS הרצויה: l_d
5.....	שיערוך המחיר הדרוש על מנת לזכות ברמה l_d : $b_{UCS}(l_d)$
6.....	קביעת Bid Bundle
6.....	קביעת Price per Impression ממוצע לקמפיין
6.....	קביעת ה-Bid Bundle עצמו
8.....	ארכיטקטורה
8.....	Classes
8.....	Datasets

הקדמה

נשים לב להבחנה הבאה הנובעת מה-specification של המשחק:

$$\frac{R_i}{10 \cdot Q} < B_i \cdot 10^3 < R_i \cdot Q$$

כאשר Q הוא ה-Quality העדכני, B_i הוא התקציב שאנו דורשים עבור הקמפיין ה- i ו- R_i הוא ה-reach של הקמפיין ה- i . כמו כן, עבור הקמפיין ה- i נגדיר את D_i ו- S_i להיות קבוצת הימים שבהם הקמפיין פעיל וקבוצת הסגמנטים אליהם מופנה הקמפיין בהתאמה.

כעת, נגדיר את פונקציית האינדיקטור הבאה:

$$\psi_{s,d,i} = \begin{cases} 1, & s \in S_i, d \in D_i \\ 0, & o.w. \end{cases}$$

ונגדיר מדד למידת הביקוש של סגמנט s ביום d :

$$demand(s, d) = \frac{\sum_{i \in C} \frac{1}{|S|} \cdot \psi_{s,d,i} \cdot \frac{R_i}{|D_i|}}{\sum_{s' \in S} \sum_{i \in C} \frac{1}{|S'|} \cdot \psi_{s',d,i} \cdot \frac{R_i}{|D_i|}}$$

כאשר S היא קבוצת הסגמנטים כולה ו- C היא קבוצת כל הקמפיינים שהמשחק הציע במשחק הנוכחי.

אם d' הינו היום הנוכחי, אזי נוסחה זו היא **מדויקת** עבור כל $d \leq d' + 2$. אולם, בימים המקיימים $d > d' + 2$, **עשויים** להתחיל קמפיינים אשר יעלו את מידת הביקוש של סגמנט מסוים, ועליהם אין לנו כל מידע ביום d' בו מתבצע החישוב. לכן, נרצה להוסיף גורם תיקון למדד הנ"ל. גורם תיקון זה ישערך את הביקוש העתידי שיתווסף לכל סגמנט ע"י קמפיינים עתידיים. גורם תיקון זה יחושב כקמפיין דמה שיסמלך את אי הידיעה על פי ההיסטוריה.

נתייחס ל $\sigma(s, \vec{d})$ כקמפיין אשר פונה לסגמנט s (מסוים) בלבד. בעבור קמפיין זה, נחשב את הסטטיסטיקות לפי הימים הקודמים (הרלוונטיים רק עבור אותו סגמנט s). משך הקמפיין עבור קמפיין זה (ה- D_i שלו) ייקבע על פי ממוצע מספר הימים של קמפיינים שהתחילו ביום ה- \vec{d} על בסיס סטטיסטיקות של משחקים קודמים, וה- $reach$ שלו (ה- R_i שלו) הוא ממוצע של ה- $reach$ של קמפיינים שהתחילו ביום ה- \vec{d} על בסיס סטטיסטיקות קודמות מוכפל בהסתברות שקמפיין שמתחיל ביום ה- \vec{d} יפנה לסגמנט ה- s על בסיס סטטיסטיקות קודמות.

לכן, בנוסחה $demand(s, d)$ נחליף את C ב-

$$C' = C \cup \{\sigma(s, \vec{d}) | d' + 2 < \vec{d} \leq d\}$$

נרחיב את ההגדרה הנ"ל לקבוצת סגמנטים $S' \subseteq S$ כך:

$$demand(S', d) = \sum_{s' \in S'} demand(s', d)$$

וכן עבור קבוצת ימים D' כך:

$$demand(S', D') = \frac{1}{|D'|} \sum_{d \in D'} demand(S', d)$$

הערה: מכיוון שאנו לא יודעים מהם הקמפיינים שמוקצים ביום ה-0 ליתר השחקנים, נשערך אותם באופן דומה ונוסיף לקבוצה C' .

