Aalto-yliopisto
Perustieteiden
korkeakoulu

Koneoppimismallin hyödyntäminen junan vikaantumisen arvioinnissa

Ville Tuominen

4.12.2019

Ohjaaja: *TkT Otto Sormunen*

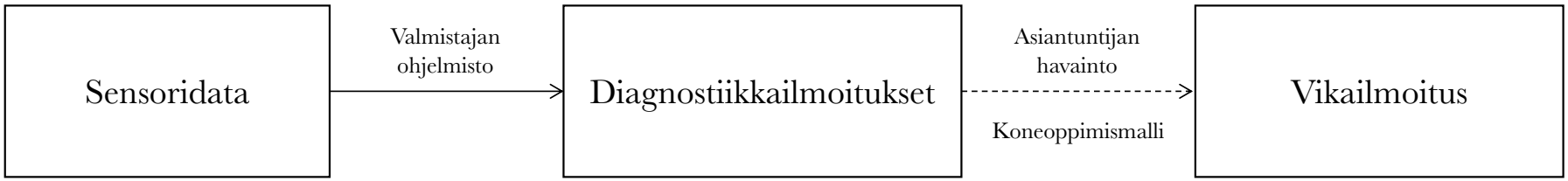
Valvoja: *Prof. Antti Punkka*

Systeemianalyysin
laboratorio

Tavoitteet

- Tavoitteena luoda koneoppimismalli, jolla voidaan luokitella junan lähettämät diagnostiikkaviestit päivittäisellä tarkkuudella sen mukaan, onko junassa todellinen vikatilanne vai ei
- Jotta ennuste olisi käyttökelpoinen vikojen toteamisen kannalta, ilmoituksia halutaan luokitella mahdollisimman vähän väärin oikeiksi vioiksi.

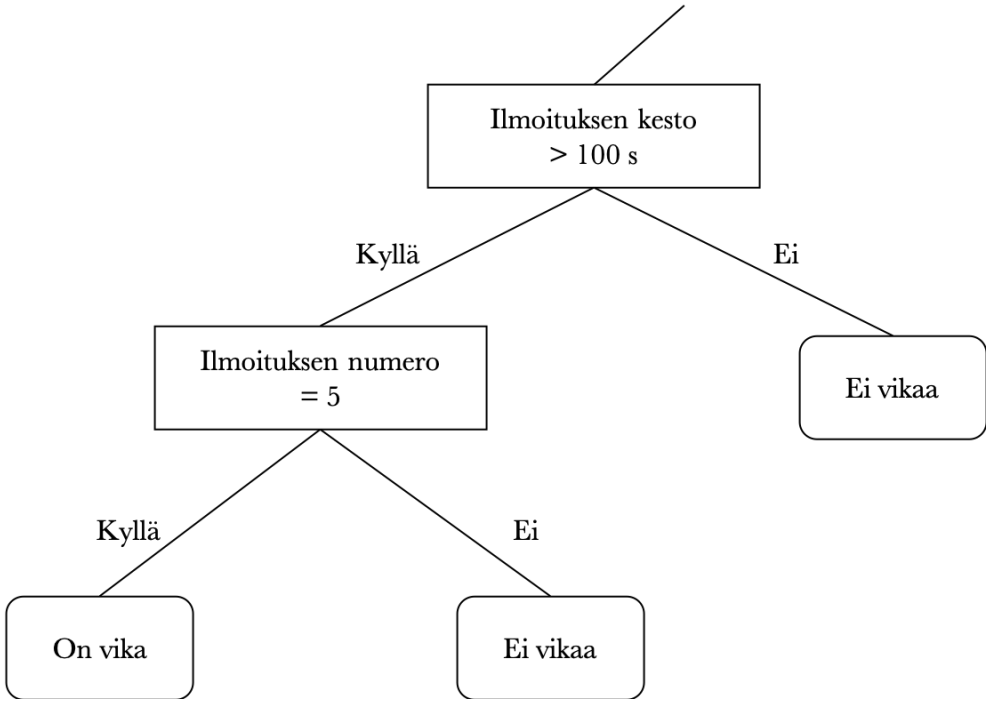
Taustaa



- **Diagnostiikka:** junan lähettämät diagnostiikkailmoitukset (kesto, alkamisaika, komponentti)
- **Vikailmoitukset:** ihmisten kirjaamat havainnot vioista (havaintoaika, vapaa kuvaus, korjauksen tiedot)
- Kaikkien diagnostiikkaviestien syntyperusteita ei tiedetä tarkalleen
  - Läheskään kaikista ilmoituksista ei seuraa suoraan vikaa
  - Ei pääsyä sensoridataan
- Vikailmoitusdata, johon junan diagnostiikkailmoituksia verrataan, on osittain puutteellista
  - Kaikista vikaantumistilanteista ei ole vikailmoituksia tai ilmoitus on tehty paljon myöhemmin

Päätöspuu

- Luokitteluun ja regressioon sopiva koneoppimismalli
- Lopputuloksena yksinkertaisista ehdoista koostuva binääripuu
- Täydellisen puun luonti NP-täydellinen ongelma
- Ongelmana ylisovittaminen koulutusdataan



Päätöspuu

- Päätöspuun luontiin eri heuristisia algoritmeja, esim. C4.5 ja CART
- CART-algoritmi rekursiivisesti (Steinberg, 2009):

  1. Jaetaan data kahteen osaan jakokriteerin perusteella (esim. Ginin epäpuhtausmitta) datan parhaiten erottelevan muuttujan kautta.

