

Arto Meriläinen

Toimintapisteiden käyttö ja tunnistaminen koneoppimismenetelmillä

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Kandidaatintyö

3.5.2010

Vastuupettaja:

TkT. Markus Turunen

Työn ohjaaja:

TkT. Harri Valpola



Aalto-yliopisto
Teknillinen korkeakoulu

Tekijä: Arto Meriläinen

Työn nimi: Toimintapisteiden käyttö ja tunnistaminen
koneoppimismenetelmillä

Päivämäärä: 3.5.2010

Kieli: Suomi

Sivumäärä:5+36

Tutkinto-ohjelma: Bioinformaatioteknologia

Vastuuopettaja: TkT. Markus Turunen

Ohjaaja: TkT. Harri Valpola

Jokainen järjestelmä voidaan jakaa toimintapisteisiin, joissa järjestelmän toiminta on erilaista. Järjestelmä ja toimintapisteet voivat olla miltein mitä vain mielekkäästi hahmotettavia käsitteitä. Autolla ajon voidaan nähdä jakautuvan toimintapisteisiin ”sula tie” ja ”liukas tie”: liukkaalla tiellä auton reagointi ohjaukseen on merkittävästi erilaista suhteessa reagointiin sulalla tiellä. Ongelma on ihmiselle intuitiivisesti hyvin helppo ymmärtää, mutta koneoppimisjärjestelmälle näinkin yksinkertaista toimintapisteiden tunnistamista voidaan pitää ylivoimaisena perinteisillä lähestymistavoilla.

Tiedossa on useita erilaisia menetelmiä, jotka mahdollistavat tunnettujen toimintapisteiden käytön ja halutun toiminnan tuottamisen eri toimintapisteissä. Tutkielmassa kuvataan joitakin näistä menetelmistä ja esimerkitapauksia niiden käytöstä. Tässä kandidaatintyössä havaitaan useiden eri aivoalueiden kykenevän sekä reagoimaan erilaisiin toimintapisteisiin että oppimaan toimintapisteet. Nämä mekanismit palvelevat muita aivoalueita ilmoittamalla eliön toimintapisteen, mikä mahdollistaa laajan adaptaation erilaisiin tilanteisiin.

Tutkielmassa esitellään yksinkertaisten toimintapisterajojen manuaalista opettamista koneoppimismenetelmille. Tällä menetelmällä voidaan laajentaa yksinkertaisissa sovelluksissa koneen toimintaa merkittävästi. Työssä tutustutaan myös kolmeen erilaiseen koneoppimismenetelmään, jotka kykenevät tunnistamaan automaattisesti järjestelmän toimintapisteen vaihtumisen ja järjestelmän käyttäytymisen eri toimintapisteissä.

Avainsanat: koneoppiminen, ohjaamaton oppiminen, korrelaatorakenne, Boltzmannin kone

Sisältö

Sisällysluettelo	iii
Termien suomennokset	v
1 Johdanto	1
2 Tunnettujen toimintapisteiden hyödyntäminen	3
2.1 Sekoitemallit	3
2.1.1 Asiantuntijain kirjo	3
2.1.2 Hierarkkinen asiantuntijain kirjo	4
2.1.3 Piilotetut Markovin mallit	5
2.2 Sovelluskohteita ja esimerkkejä	6
2.2.1 Luokittelu	7
2.2.2 EEG-käyrän analysointi	7
2.2.3 Liikkumistyylin mallintaminen	8
2.3 Toimintapisteiden havaitseminen aivoissa - esimerkkejä	8
2.3.1 Hippokampuksen yhdistävä koodaus	9
2.3.2 Hippokampuksen paikka-, päänsuunta- ja hilasolut toimintapisteiden määrääjinä	9
2.3.3 Aivokuori - Biasoidun kilpailun malli	10
3 Ohjaamattoman oppimisen tilastollisia menetelmiä	12
3.1 Pääkomponenttianalyysi	12
3.2 Kanoninen korrelaatioanalyysi	13
3.2.1 Määritelmä	13
3.2.2 Iteratiivinen kanoninen korrelaatioanalyysi	13
4 Toimintapisteiden paljastaminen	17
4.1 Manuaalinen toimintapisteiden määrittäminen	17

4.1.1	Esimerkki: Univaiheen analysointi	17
4.1.2	Esimerkki: Luokittelu	18
4.2	Veräjäyty ehdollinen rajoitettu Boltzmannin kone	19
4.2.1	Toimintaperiaate	19
4.2.2	Kouluttaminen	21
4.2.3	Sovellukset ja käyttö toimintapisteen hakuun	22
4.3	Puolirajoitettu Boltzmannin kone ja syvät uskomusverkot	22
4.3.1	Toimintaperiaate	23
4.3.2	Sovellukset ja toimintapistetulkinta	23
4.4	Ulkotulon ja laajennetun kanonisen korrelaatioanalyysin käyttö toimintapisteiden estimointiin	25
4.4.1	Toimintaperiaate	25
4.4.2	Menetelmän laajentaminen	26
4.4.3	Alustaminen ja aineiston esikäsittely	26
4.4.4	Menetelmän 1. ilmentymä	27
4.4.5	Menetelmän 2. ilmentymä	28
4.4.6	Jälkikäsittely	30
4.4.7	Koneoppimisnäkökulma	30
5	Yhteenveto ja pohdinta	32

Termien suomennokset

Canonical Correlation Analysis	Kanoninen korrelaatioanalyysi
Deep Belief Net	Syvä uskomusverkko
Gated Factorized Conditional Restricted Boltzmann Machine	Veräjäyty, tekijöity, ehdollinen rajoitettu Boltzmannin kone
Gated Conditional Restricted Boltzmann Machine	Veräjäyty ehdollinen rajoitettu Boltzmannin kone
Gibbs-sampling	Gibbs-otanta
Grid Cells	Hilasolut
Head Direction Cells	Päänsuuntasolut
Hidden Markov Model	Piilotettu Markovin malli
Hierarchical Mixture of Experts	Hierarkkinen asiantuntijain kirjo
Input-Output Hidden Markov Model	Syöte-lähtö piilotettu Markovin malli
Mixture Models	Sekoitemallit
Mixture of Experts	Asiantuntijain kirjo
Place Cells	Paikkasolut
Principal Component Analysis	Pääkomponenttianalyysi
Semi-restricted Boltzmann Machine	Puolirajoitettu Boltzmannin kone

Luku 1

Johdanto

Maailma koostuu toimintatiloista. Jossakin tilanteessa suotuista toimintaa voi olla toisessa tilanteessa virheellinen. Esimerkiksi autoa ajaessa sääolosuhteet vaikuttavat kriittisesti ajotyyliin; liukkaalla ajettaessa auton käyttäytyminen on aivan erilaista suhteessa sulaan tiehen, joten lopputulos olisi haitallinen, mikäli ajaja jättäisi huomiotta sääolosuhteen määräämän toimintatilan. Näitä erilaisia toimintatiloja nimitetään toimintapisteiksi.

Monet perinteiset koneoppimismenetelmät olettavat opittavan aineiston muodostuneen yhden tietyn toimintapisteiden ympäristössä. Useissa tapauksissa mallin rakenne onkin etukäteen täysin tiedossa, jolloin tarpeellista on ainoastaan mallin parametrien määrittäminen käyttäen saatavilla olevaa aineistoa. Vaikka tämänkaltaisen lähestymistapa on mielekäs monissa tapauksissa, voivat lähestymistavan sisältämät rajoitukset olla joissakin tapauksissa liikaa. Aineisto ei kuitenkaan aina ole muodostunut vain yhdessä toimintapisteessä, eikä toimintapisteiden määrää tai sopivaa toimintaa etukäteen tunneta. Ihmisresursseja ei mahdollisesti haluta käyttää ongelman tarkkaan määrittelyyn. Toimintapisteiden määrittäminen etukäteen voi olla myös ihmiselle ylivoimaista ongelman luonteen takia.

Tunnettaessa toimintapisteet ja niitä vastaavat mallit voidaan haluttu toiminta saavuttaa käyttämällä asiantuntijain kirjo -menetelmää (mixture of experts). Menetelmä tunnistaa toimintapisteet ja antaa sitä vastaavan mallin muodostaa toiminnan. Täten voidaan todeta, että mallin ollessa tiedossa sen käyttäminen ei ole ongelmallista.

Vaikka mallin käyttö on suoraviivaista, ei mallin muodostaminen ole yhtä helppoa. Mikäli toimintapisteet tunnetaan etukäteen, voidaan jokaiseen toimintapisteeseen sopiva malli kouluttaa erikseen, jolloin lähestytään hallittavissa olevia perinteisiä koneoppimisongelmia. Sama pätee myös toisin päin: Mikäli jokaiseen sopivat mallit tunnetaan, voidaan valikointi toteuttaa yksinkertaisesti kokeilemalla. Tilanne on huomattavasti monimutkaisempi, mikäli sekä mallit että toimintapisteet ovat tuntemattomia.

Tässä kandidaatintyössä tutustutaan toimintapisteiden hyödyntämiseen sekä erilaisiin menetelmiin, jotka mahdollistavat tuntemattomien toimintapisteiden ja niitä

vastaavien mallien kouluttamisen käyttäen saatua aineistoa. Tutkielmassa pyritään havainnollistamaan aivoissa tapahtuvan oppimisen merkitys, sekä kuvaamaan algoritmien sovelluskohteita. Työssä käsitellään laajemmin ainoastaan koneoppimismenetelmiä, joilla on suora yhteys toimintapisteiden estimointiin tai käyttöön.

Tutkielma on jaettu kolmeen käsittelylukuun. Ensimmäisessä käsittelyluvussa tarkastellaan tunnettujen toimintapisteiden käyttämistä sekä niiden käyttökohteita. Toisessa käsittelyluvussa tutustutaan muutamaa yleiseen analysointimenetelmään: pääkomponenttianalyysiin ja kanoniseen korrelaatioanalyysiin. Menetelmien hyödyt käyvät ilmi kolmannessa käsittelyluvussa, jossa tutustutaan toimintapisteiden estimointiin. Luvussa tutustutaan muutamaa erilaiseen malliin, joiden valinnassa on kiinnitetty huomiota mallien tuoreuteen, erilaiseen ja selkeyteen.

Luku 2

Tunnettujen toimintapisteiden hyödyntäminen

2.1 Sekoitemallit

Sekoitemallit (mixture models) ovat koneoppimismenetelmiä, joilla kyetään analysoimaan ja käyttämään aineistoa, joka on muodostunut usean eri mallin pohjalta. Tässä luvussa tutkitaan erilaisia sekoitemalleja, jotka kykenevät hyödyntämään opittuja toimintapisteitä tavoitellun toiminnan saavuttamiseen.

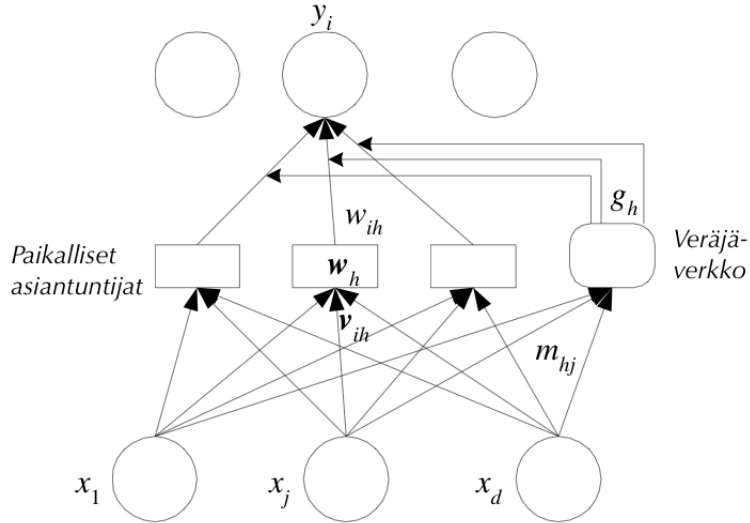
2.1.1 Asiantuntijain kirjo

Asiantuntijain kirjo (mixture of experts) on tunnettu koneoppimismenetelmä toimintapisteiden käyttöön. Menetelmässä käytetään ulostulosignaalia muodostettaessa useita eri asiantuntijoita, jotka voivat tuottaa ulostuloarvonsa millä tahansa koneoppimismenetelmällä. Eri asiantuntijoiden paino signaalia muodostaessa määräytyy erillisen veräjäverkon (gating network) antaman arvon pohjalta. Sekä veräjäverkko että asiantuntijat saavat samat sisääntulosignaalit. Kuvassa 2.1 esitetään kaavamaisesti mallin rakenne. [1, sivut 296-299]

Matemaattisesti ulostulosignaali määräytyy seuraavasti:

$$y(\mathbf{x}) = \sum_{h=1}^H w_h(\mathbf{x})g_h(\mathbf{x}), \quad (2.1)$$

jossa x kuvaa sisääntulovektoria, y kuvaa ulostulovektoria, g_h kunkin asiantuntijan painoa ja w_h kunkin asiantuntijan antamaa arvoa. Mikäli asiantuntijafunktiot w_h ja veräjäverkkofunktio g_h ovat tiedossa, on ulostuloarvon laskeminen hyvin suoraviivaista: sisääntulovektorilla lasketaan $\mathbf{x}$ funktioiden $g_h(\mathbf{x})$ ja $w_h(\mathbf{x})$ arvot, jonka jälkeen arvot sijoitetaan suoraan kaavaan 2.1. On syytä huomata, että funktiot $g_h(\mathbf{x})$ ja $w_h(\mathbf{x})$ voivat tuottaa ulostuloarvonsa millä tahansa menetelmällä.