קביעת ה- Campaign Opportunity Bid

נגדיר $\tilde{B}_i = demand(S_i, D_i) \cdot R_i \cdot \alpha$, כאשר $\alpha \sim \mathcal{N}(\mu, \sigma^2)$ עבור μ ו- σ^2 אותם נמצא בהמשך. α הוא גורם שנועד למנוע מהסוכן להיות צפוי בבחירותיו. נסתכל על התנאי הבא:

$$\frac{R_i}{10 \cdot Q} < \tilde{B}_i \cdot 10^3 < R_i \cdot Q$$

אם התנאי לא מתקיים, נעדכן את ערכו של \tilde{B}_i באופן הבא:

$$\tilde{B}_i = R_i \cdot \frac{Q}{10^3} - \varepsilon \quad \text{אם } \tilde{B}_i \cdot 10^3 \geq R_i \cdot Q \quad \text{אז נגדיר:}$$

$$\tilde{B}_i = \frac{R_i}{10^4 \cdot Q} + \varepsilon \quad \text{אם } \tilde{B}_i \cdot 10^3 \leq \frac{R_i}{10 \cdot Q} \quad \text{אז נגדיר:}$$

שערוך הרווחיות של קמפיין

נשתמש באלגוריתם AdaBoost עם *decision stumps* (עצי החלטה בגודל 1) שיתבסס על היסטוריית משחקים קודמים תוך שימוש ב-*features* הבאים:

- היום d
- התקציב \tilde{B}_i
- יום התחלת הקמפיין
- יום סיום הקמפיין
- הסגמנט S_i
- מספר ה-*impressions* הדרושים R_i
- קבוע ה-*video*
- קבוע ה-*mobile*
- אחוז השלמת הקמפיין
- הרווח כתוצאה מהקמפיין
- $demand(S_i, D_i)$

ההחלטה +1 (אם הקמפיין רווחי) או -1 (אם הקמפיין לא רווחי) תתבצע כדלהלן:

1. אם הושלמו לפחות 100% מהקמפיין והרווח מהקמפיין גדול מאפס אז ההחלטה היא 1.
2. אם הושלמו לפחות 90% מהקמפיין והרווח מהקמפיין גדול מ- $\frac{\tilde{B}_i}{R_i} \cdot 0.02$ אז ההחלטה היא 1.
3. אם הושלמו לפחות 80% מהקמפיין והרווח מהקמפיין גדול מ- $\frac{\tilde{B}_i}{R_i} \cdot 0.1$ אז ההחלטה היא 1.
4. בכל מקרה אחר, ההחלטה היא -1.

קביעת ה- Campaign Opportunity Budget: B_i בפועל:

$$B_i = \begin{cases} \tilde{B}_i & \text{the campaign is profitable} \\ \text{maximal valid budget} = R_i \cdot \frac{Q}{10^3} - \varepsilon & \text{o. w.} \end{cases}$$

קביעת ה-UCS Bid

ה-bid UCS ביום ה-d ייקבע בשני שלבים:

- קביעת רמת ה-UCS הרצויה ביום ה-d, l_d .
- שיערוך המחיר הדרוש על מנת לזכות ברמה זו: $b_{UCS}(l_d)$.

קביעת רמת ה-UCS הרצויה: l_d

נקבע את רמה זו כך: יהי $I_{target,i}$ כמות ה-impressions שאנו מעוניינים להשיג בקמפיין ה-i ויהי $I_i(d)$ כמות ה-impressions שהושגו עד היום ה-d (לא כולל) בקמפיין ה-i. אז נגדיר:

$$Accuracy(l_d) = \operatorname{argmax}_{i \in C} \left[\frac{I_{target,i} - I_i(d)}{\text{DaysToGo}(i)} \cdot \frac{1}{|S_i|} \right]$$

כאשר פונקציית ה-ceiling "מעגלת" מעלה לרמת ה-UCS הבאה ו- $Accuracy(l)$ היא מידת הדיוק של הדרגה ה-l.