  2. Jatketaan jakoa kunnes data loppuu tai kaikki jäljellä olevat alkiot kuuluvat samaan luokkaan.

Ginin epäpuhtausmitta solmussa *t*:

$$i(t) = 1 - \sum_{j=1}^{J} p_t^2(j)$$

$p_t(j)$ on luokan j osuus solmussa t

Epäpuhtauden pieneneminen jaolle *s*:

$$\Delta i(s,t) = i(t) - p_{tL} i(t_L) - p_{tR} i(t_R)$$

$i(t_L)$ ja $i(t_R)$ alisomujen epäpuhtausmitta

Haetaan jako *s*, jonka epäpuhtauden pieneneminen on suurin

Päätöspuu

- CART-Algoritmi (Steinberg, 2009):

  3. Yksinkertaistetaan valmista puuta karsimalla solmuja (esim. hinta-kompleksisuuskarsiminen), jolloin vähiten merkitsevät jakokohdat poistetaan. Tallennetaan karsitut puut.

  4. Valitaan karsituista puista pienikokoisin puu, jonka luokitteluvirhe on pienin.

Hinta-kompleksisuus: $$Ra(T) = R(T) + a\|T\|$$

$R(t)$ mallin takaisinsijoitusvirhe koulutusdataan
$|T|$ päätesolmujen määrä
a puun koko rajoittava sakko

Satunnaismetsä

- Luodaan useita, esimerkiksi satoja, päätöspuita (Breiman, 2001):
  - Valitaan lähtödatasta satunnaisesti joukko, johon sovitetaan päätöspuu.
  - Jakokohdissa tarkasteltavat muuttujat valitaan satunnaisesti kaikkien muuttujien joukosta. Jaoista valitaan paras epäpuhtausmitan avulla.
  - Valmista puuta ei karsita.
- Luokittelussa satunnaismetsän lopullinen ennuste saadaan joko puiden enemmistön tai keskiarvon perusteella
- Puiden määrän kasvaessa yleistysvirhe suppenee
  - Ylisovittaminen katoaa käytännössä kokonaan

Tulokset

- Parhaimmat tulokset saatiin, kun tarkastellaan mahdollisimman tarkkaan rajattua tilannetta (yhtä komponenttia)
  - Vaatii asiantuntemusta sekä diagnostiikka- että vikailmoitusten suodattamiseksi
- Komponenttien käyttömääristä tai junan ajomäärästä ei ollut hyötyä

- Testidata 2kk, koulutusdata 7kk
- Diagnostiikkailmoituksia muutamia tuhansia per komponentti
- Pythonin scikit-paketin RandomForestClassifier

Tulokset

- A:lle ja C:lle melko hyvä tulos
- B on kokonaisuus useita komponentteja

- Esitettynä paras ennuste, joka saatu käyttämällä tiettyä päätöspuiden määrää (<10) satunnaismetsässä
- Suurella puiden määrällä nämä tulokset ovat hieman huonompia, mutta tällöin mallin pitäisi yleistyä kaikille testidatoille selvästi paremmin

Komponentti A		
Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**100% (90%)**	83% (10%)
On vika	0% (0%)	**17% (100%)**

Komponentti B		
Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**99% (88%)**	91% (12%)
On vika	1% (55%)	**9% (45%)**

Komponentti C		
Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**100% (89%)**	82% (11%)
On vika	0% (0%)	**18% (100%)**

Tulokset

Puiden määrä < 10

Komponentti A

Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**100% (90%)**	83% (10%)
On vika	0% (0%)	**17% (100%)**

Komponentti B

Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**99% (88%)**	91% (12%)
On vika	1% (55%)	**9% (45%)**

Komponentti C

Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**100% (89%)**	82% (11%)
On vika	0% (0%)	**18% (100%)**

Puiden määrä = 1 000

Komponentti A

Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**100% (89%)**	11% (10%)
On vika	0% (0%)	**8% (100%)**

Komponentti B

Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**98% (88%)**	91% (12%)
On vika	2% (58%)	**9% (41%)**

Komponentti C

Ennuste	Todellinen	
	Ei vikaa	On vika
Ei vika	**98% (88%)**	86% (12%)
On vika	2% (50%)	**14% (50%)**

Yhteenveto

- Diagnostiikkailmoituksille ei pystytty luomaan yhtä hyvää mallia kuin mitä sensoridatalle olisi mahdollista (vrt. esim. Sun et al. (2007), Jegadeeshwaran et al. (2013))
- Vikailmoitusdatan puutteellisuus vaikeuttaa sekä mallin luontia että ennusteen validointia
- Osassa tapauksia pystyttiin luomaan melko hyvä malli tarkkaan rajatulle komponentille, joskin mallit hieman ylisovitettuja
- Suurella puiden määrällä koulutettua satunnaismetsää, joka ei ennusta vikatilanteita turhaan, voi käyttää vikojen havaitsemiseen muiden keinojen ohella

Viitteet

- Steinberg, D., & Colla, P. (2009). CART: classification and regression trees. *The top ten algorithms in data mining*, *9*, 179.
- Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.
- Sun, W., Chen, J., & Li, J. (2007). Decision tree and PCA-based fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, *21*(3), 1300-1317.
- Jegadeeshwaran, R., & Sugumaran, V. (2013). Comparative study of decision tree classifier and best first tree classifier for fault diagnosis of automobile hydraulic brake system using statistical features. *Measurement*, *46*(9), 3247-3260.