Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Προηγμένα Θέματα Αλγορίθμων

Αλγόριθμοι Δικτύων και Πολυπλοκότητα

Εαρινό εξάμηνο 2021 – 2022

(ΕΜΠ – ΑΛΜΑ)

Διδάσκοντες: Α. Παγουρτζής - Δ. Φωτάκης - Θ. Λιανέας

1η σειρά ασκήσεων

Προθεσμία παράδοσης: 12/5/2022

Άσκηση 1

Θεωρούμε την online εκδοχή του Projected Gradient Descent. Έστω κυρτό σύνολο $S \subseteq \mathbb{R}^d$. Ο αλγόριθμος OPGD εξελίσσεται σε χρονικό ορίζοντα μήκους $T \in \mathbb{N}$ και χρησιμοποιεί βήμα $\eta > 0$.

Θεωρούμε μια αυθαίρετα επιλεγμένη αρχική λύση $\boldsymbol{x}_1 \in S$. Για κάθε χρονική στιγμή $t = 1, \dots, T$, ο αλγόριθμος OPGD:

1. Λαμβάνει ως είσοδο κυρτή συνάρτηση $f_t : S \to \mathbb{R}_+$, την οποία εφαρμόζει στην τρέχουσα λύση $\boldsymbol{x}_t$. Το κόστος του αλγόριθμου για το βήμα t είναι $f_t(\boldsymbol{x}_t)$.
2. Θέτει $\boldsymbol{y}_{t+1} = \boldsymbol{x}_t - \eta \nabla f_t(\boldsymbol{x}_t)$
3. Ενημερώνει την τρέχουσα λύση σε $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in S} \|\boldsymbol{x} - \boldsymbol{y}_{t+1}\|$, προβάλλοντας το $\boldsymbol{y}_{t+1}$ στο S με βάση την Ευκλείδεια απόσταση.

'Εστω $B = \max_{\boldsymbol{x},\boldsymbol{y} \in S} \|\boldsymbol{x} - \boldsymbol{y}\|$ η διάμετρος του S και $G = \max_t \max_{\boldsymbol{x} \in S} \|\nabla f_t(\boldsymbol{x})\|$ ένα άνω φράγμα στο Ευκλείδειο μέτρο του gradient των συναρτήσεων f_t.

1. Να δείξετε ότι για βήμα $\eta = \frac{B}{G\sqrt{T}}$,

$$\sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in S} \sum_{t=1}^{T} f_t(\boldsymbol{x}) \leq BG\sqrt{T}$$

   Η ποσότητα στο αριστερό μέλος της παραπάνω ανισότητας είναι γνωστή ως *regret* του αλγόριθμου (ως προς τη βέλτιστη λύση για το άθροισμα των συναρτήσεων f_t που υπολογίζεται εκ των υστέρων).

2. Να επιλέξετε χρονικά μεταβαλλόμενο βήμα η_t (πλέον το βήμα εξαρτάται από τη χρονική στιγμή t, όχι από το μήκος του χρονικού ορίζοντα T) ώστε να έχουμε regret $O(BG\sqrt{T})$, χωρίς εκ των προτέρων γνώση του μήκους T του χρονικού ορίζοντα.
3. Αν οι συναρτήσεις f_t είναι α-ισχυρά κυρτές, να επιλέξετε βήμα η_t (ως συνάρτηση των α και t) ώστε να έχουμε regret $O(G^2 \ln(T)/\alpha)$.

Άσκηση 2

Έστω το Γραμμικό Πρόγραμμα (Π):

$$\begin{array}{lrcl}
\text{min} & -10x_1 + 57x_2 + 9x_3 + 24x_4 & & \\
\text{s.t.} & 0.5x_1 - 5.5x_2 - 2.5x_3 + 9x_4 + x_5 & = & 0 \\
 & 0.5x_1 - 1.5x_2 - 0.5x_3 + x_4 + x_6 & = & 0 \\
 & x_1 + x_7 & = & 1 \\
 & x_1, x_2, x_3, x_4, x_5, x_6, x_7 & \geq & 0
\end{array}$$

(α) Να λύσετε (δύο φορές) το (Π) με τον αλγόριθμο Simplex. Να ξεκινήσετε με βασικές μεταβλητές τις x_5, x_6, x_7. Να ακολουθήσετε τους παρακάτω κανόνες για την εναλλαγή στηλών (pivoting) στη βάση (την πρώτη φορά τον ένα, την δεύτερη τον άλλο):