שיערוך המחיר הדרוש על מנת לזכות ברמה l_d : $b_{UCS}(l_d)$

תחילה נשערך את המחיר הצפוי $b_{UCS}(l_d)_{pred}$ על-ידי אלגוריתם רגרסיה, [למשל SVM (Support Vector Machine) עם kernel אופטימלי, כדוגמת RBF, אותו נחשב בהמשך] שישתמש ב-features הבאים:

- היום d.
- $demand(s, d)$ לכל סגמנט s (נזכיר: חישוב זה מדויק עבור היום ה-d כשהוא מבוצע ביום ה-d-1)
- אחוז השחקנים שיחזיקו בקמפיין פעיל ביום ה-d מתוך כלל השחקנים.
- אחוז השחקנים שיחזיקו בקמפיין שיומו האחרון הוא d מתוך כלל השחקנים.

אלגוריתם זה ישערך את הערך הבא של $b_{UCS}(l_d)_{pred}$ בסדרת המחירים של l_d .

לאחר מכן, ה-bid ייקבע ע"י שימוש ב-exponential smoothing כדלהלן:

נסמן ב- l_{d-1} את הרמה הדרושה ביום ה-d-1 וב- \widetilde{l}_{d-1} את הרמה בפועל ביום ה-d-1. נסמן: $\Delta = l_{d-1} - \widetilde{l}_{d-1}$.

קעת נקבע את $b_{UCS}(l_d)$ באופן הבא:

- אם $l_d = 8$ אז $b_{UCS}(l_d) = 0$.
- אחרת:

○ אם $1 \leq l_d + \Delta \leq 8$:

$$b_{UCS}(l_d) = (1 - \alpha) \cdot b_{UCS}(l_d)_{pred} + \alpha \cdot b_{UCS}(l_d + \Delta)_{pred}$$

○ אם $l_d + \Delta < 1$:

$$b_{UCS}(l_d) = [(1 - \alpha) \cdot b_{UCS}(l_d)_{pred} + \alpha \cdot b_{UCS}(1_d)_{pred}] \cdot \beta$$

○ אם $l_d + \Delta \geq 8$ אז:

$$b_{UCS}(l_d) = [(1 - \alpha) \cdot b_{UCS}(l_d)_{pred} + \alpha \cdot b_{UCS}(8_d)_{pred}] \cdot \gamma$$

כאשר: $0 \leq \alpha \leq 1$, $\beta > 1$, $0 \leq \gamma < 1$ ואת α, β, γ נגדיר בהמשך באופן האופטימלי שנמצא.

קביעת Bid Bundle

קביעת Price per Impression ממוצע לקמפיין

יהי $I_{target,i}$ כמות ה-impressions שאנו מעוניינים להשיג בקמפיין ה- i , כאשר גודל זה נקבע כך שהוא ימקסם את הפונקציה הבאה:

$$f(I, \bar{p}) = TotalProfit(I) \cdot Q^\alpha(I)$$

כאשר \bar{p} הוא המחיר הממוצע ל-impression α -י הוא פרמטר שנועד לתת משקל שונה ל- Q וערכו האופטימלי ייקבע בהמשך.

הערה: אנו מניחים כי מקסימום לפונקציה f ישיג בו-זמנית רווח כספי גדול יחד עם שימור $Quality\ rate$ גבוה. כאמור, נרצה למקסם את הפונקציה:

$$f(I, p) = (ERR(I) \cdot B_i - \bar{p} \cdot I) \cdot ((1 - \eta)Q_{old} + \eta \cdot ERR(I))^\alpha$$

על-מנת למצוא מקסימום לפונקציה זו, נשתמש בתנאי היריסטי זה:

$$\bar{p} = \frac{demand(S_i, D_i) \cdot B_i}{I_{target,i}}$$

כלומר, נקבל:

$$f(I, \bar{p}) = (ERR(I) \cdot B_i - p \cdot demand(S_i, D_i) \cdot B_i) \cdot ((1 - \eta)Q_{old} + \eta \cdot ERR(I))^\alpha$$

$$f(I) = B_i \cdot (ERR(I) - demand(S_i, D_i)) \cdot ((1 - \eta)Q_{old} + \eta \cdot ERR(I))^\alpha$$

ה- I שימקסם את הפונקציה האחרונה הוא $I_{target,i}$ המבוקש. נמצא אותו ע"י brute force.

