

Istraživački projekti - HOW TO

Stefan Nožinić (stefan@lugons.org)

January 7, 2021

Uvod

- Cilj je učenje istraživačkog procesa kroz:
- Planiranje
- Implementaciju
- Evaluaciju
- Dokumentovanje
- Prezentaciju rezultata

Kako odabrati projekat?

- Prvi predlog nikada nije i ne treba da bude krajnji predlog
- Krajnji predlog mora biti formiran NAJKASNIJE do letnjeg seminara uz konsultaciju sa saradnicima i rukovodiocem
- Što se pre sastavi krajnji predlog - više vremena za implementaciju i dodatna ispitivanja
- Projekat za cilj mora imati istraživanje, ispitivanje nečega

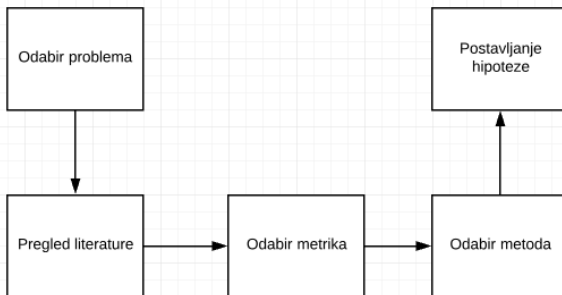
Naučna metodologija

- Postavljanje hipoteze - pretpostavke - zimski seminar
- Pravljenje plana implementacije i evaluacije - period posle zimskog a pre letnjeg
- Implementacija - letnji seminar
- Verifikacija implementacije - letnji seminar
- Prikupljanje rezultata - kraj letnjeg seminara
- Pisanje naučnog rada kao izveštaja o rezultatima istraživanja - poslednji dani letnjeg seminara i jesenji seminar

Smernice za postavljanje hipoteze

- Odabir problema koji želimo da rešimo
- Odabir skupa algoritama za koje želimo da uradimo evaluaciju (mogu biti i neki novi algoritmi)
- Odabir metrika uz pomoć kojih merimo uspešnost određenog algoritma za rešavanje problema
- Rezultati se onda dobijaju prostom komparacijom algoritama iz ovoga skupa
- Pretpostavka: neki podskup gornjeg skupa će pokazati bolje rezultate nego ostali po odabranim metrikama

Smernice za postavljanje hipoteze



Primeri

- Problem: prepoznavanje lica
- Metrike: preciznost, tačnost, odziv, brzina, količina potrebnog prostora
- Algoritmi: eigenface, neuronske mreže, linearna klasifikacija, ...
- Hipoteza: eigenface će pokazati najbolju preciznost, odziv, tačnost dok će linearna klasifikacija biti najbrža...

Primeri

- Problem: Raspoređivanje procesa na paralelnom procesoru
- Metrike: ukupno vreme kada procesor ne radi ništa, zauzeće memorije za meta-podatke, brzina, zavisnost potrebnih procesora od broja procesa
- Algoritmi: algoritmi za bojenje grafa...
- Hipoteza: Neki algoritam X će imati najmanje idle vreme dok će Y imati najveće idle vreme ali zauzima više prostora za njegov rad....

Primeri

- Problem: Implementacija sistema datoteka
- Metrike: Proširivost, minimalna veličina bloka, fragmentnost, broj skokova HDD-a...
- Algoritmi: EXT4, NTFS, neki custom file system...
- Hipoteza: "Moj custom file system ima najmanji broj skokova HDD-a a zadržava fragmentnost kao EXT4"

Primeri

- Problem: Numerička simulacija promene boje elektrohromatskih materijala
- Metrike: tačnost, stabilnost, brzina, ...
- Algoritmi: Ojlerov algoritam, Runga-Kuta algoritam, Ojlerov implicitni algoritam, imitacija procesa bez numeričkog simuliranja sistema...
- Hipoteza: Imitacija procesa pokazuje dovoljno realne rezultate za materijale X i Y za napon od U_1 do U_2 dok radi brže od numeričkih metoda...

Primeri

- Problem: Izrada jezika
- Metrike: prosečan broj linija za rešavanje nekog skupa problema, ekspresivnost
- Jezici: taj novi custom jezik, C, Haskell, LISP, Java, ...
- Hipoteza: Potrebno je najmanje linija koda u custom jeziku da se napravi X

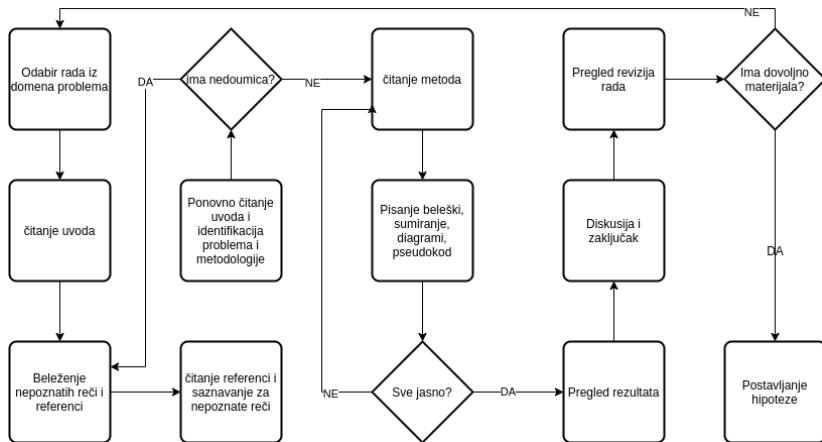
Referentna vrednost

- Moramo uvek porediti sa nekom referentnom vrednošću
- Komparacijom saznajemo da li je nešto bolje ili lošije
- Ako ne poredimo sa referentnom vrednošću, rad ne znači ništa!
- Zbog ovoga, skup algoritama (metoda, jezika, ...) mora uvek imati više od jednog elementa
- Rad u kom se ispostavi da je vaš algoritam lošiji od drugih je opet validan rad
- Bolje imati rad koji objašnjava zašto je nešto loše nego rad koji ne poredi i ne objašnjava ništa

Kako do ideje?

- Njutnova jabuka ni Njutnu nije pala na glavu pa neće ni nikom drugom!
- Pokušajte da uvidite sopstvene probleme
- čitanje tehničkih blogova, radova, arxiv, esveske.github.io, ...
- Ako radite na nekom projektu sada, mora da postoji prostora za inovaciju
- Beleženje ideja u svesku

Smernice za čitanje naučnih radova



Teorija

- Teorija je izuzetno važna da razumete oblast i problem kojim se bavite i da kasnije lakše možete da isplanirate implementaciju projekta
- Potrebno upoznavanje sa svom matematikom iza odabranih algoritama (konceptata)
- Primeri: statistika, teorija grafova, linearna algebra, ...
- Ako se bavite simulacijom: potrebno je upoznavanje sa datim modelima datog fenomena
- Ako se bavite izradom jezika: leksička analiza, konačni automati, parsiranje, AST, ...
- Poznavanje teorije pomaže da se bolje definišu metrike

Pretraga algoritama

- Velika je verovatnoća da je još neko pokušao da reši dati problem
- Pre inoviranja novog algoritma, pretražiti postojeće
- Google Scholar i literatura iz date oblasti