Kuva 2.1: Asiantuntijain kirjo -menetelmän kaavamainen rakenne. Sekä kaikki asiantuntijat, että veräjäverkko saavat samat sisääntuloarvot. Veräjäverkon ulostulosarvo määrää asiantuntijoiden painoarvot. [1, kuva 12.11, muokattu]

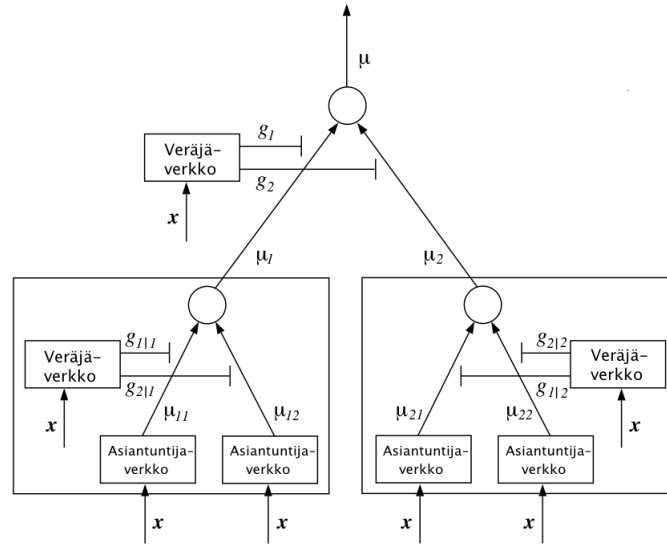
Veräjäverkon ja asiantuntijoiden saadessa samat sisääntulosignaalit voivat asiantuntijat toimia joko yhteistyössä tai kilpaillen. Käytännössä toiminta riippuu täysin w_h :n koulutusvaiheesta saamista arvoista. Yhteistyössä toimivat asiantuntijat saavat kaikki suhteellisen paljon painoarvoa ulostuloarvolleen veräjäverkolta. Sen sijaan kilpaillen toimimaan opetetut asiantuntijat saavat kaiken painoarvon muiden asiantuntijoiden jäädessä huomiotta. On syytä huomioida yhteistyössä toimivien asiantuntijoiden olevan huomattavasti haastavampia analysoitavia toimintapisteiden näkökulmasta. Mikäli asiantuntijat toimivat kilpaillen, toimii jokainen asiantuntijoista omissa toimintapisteissään. [2] [1, sivut 299-300]

2.1.2 Hierarkkinen asiantuntijain kirjo

Hierarkkinen asiantuntijain kirjo (Hierarchical Mixture of Experts) laajentaa edellä esitettyä asiantuntijain kirjoa käyttämällä jokaisena asiantuntijana hierarkkisen asiantuntijain kirjoa. Tällöin muodostuu rekursiivinen malli, joka sisältää aina syvemmälle mentäessä uuden hierarkkisen asiantuntijain kirjon. Kaavamainen kuva mallista on esitetty kuvassa 2.2.

Vaikka malli onkin rakenteeltaan varsin selkeä, voidaan sen toimintaa analysoida usealla eri tavalla. Malli voidaan tulkita laajennettuna päätöspuuna, jossa sisääntulo syötetään kuhunkin lehteen. Kuka veräjäverkko toimii päätössolmuna, joka yhdistää asiantuntijoidensa ulostulot. Juureen saapunut tieto sisältää painotetun keskiarvon eri lehdistä saaduista signaaleista. Verrattuna päätöspuihin menetelmällä saavutetaan diskreetin arvon asemasta jatkuva lopputulos. [1, sivut 300-301]

Hierarkkinen asiantuntijain kirjo luo toimintapisteiden kannalta kiinnostavan ulottu-



Kuva 2.2: Hierarkkinen asiantuntijain kirjo. Kukin asiantuntija koostuu omasta kirjostaan, mikä tuottaa rekursiivisen rakenteen. [3, muokattu]

vuuden. Ylimmän tason veräjäverkko voi tulkita tilannetta hyvin kokonaisvaltaisesti antaen määräysvallan kokonaan tietylle asiantuntijalle, joka vastaa tilanteeseen parhaalla mahdollisella tavalla. Tietty asiantuntija taas tulkitsee tilanteen omasta näkökulmastaan, ja antaa määräysvallan jollekin omalle asiantuntijalleen (tai sekoittaa asiantuntijoidensa ulostuloja). Tasojen määrä voi olla hyvin suuri, jolloin aina vain yksityiskohtaisempi toiminta on mahdollista.

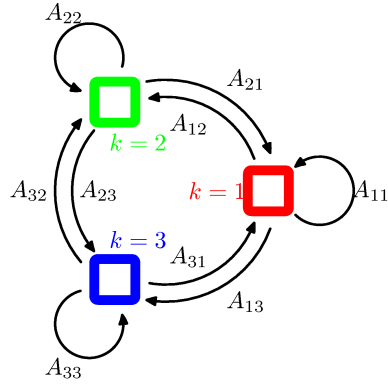
Yleistyksiä menetelmän käytölle on lukuisia. Esimerkkinä hierarkisen asiantuntijain kirjon toiminnasta voi pitää valintaa siivoamisen ja ruokailun välillä. Ylimmän tason veräjäverkko päättää toimintapisteen siivoamisen ja ruokailun väliltä. Alemman tason veräjäverkot ja asiantuntijat päättävät tarkemman toiminnan valitussa tilanteessa.

2.1.3 Piilotetut Markovin mallit

Piilotetut Markovin mallit (Hidden Markov Models) tarjoavat keinon sekvensseistä koostuvan aineiston analysointiin. Tässä luvussa käsitellään menetelmän suorittamista diskreetissä tapauksessa, jossa toimintapistet ja toimintapisteiden tuottamat arvot ovat diskreettejä muuttujia.

Malli koostuu mallin ulkopuolelle näkyvistä ja näkymättömistä muuttujista. Näkymättömät muuttujat määrittävät siirtymätodennäköisyyksiä eri toimintapisteiden välillä. Näkyvät muuttujat vastaavasti määrittävät todennäköisyyksiä rajatulle määrälle eri ulostuloarvoja. Ulostuloarvot riippuvat täten toimintapisteestä ja näkyvien muuttujien arvoista. Kuvassa 2.3 esitetään kaavamaisesti toimintapisteen vaihtuminen kolmen toimintapisteen tapauksessa. [4, sivut 610-615]

Kun siirtymätodennäköisyydet ja eri toimintapisteissä saatavat arvot ja niiden ulos-



Kuva 2.3: Tilasiirtymät piilotetuissa Markovin malleissa. Kuvan neliöt kuvaavat eri tiloja, nuolet siirtymiä tiloista toiseen. [4, kuva 13.6]

tulotodennäköisyydet on tiedossa, saadaan ulostuloarvo helposti:

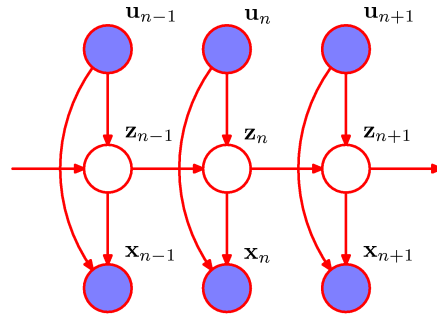
1. Päätetään aloitustoimintapiste. Ennustetta tehdessä tämän tulee olla jo etukäteen tiedossa, mutta aloitustoimipiste voidaan myös arpoa, mikäli ilmiötä halutaan vain mallintaa.
2. Arvotaan painotetusti ulostuloarvo nykyisessä toimintapisteessä.
3. Arvotaan painotetusti uusi toimintapiste.
4. Siirry kohtaan 2. Tätä toistetaan kunnes haluttu määrä askeleita on suoritettu.

Piilotettu Markovin malli soveltuu mainiosti ennustamiseen ja aineiston mallintamiseen. Sen sijaan se ei kykene muodostamaan vastaanvanlaista sisääntulo-ulostulo-ristiviittausta kuin asiantuntijain kirjo -menetelmä.

Vaikka perusmuotoinen piilotettu Markovin malli ei tarjoa keinoa tehdä ennustetta sisääntulevien arvojen välillä, on mallia kehitetty monilla eri tavoin. Kiinnostavana esimerkkinä voidaan pitää Bengion ja Fransconin esittelemään menetelmää, syöte-lähtö piilotettu Markovin malli (Input-Output Hidden Markov Model) [5]. Malliin on lisätty mahdollisuus antaa sisääntuloarvoja, jotka vaikuttavat ulostuloarvoihin että mallin toimintapisteeseen. Vaikka mallin toiminta lähenee asiantuntijain kirjoja, voidaan merkittävänä etuna pitää mallin toimintapisteen riippuvuutta edellisestä toimintapisteestä. Kuvassa 2.4 esitetään laajennetun piilotetun Markovin mallin rakenne.

2.2 Sovelluskohteita ja esimerkkejä

Edellisessä luvussa tehtiin katsaus yleisimpiin koneoppimismenetelmiin, jotka kykenevät hyödyntämään opittuja toimintapisteitä. Tässä luvussa tutkitaan joitakin sovelluksia, joihin menetelmiä voidaan soveltaa.



Kuva 2.4: Rakennekuvio syöte-lähtö piilotetusta Markovin mallista. Tunnetut u_n solmut kuvaavat sisääntulevia arvoja, jotka vaikuttavat toimintapisteeseen z_n . Toimintapisteeseen vaikuttaa myös aiempi toimintapiste z_{n-1} . Havaittaviin ulostuloarvoihin vaikuttavat sekä sisääntulleet arvot u_n että toimintapiste z_n . [4, kuva 13.18]

2.2.1 Luokittelu

Luokittelu on yksi yleisimmistä koneoppimisen ongelmista. Luokittelussa halutaan tiettyjen mittausten perusteella päätellä automaattisesti jotakin aineistosta. Esimerkiksi kehittyneet roskapostisuodattimet toteuttavat aineistolle testejä, jotka saavat binäärisen arvon (1 tai 0). Roskapostisuodatin käyttää näitä testituloksia määrittääkseen sähköpostiviestin luokan roskapostin ja toivotun sähköpostin väliltä.

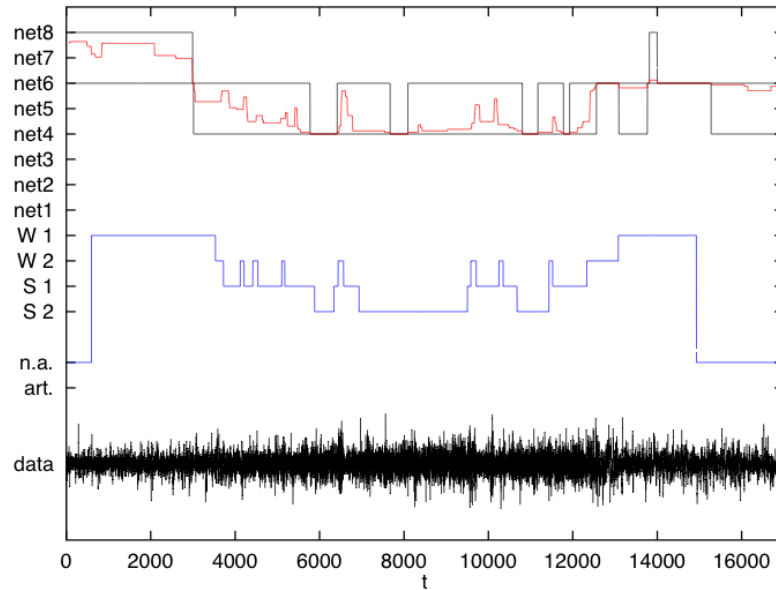
Luokittelu voidaan toteuttaa hyvinkin yksinkertaisilla menetelmillä, mutta monimutkaisissa tehtävissä voi olla tarpeen käyttää asiantuntijain kirjo -menetelmää. Esimerkkejä tämänkaltaisesta tilanteesta tuottaa jo yksinkertainen ehto, jossa luokittelusääntö (toiminta tietyssä toimintapisteessä) tulee tuottaa jonkin toisen säännön pohjalta (toimintapisteen valinta). On oleellista huomata, että yksinkertainen neuroverkko epäonnistuu suurella todennäköisyydellä tämänkaltaisissa tehtävissä [2].

2.2.2 EEG-käyrän analysointi

Tässä esimerkissä keskitytään tunnettujen toimintapisteiden käyttämisen sijaan toimintapisteiden oppimiseen. Tässä yhteydessä ei vielä mennä syvällisemmin käytettäviin koulutusalgoritmeihin, vaan painopisteenä on esitellä yksi toimintapisteajattelun sovelluskohde.

