

Tehtävä 1. Koneoppimista (2 pistettä)

Lataa ja asenna Weka-ohjelmisto. Ohjeita aloittamiseen löytyy kurssin kotisivulta (suora linkki: www.cs.helsinki.fi/node/81056). Lataa myös tiedosto `spirals.arff` osoitteesta www.cs.helsinki.fi/webfm_send/1477 (linkki löytyy myös kurssin sivulta).

Opeta Weka-ohjelmistoa käyttäen monikerrosperspeptronin luokitteluun `spirals`-aineiston esimerkkejä oikeisiin luokkiin (tiedostossa muuttuja `Class`, jonka arvot ovat 'a' ja 'b') x - ja y -koordinaattien mukaan (tiedostossa muuttujat `A1` ja `A2`).

Huomaa, että perusasetuksilla luokitin tuottaa huonon luokittelutuloksen. Jos käytät testivirheen määrittämiseen esimerkiksi 'Percentage split % 66' -asetusta, luokittelutarkkuus on vain 52.9% (36 oikein, 32 väärin). Valitsemalla 'Visualize classifier errors' ja valitsemalla x -akselin muuttujaksi `A1` ja y -akselin muuttujaksi `A2` näet päätösrajapinnan, joka määrää luokituspäätökset. Huomaat, että luokitukset menevät päin mäntyä.

- a) (1 piste) Kokeile muuttaa monikerrosperspeptronin asetuksia, esimerkiksi piilokerrosten määrää (kenttä `hiddenLayers`, johon merkitään neuronien lukumäärät piilokerroksissa pilkuilla eroteltuna) ja opetusaskelten määrää (kenttä `trainingTime`). Saatko luokitteluvirheeksi alle 10%?
- b) (1 piste) Kokeile myös lähimmän naapurin luokitinta (`Lazy`→`IB1`) ja päätöspuuluokitinta (`Trees`→`J48`). Tutki niiden tuottamia päätösrajapintoja ja pohdi kunkin luokittimen kohdalla, miksi ne mahdollisesti toimivat tai eivät toimi hyvin.

Tehtävä 2. Hahmontunnistus (2 pistettä)

Lataa kurssin kotisivulta paketti `surf-package.zip`, pura se, siirry syntyneeseen hakemistoon ja suorita komento `ant example`. (Toimii ainakin laitoksen Linux-koneissa.) Tuloksena ruudulle pitäisi ilmestyä luennolla esitetty kasvontunnistusesimerkki. Kuvassa olevat punaiset täplät vastaavat löytyneitä piirteitä ja kuvien väliset vihreät ja siniset viivat vastaavat piirrepareja, jotka ovat niin samankaltaisia, että ne voidaan tulkita samoiksi hahmoiksi.

- a) (1 piste). Kokeile vaihtaa `example`-hakemistossa olevat tiedostot `img1.jpg` ja `img2.jpg` toisiin kuviin ja aja ohjelma uudelleen. Kuvien tulisi olla samankorkuiset.

Kokeile vähintään viittä muuta kuvaparia, joista osassa näkyy sama rakennus, sama henkilö ja/tai sama esine. Kokeile myös kuvapareja, jotka *eivät* esitä samaa kohdetta. Yritä löytää sekä esimerkkejä, joissa SURF-piirteisiin perustuva tunnistus toimii (eli samaa esittävistä kuvista löytyy useita yhteensopivia piirteitä) että esimerkkejä, joissa se ei toimi.¹

- b) (1 piste). Valitse yksi (tai useampi) kuva ja muokkaa sitä

- * säätämällä valoisuutta, kontrastia tai väritasapainoa,
- * kallistamalla kuvaa (rotaatio),
- * rajaamalla ja skaalaamalla uudelleen,
- * lisäämällä kohinaa (noise) tai sumentamalla (blur).

Voit käyttää haluamaasi kuvankäsittelyohjelmaa (iPhoto, GIMP, Photoshop, jne).

Toimiiko SURF, kun sovellat sitä alkuperäiseen ja muokattuun kuvaan?

Vaihtoehtoisesti (hauskempi tapa): Ota itse kameralla (tai puhelimella) kuvapareja, jotka esittävät samaa esinettä, rakennusta tai henkilöä, ja kokeile toimiiko SURF.

Pohdi millaiset muutokset kuvissa ja olosuhteissa aiheuttavat ongelmia SURF:iin perustuvalla kuvantunnistukselle ja millaiset taas eivät.

Tehtävä palautetaan kirjoittamalla lyhyt selostus siitä, mitä huomasit: selitä mielestäsi olennaiset huomiot, joita teit kokeista. Liitä selostukseen vähintään viisi kuvaparia. Palautettavan tiedoston muoto on oltava PDF. Käytä raportissa matalaresoluutioisia kuvia, jotta PDF-tiedoston koko pysyy alle 2 MB. Palautus sähköpostitse laskariohjaajalle osoitteeseen `etunimi.sukunimi@helsinki.fi`. Vaihtoehtoisesti voit tulostaa raportin ja toimittaa sen laskariohjaajalle tai luennoijalle (vaikka luennolla).