1. (i) ως νέα βασική μεταβλητή επιλέγεται εκείνη με το ελάχιστο ανηγμένο κόστος, και (ii) αν υπάρχουν δύο ή περισσότερες βασικές μεταβλητές υποψήφιες για έξοδο από τη βάση, επιλέγεται εκείνη με τον ελάχιστο δείκτη.
2. (i) ως νέα βασική μεταβλητή επιλέγεται εκείνη (από τις μεταβλητές με αρνητικό ανηγμένο κόστος) με τον ελάχιστο δείκτη, και (ii) αν υπάρχουν δύο ή περισσότερες βασικές μεταβλητές υποψήφιες για έξοδο από τη βάση, επιλέγεται εκείνη με τον ελάχιστο δείκτη.

(β) Να διατυπώσετε το δυϊκό πρόγραμμα (ΔΠ) του (Π). Να διατυπώσετε τις complementary slackness συνθήκες για τα (Π) και (ΔΠ), και να τις χρησιμοποιήσετε για τον υπολογισμό της βέλτιστης λύσης του (ΔΠ).

Άσκηση 3

Έστω $m \times n$ πίνακας A. Συμβολίζουμε με $\Delta^n = \{\boldsymbol{x} \in \mathbb{R}_{\geq 0} : \|\boldsymbol{x}\|_1 = 1\}$ το simplex n διαστάσεων (αντίστοιχα, Δ^m είναι το simplex m διαστάσεων). Για $k \in \mathbb{N}$, συμβολίζουμε $[k] = \{1, \dots, k\}$.

(α) Να εκφράσετε το πρόβλημα βελτιστοποίησης $P = \min_{\boldsymbol{x}\in\Delta^n} \max_{j\in[m]} (A\boldsymbol{x})_j$ στο Δ^n ως Γραμμικό Πρόγραμμα ($(A\boldsymbol{x})_j$ είναι η j-οστή συντεταγμένη του n-διανύσματος $A\boldsymbol{x}$). Ο βασικός περιορισμός είναι $\boldsymbol{x} \in \Delta^n$ και η αντικειμενική συνάρτηση είναι $c(\boldsymbol{x}) = \max_{j\in[m]} (A\boldsymbol{x})_j$, την οποία επιδιώκουμε να βελτιστοποιήσουμε.

(β) Να εκφράσετε το πρόβλημα βελτιστοποίησης $D = \max_{\boldsymbol{y}\in\Delta^m} \min_{i\in[n]} (A^T\boldsymbol{y})_i$ στο Δ^m ως Γραμμικό Πρόγραμμα.

(γ) Να δείξετε ότι τα Γραμμικά Προγράμματα που αντιστοιχούν στα προβλήματα βελτιστοποίησης P και D είναι δυϊκά μεταξύ τους. Ως εκ τούτου $P = D$.

(δ) Να δείξετε ότι:

$$\max_{j\in[m]} (A\boldsymbol{x})_j = \max_{\boldsymbol{y}\in\Delta^m} \boldsymbol{y}^T A\boldsymbol{x}\,,\ \text{και άρα}\ \min_{\boldsymbol{x}\in\Delta^n} \max_{j\in[m]} (A\boldsymbol{x})_j = \min_{\boldsymbol{x}\in\Delta^n} \max_{\boldsymbol{y}\in\Delta^m} \boldsymbol{y}^T A\boldsymbol{x}\,,$$

και αντίστοιχα:

$$\max_{\boldsymbol{y}\in\Delta^m} \min_{i\in[n]} (A^T\boldsymbol{y})_i = \max_{\boldsymbol{y}\in\Delta^m} \min_{\boldsymbol{x}\in\Delta^n} \boldsymbol{x}^T A^T \boldsymbol{y}$$

Ως συμπέρασμα, καταλήγουμε πως:

$$\min_{\boldsymbol{x}\in\Delta^n} \max_{\boldsymbol{y}\in\Delta^m} \boldsymbol{y}^T A\boldsymbol{x} = \max_{\boldsymbol{y}\in\Delta^m} \min_{\boldsymbol{x}\in\Delta^n} \boldsymbol{y}^T A\boldsymbol{x}\,,$$

που είναι γνωστό ως *Minimax Θεώρημα του von Neumann*.