Ihmisen unirytmistö koostuu useista eri univaiheista, joita edustaa eri tietoisuuden taso. Normaali unirytmistö liikkuu sujuvasti valvetilasta syvään uneen ja siitä yhä REM-uneen (Rapid Eye Movement, REM), joten vaiheiden voidaan nähdä linkittyneen hyvin vahvasti. Unta on voitu analysoida aivosähkökäyrän (elektroenkefalografia, EEG), silmäsähkökäyrän (elektro-okulografia, EOG) ja lihassähkökäyrän (elektromyografia, EMG) avulla. Näistä analysointimenetelmistä ensimmäinen tuottaa hyvin paljon arvokasta informaatiota univaiheesta, sillä sen muutoksista ammattilainen kykenee tunnistamaan univaiheen, mistä on hyötyä erityisesti univaikeuksien tulkinnessa. [6]

Univaiheen vaihtelut voidaan tulkita erilaisina toimintapisteinä, joita edustavat mallit tuottavat erilaisen aivosähkökäyrän. Koska univaiheet ovat vahvasti linkittyneet toisiinsa, voidaan piilotettua Markovin mallia soveltaa ongelmaan. Menetelmä tunnistaa toimintapisteen vaihtelun ohjaamattomasti ilman asiantuntijan apua koulutuksessa. Kuva 2.5 esittää asiantuntijan ja rakennetun mallin analyysit. [7]



Kuva 2.5: Aivosähkökäyrän analysointi piilotetulla Markovin mallilla. Alempi kaavio näyttää univaiheen asiantuntijan analyysin perusteella, ja ylempi kaavio näyttää koneoppimismenetelmän tulkitseman univaiheen. [7]

2.2.3 Liikkumistyylin mallintaminen

Taylor ja Hinton esittävät hyvin poikkeuksellisen sovelluksen toimintapisteiden käytölle mallintamalla ihmisen liikkumistyyliä. Sovelluksessa koulutetaan veräjäyty, tekijöity, ehdollinen rajoitettu Boltzmannin kone aineistolla, joka sisältää erilaisia kävelytyylejä, joista kukin kävelytyyli kuvaa yhtä toimintapistettä. Koulutetun mallin avulla saadaan kehitettyä sopivilla parametreilla alkuperäistä liikkumistyyliä vastaava lopputulos lähes kaikissa tapauksissa. [8]

2.3 Toimintapisteiden havaitseminen aivoissa - esimerkkejä

Arkipäiväinen ihmisen toiminta viestii aivojen sekä kykenevän tunnistamaan toimintapisteen että oppimaan toimintapistettä vastaavan mallin samanaikaisesti. Aivot pystyvät selvittämään ongelmia maailmassa, joka sisältää tuhansia erilaisia toimintapisteitä käyttämällä jatkuvasti samaa rajapintaa, elimistöämme. On siis täysin

selvää, että aivot kykenevät valitsemaan saatujen ärsykkeiden pohjalta sopivan toimintapisteen ja käyttäytyvän asianmukaisesti. Voidaankin siis päätellä aivojen kykenevän hyödyntämään opittuja toimintapisteitä. On kuitenkin huomattava, ettei malleja tai toimintapisteitä ole koodattu vastasyntyneen aivoihin, minkä seurauksena aivoissa voidaan olettaa olevan oppimismekanismi, jonka avulla toimintapisteet voidaan löytää.

Tässä luvussa tutustutaan aivoalueisiin, joissa tiedetään tapahtuvan toimintapisteiden hyödyntämistä ja oppimista. Luvussa esitellään aivoissa esiintyvää toimintaa menemättä oppimismekanismien yksityiskohtiin.

2.3.1 Hippokampuksen yhdistävä koodaus

Vauriot hippokampuksessa ovat antaneet viitteitä kyvystä muodostaa episodinen muisti, joka tallentaa eri aivoalueiden tilan ja tilamuutokset tietyllä aikahetkellä. Muisteltaessa jotakin tapahtumaa hippokampus toistaa koetun tilan aktivoimalla aivoalueita. [9]

Vaikka ajatus onkin teoriassa hyvin yksinkertainen, sisältää se useita ongelmia. Mikäli vain nykyinen subjektiivinen tilanne tulee tallettaa muistiin, onnistuu se edellä kuvatulla menetelmällä. Mikäli tilanne sisältää relaatioita useamman tekijän välillä, ei niitä kuitenkaan pystytä selvittämään yhtä helposti. Mikäli Mika on esimerkiksi lainannut kirjaansa Heikille, ei pelkkä sanojen ”Mika”, ”kirja” ja ”Heikki” tallentaminen muistiin riitä kuvaamaan tilannetta täydellisesti. Tämä on kuitenkin selvitetävissä rooliin sitomisella (role binding). Tämänkaltaisen sitominen edellyttää hippokampukselta yhdistävää koodaamista (conjunctive coding). [10]

Vaikka esimerkki onkin hyvin käytännönläheinen, se kuvaa hyvin millä tavoin toimintapisteet näkyvät hippokampuksessa. Kun puhutaan kirjan lainaamisesta, on tiedossa välittömästi toimintapiste, joka kuvaa lainaustapahtumaa. Jos Mika olisi varastanut kirjan Heikiltä, olisi tilanne tallettunut hippokampukseen eri tavoin, jolloin myös toimintapiste olisi ollut erilainen.

2.3.2 Hippokampuksen paikka-, päänsuunta- ja hilasolut toimintapisteen määrääjinä

Hippokampuksen tiedetään osallistuvan myös suuntavaistoon. Hippokampus sisältää paikkasoluja (place cells), jotka ovat aktiivisia eliön ajatellessa tiettyä paikkaa tai eliön ollessa tietyssä paikassa. Rottia tutkittaessa on voitu löytää hyvin tarkasti rotan sijainti labyrintissa tutkimalla paikkasolujen aktivaatiota. [11] Toimintapiste-tulkinta näille soluille on varsin selkeä. Solujen aktivaatio ilmaisee muille aivoalueille tietyn eliön sijainnin. Hippokampus toimii siis tässä mallissa veräjäverkkona.

Hippokampus sisältää toisen solupopulaation, jonka solut ovat aktiivisia vain tietyssä tilanteessa. Rotilla tehtyt kokeet osoittavat, että hippokampus sisältää soluja, jotka ovat vahvasti riippuvaisia ainoastaan pään suunnasta. Nämä päänsuuntasolut (head

direction cells) ovat tulkittavissa myös toimintapisteen määrääjinä, jotka ilmaisevat muille aivoalueille nykyisen katselusuunnan. [12]

Edellisiä solupopulaatioita mutkikkaampia ovat hilasolut (grid cells). Kuten paikkasolujen aktivaatio, myös hilasolujen aktivaatio on riippuvaista eliön sijainnista. Hilasolut eivät kuitenkaan ole aktiivisia vain yhdessä paikassa, vaan solujen muodostama yhteinen aktivaatiokuvio ilmaisee eliölle sen nykyisen sijainnin. Tutkimukset osoittavat samanaikaisesti aktivoituvien solujen aktivoituvan joka kerta samalla tavoin, joten hilasolujen aktivaatiota voidaan pitää paikkasoluja vastaavana mekanismina eliön paikan ilmaisemisessa. Rotilla tehdyt tutkimukset osoittavat kahden identtisen paikan tuottavan erilaisen vastineen, mikäli rotta itse kulkee paikkojen välillä tiedostaen siirtymisensä. Kuitenkin merkittäviä päällekkäisyyksiä esiintyy aktivoituvissa soluissa. Kuvassa 2.6 esitetään hilasolujen aktivaatiota rotalla sen kulkiessa kahden identtisen huoneen välillä. [13]

Hilasolujen toimintapistetulkinta on vastaava kuin paikkasolujen. Kun hippokampuksen hilasoluissa esiintyy tietty kuvio, saavat muut aivoalueet tiedon nykyisestä sijainnista. Sijaintitieto taas vaikuttaa oleellisesti eliön toimintapisteeseen.

2.3.3 Aivokuori - Biasoidun kilpailun malli

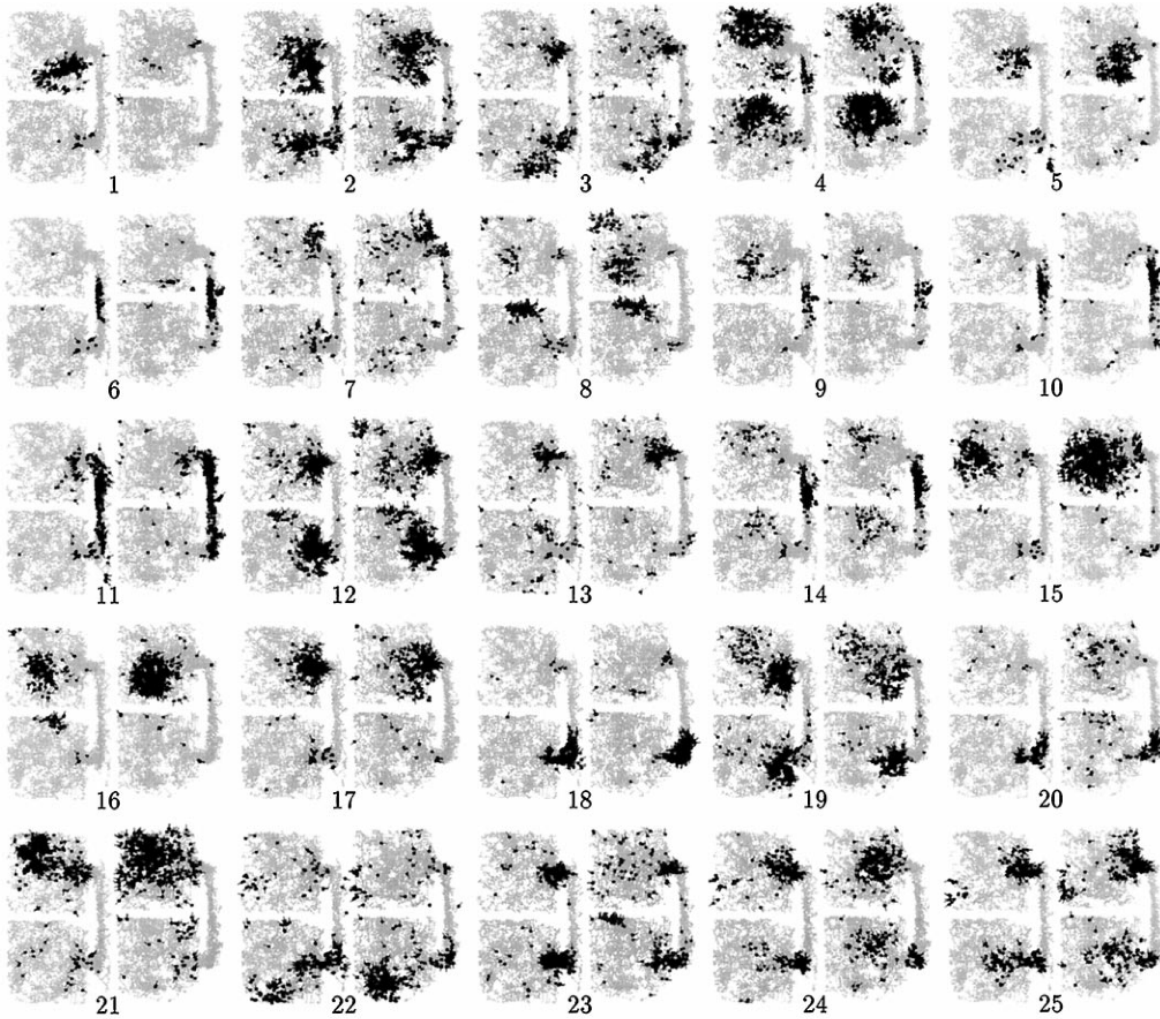
Aivokuorella esiintyy useita ilmiöitä, jotka viittaavat toimintapisteiden olemassaoloon. Selkeimpänä esimerkkinä voidaan pitää valikoivaa tarkkaavaisuutta. Arkipäiväiset kokemukset viittaavat siihen, että huomion kiinnittyessä johonkin asiaan suodatetaan automaattisesti epäoleellisia ärsykeitä.

Tarkkaa selitystä ilmiölle ei kirjoitushetkellä tunneta. Tutkimukset näköaivokuoren V4-alueella viittaavat aivojen kykenevän valikoimaan ärsykeitä keskittymisen mukaan [14]. Myös keskittymiskyvyn on ehdotettu aiheutuvan samanlaisen ilmiön kautta [15]. Molemmissa tapauksissa on sovellettu biasoidun kilpailun mallia.

Biasoidun kilpailun mallissa alhaalta-ylöspäin -vaikutukset (bottom-up) aktivoivat biasoinnin, joka näkyy ylhäältä-alas -vaikutuksena (top-down) vaimentaen epäoleellisia ärsykeitä. Mallilla voidaan selittää useita aivoissa tapahtuvia ilmiöitä. Mallin pohjalta tuotetut näköaivokuoren simulaatiot tuottavat hyvin realistisen lopputuloksen [16]. On syytä havaita biasoidun kilpailun mallin olevan hyvin analoginen toimintapisteajattelun kanssa. Ylemmän tason biasointi määrää alemmille tasoille toimintapisteen.

