



 ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
 ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
 ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνητή Νοημοσύνη

Εισαγωγή

Γιώργος Στάμου





Τεχνητή Νοημοσύνη

2

Διδάσκοντες



Γιώργος
Στάμου



Αλέξανδρος
Χορταράς



Αντώνης
Κορκοφίγκας



Αλέξης
Μανδαλίός



Έντι
Ντερβάκος



Γιώργος
Φιλανδριάνος

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΚΑΙ ΜΑΘΗΣΗΣ
ails.ece.ntua.gr


 ARTIFICIAL INTELLIGENCE & LEARNING SYSTEMS LABORATORY




Τεχνητή Νοημοσύνη

3

Υλικό μαθήματος

- Ιστότοπος ([mycourses](#))
- Mailing list (ail_course@ails.ece.ntua.gr)
- Βασικά συγγράμματα
- Handouts (~διαφάνειες διαλέξεων), σημειώσεις
- Βιβλιογραφικές αναφορές

Ώρες διαλέξεων

Πέμπτη 15:15-17:00, Αμφ.3 Ν. Κτ. ΣΗΜΜΥ
 Παρασκευή 12:45-14:30, Αμφ.3 Ν. Κτ. ΣΗΜΜΥ

Βαθμολόγηση

3 σειρές ασκήσεων (1.2 μονάδες)
 Εργαστηριακά θέματα, παρουσιάσεις (1.8 μονάδες)
 Εξετάσεις (8 μονάδες)



Τεχνητή Νοημοσύνη - Αντικείμενο

4

Μελέτη διαδικασίας σκέψης και συλλογισμού ή της ευφυούς συμπεριφοράς ή κατασκευή μηχανών που αντιμετωπίζουν προβλήματα που προς το παρόν τα επιλύουν καλύτερα οι άνθρωποι

Στόχος της ΤΝ είναι η ανάπτυξη συστημάτων που:

- Σκέφτονται όπως οι άνθρωποι
- Συμπεριφέρονται όπως οι άνθρωποι
- Σκέφτονται λογικά
- Αντιδρούν λογικά



Τεχνητή Νοημοσύνη - Εφαρμογές

5

- Παίγνια
- Απόδειξη θεωρημάτων
- Δημιουργικά συστήματα
- Αυτόνομα robot
- Σχεδιασμός ενεργειών και χρονοπρογραμματισμός
- Ευφυείς υπηρεσίες διαδικτύου
- Ιατρική, νομικές επιστήμες, βιοτεχνολογία, γεωπονία, διαστημική τεχνολογία, αρχαιολογία, ...
- Όραση υπολογιστών, επεξεργασία φυσικής γλώσσας, ανάλυση δεδομένων, εξόρυξη γνώσης, ...



Τεχνητή Νοημοσύνη - Αξιολόγηση

6

Δοκιμασία Turing

Ο υπολογιστής περνά τη δοκιμασία αν ένας άνθρωπος εξεταστής, αφού αλληλεπιδράσει με το σύστημα, δεν μπορεί να συμπεράνει αν οι αποκρίσεις προέρχονται από άνθρωπο ή από μηχανή

Ικανότητες μηχανών για να περάσουν τη δοκιμασία

- ▶ Αντίληψη (επεξεργασία σημάτων από αισθητήρες, αλληλεπίδραση με άνθρωπο,...)
- ▶ Γνώση, συλλογισμός, διορατικότητα, προσαρμοστικότητα, δημιουργικότητα,...