קביעת ה-Bid Bundle עצמו

עבור הקמפיין ה- i ביום ה- d נרצה לקבוע את כמות ה-targeted impressions שאנו מעוניינים להשיג ביום $d + 1$, ונסמן ערך זה ב- $I_{t,i,d+1}$.

$$I_{t,i,d+1} = \max \left\{ |S_i| \cdot Accuracy(l_{d+1} - 1), \left(I_{target,i} - I_i(d + 1) \right) \cdot Accuracy(l_{d+1} - 1) \right\}$$

נמין כעת את הסגמנטים ששייכים ל- S_i בסדר עולה בהתאם ל- $demand(s, d + 1)$.

נבחר את k הסגמנטים הראשונים בסדרה הממויינת שגודלם המצרפי כפול $Accuracy(l_{d+1} - 1)$ הוא $I_{t,i,d+1}$ לפחות. לכל אחד מהסגמנטים s_j הללו נייצר bid מהצורה:

$$\left\{ \begin{array}{l} \text{price per impression: } \bar{p} \cdot demand(s_j, d + 1), \\ \text{number of impressions limit: } size \cdot Accuracy(l_{d+1} - 1), \\ \text{total price limit: } \bar{p} \cdot demand(s_j, d + 1) \cdot size \cdot Accuracy(l_{d+1} - 1) \end{array} \right\}$$

כלומר נקבל את סדרת ה-bid-ים הבאה:

$$\left\{ \begin{array}{l} \bar{p} \cdot demand(s_1, d + 1), \\ |s_1| \cdot Accuracy(l_{d+1} - 1), \\ \bar{p} \cdot demand(s_1, d + 1) \cdot |s_1| \cdot Accuracy(l_{d+1} - 1) \end{array} \right\}$$

$$\left\{ \begin{array}{c} \bar{p} \cdot demand(s_2, d + 1), \\ |s_2| \cdot Accuracy(l_{d+1} - 1), \\ \bar{p} \cdot demand(s_2, d + 1) \cdot |s_2| \cdot Accuracy(l_{d+1} - 1) \end{array} \right\}$$

...

$$\left\{ \begin{array}{c} \bar{p} \cdot demand(s_k, d + 1), \\ (I_{t,i,d+1} - \sum_{j=1}^{k-1} |s_j|) \cdot Accuracy(l_{d+1} - 1), \\ \bar{p} \cdot demand(s_k, d + 1) \cdot (I_{t,i,d+1} - \sum_{j=1}^{k-1} |s_j|) \cdot Accuracy(l_{d+1} - 1) \end{array} \right\}$$