Näköaivokuoren V1-alueelle annetulla ärsykkeellä voidaan tuottaa näkökenttään staattinen fosfeeni (näköaistimus ilman valonlähdettä). V5-alueella tuotetun ärsykeen tiedetään vaikuttavan fosfeenin liikkumiseen näkökentässä. Täten voidaan päätellä V5-alueen ilmaisevan toimintapisteen (nykyisen liikkumissuunnan) V1-alueelle, minkä ansiosta tietoisesti havaittu kokemus on yhtenäinen. [17]

On syytä havaita toimintapisteajattelun olevan kerrostettavissa, jolloin toimintapistet muodostavat hierarkkisen rakenteen. Esimerkiksi ruokaillessa hierarkian ylin taso määrää alemmille tasoille toimintapisteen. Jokin alemmista tasoista vastaa kar-



Kuva 2.6: Hilasolujen aktivaatio. Kuvassa näkyy 25 hilasolun aktivaatio rotan kulkiessa kahdessa identtisessä huoneessa ja niitä yhdistävässä käytävässä. Koe oli jaettu kahteen osioon, jonka puolivälissä identtisten laatikoiden paikkaa vaihdettiin. Vasemmalla näytetään aktivaatiot kokeen ensimmäiseltä puoliskolta, ja oikealla näytetään aktivaatiot kokeen jälkimmäisellä puoliskolla. Harmaalla esitetyt pisteet kuvaavat rotan sijaintia ja kulkua labyrintissä ilman solun aktivaatiota. Mustalla esitetty kuvaa solun aktivaatiota kyseisessä sijainnissa. [13]

kealla tasolla toiminnasta ("leikkaan ruokaa, siirrän ruokaa suuhun") jonkin vielä alemman tason vastatessa tarkemmasta toiminnasta ("liikutan veistä, liikutan kättä"). Nämäkin tasot ovat jaettavissa useisiin alatasoihin.

Luku 3

Ohjaamattoman oppimisen tilastollisia menetelmiä

Tässä luvussa tutustutaan menetelmiin, jotka ovat välttämättömiä luvussa 4 esitettävien menetelmien ymmärtämisessä. Luvussa esitellään kaksi tunnettua menetelmää: pääkomponenttianalyysi (Principal Component Analysis, PCA) ja kanoninen korrelaatioanalyysi (Canonical Correlation Analysis, CCA). Näistä ensimmäinen on laajasti käytössä oleva tekniikka, joka mahdollistaa moniulotteisen aineiston ulottuvuuksien vähentämisen. Jälkimmäinen tekniikka taasen on oiva apuväline pyrkiessä löytämään korreloivia piirteitä kahden eri aineiston välillä.

Menetelmiä esitellessä pääpainona on menetelmien perusperiaatteiden ymmärtäminen. Kanonisen korrelaatioanalyysin osalta esitellään myös perusmuotoinen iteratiivinen menetelmä, joka on pääosassa luvussa 4 esitettävässä parannetussa kanonisessa korrelaatioanalyysissä.

3.1 Pääkomponenttianalyysi

Pääkomponenttianalyysiä käytetään aineiston ulottuvuuksien pienentämiseen pyrkien menettämään mahdollisimman vähän informaatiota. Menetelmässä oletetaan informaation kuvautuvan aineistossa variaationa. Näin ollen etsimällä aineistosta sellaiset komponentit, joissa variaatiota on eniten, löydetään myös komponentit, jotka sisältävät informaatiota eniten. Kun käytettäviä komponentteja lisätään, saadaan lopulta selitettyä kaikki aineiston variaatio. Usein ei kuitenkaan haluta etsiä kuin muutama pääkomponentti, joissa informaatiota esiintyy eniten. Kun pääkomponentit on löydetty, projisoidaan kaikki aineisto näille komponenteille. Tämän jälkeen ulottuvuuksien määrä vastaa valittujen pääkomponenttien määrää. [1, sivut 108-109]

Kaikkien aineiston komponenttien keskinäistä variaatiota kuvaa kovarianssimatriisi (Σ). Kovarianssimatriisin ominaisvektorit kuvaavat ominaisarvojen suuruusjärjestyksessä suuntia, joissa variaatiota on eniten. Tällöin aineisto voidaan projisoida

k :hon ulottuvuuteen:

$$Z = W^T(X - M), \quad (3.1)$$

jossa Z kuvaa projisoitua aineistoa, W kuvaa kovarianssimatriisin k :ta suurinta ominaisarvoa vastaavaa ominaisvektoria, X aineistoa alkioita ja M aineiston keskiarvoa. [1, sivut 109-110]

3.2 Kanoninen korrelaatioanalyysi

Kanoninen korrelaatioanalyysi on H. Hotellingin vuonna 1936 kehittämä menetelmä [18], joka on hyvin käytetty tilastotieteessä lineaaristen vastaavuuksien löytämiseen [19]. Menetelmässä pyritään kertomaan aineiston alkioita siten, että aineistojen välinen korrelaatio olisi mahdollisimman suuri.

Menetelmälle löytyy useita käyttökohteita. Mikäli aineistot voidaan viedä lineaarikuvauksella samalle suoralle, voidaan aineiston muuttujien välisten suhteiden olettaa olevan täysin samat. Jos kanonisessa korrelaatiossa paljastuu yksittäisiä pisteitä, joita ei voida kiertää samalle suoralle, voidaan muuttujien olettaa kuvaavan eri asioita. Tätä ominaisuutta voitaisiin käyttää esimerkiksi kahden erilaisen testin vertailuun. Mikäli käytettävissä on kaksi rinnakkaista tutkimusmenetelmää, jotka tuottavat moniulotteista aineistoa, voidaan kanonisella korrelaatioanalyysillä löytää tutkimusmenetelmien yhtenevyydet. Jos aineisto on kerätty ajallisesti tai paikallisesti muuttuvista ominaisuuksista, voivat muutokset korreloivuudessa indikoida muuttunutta toimintapistettä. Tähän ajatukseen palataan luvussa 4.

3.2.1 Määritelmä

Matemaattisesti kanoninen korrelaatioanalyysi voidaan kuvata seuraavasti [19]:

$$\max_{\mathbf{a}, \mathbf{b}} \text{cor}(\mathbf{ax}, \mathbf{by}), \quad (3.2)$$

missä $\mathbf{x}$ ja $\mathbf{y}$ kuvaavat vertailtavia aineistoja, $\mathbf{a}$ lineaarikuvausta $\mathbf{x}$:lle ja $\mathbf{b}$ lineaarikuvausta $\mathbf{y}$:lle.

Määritelmän pohjalta voidaan muodostaa erilaisia tapoja lineaarikuvausten $\mathbf{a}$ ja $\mathbf{b}$ ratkaisemiseksi. Muuttujat voidaan ratkaista analyttisesti käyttämällä lineaarialgebraa [19]. Muuttujat voidaan ratkaista myös iteratiivisesti [18][20].

3.2.2 Iteratiivinen kanoninen korrelaatioanalyysi

Iteratiivista kanonista korrelaatioanalyysiä tehdessä projisoidaan vertailtava aineisto hypertasolle, joka on muodostettu käyttäen toista vertailtavaa aineistoa. Tämä toistetaan myös toiselle vertailtavalle aineistolle.

On syytä havaita, että iteratiivinen kanoninen korrelaatioanalyysi voidaan toteuttaa usealla eri tavalla [18] [20] [21] [22]. Tässä yhteydessä tutustutaan menetelmään, jonka paranneltua versiota käsitellään luvussa 4. Tässä esiteltävä menetelmä on kuvattu lähteen [20] pohjalta.

Seuraavassa esitettävässä iteratiivisen algoritmin yhteenvedossa oletetaan aineiston koostuvan useasta riippumattomasta ilmentymästä, joista kukin ilmentymä sisältää moniulotteisen tuloksen. Käytännössä tämä voisi tarkoittaa kahta erillistä anturia (X ja Y), joista molemmat antavat kaksi ulostuloarvoa (kaksi ulottuvuutta). Useampi riippumaton ilmentymä voidaan käsittää siten, että antureiden antamat ulostuloarvot muuttuvat ajan suhteen. Yhteenvedossa esitettävissä kaavoissa tämä esitetään funktiona t :n suhteen. On syytä huomata, että tämän seurauksena kaikki ulostulovektoritkin ovat funktioita t :n suhteen.

Menetelmä koostuu neljästä vaiheesta:

1. **Keskitys** Vertailtavat aineistot keskitetään. Käytännössä tämä tarkoittaa aineiston kunkin alkion vähentämistä aineiston keskiarvolla:

$$X(t) = X(t) - \overline{X(t)} \quad (3.3)$$

$$Y(t) = Y(t) - \overline{Y(t)} \quad (3.4)$$

2. **Valkaisu** Aineistot valkaistaan. Tällä operaatiolla tarkoitetaan aineiston komponenttien variaation pakottamista lukuun 1 ja kovarianssien poistamista. Valkaisua voidaan siis pitää operaationa, jossa molempien aineistojen kovarianssimatriisi pakotetaan yksikkömatriisiksi.
3. **Projektiio** Tässä vaiheessa ratkaistaan aineiston projektiot $\mathbf{a}(t)$ ja $\mathbf{b}(t)$. Nämä projektiot kuvaavat aineiston sovitumista hypertasoille, jotka ovat mahdollisimman lähellä toisiaan. On syytä huomioida ensimmäisellä iteraatiokierroksella matriisien $\mathbf{W}_X$ ja $\mathbf{W}_Y$ sisältävän satunnaisia pieniä arvoja. Projektiio tapahtuu yksinkertaisesti:

$$\mathbf{a}(t) = \mathbf{W}_X \mathbf{X}(t) \quad (3.5)$$

$$\mathbf{b}(t) = \mathbf{W}_Y \mathbf{Y}(t) \quad (3.6)$$

4. **Projektiomatriisien estimointi** Käyttämällä edellisessä vaiheessa saatuja aineiston projektiota $\mathbf{a}(t)$ ja $\mathbf{b}(t)$ ratkaistaan uudet matriisit $\mathbf{W}_X$ ja $\mathbf{W}_Y$. On syytä huomata, että estimoidessa uutta projektiomatriisia $\mathbf{W}_X$ käytetään projektiota $\mathbf{b}(t)$ ja vastaavasti $\mathbf{W}_Y$:tä estimoidessa käytetään projektiota $\mathbf{a}(t)$. Projektiomatriisit voidaan estimoida kahdella eri lähestymistavalla. Ensimmäisessä tavassa aineistosta ratkaistaan ulottuvuus kerrallaan vektorien $\mathbf{a}(t)$ ja $\mathbf{b}(t)$ arvo. Tällöin menetelmää toistetaan kunnes kaikkien ulottuvuuksien $\mathbf{a}(t)$ ja $\mathbf{b}(t)$ ilmentymät on ratkaistu. Ensimmäisessä lähestymistavassa W_X ja W_Y ovat vektoreita (matriisiesityksenä $1 \times n$ kokoinen matriisi). On syytä havaita vaiheen 3 tuottavan tällöin $a(t)$:lle skalaarimuotoisen ulostulon. Soveltaessa

ensimmäistä lähestymistapaa tulee käyttää seuraavia kaavoja estimointiin:

$$W_X = \sum_t \mathbf{X}(t)^T * b(t) \quad (3.7)$$

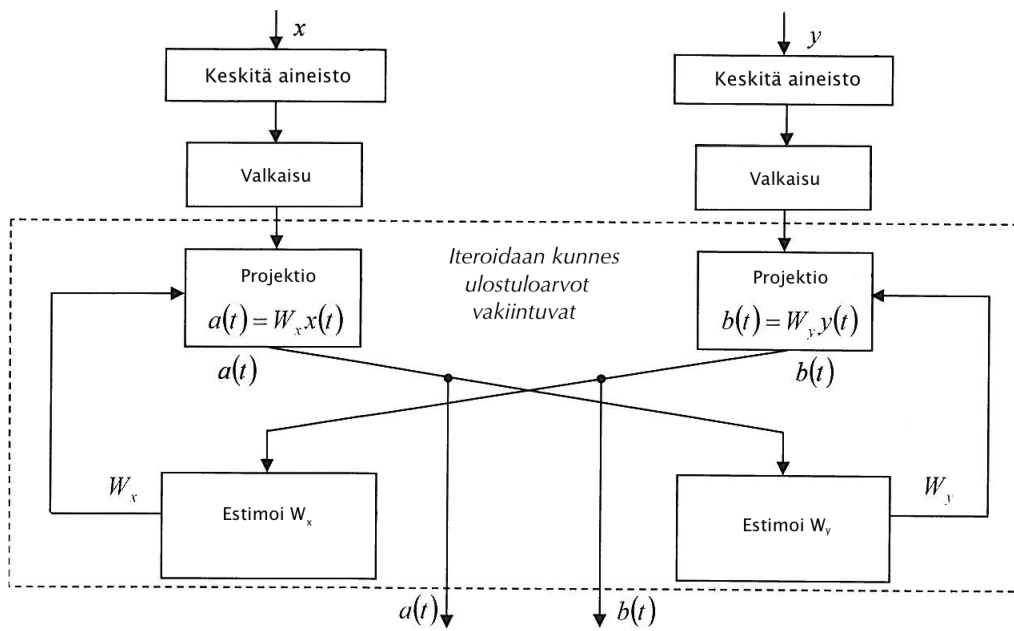
$$W_Y = \sum_t \mathbf{Y}(t)^T * a(t) \quad (3.8)$$

Toisessa lähestymistavassa ratkaistaan kerralla kaikkien ulottuvuuksien $\mathbf{a}(\mathbf{t})$ ja $\mathbf{b}(\mathbf{t})$ ilmentymät. Käytännössä tämä on varsin yksinkertainen laajennus edelliseen: W_X ja W_Y ovat matriiseja, jonka rivit sisältävät riveillään kutakin aineiston ulottuvuutta vastaavan projektiokertoimen. Tällöin vaiheessa kolme saadaan $\mathbf{a}(\mathbf{t})$:stä vektori ($n * 1$ kokoinen matriisi), joka kuvaa kaikkien ulottuvuuksien projektioita riveillään. Soveltaessa toista lähestymistapaa käytetään estimoinnissa seuraavia kuvauksia:

$$W_X = \sum_t b(t) * \mathbf{X}(t)^T \quad (3.9)$$

$$W_Y = \sum_t a(t) * \mathbf{Y}(t)^T \quad (3.10)$$

Kahta viimeistä vaihetta toistetaan, kunnes projektiot $\mathbf{a}(\mathbf{t})$ ja $\mathbf{b}(\mathbf{t})$ eivät enää muutu merkittävästi. Menetelmä on esitetty kaavamaisesti kuvassa 3.1.