Τεχνητή Νοημοσύνη - Syllabus



Ενότητα 1. Αλγόριθμοι αναζήτησης (3 διαλέξεις, ασκήσεις)

- ▶ **Περιγραφή Προβλημάτων** σε Χώρο Καταστάσεων, Αλγόριθμοι Αναζήτησης Λύσης
- ▶ **Αλγόριθμοι Τυφλής Αναζήτησης** (Αναζήτηση Πρώτα σε Βάθος, Αναζήτηση Πρώτα σε Πλάτος, Αναζήτηση Επαναληπτικής Εκβάθυνσης, Αναζήτησης Διπλής Κατεύθυνσης)
- ▶ **Αλγόριθμοι Πληροφοριμένης Αναζήτησης** (Ευριστικοί Μηχανισμοί, Αλγόριθμοι Αναρρίχησης Λόφων, Προσομοιόμενη Ανόπτηση, Αναζήτηση με Απαγορευμένες Καταστάσεις, Αναζήτηση Πρώτα στο Καλύτερο, Αλγόριθμος A*)
- ▶ **Αλγόριθμοι για παιίνια** (Αλγόριθμος Minimax, Αλγόριθμος AlphaBeta)

Τεχνητή Νοημοσύνη - Syllabus



Ενότητα 2. TN και Λογική (6 διαλέξεις, ασκήσεις, εργαστήριο)

- ▶ **Προτασιακή Λογική** (Σύνταξη, Σημασιολογία, Ερμηνεία, Ικανοποιημότητα, Μοντέλα Γνώσης, Καπτηγορηματική και Υπονοούμενη Γνώση, Λογική Συμπερασματολογία, Αλγόριθμοι Συλλογιστικής, SAT solvers, Resolution)
- ▶ **Λογική Πρώτης Τάξης** (Διαχείριση Μεταβλητών και Ποσοσδεικτών, Αλγόριθμος Ανάλυσης για Λογική Πρώτης Τάξης, Δισεπίλυσμότητα)
- ▶ **Λογικός Προγραμματισμός** (Λογική Horn, Ανάλυση SLD, η Γλώσσα PROLOG, Γεγονότα και Κανόνες, Στρατηγικές Αναζήτησης, Άρνηση σαν Αποτυχία, Συστήματα Παραγωγής, Έμπειρα Συστήματα)
- ▶ **Δομημένες Αναπαραστάσεις Γνώσης** (Αντικείμενα, Πλαίσια, Σημασιολογικά Δίκτυα, Γράφοι Γνώσης, Περιγραφικές Λογικές)

Τεχνητή Νοημοσύνη - Syllabus



Ενότητα 3. Αβεβαιότητα (3 διαλέξεις, ασκήσεις)

- ▶ **Πιθανοτικές Προσεγγίσεις** (Θεωρία Πιθανοτήτων, Κανόνας του Bayes, Ταξινομής Bayes, Δίκτυα Πίστης, Ευλογοφάνεια, Θεωρία Dempster-Shafer)
- ▶ **Ασαφής Λογική** (Ασαφή Σύνολα, Συναρτήσεις Συμμετοχής, Ασαφείς Διαμερίσεις, Ασαφή Συστήματα, Ασαφής Συσταδοποίηση)

Ενότητα 4. Μηχανική μάθηση (5 διαλέξεις, ασκήσεις, εργαστήριο)

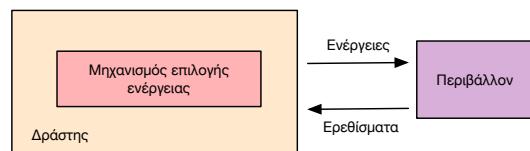
- ▶ **Προσαρμοζόμενοι Δράστες** (Προβλήματα Μάθησης, Ορολογία, Σενάρια Μάθησης, PAC learning)
- ▶ **Προβλήματα και Μοντέλα Μάθησης** (Γραμμική Ταξινόμιση, Percetrons, Βέλτιστα Υπερεπίπεδα, Μηχανές Διανυσμάτων Υποστήριξης (SVM), Μη-Γραμμική Διαχωρισμότητα, Συναρτήσεις Πυρήνων, Νευρωνικά Δίκτυα Πολλών Επιπέδων, Βαθιά Μάθηση)

Εννοιολογικό πλαίσιο



TN και ανάπτυξη ευφυών δραστών

Ορθολογικός δράστης



Ορθολογικός δράστης - Λειτουργία



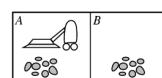
Πρόβλημα

Φτάνεις σε ένα εστιατόριο για φαγητό.
Αποφάσισε αν θα περιμένεις για τραπέζι ή θα φύγεις.