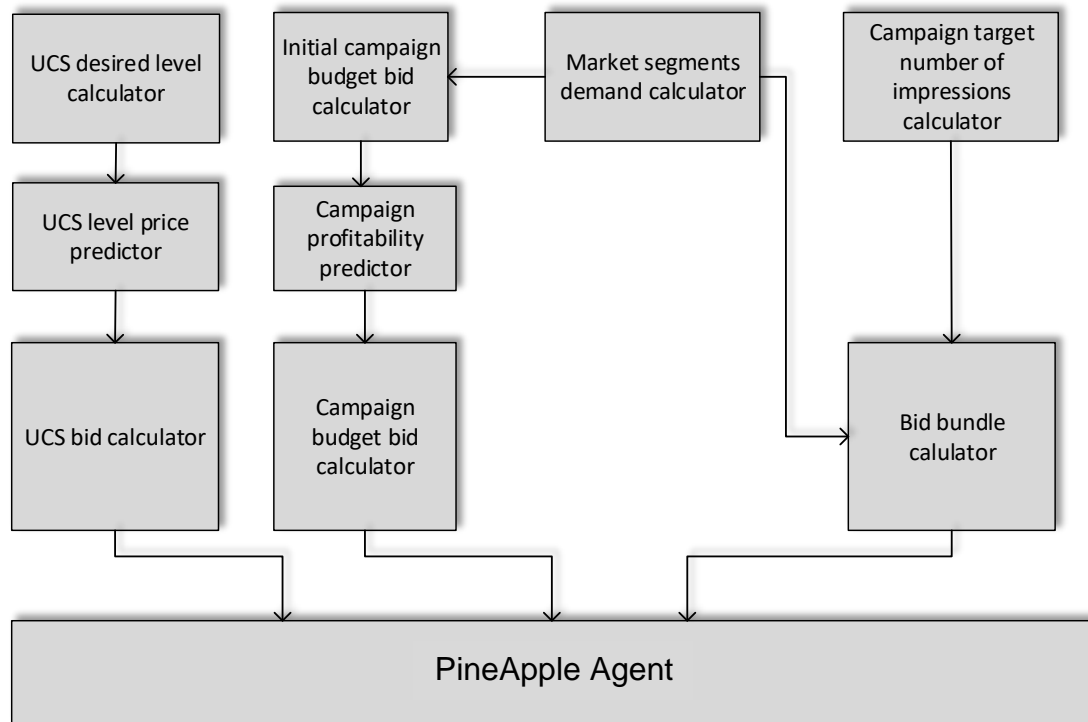
בסגמנט ה-k, כפי שנראה, ייתכן ונמלא רק חלק ממנו, על מנת לא לקבל יותר targeted impressions מאשר שהתכוונו.

נשים לב כי מתקיים:

$$\sum_{j=1}^k size(s_j) \cdot \frac{p_j}{demand(s_j, d + 1)} = \left[p_j \equiv \bar{p} \cdot demand(s_k, d + 1) \right] = \bar{p} \cdot I_{t,i,d+1}$$

כנדרש.

ארכיטקטורה



Classes

- CalcTargetedImp - אחראית על חישוב ה-targeted impressions שאנו מעוניינים להשיג.
- CalcPricePerImp - אחראית על קביעת כמות ה-impressions שאנו מעוניינים להשיג.
- CalcBidBundle - אחראית על קביעת ה-Bid Bundle.
- CalcSegDemand - אחראית על חישוב הביקוש של סגמנט ספציפי s במשך המשחק.
- CalcInitBudgetBid - אחראית על קביעת ה-Budget התחלתי עבור הקמפיין.
- CampaignProfitabilityPredictor - אחראית על שערך הרווחיות של קמפיין.
- CalcBudgetBid - אחראית על קביעת ה-Campaign Opportunity Budget בפועל, לאחר התחשבות בכמות הרווחיות של הקמפיין.
- CalcUCSDesired - אחראית על קביעת רמת ה-UCS הרצויה.
- UCSLevelPricePredictor - אחראית על שערך המחיר הצפוי עבור רמת ה-UCS הרצויה.
- CalcUCSBid - אחראית על חישוב ה-bid עבור רמת ה-UCS הרצויה.

Datasets

- סטטיסטיקות ממשחקים קודמים על מנת שנוכל לעשות predictions ו-classifications על סמך ההיסטוריה. סטטיסטיקות אלו יכללו מידע ממשחקים שאנחנו הרצנו, מידע שנאסוף בין משחק למשחק לאורך התחרות, וכן מידע שנאסוף בזמן המתרחש ביום המשחק עצמו.

- מידע אודות הקמפיינים השונים שהוקצו לאורך ימי המשחק: בכמה הם נקנו, לאיזה סגמנטים הם פנו, מה היה הreach, מהו משך זמנם, מה היה הרווח מאותם קמפיינים, מה היה אחוז ההשלמה שלהם, ומה היה השינוי באיכות לפני ואחרי.
- מחירים של רמות UCS בכל יום.
- עומסים של סגמנטים בימים שונים (מבחינת ביקוש עבור אותו סגמנט על סמך העבר, ועל סמך הסטטיסטיקות במסמך המשחק).
- כמה סוכנים מחזיקים בקמפיין בכל יום.
- כמה סוכנים שונים זוכים ביותר קמפיינים מהשאר.