Kuva 3.1: Iteratiivinen kanoninen korrelaatioanalyysi. Vertailtavat aineistot keskitetään ja valkaistaan. Tämän jälkeen aineistoja pyritään projisoimaan toisilleen. Kun $\mathbf{a}(t)$ ja $\mathbf{b}(t)$ eivät enää muutu, ilmaisevat ne lähtöaineistojen arvoja projisoituna mahdollisimman läheisille hypertasoille. [20, muokattu]

Luku 4

Toimintapisteiden paljastaminen

Luvussa 2 todettiin toimintapisteiden käytön olevan helppoa. Luvussa esitettiin myös muutama esimerkki menetelmien käytöstä, sekä sen merkityksestä aivojen toiminnalle. Tässä luvussa esitetään miten toimintapisteet voidaan tunnistaa ja toimintapisteitä vastaavat mallit kouluttaa. Luvussa käsitellään yksinkertaista ratkaisua, jossa ihminen määrittää toimintapisteet etukäteen. Luvussa otetaan myös esille kaksi Boltzmannin koneisiin pohjautuvaa ratkaisumallia, jotka molemmat kykenevät paljastamaan toimintapisteet hieman eri toimintaperiaatteilla. Lisäksi luvussa tutustutaan menetelmään, joka pyrkii löytämään aineistosta toimintapisteet suoraan yhdistelemällä yksinkertaisia tilastollisia menetelmiä.

4.1 Manuaalinen toimintapisteiden määrittäminen

Ihmisen todettiin kykenevän ratkaisemaan toimintapisteiden useissa eri tapauksissa. Luvun 2.2 kaikki esimerkit, luokittelu, univaiheen seuranta ja liikkumistyylin määrittäminen ovat sopivan alan asiantuntijan ratkaistavissa, joten teoriassa toimintapisteiden määrittäminen on siirrettävissä konepohjaisesta ratkaisemisesta ihmispohjaiseksi.

Määriteltäessä toimintapisteitä manuaalisesti valitaan käytettävä malli ilmiön rakenteen mukaisesti. Kaikki ilmiöön liittyvä tieto luonnollisesti on syytä ottaa huomioon mallia kehitettäessä. Mikäli siirtymät eri toimintapisteiden välillä eivät tapahdu yhtä todennäköisesti, on syytä käyttää piilotettua Markovin mallia. Jos toimintapisteet siirtyvät yhtä todennäköisesti tilasta toiseen, voi asiantuntijain kirjo -menetelmä olla parempi ratkaisu.

4.1.1 Esimerkki: Univaiheen analysointi

Luvussa 2 esitettiin univaiheiden olevan tulkittavissa toimintapisteinä. Samassa yhteydessä esitettiin miten toimintapisteajattelua kyettiin hyödyntämään univaiheita analysoidessa. Vaikka esitelty malli pohjautuikin aineistolähtöiseen toimin-

tapisteiden etsintään, voidaan toimintapisteajattelua soveltaa myös manuaalisen toimintapiste-etsinnän yhteydessä. On syytä huomata, että tämä esimerkki on täysin teoreettinen, eikä tässä esimerkissä mennä tarkemmin matemaattisiin yksityiskohtiin. Esimerkin tavoitteena on selventää univaiheen analysoinnin olevan toteutettavissa automatisoidusti asiantuntijan opetettua tietokonemallin.

Univaiheen analysointi voidaan toteuttaa manuaalisesti piilotetulla Markovin mallilla, jossa kukin univaihe muodostaa yhden toimintapisteen. Kutakin toimintapistettä taas vastaa malli, joka tuottaa vain tietynlaista aineistoa. Asiantuntijan tehtävä on määrätä todennäköisyydet, joilla toimintapiste (univaihe) vaihtuu toiseksi. Kunkin eri univaiheen malli voidaan kouluttaa yksitellen käyttämällä referenssiaineistoa, joka voi olla yksinkertaisesti EEG-käyrä yön ajalta. Asiantuntija erittelee referenssiaineistosta univaiheet, jonka jälkeen eriteltyä aineistoa käytetään kouluttamaan kutakin univaihetta vastaava malli.

Teoriassa edellä esitetty malli voisi tuottaa järkevän univaiheita simuloivan EEG-käyrän. Kiinnostuksen kohde on kuitenkin käyttää saatua tietoa univaiheen tunnistamiseen aineiston tuottamisen sijaan. Mallin kääntäminen vastakkaiseksi on kuitenkin mahdollista toimintapisteiden ja niiden välisten suhteiden ollessa tiedossa.

Ensimmäinen toimintapisteen määrittäminen joudutaan tekemään vain uuden aineiston ja mallien pohjalta. Mitä lähempänä aineisto on jotakin mallia, sitä todennäköisemmin aineisto on generoitunut mallin kuvaamassa toimintapisteessä. Tämän jälkeen toimintapisteen ei odoteta muuttuneen ennen kuin aineiston arvot eivät ole enää riittävän todennäköisesti mallin tuottamia. Uutta oletettua toimintapistettä arvioitaessa tarkastellaan siirtymätodennäköisyyksiä, sekä todennäköisyyksiä, joilla aineisto olisi tuotettu jossakin toimintapisteessä. Voidaan siis todeta, että piilotetun Markovin mallin parametrit ovat teoriassa riittäviä estimoitaessa univaihetta.

4.1.2 Esimerkki: Luokittelu

Manuaalinen toimintapisteen määrittäminen voi tulla kysymykseen luokittelussa. Mikäli luokittelijan tulee toteuttaa luokittelua useissa merkittävästi erilaisissa tilanteissa, perinteiset yksinkertaiset neuroverkkoratkaisut eivät todennäköisesti tuota toivottua tulosta [2], jolloin on tarpeen käyttää asiantuntijain kirjo -menetelmää apuna.

Otetaan esimerkiksi yksinkertainen tilanne, jossa esineitä luokitellaan kahteen eri huoneeseen riippuen esineen väristä. Esineen muut ominaisuudet määräävät esineen sijainnin huoneessa. Esineiden toivotut sijainnit eivät kuitenkaan ole samat eri huoneissa, vaan kummallakin huoneella on omanlainen järjestyksensä.

Tässä yksinkertaisessa luokittelusovelluksessa on perusteltua käyttää asiantuntijain kirjo -menetelmää. Eri huoneet muodostavat oman toimintapisteensä, joissa esineiden sijoittelusäännöt ovat erilaiset.

4.2 Veräjäyty ehdollinen rajoitettu Boltzmannin kone

Veräjäyty ehdollinen rajoitettu Boltzmannin kone (Gated Conditional Restricted Boltzmann Machine, Gated CRBM) kehitettiin alun perin ohjaamattomaan kuvan muutosten oppimiseen ja mallintamiseen. Menetelmälle annetaan alkuperäinen ja muuttunut kuva. Menetelmä tunnistaa miltä osin kuva on muuttunut, jonka jälkeen menetelmä kykenee hyödyntämään oppimaansa tuottaakseen vastaavan muunnoksen uudelle kuvalle. Muutokset voivat sisältää esimerkiksi liikkumista, suodattamista tai jonkin muun merkittävän muutoksen. Vaikka esimerkkien yhteydessä puhutaankin kuvista, on syytä huomata menetelmän toimivan millä tahansa aineistolla. [23]

Erityisen kiinnostavaksi menetelmän tekee sen laajennettavuus. Rajoitetut Boltzmannin koneet ovat syvien uskomusverkkojen (Deep Belief Nets) perusrakenneosia. Näitä käyttämällä piilotetut neuronit toimivat sisääntuloina uusille piilotetuille neuroneille. Kouluttaminen voidaan suorittaa jo koulutettujen neuronien päälle, jolloin saavutetaan uusia ylempiä toimintapisteen tunnistavia tasoja. Tässä luvussa ei hyödynnetä tätä mahdollisuutta, mutta on syytä laittaa merkille menetelmän laajennettavuus. Luvussa 4.3 palataan ajatukseen käyttämällä toista Boltzmannin koneen muunnelmää. [24, s. 634] [23]

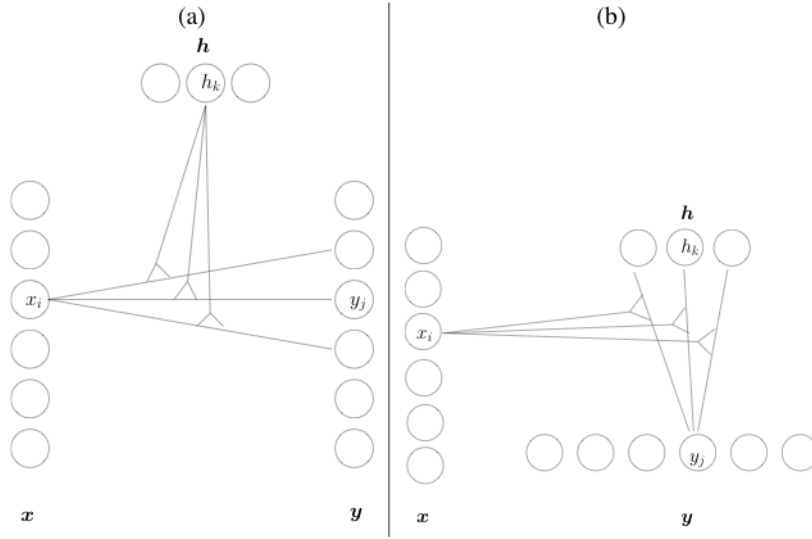
4.2.1 Toimintaperiaate

Tarkastellaan ensin veräjäytyä Boltzmannin konetta, joka on jo opetettu toimimaan halutusti. Veräjäyty Boltzmannin kone koostuu kolmesta omasta tasostaan: sisääntuloneuronit, joiden arvot riippuvat syötettävästä aineistosta; piilotetut neuronit, joiden arvot riippuvat sisääntuloneuronien arvoista ja ulostuloneuroneista, joiden arvot riippuvat sisääntuloneuroneista ja piilotetuista neuroneista. On syytä huomata, että piilotettujen neuronien arvot ovat binäärisiä (voivat saada vain arvot 0 tai 1), kun taas sisääntuloneuronien ja ulostuloneuronien arvot ovat jatkuvia. Kuvassa 4.1 esitetään mallin arkkitehtuuri.

Perusperiaate on minimoida energiefunktio E , jonka arvot riippuvat annetusta aineistosta, sekä neuronien välisistä painoista, jotka määräytyvät opetuksen pohjalta. Opetukseen tai sisääntulevien neuronien tilaan ei voida koskea, joten etsittäessä minimiä tulee ulostuloneuronien tila määrittää sopivasti. Kaikkien arvojen ollessa binäärisiä energiefunktio E määräytyy seuraavasti [23]:

$$E(\mathbf{y}, \mathbf{h}; \mathbf{x}) = \sum_{ijk} W_{ijk} x_i y_j h_k - \sum_{jk} W_{jk}^{\mathbf{y}\mathbf{h}} y_j h_k - \sum_j W_j^{\mathbf{y}} y_j - \sum_k W_k^{\mathbf{h}} h_k, \quad (4.1)$$

jossa x kuvaa kaikkien sisääntuloneuronien joukkoa, h kaikkien piilotettujen neuronien joukkoa, y ulostulovektorien joukkoa ja W eri neuronien välisiä painoja. Vaik-



Kuva 4.1: Veräjäidyn ehdollisen rajoitetun Boltzmannin koneen arkkitehtuuri. Vasemmalla esitetään Boltzmannin koneen käyttöä veräjäitynä regressiona, jossa kukin piilotettu neuroni muuttaa hieman ulostulevan kuvan rakennetta. Oikealla esitetään sama malli siten, että sisääntulevat yksiköt toimivat veräjäverkkona, jotka määräävät miten paljon kukin piilotettu neuroni saa vaikuttaa kuvaan. [23]

ka minimointiongelma vaikuttaakin suhteellisen yksinkertaiselta, ei ongelma ole ratkaistavissa analyttisillä työkaluilla. Tämän takia joudutaan turvautumaan iteratiiviseen algoritmiin, joka pyrkii löytämään lokaalin minimin energiafunktioista. Tätä algoritmia kutsutaan Gibbs-otannaksi. [24]

Gibbs-otannassa oletetaan kullakin neuronilla olevan tietty todennäköisyysjakauma, jonka mukaisesti sen aktivaatio toteutuu. Gibbs-otannan alkutilassa kunkin neuronin oletetaan olevan passiivinen. Kukin muutettavissa oleva neuroni käydään luonnollisessa järjestyksessä läpi, jolloin lasketaan todennäköisyys, jolla neuroni olisi aktiivinen olettaen sen hetkisen tilan olevan pysyvä (piilotetulle neuronille $p(h_k = 1|\mathbf{x}, \mathbf{y})$ ja ulostuloneuronille $p(y_j = 1|\mathbf{x}, \mathbf{h})$). Tätä toistetaan kunnes neuronien tilat ovat konvergoituneet. [24]

Mikäli oletetaan neuronien tilojen olevan vain binäärisiä, saadaan seuraavat yhdistelmätodennäköisyydet tai päivityssäännöt [23]:

$$p(h_k = 1|\mathbf{x}, \mathbf{y}) = \frac{1}{1 + \exp(-\sum_{ij} W_{ijk} x_i y_j)} \quad (4.2)$$

$$p(y_j = 1|\mathbf{x}, \mathbf{h}) = \frac{1}{1 + \exp(-\sum_{ik} W_{ijk} x_i h_k)}, \quad (4.3)$$

joissa x kuvaa kaikkien sisääntuloneuronien joukkoa, h kaikkien piilotettujen neuronien joukkoa, y ulostulovektorien joukkoa ja W eri neuronien välisiä painoja.