Παράγοντες που επηρεάζουν την απόφαση

Εναλλακτικές: υπάρχει κάποιο εστιατόριο κοντά;
Ποιότητα αναμονής: υπάρχει άνετος χώρος αναμονής (π.χ. μπαρ);
Ημέρα: Είναι σήμερα ΣΚ ή κάποια γιορτή;
Πείνα: Είσαι πολύ πεινασμένος;
Πληρότητα: Πόσοι τρώνε στο εστιατόριο (κανείς, αρκετοί, είναι γεμάτο);
Τιμή: πόσο ακριβό είναι το εστιατόριο (φτηνό, μέσο, ακριβό);
Καιρικές συνθήκες: Βρέχει έξω;
Κράτηση: Έχεις κάνει κράτηση;
Είδος εστιατορίου: Ινδικό, πιτσαρία, γκουρμέ;
Εκτίμηση αναμονής: εκτίμηση σε λεπτά (0-10, 10-30, 30-60, >60);

Παράδειγμα ορθολογικού δράστη



Αντιλήψεις: Θέση και περιεχόμενο π.χ. [Α, Σκονισμένο]

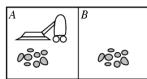
Ενέργειες: Αριστερά, Δεξιά, Αναρρόφηση

Παράδειγμα ορθολογικού δράστη



13

Ακολουθία αντιλήψεων	Ενέργεια
[A, Καθαρό]	Δεξιά
[A, Σκονισμένο]	Αναρρόφηση
[B, Καθαρό]	Αριστερά
[B, Σκονισμένο]	Αναρρόφηση
[A, Καθαρό], [A, Καθαρό]	Δεξιά
[A, Καθαρό], [A, Σκονισμένο]	Αναρρόφηση
[A, Καθαρό], [A, Καθαρό], [A, Καθαρό]	Δεξιά
[A, Καθαρό], [A, Καθαρό], [A, Σκονισμένο]	Αναρρόφηση



Ιδανικοί ορθολογικοί δράστες



14

Ορθολογικός δράστης = Αρχιτεκτονική + Πρόγραμμα

Αρχιτεκτονική

- ▶ Επιτρέπει τα ερεθίσματα να διοχετεύονται στο Πρόγραμμα
- ▶ Εκτελεί το Πρόγραμμα
- ▶ Μετατρέπει τα αποτελέσματα του Προγράμματος σε μηχανισμούς δράσης

Πρόγραμμα

- ▶ Ορίζει την απεικόνιση [ακολουθία ερεθισμάτων -> ενέργεια]

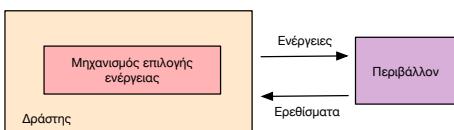
Ιδανικός Ορθολογικός Δράστης (Ideal Rational Agent)

- ▶ Ο δράστης ο οποίος για κάθε πιθανή κατάσταση και ερεθίσματα έχει τη δυνατότητα και τη γνώση να ενεργήσει ώστε να μεγιστοποιήσει την επίδοσή του

Πλαίσιο δράσης



15



Περιβάλλον, Αισθητήρες, Δράσεις, Επίδοση

Αφαίρεση

Καθορισμός σημαντικών στοιχείων



Τυπικός ορισμός προβλήματος

Πλαίσιο δράσης - Παράδειγμα



16

Αυτόματος οδηγός ταξί

[Περιβάλλον, Αισθητήρες, Δράσεις, Επίδοση]