¹Lisävinkki: ohjelma näyttäisi (ehkä) toimivan joskus paremmin, jos `src/com/stromberglabs/jopensurf/SurfCompare.java`-tiedoston riville 66 vaihtaa totuusarvoksi `private boolean mUpright = true`.

Tehtävä 3. Luonnollisen kielen käsittely (2 pistettä)

Tässä tehtävässä tutustutaan luonnollisen kielen käsittelyyn OpenNLP -kirjaston avulla. Lataa kurssisivulta valmis Java-pohja tehtävää varten. Pohja on Maven-yhteensopiva, joten voit käyttää sitä haluamassasi kehitysympäristössä.

- a) (1p) Simuloi ensin kynällä ja paperilla CYK-algoritmia lauseeseen “Google bought DeepMind for \$500M in January”, kun käytössä on seuraavat kielioppisäännöt:

$S \rightarrow NP VP$	$PP \rightarrow P NP$	$N \rightarrow \text{January}$
$NP \rightarrow NP PP$	$NP \rightarrow N$	$V \rightarrow \text{bought}$
$VP \rightarrow VP PP$	$N \rightarrow \text{Google}$	$P \rightarrow \text{for}$
$VP \rightarrow V NP$	$N \rightarrow \text{DeepMind}$	$P \rightarrow \text{in}$
$VP \rightarrow V$	$N \rightarrow \$500M$	

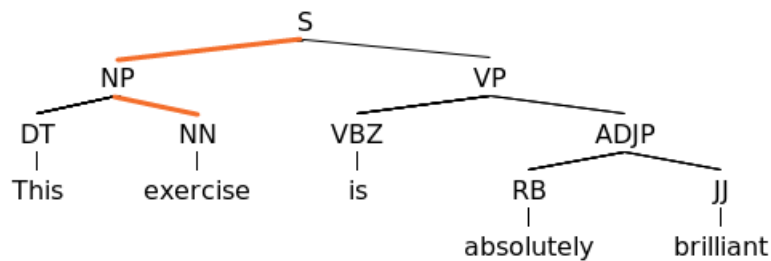
Oheassa CYK-algoritmissa laskettava taulukko, johon on valmiiksi täytetty kaksi alinta riviä. Täytä itse ruudut, joissa on '·'.

·						
·	·					
·	·	·				
·	·	·	·			
·	·	·	·	·		
S	VP		PP		PP	
N,NP	V,VP	N,NP	P	N,NP	P	N,NP
Google	bought	DeepMind	for	\$500M	in	January

Esimerkiksi vasemmanpuoleisen sarakkeen ensimmäinen täyttämättömään ruutuun (1,3) saadaan S soveltamalla sääntöä $S \rightarrow NP VP$ ruutuihin (1,1) ja (2,3).

Kun olet täyttänyt taulukon, rakenna lauseen kaikki jäsenyspuut laajentamalla juurisolmun S.

- b) (1p) Täydennä tiedoston *Extractor.java* metodi *extractSubject* seuraavasti: Annettuna on lauseen jäsennyspuu. Tehtävänäsi on yrittää etsiä lauseen subjekti eli tekijä. Eräs heuristinen menetelmä (ei toimi aina) löytää subjekti on seuraava: oletetaan, että lauseella on jäsennyspuu, jonka juurella S on ainakin lapset NP (*noun phrase*) ja VP (*verb phrase*). Valitaan juuren lapsi NP (voit olettaa, että se on aina jäsennyspuussa). Nyt subjekti on ensimmäinen substantiivi, joka löytyy tästä alipuusta (ks. alla oleva kuva). Substantiiveja ovat sanat, joiden POS-tag on NN, NNP, NNPS tai NNS. Toteuta haku **leveyshaulla**. Voit testata metodisi toimivuutta tehtäväpohjasta löytyvillä valmiilla testeillä.



Kuva 1: Lauseen “*This exercise is absolutely brilliant*” jäsennyspuu

Toteuta seuraavaksi tiedoston *Main.java* main-metodiin seuraava logiikka hyödyntäen luokasta *NLPUtils* löytyviä metodeja: pohjassa tunnistetaan kaikki Franz Kafkan kirjan *The Metamorphosis* lauseet. Hylkää kaikki lauseet, joissa ei esiinny sanaa “Gregor” (vihje: *String*-olion *contains*-metodi). Jokaiselle muulle lauseelle, tuota lauseen jäsennyspuu (ohita lause, jos jäsentäminen ei onnistu) ja käytä aiemmin tekemääsi metodia lauseen subjektin selvittämiseen. Tulosta kaikki lauseet, joiden subjekti on “Gregor”. Tarkastele lauseita ja selvitä niiden avulla, mitä Gregor tekee.²

²vrt. <http://hint.fm/seer/#left=google%20is&right=artificial%20intelligence%20is>