On syytä kuitenkin huomata sääntöjen rajoittavan huomattavan paljon käytettävää kuva-aineistoa. Olettaessa ulostuloarvojen olevan binäärisiä oletetaan samalla kuvien olevan täysin mustavalkoisia. Tämän takia on syytä muuttaa ulostuloneuro-nien jakauma normaalijakautuneeksi [23]:

$$p(y_j = 1|\mathbf{x}, \mathbf{h}) = \mathcal{N}(y_j; v \sum_{ik} x_i h_k W_{ijk} + W_j^y, v^2), \quad (4.4)$$

jossa x_j kuvaa sisääntulevaa arvoa, y_j kuvaa ulostuloarvoa, h_k piilotetun muuttujan arvoa, W_{ijk} linkkien painoja ja v varianssia. Tällöin myös energiafunktion muoto muuttuu:

$$E(\mathbf{y}, \mathbf{h}; \mathbf{x}) = \frac{1}{2v^2} \sum_j (y_j - W_j^y)^2 - \frac{1}{v} \sum_{ijk} W_{ijk} x_i y_j h_k - \sum_k W_k^h h_k, \quad (4.5)$$

jossa muuttujat on määritetty vastaavasti kuin edellä.

On syytä havaita, ettei myöskään todennäköisyyksiä $p(y_j = 1|\mathbf{x}, \mathbf{h})$ ja $p(h_k = 1|\mathbf{x}, \mathbf{y})$ voida ratkaista analyttisesti, vaan Gibbsin näytteistys on tässäkin tapauksessa pakollinen operaatio. [24]

Edellä esiteltiin kahden hyvin tavallisen todennäköisyysjakauman käyttöä. On syytä havaita, että nämä jakaumat eivät suinkaan ole ainoat vaihtoehdot, vaan ehdolliset todennäköisyydet voidaan ratkaista mielivaltaisilla jakaumilla. [23]

4.2.2 Kouluttaminen

Koulutettaessa koneoppimismenetelmiä käytetään usein suurimman uskottavuuden (Maximum Likelihood, ML) -estimaattia, jossa pyritään maksimoimaan todennäköisyys, että annetulla aineistolla malli tuottaisi halutun vasteen. Veräjöidyssä Boltzmannin koneessa vastaavana maksimoitavana suureena voidaan pitää keskimääräistä ehdollista logaritmista uskottavuutta [23]:

$$L = \frac{1}{N} \sum_{\alpha} \log p(\mathbf{y}^{\alpha}, \mathbf{h}; \mathbf{x}^{\alpha}), \quad (4.6)$$

jossa N kuvaa näytteiden määrää ja $\mathbf{x}^{\alpha}$ ja $\mathbf{y}^{\alpha}$ kuvaavat jokaista näyteparia. Funktion maksimointi voidaan tehdä millä tahansa sopivalla menetelmällä. Maksimointi voidaan esimerkiksi suorittaa funktion gradienttiin pohjautuvalla menetelmällä, jossa kuljetaan askeleittain funktion suurinta kasvusuuntaa (gradienttia) kohti. Tässä tapauksessa gradienttifunktioksi saadaan:

$$\frac{\partial L}{\partial \mathbf{W}} = \sum_{\alpha} \left(\frac{\partial E(\mathbf{y}^{\alpha}, \mathbf{h}; \mathbf{x}^{\alpha})}{\partial \mathbf{W}} \right)_{\mathbf{h}} - \sum_{\alpha} \left(\frac{\partial E(\mathbf{y}, \mathbf{h}; \mathbf{x}^{\alpha})}{\partial \mathbf{W}} \right)_{\mathbf{h}, \mathbf{y}}, \quad (4.7)$$

jossa $\mathbf{W}$ kuvaa kaikkein linkkien painoja (myös biasointipainot huomioiden).

4.2.3 Sovellukset ja käyttö toimintapisteen hakuun

Kuten voidaan olettaa, opitun muutoksen analysointi on myös mahdollista. Aineistosta voidaan esimerkiksi hakea näytteet, joiden välinen muutos on mahdollisimman pieni. Tällä tavoin on onnistuneesti ryhmitelty luonnollisesti tuotettuja numeroita. Kokeessa menetelmälle annettiin luonnollisesti tuotettu mallinumero, jonka jälkeen menetelmä kävi aineiston läpi. Menetelmä muodosti annetun näytteen ja kunkin aineistossa olevan näytteen välille etäisyyssuureen käyttäen vaadittua muutosta etäisyyssmittarina. Kuvassa 4.2 esitetään saatuja tuloksia. [23]



Kuva 4.2: Veräjäidyn ehdollisen rajoitetun Boltzmannin koneen käyttö samanlaisen kuvan löytämisessä. Vasemmassa ylälaidassa on esitetty kuva, jota vastaavia kuvia on haluttu lähdeaineistosta löytää. Ylärivin muut viisi kuvaa koostuvat koneoppimismenetelmän löytämistä parhaista vastaavuuksista käyttäen muutoksen suuruutta mittana. Alempi rivi koostuu euklidisen etäisyyden (kuvien puhtaan vastaavuden) antamista vastaavuuksista. [23, kuva 11.5]

On merkittävää huomata, että menetelmä kykenee havaitsemaan sekä aineiston käyttäytymisen toimintapisteissä että toimintapisteiden muutokset. Mikäli aineiston näytteet ovat muodostuneet esimerkiksi ajan suhteen, voidaan menetelmällä tunnistaa tilanne, jossa opittu ennuste näytteiden muuttumiselle ei pädekään odotetusti. Näitä yllättäviä muutoksia voidaan pitää toimintapisteen muuttumisen merkkeinä. Toisaalta itse päättelysääntö kuvaa järjestelmän ominaista toimintaa tietystä toimintapisteessä.

4.3 Puolirajoitettu Boltzmannin kone ja syvät uskomusverkot

Tässä luvussa käsitellään myös yhtä Boltzmannin koneen erikoistapausta, puolirajoitettua Boltzmannin konetta (Semi-restricted Boltzmann Machine, SRBM), joka tuottaa tiedon toimintapisteen muuttumisesta varsin erilaisella menetelmällä verrattuna edellisessä luvussa esiteltyyn erikoistapaukseen. Edellisessä luvussa todettiin rajoitettujen Boltzmannin koneiden olevan syvien uskomusverkkojen perusrakenneosia. Tässä luvussa edetään puolirajoitettuun Boltzmannin koneeseen, jota voidaan vastaavasti käyttää syvän uskomusverkon rakenneosana. Tätä ominaisuutta hyödynnetään toimintapisteiden tunnistamisessa.

Tässä luvussa ei tutustuta syvällisesti puolirajoitetun Boltzmannin koneen tai syvien uskomusverkkojen rakenteeseen. Myöskään näiden mallien opettamista ei tässä

yhteydessä käsitellä. Luvun tavoitteena on antaa vaihtoehtoinen näkemys Boltzmannin koneen käytöstä toimintapisteiden etsimiseen. Syvistä uskomusverkoista ja puolirajoitetusta Boltzmannin koneesta löytyy lisätietoa lähteistä [25], [26] ja [27].

4.3.1 Toimintaperiaate

Tutustutaan ensin puolirajoitetun Boltzmannin koneen rakenteeseen. Puolirajoitettu Boltzmannin kone koostuu kahdesta eri neuronipopulaatiosta: näkyvästä ja näkymättömästä. Kaikista näkyvistä neuroneista on suuntaamaton yhteys jokaiseen näkymättömään neuroniin. Lisäksi näkyvät neuronit on linkitetty toisiinsa, mikä mahdollistaa luonnollisemman liukuman kahden vierekkäisen arvon välille. Tätä voidaan havainnollistaa luonnollisella kuvalla, jossa yksittäinen piste voidaan approksimoida vierekkäisten pikselien perusteella. Kuvassa 4.3 kuvataan puolirajoitetun Boltzmannin koneen arkkitehtuuri. [25]

Puolirajoitetun Boltzmannin koneen rakenne voidaan kuvata energiafunktiona [25]:

$$E(\mathbf{v}, \mathbf{h}) = - \sum_i b_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij} - \sum_{i < i'} v_i v_{i'} L_{ii'}, \quad (4.8)$$

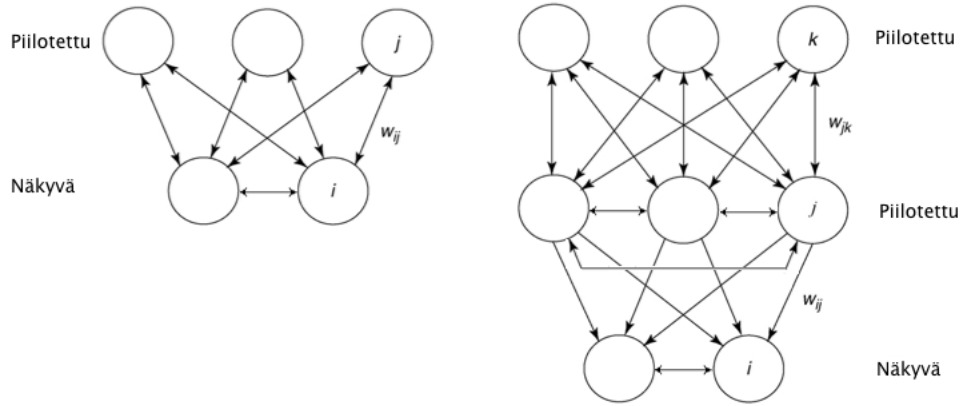
jossa v_i kuvaa näkyvää neuronua, h_j piilotettua neuronua, w_{ij} näkyvän ja piilotetun neuronin välistä painoa ja $L_{ii'}$ kahden näkyvän neuronin painoa. On syytä huomata energiafunktion vastaavan hyvin vahvasti rajoitetun Boltzmannin koneen energiafunktia. Ainoana erona voidaan pitää viimeistä termiä, mikä yhdistää alemman tason neuronit keskenään.

Aineiston rakenteen tunnistamisen kannalta yksittäinen binääriverkko ei kuitenkaan ole tehokas. Menetelmää voidaan laajentaa syväksi uskomusverkoksi, jossa piilotettuja tasoja on useita. Menetelmän kouluttaminen tapahtuu käytännössä täysin samoin kuin yksittäisen puolirajoitetun Boltzmannin koneen. Kun alin piilotettu taso on koulutettu, lisätään tason päälle uusi piilotettu taso. Tässä yhteydessä toisiksi ylimmän tason neuronien välille lisätään yhteydet, jonka jälkeen uusi piilotettu taso voidaan kouluttaa. Kuvassa 4.3 kuvataan puolirajoitettujen Boltzmannin koneiden käyttöä syvän uskomuksen verkkojen rakentamisessa. [25]

4.3.2 Sovellukset ja toimintapistetulkinta

Koulutettaessa verkkoa jokainen uusi taso löytää piirteitä alemman tason syötteestä [25]. Alimmalla piilotetulla tasolla saattaa tapahtua yksinkertaista suodattamista ylemmän tason huolehtiessa kokonaisuuden järjestyksestä. Mikäli verkolle on koulutettu esimerkiksi mustavalkoisina kuvina kuvat neliö ja ympyrä, saattavat tietyt ylimmän tason neuronit ilmaista puhtaasti näytetyn kuvion muotoa abstraktimmalla tasolla.