Άσκηση 4

Θεωρούμε πρόβλημα εξισορρόπησης φορτίου (load balancing) με m παράλληλες μηχανές και n εργασίες. Κάθε εργασία $i \in [n]$ χαρακτηρίζεται από m-διάνυσμα $\boldsymbol{p}_i = (p_{i1}, \dots, p_{im})$, όπου $p_{ij} \in \mathbb{N}$

είναι το φορτίο που επιφέρει η ανάθεση της εργασίας i στη μηχανή j. Το μέγιστο φορτίο $L^{\max}(\varphi)$ μιας ανάθεσης $\varphi : [n] \to [m]$ εργασιών σε μηχανές είναι $L^{\max}(\varphi) = \max_{j\in[m]} L^j(\varphi)$, όπου

$$L^j(\varphi) = \sum_{i:\varphi(i)=j} p_{ij}$$

είναι το συνολικό φορτίο της μηχανής j στην ανάθεση φ.

1. Να διατυπώσετε το πρόβλημα του υπολογισμού μιας ανάθεσης φ^* με ελάχιστο $L^{\max}(\varphi^*)$ ως πρόβλημα Ακέραιου Γραμμικού Προγραμματισμού. Να δώσετε παράδειγμα όπου η βέλτιστη λύση του Ακέραιου Γραμμικού Προγράμματος μπορεί να είναι m φορές μεγαλύτερη της βέλτιστης λύσης του Γραμμικού Προγράμματος που προκύπτει ως "χαλάρωση" του Ακέραιου Προγράμματος, αν αντικαταστήσουμε τους περιορισμούς ακεραιότητας με περιορισμούς μη-αρνητικότητας για τις μεταβλητές που περιγράφουν τη βέλτιστη ανάθεση.

2. Έστω L^* το ελάχιστο μέγιστο φορτίο μιας μηχανής στη βέλτιστη κλασματική ανάθεση. Θεωρούμε το Γραμμικό Πρόγραμμα που υπολογίζει μια κλασματική ανάθεση των εργασιών στις μηχανές ώστε κάθε μηχανή να έχει συνολικό φορτίο (στην κλασματική ανάθεση) το πολύ L^*. Γνωρίζοντας εκ των προτέρων το L^*, αποκλείουμε την ανάθεση εργασιών i σε μηχανές j όταν $p_{ij} > L^*$, ώστε να αποφύγουμε το παράδειγμα του (1) (αυτή η τεχνική είναι γνωστή ως *parameter pruning*). Να δείξετε ότι (i) το L^* μπορεί να υπολογιστεί με δυαδική αναζήτηση, και (ii) ότι υπάρχει μια εφικτή λύση στο Γραμμικό Πρόγραμμα που διατυπώνεται με βάση το L^* όπου το πολύ $n + m$ μεταβλητές έχουν θετική τιμή.

Άσκηση 5

Θεωρούμε τον παρακάτω αλγόριθμο προσέγγισης για το πρόβλημα Vertex Cover σε ένα γράφημα $G(V, E)$, όπου κάθε κορυφή $v \in V$ έχει βάρος $w(v) > 0$. Το ζητούμενο είναι να υπολογίσουμε ένα σύνολο κορυφών $C \subseteq V$, με ελάχιστο συνολικό βάρος, ώστε κάθε ακμή $e \in E$ να έχει τουλάχιστον ένα άκρο της στο C.

Να δείξετε ότι ο αλγόριθμος επιτυγχάνει λόγο προσέγγισης 2. Ειδικότερα, να δείξετε ότι για τα C και $c(e)$, $e \in E$, στο τέλος του αλγορίθμου, ισχύουν τα εξής:

1. Το C καλύπτει όλες τις ακμές.

2. Το συνολικό βάρος του C είναι μικρότερο ή ίσο του $2\sum_{e\in E} c(e)$.

3. Το συνολικό βάρος της βέλτιστης λύσης είναι μεγαλύτερο ή ίσο του $\sum_{e\in E} c(e)$.
   Υπόδειξη: Χρησιμοποιήστε LP duality.