- ▶ Περιβάλλον: Δρόμοι, αυτοκίνητα, πεζοί, πελάτες
- ▶ Αισθητήρες: Κάμερες, αισθητήρες υπερήχων, κοντέρ, GPS, οδόμετρο, αισθητήρες λειτουργίας αυτοκινήτου, εγκέφαλος αυτοκινήτου
- ▶ Δράσεις: Γκάζι, φρένο, τιμόνι, φώτα, κόρνα
- ▶ Δείκτες Επίδοσης: Ασφαλής, γρήγορη, άνετη, χωρίς παραβάσεις μεταφορά

Πλαίσιο δράσης - Παράδειγμα



17

Σύστημα υποστήριξης ιατρικής διάγνωσης

- ▶ Περιβάλλον: Ασθενής, ιατροί, πληροφοριακό σύστημα νοσοκομείου
- ▶ Αισθητήρες: Πληκτρολόγιο (εισαγωγή δεδομένων που αφορούν τον ασθενή), βάσεις ιατρικών δεδομένων
- ▶ Δράσεις: Διάλογοι με το ιατρικό προσωπικό, αυτόματη ενημέρωση βάσεων δεδομένων
- ▶ Δείκτες Επίδοσης: Ανάρρωση ασθενούς, Ελαχιστοποίηση κόστους, εξαγωγή γνώσης

Κατηγοριοποίηση Δραστών



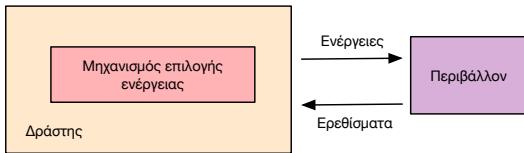
18

Κατηγοριοποίηση ανάλογα με τη δομή

- ▶ Δράστες βασισμένοι σε πίνακες (Table Driven Agents)
- ▶ Απλοί ανακλαστικοί δράστες (Simple Reflex Agents)
- ▶ Ανακλαστικοί δράστες με μοντέλο (Reflex Agents with Internal State)
- ▶ Δράστες επιδιώκησης στόχου (Goal-based Agents)
- ▶ Δράστες βασισμένοι στη χρησιμότητα (Utility-based Agents)
- ▶ Δράστες με μάθηση (Learning-based agents)

Δράστες βασισμένοι σε πίνακα

19



Μεθοδολογία

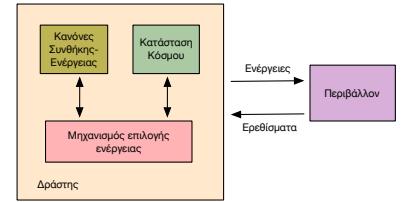
- Χρήση πινάκων για αντιστοίχιση ερεθισμάτων που λαμβάνει ο δράστης από το περιβάλλον με την κατάσταση του δράστη με τις ενέργειες που πρέπει να εκτελεσθούν

Απλοί ανακλαστικοί δράστες

20

Μεθοδολογία

- Αναγνώριση συχνών και επαναλαμβανόμενων σχέσεων μεταξύ ερεθισμάτων και ενέργειών τους
- Μοντελοίση των σχέσεων με τη μορφή κανόνων συνθήκης-ενέργειας (condition-action rules)
- Επιλογή του κατάλληλου κανόνου, και στη συνέχεια εφαρμογή της αντίστοιχης ενέργειας



```

Function Simple-Reflex-Agent(percept)
static: rules, /*condition-action rules*/
state ← Intercept_input(percept)
rule ← Rule_match(state, rules)
action ← Rule_Action(rule)
return(action)