Syvissä uskomusverkoissa pätee neuronien todennäköisyystulkinta, mikä tekee toimintapisteen päättelystä yksinkertaista. Tunnettaessa näkyvän tason neuronien ak-



Kuva 4.3: Vasemmalla esitetään puolirajoitetun Boltzmannin koneen arkkitehtuuri. Näkyvät neuronit on linkitetty suuntaamattomasti keskenään ja näkymättömien neuronien kanssa. Näkymättömien neuronien välillä ei ole linkkejä. Oikealla esitetään puolirajoitetuista Boltzmannin koneista rakennetun syvän uskomuseverkon arkkitehtuuri, jossa vain ylimmän tason neuronien väliltä puuttuvat linkit. [26, muokattu]

tivaatiot voidaan ensimmäisen piilotetun tason neuronien aktivaatiotodennäköisyydet laskea. Kun ensimmäisen piilotetun tason neuronien aktivaatiotodennäköisyydet ovat tunnettuja, voidaan tutkia toisen tason neuronien aktivaatioita. Tätä toistetaan kunnes kaikkien tasojen neuronien aktivaatiotodennäköisyydet ovat tiedossa. Ylimmän tason aktivoituneet neuronit kuvaavat toimintapisteitä. [25]

Mikäli puolirajoitetuista Boltzmannin koneista muodostuvaa verkkoa halutaan käyttää (opittujen) kuvioiden tuottamiseen, voidaan tämäkin toteuttaa. Ylimmän tason neuronien aktiivisuudet arvotaan tai alustetaan halutuiksi. Tämän jälkeen kahden ylimmän tason välillä suoritetaan Gibbs-otanta, jolla pyritään löytämään kaavassa 4.8 kuvatun energiefunktion minimi. Gibbs-otanta voidaan suorittaa tässä yhteydessä kuitenkin poikkeavasti painottaen näytteistystä vain toiseksi ylimmän tason neuroneihin. Gibbs-otantaa jatketaan taso kerrallaan alaspäin. Tätä voidaan jatkaa näkyvälle tasolle asti, mutta menetelmän ulostuloarvon voi laajentaa reaaliarvoiseksi käyttämällä alimman piilotetun tason aktivointitodennäköisyyttä varsinaisen aktivaation sijaan. [25]

Kokonaisuudessaan syvän uskomuksen verkot mahdollistavat abstraktiohierarkioiden tuottamisen. Alimman tason neuronit oppivat yksittäisiä piirteitä ylemmien tasojen tunnistamiseksi kokonaisuuden. Kääntäen ylemmän tason neuronien aktivaatiot antavat määräksiä alemman tason neuroneille. Analogioita tälle tilanteelle löytyy lukuisia. Yrityksessä johtaja näkee kokonaiskuvan ja parhaimmillaan vaistoa projektit, joihin on syytä panostaa. Tämä näkyy alemmalla tasolla käskyinä, minkä seurauksena alempi taso toteuttaa käskyn suhteellisen riippumattomasti ylemmästä tasosta. Tätäkin yksinkertaisemmin ihmismassaa voidaan pyytää muodostamaan suorakulmio tai neliö määräämättä yksittäisten ihmisten sijaintia muodostettavassa kuviossa.

4.4 Ulkotulon ja laajennetun kanonisen korrelaatio-analyysin käyttö toimintapisteiden estimointiin

Edelliset luvut esittelivät menetelmiä, joiden rakenteessa oli varsin suora yhteys toimintapisteisiin. Tässä luvussa tutustutaan erilaiseen menetelmään, jossa lopputulos ei ole aina aivan yhtä selkeä toimintapistetulkinnaltaan. Menetelmä kuitenkin kykenee paljastamaan aineistosta potentiaalisia toimintapisteitä, sekä muuttujat, jotka määrittävät toimintapisteen muutoksia. Luvussa esitettävä menetelmä on kehitetty Aalto-yliopiston teknillisessä korkeakoulussa. Menetelmän on patentoinut ZenRobotics Oy, ja tämä luku on kirjoitettu ZenRobotics Oy:n patenttihakemuksen pohjalta [20].

Luvussa edetään toimintapisteen löytämisestä yhdessä erikoistapauksessa useisiin lisäyksiin, jotka muuttavat menetelmää merkittävästi. Erikoistapauksessa esitelty menetelmä on vain harvoin käyttökelpoinen, mutta se antaa ymmärrettävän kuvan menetelmästä.

4.4.1 Toimintaperiaate

Tarkastellaan aluksi yksinkertaista tilannetta, jossa pyritään löytämään mustan pisteen liikkumissuuntaa tutkittaessa kahta harmaataustaista kuvaa (ennen liikkumista ja liikkumisen jälkeen). Kuvia analysoidessa voidaan kuvien pikselit muuttaa yksinkertaisesti vektoreiksi, joiden ulottuvuuksien määrä riippuu kuvian leveydestä ja korkeudesta. On syytä huomata, ettei tässä yhteydessä muodosteta *leveys* x *korkeus* kokoista matriisia vaan vektorin, jonka pituus on *leveys* * *korkeus*. Vektorissa yksittäiset alkiot kuvaavat sitä vastaavan pikselin intensiteettiä asteikolla 0 (valkoinen) - 1 (musta). Kuvataan vektoreita $\mathbf{x}$:llä (alkutila) ja $\mathbf{y}$:llä (lopputila). Otetaan vektorien välillä ulkotulo:

$$\mathbf{z} = \mathbf{x}\mathbf{y}^T, \quad (4.9)$$

jossa muodostunut $\mathbf{z}$ kuvaa matriisia, jonka koko on *leveys* * *korkeus* x *leveys* * *korkeus*. Muodostuneen matriisin koko on siis ilmiömäisen suuri. Mikäli käsiteltäisiin kuvaa, jonka koko on 10 * 10, saataisiin muodostuneen matriisin alkioden määräksi 10000.

Ulkotulon tulkitseminen on tässä yhteydessä varsin yksinkertaista. Ulkotuloa otettaessa kerrotaan jokaisella $\mathbf{x}$ -vektorin alkioilla vektorin $\mathbf{y}$ alkioita. Kukin sarake (ja rivi) kuvaa siten molemmista kuvista muodostunutta yhdistelmää tietyn pikselin suhteen. Mikäli kerrotaan pienellä arvolla (harmaalla pikselillä) suurta arvoa (musta pikseli), saadaan lopputulokseksi myös harmaa pikseli. Vastaavasti pienellä arvolla kerrotaessa toista pientä arvoa saadaan myös hyvin pienen arvon. Sen sijaan suhteellisen suurella intensiteettiarvolla kertoessa toista suurta intensiteettiarvoa muutokset ovat vähäisiä.

Saatua matriisia voidaan käsitellä laajempaan aineistona, jonka jokainen sarake kuvaa yhtä vektoria. Tästä aineistosta voidaan selvittää pääkomponenttianalyysillä

suurimmat vaihtelun suunnat, jotka kuvaavat pikselin liikkumista aineistossa. Vastaavasti kuin luvussa 4.2 voidaan tätä tietoa käyttää toimintapisteen määrittämiseen. Yksittäinen kuvassa tapahtuva muutos kyetään löytämään, jota hyödyntämällä voidaan muodostaa ennuste tulevasta kuvasta. Mikäli ennuste ei täsmää odotetun kanssa, on toimintapiste saattanut muuttua.

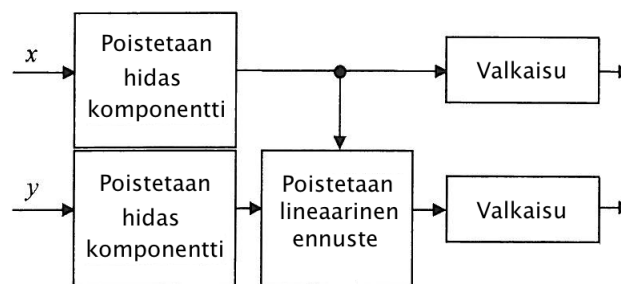
4.4.2 Menetelmän laajentaminen

Edellä kuvattu menetelmä sisältää useita heikkouksia. Aineistoa varten tehtiin useita oletuksia, ja jo puolivälissä voitiin havaita käsiteltävien alkioden määrän kasvavan suuresti. Tätä varten menetelmä sisältää useita eri tekniikoita aineiston esikäsittelyyn. Myös vaihtoehtoinen tapa käsitellä alkioita on olemassa.

Ensimmäinen laajennus koostuu raaka-aineiston esikäsittelystä. Koska tavoitteena on löytää toimintapisteen ja niihin vaikuttavia muuttujia, suodatetaan aineistosta kiinnostavat piirteet esiin. Toinen laajennus sisältää ulkotulon jälkeen kerrotun aineiston vaihtoehtoisen käsittelyn muilla ohjaamattoman oppimisen menetelmillä. Toiseen laajennukseen sisältyy myös vaihtoehtoinen tapa edetä ilman ulkotuloa pitäen alkioden määrän pienempänä. Kolmas laajennus keskittyy aineiston jälkikäsittelyyn sopivalla tekniikalla. Kuten havaittiin, aineiston laadusta riippuen lisäykset eivät aina ole välttämättömät, mutta useimmissa tapauksissa niiden käyttö on suositeltavaa.

4.4.3 Alustaminen ja aineiston esikäsittely

Esikäsittelyn yhteydessä luodaan pohja myöhemmälle aineiston työstölle. Koska menetelmää sovelletaan tilanteissa, joissa halutaan löytää erilaiset toimintapisteen, voidaan toimintapistettä kuvaamattomat ylimääräiset piirteet suodattaa pois. Kuvassa 4.4 esitetään kaavamaisesti aineiston esikäsittely.



Kuva 4.4: Alustamisen vaiheet. Ensimmäisessä vaiheessa järjestelmälle syötettävistä komponenteista poistetaan hitaasti muuttuvat piirteet. Seuraavaksi komponenttien välinen lineaarinen korrelaatio poistetaan. Viimeisessä vaiheessa sääntö tulevat komponentit valkaistaan. [20, muokattu]

Järjestelmälle syötettävissä komponenteissa voi esiintyä hitaita piirteitä, jotka muut-

tavat aineiston keskiarvoa ajan suhteen. Nämä hitaat piirteet voidaan löytää käyttämällä hitaiden piirteiden analyysiä (Slow Feature Analysis) [28]. Tämän jälkeen hitaasti muuttuvat piirteet voidaan poistaa syötettävästä aineistosta.

Mikäli aineistossa esiintyy joidenkin komponenttien välillä aina lineaarista korrelaatiota, poistetaan lineaarinen korrelaatio ennen seuraavaa vaihetta. Lineaarinen korrelaatio itsessään ei sisällä informaatiota muuttujien toimintapisteistä. Mikäli korrelaatioissa tapahtuu muutoksia, on informaatio sen sijaan arvokasta; erilainen korrelaatio saattaa olla ominaista tietyssä toimintapisteessä.

Aineiston komponentit voidaan tämän jälkeen valkaista. Valkaisu kuvataan matemaattisesti operaationa, jossa aineisto käsitellään siten, että sen kovarianssimatriisi on yksikkömatriisi [28]. Käytännössä operaatio voidaan tulkita siten, että muuttujissa tapahtuvat muutokset ovat samassa asemassa, vaikka reaaliuutos olisi ollut huomattavan suuri.

Näiden muokkausten jälkeen aineistoon voidaan tarvittaessa lisätä epälineaarinen komponentti. Epälineaarinen komponentti näkyy yhtenä toimintapisteinä, joka voidaan asettaa alkuarvoksi hakiessa muita toimintapisteitä. Tämän operaation on havaittu nopeuttavan huomattavasti toimintapisteiden löytämistä.

4.4.4 Menetelmän 1. ilmentymä

Menetelmän ensimmäisessä ilmentymässä pyritään erottamaan toimintapisteiden vaihtuminen ja niiden sijainnit annetuista muuttujista. Tämä ilmentymä vastaa hyvin pitkälti aikaisemmin esiteltyä menetelmän toimintaperiaatetta yksinkertaisessa tilanteessa.

Esikäsitellyille signaaleille toteutetaan ulkotulo (kaava 4.9). Saadun matriisin kukin rivi ja sarake sisältää koko toisen aineiston kerrottuna yhdellä toisen aineiston arvoista. Mikäli aineisto on valkaistu, korostavat vaihtelut toisiaan samassa "mittakaavassa". Näiden vaihtelujen tunnistaminen voidaan tehdä pääkomponenttianalyysillä kuten aiemmassa esimerkissä. Vaihtelut voidaan toteuttaa myös tutkimalla lokaalia varianssia, käyttämällä lineaarista hitaiden piirteiden analyysiä (Linear Slow Feature Analysis) tai soveltamalla kanonista korrelaatioanalyysiä. Käytettävä menetelmä riippuu oletetusta muutoksen luonteesta.

Piikit lokaalissa varianssissa indikoivat muutosta toimintapisteessä. Myös selkeästi erilaiset paikalliset varianssit antavat viitteitä erilaisesta toimintapisteestä. Kanonisella korrelaatioanalyysillä voidaan etsiä jonkin ulkotulolla saadun aineiston ja jonkin aineistoon liittyvän muuttujan välistä korrelaatiota.

Menetelmiä voi myös yhdistellä. Pääkomponenttianalyysillä löydetuille vektoreille voidaan projisoida aineisto, jonka jälkeen lokaalia varianssia voi tutkia matalaulottuvuuksisessa avaruudessa.