Να βρείτε ακόμη ένα (κατά το δυνατόν απλούστερο) γράφημα όπου ο αλγόριθμος υπολογίζει ένα Vertex Cover με συνολικό βάρος διπλάσιο από αυτό της βέλτιστης λύσης.

WeightedVertexCover($G(V, E)$, w)
    $C \leftarrow \emptyset$;
    $\forall v \in V$, $t(v) \leftarrow w(v)$;
    $\forall e \in E$, $c(e) \leftarrow 0$;
    while C δεν είναι vertex cover **do**
        $e = \{u, v\}$ μια ακάλυπτη ακμή;
        $\delta \leftarrow \min\{t(u), t(v)\}$;
        $t(u) \leftarrow t(u) - \delta$;
        $t(v) \leftarrow t(v) - \delta$;
        $c(e) \leftarrow \delta$;
        $C \leftarrow C \cup \{z \in \{u, v\} \mid t(z) = 0\}$;
    return(C);

Άσκηση 6

Αποδείξτε ότι ο λόγος προσέγγισης που επιτυγχάνει ο Greedy αλγόριθμος για το Weighted Set Cover είναι στην πραγματικότητα $H_{|S_{\max}|}$, όπου $S_{\max}$ το σύνολο εισόδου με τη μεγαλύτερη πληθικότητα.

Υπόδειξη: αποδείξτε ότι τα στοιχεία οποιουδήποτε συνόλου $S \in \mathcal{S}$ καλύπτονται από τον Greedy με συνολικό κόστος το πολύ $H_{|S|} \cdot cost(S)$, δηλαδή $\sum_{e \in S} price(e) \leq H_{|S|} \cdot cost(S)$.

Άσκηση 7

(α) Σχεδιάστε απλό f–προσεγγιστικό αλγόριθμο για το πρόβλημα Cardinality Set Cover, όπου f είναι το μέγιστο πλήθος συνόλων στα οποία μπορεί να ανήκει κάποιο στοιχείο. Αποδείξτε την ορθότητά του και τον λόγο προσέγγισης που επιτυγχάνει. Βρείτε tight example για τον λόγο προσέγγισης του αλγορίθμου.

Υπόδειξη: παρατηρήστε ποιο είναι το f στο πρόβλημα (Cardinality) Vertex Cover *και γενικεύστε κατάλληλα.*

(β) Σχεδιάστε f–προσεγγιστικό αλγόριθμο για το πρόβλημα Weighted Set Cover. Αποδείξτε την ορθότητά του και τον λόγο προσέγγισης που επιτυγχάνει. Βρείτε tight example για τον λόγο προσέγγισης του αλγορίθμου.

Υπόδειξη: γενικεύστε την ιδέα του degree-weighted αλγορίθμου για το Weighted Vertex Cover. *Εναλλακτικά, μπορείτε να αποδείξετε τον λόγο προσέγγισης προσαρμόζοντας την τεχνική της άσκησης 5.*

Άσκηση 8

(α) Συμπληρώστε την απόδειξη που θα βρείτε στις διαφάνειες για τον λόγο προσέγγισης $5/3$ για το πρόβλημα Metric TSP$_{(s,t)-path}$. Εξηγήστε τον ρόλο του όρου $c_{s,t}$ στην ανάλυση καθενός από τους δύο επιμέρους αλγορίθμους.

(β) Δώστε tight example για τους επιμέρους αλγορίθμους, καθώς και για τον συνολικό αλγόριθμο.

Άσκηση 9

Θεωρήστε τον εξής αλγόριθμο για το πρόβλημα Knapsack: τα στοιχεία ταξινομούνται σε φθίνουσα σειρά $p(i)/w(i)$ και εισάγονται με αυτή τη σειρά στο σακίδιο. Όποια στοιχεία δεν χωρούν απορρίπτονται.

(α) Δείξτε ότι ο λόγος προσέγγισης αυτού του αλγορίθμου δεν φράσσεται από καμμία σταθερά.

(β) Δείξτε ότι με την εξής απλή τροποποίηση ο αλγόριθμος γίνεται $(1/2)$-προσεγγιστικός: επιλέξτε την καλύτερη λύση μεταξύ της παραπάνω διαδικασίας και του να πάρει κανείς μόνο το στοιχείο με τη μεγαλύτερη αξία.

(γ) Γενικεύστε την ιδέα του (β) ώστε να πάρετε ένα PTAS για το πρόβλημα.