```

Δράστες με μοντέλο

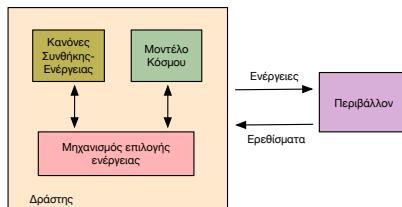
21

Παρατήρηση

Υπάρχουν μηχανισμοί καταστάσεων του περιβάλλοντος που μπορεί να παράγουν τα ίδια ερεθίσματα

Μεθοδολογία

- Διατήρηση ενός μοντέλου εσωτερικής κατάστασης που επιτρέπει την διαφοροποίηση των περιπτώσεων και την επιλογή των δράσεων



```

Function Reflex-Agent-With-State(percept)
static: state, /*description of the current state*/
rules /* set of condition-action rules */
state ← Update_State(state, percept)
rule ← Rule_Match(state, rules)
action ← Rule_Action(rule)
state ← Update_State(state, action)
return(action)

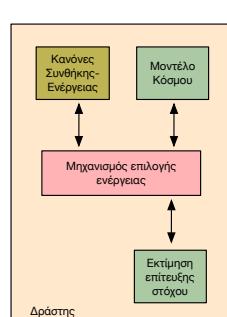
```

Δράστες επιδίωξης στόχου

22

Παρατήρηση

Ο δράστης έχει στόχους που πρέπει να επιτύχει (όχι απλά να εκτελέσει μια ενέργεια)



Μεθοδολογία

- Η γνώση των στόχων συνδυάζεται με όλες τις άλλες πληροφορίες ώστε να επιλεγεί η κατάλληλη ενέργεια
- Η επιλογή είναι αποτέλεσμα μιας διαδικασίας έρευνας (search) και αλγόριθμων σχεδιασμού ενεργειών (planning)

Δράστης επιδίωξης στόχου

23

```

Function Goal-Based-Agent(percept)
static: state, /* description of the current world state */
rules /* set of condition-action rules */
goal /* set of specific success states */
state ← Update_State(state, percept)
rule ← Rule_Match(state, rules)
action ← Rule_Action(rule)
state ← Update_State(state, action)
if (state in goal) then
    return (action)
else
    percept ← Obtain_Percept(state, goal)
    return(Goal-Based-Agent(percept))

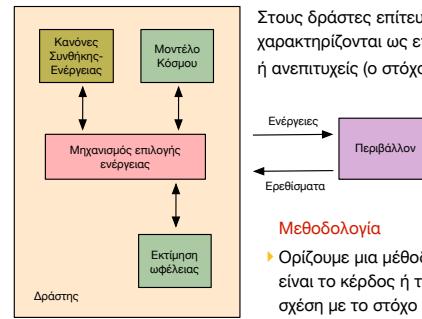
```

Δράστης αφέλειας

24

Παρατήρηση

Στους δράστες επίτευξης στόχου οι καταστάσεις χαρακτηρίζονται ως επιτυχείς (τελική επίτευξη στόχου) ή ανεπιτυχείς (ο στόχος δεν έχει ακόμη επιτευχθεί)



Μεθοδολογία

- Ορίζουμε μια μέθοδο για να διακρίνουμε πόσο είναι το κέρδος ή το σκορ κάθε κατάστασης σε σχέση με το στόχο
- Χρησιμοποιούμε μία συνάρτηση που ονομάζουμε συνάρτηση αφέλειας (utility function)
- Αξιολογούμε με αυτή τις εναλλακτικές ενέργειες

Δράστης αφέλειας

25

```
Function Goal-Based-Agent(percept)
    static: state, /* description of the current world state */
            rules /* set of condition-action rules */
            goal /* set of specific success states */
    state ← Update_State(state, percept)
    rule ← Rule_Match(state, rules)
    action ← Rule_Action(rule)
    state ← Update_State(state, action)
    score ← Obtain_Score(state)
    if (state in goal) and Best_Score(score) then
        return(action)
    else
        percept ← Obtain_Percept(state, goal)
        return(Goal-Based-Agent(percept))
```



Δράστης με Μάθηση

26