4.4.5 Menetelmän 2. ilmentymä

Menetelmän toinen ilmentymä ratkaisee esimerkissä esiintyneen ongelman suuressa ulottuvuusmäärän kasvussa. Ulkotulon ottamisen sijaan menetelmässä pyritään hakemaan aineistosta muutoksia iteratiivisella menetelmällä suoraan projisoimalla aineistoja toisilleen. Ilmentymän lopputuloksena saadaan molempia sisääntuloja vastaavat ulostulot, jotka on pyritty projisoimaan hypertasoille. Hypertasot muotoutuvat iteroinnin myötä siten, että niille projisoidut vektorit kuvaavat muuttunutta toimintapistettä. On syytä havaita tämän vastaavan huomattavan paljon luvussa 3 esiteltyä kanonista korrelaatioanalyysia. Menetelmän toinen ilmentymä on esitetty kuvassa 4.5.

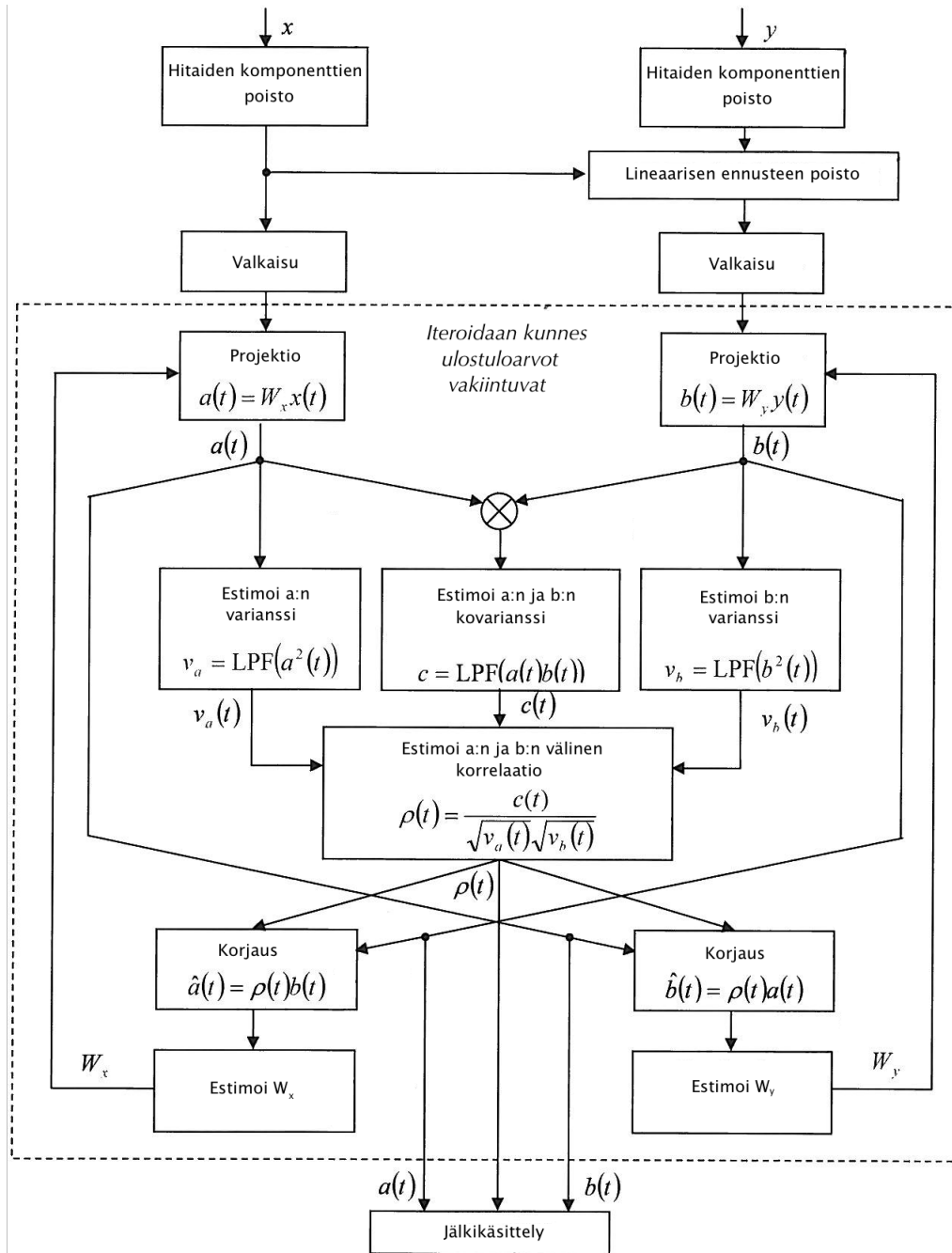
Verrattuna tavalliseen kanoniseen korrelaatioanalyysiin vaiheiden 3 ja 4 välissä ratkaistaan paikallinen korrelaatio, joka ilmaisee aineistojen vastinalkioiden välistä korrelaatiota funktion tietyllä arvolla. Paikallinen korrelaatio ρ on ratkaistavissa, mikäli aineiston muuttujien paikalliset varianssit ja kovarianssit ovat tunnettuja:

$$\rho(t) = \frac{c(t)}{\sqrt{v_a(t)}\sqrt{v_b(t)}}, \quad (4.10)$$

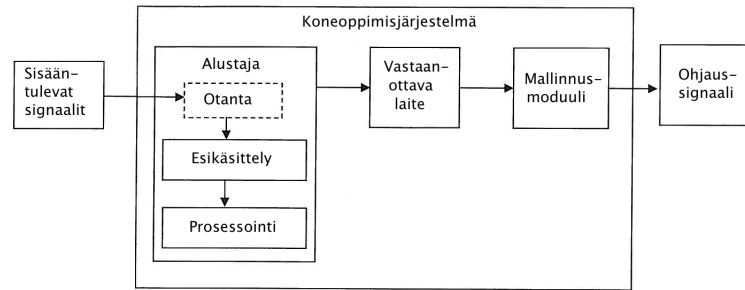
jossa $\mathbf{c}(t)$ kuvaa aineiston vastinvektorien paikallista kovarianssia, $\mathbf{v}_a$ vektorin $\mathbf{a}(t)$ paikallista varianssia ja $\mathbf{v}_b(t)$ vektorin $\mathbf{b}(t)$. Kovarianssin ratkaisemista varten aineistojen väliset vastinalkiot kerrotaan keskenään. Tulossa muodostetun aineiston paikallinen varianssi kuvaa paikallista kovarianssia. On syytä havaita, että otettaessa varianssia yksittäisestä pisteestä saadaan varianssiksi 0, sillä yksittäinen aineiston piste ei sisällä vaihtelua. Paikallinen varianssi tulee sen sijaan estimoida. Tämä voidaan tehdä esimerkiksi tutkimalla muutoksia ajanhetken t ympäristössä. Tämä voidaan tehdä myös alipäästäsuodattamalla (Low-Pass Filter, LPF) aineiston neliötä. Paikallinen kovarianssi ratkeaa vastaavasti alipäästäsuodattamalla keskenään alkioittain kerrottuja aineistoja.

Käsiteltäessä kanonista korrelaatioanalyysiä oletettiin vertailtavien aineistojen koostuvan ajallisesti tai tilallisesti muuttuvista piirteistä. Tätä indikoitiin kuvaamalla kaikkia muuttujia funktioina muuttujan t suhteen. Samalla todettiin myös ulostuloarvojen olevan funktiomuotoisia. Yhdistämällä funktiomuotoisuus paikallisen korrelaation kanssa saadaan oiva työkalu eri toimintapisteen tunnistamiseen; mikäli paikallinen korrelaatio muuttuu jossakin pisteessä merkittävästi, ovat muuttujien suhteet muuttuneet, mikä viestii mahdollisesta toimintapisteen vaihtumisesta.

Paikallista korrelaatiota käytetään lisäksi uusien projektiomatriisien muodostamiseen kertomalla projisoidut arvot arvoja vastaavilla paikallisilla korrelaatioilla. Tämä voidaan käsittää siten, että itseisarvoltaan suuri paikallinen korrelaatio ohjaa projektiota tiettyyn suuntaan. Vastaavasti itseisarvoltaan pienet lokaalin korrelaation arvot pienentävät vastaavien muuttujien merkitystä jatkossa.



Kuva 4.5: Toimintapisteiden tunnistaminen iteratiivisella proseduurilla. Vertailtavista aineistoista poistetaan hitaat komponentit, jolloin aineiston keskiarvo pysyy vakiona eri näytteiden välillä. Tämän jälkeen suorat lineaariset riippuvuudet aineistojen välillä poistetaan, minkä jälkeen aineisto valkaistaan. Valkaistut vertailtavat aineistot projisoidaan hypertasolle kertomalla aineisto W_X :llä tai W_Y :llä. Projisoitujen aineistojen välinen paikallinen korrelaatio ratkaistaan paikallisen varianssin ja paikallisen kovarianssin avulla (LPF, Low-Pass Filter, kuvaa alipäästösuodattamista). Paikallisella korrelaatiolla kerrotaan projisoidut aineistot. Käyttämällä tulon muodostamaa korjattua aineistoa muodostetaan uudet projektioatriisit W_X :llä ja W_Y . Luvussa 4.4.5 kuvataan menetelmä tarkemmin. [20, muokattu]



Kuva 4.6: Koneoppimisjärjestelmä, joka hyödyntää esiteltyä kehittämää menetelmää toimintapisteiden tunnistamiseen. Vasemmalla kuvataan sisääntulevia signaaleja, jotka syötetään koneoppimisjärjestelmälle. Koneoppimisjärjestelmän ensimmäisessä erotetaan hyväksyttävät arvot kelpaamattomista (kuvassa vaihe *otanta*). Kelpuutetut arvot esikäsitellään ja käsitellään. Tuotetut arvot tulkitaan *vastaanottavassa laitteessa*. *Mallinnusmoduulissa* muodostetaan vastinesignaali, joka näkyy *vastesignaalina*. [20, muokattu]

4.4.6 Jälkikäsitteily

Aineiston jälkikäsitteily voi sisältää hyvin erilaisia toimenpiteitä. Tavoitteena on saada aineistosta mahdollisimman selkeää ja helposti tulkittavaa. Menetelmässä esitetään aineiston jälkikäsitteilyn koostuvan normalisoinnista, ortogonalisoinnista ja ulottuvuuksien vähentämisestä.

Näiden lisäksi menetelmää käytettäessä voidaan useita muitakin operaatioita toteuttaa. Mikäli menetelmän tulos kuvaa ryhmittynyttä aineistoa, voidaan aineiston ryhmät erottaa odotuksen maksimointi (Expectation Maximization, EM) tai k-means-algoritmeilla. [1, sivut 135-144]

4.4.7 Koneoppimisnäkökulma

Mikäli aineisto sisältää viitteitä toimintapisteiden läsnäolosta, algoritmi todennäköisesti löytää ne. Sen sijaan menetelmän tulkitseminen saattaa edellyttää asiantuntijaa, mikä ei kaikissa tilanteissa ole kuitenkaan hyvä. Tämän takia joitakin valmiita suunnitelmia menetelmän käytölle on perinteisten koneoppimismenetelmien kanssa. Vaikka perinteiset koneoppimismenetelmät yleensä epäonnistuvat toimintapisteiden määrittämisessä, saattavat ne soveltua hyvin opitun toimintapisteen hyödyntämiseen kuten havaittiin luvussa 2. Kuvassa 4.6 esitetään kaavamainen esimerkki koneoppimisjärjestelmästä.

Esitelty algoritmi saa arvot jostakin ilmiöstä. Otanta-vaiheessa poistetaan suoraan selkeästi epäkelvot näytteet. Tämän vaiheen merkitys korostuu luonnollisen ilmiön ollessa kyseessä, sillä anturin antama virheellinen arvo on syytä automaattisesti jättää pois. Aineisto esikäsitellään ja käsitellään luvussa kuvatuilla menetelmillä. Saatu signaali saattaa sisältää tietoa nykyisestä toimintapistestä tai sen muuttumisesta. Tämä riippuu käytetystä menetelmäyhdistelmästä. Signaali syötetään sisääntuloar-

vona seuraavalle käsittelijälle, joka muokkaa sen sopivaksi mallinnusmoduulia varten. Mallinnusmoduuli tuottaa vastesignaalin, joka voi näkyä järjestelmän ulkopuolella esimerkiksi ohjaussignaalina.

Luku 5

Yhteenvedo ja pohdinta

Tutkielmassa on nähty toimintapisteisiin pohjautuvan ajattelun voima. Maailman voidaan nähdä koostuvan erilaisista toimintatiloista, joita on kandidaatintyön aikana nimitetty toimintapisteiksi. Intuiitiivisesti ongelma ei vaikuta ihmismielelle haastavalta. Ihmismieli kykeneekin löytämään välittömästi sopivan käyttäytymisen. Tämän ajattelun valjastaminen tietokoneelle on ollut tavoitteena jo pitkään, mutta vasta muutaman viime vuoden aikana on otettu merkittäviä edistysaskelia. Työssä käsiteltiin sekä toimintapisteiden käyttöä että niiden hakemista koneoppimismenetelmin.

Luvussa 2 todettiin tunnettujen toimintapisteiden käytölle olevan tiedossa useita esimerkkejä. Kirjallisuuden havaittiin myös sisältävän useita erilaisia käytännön sovelluksia, jotka edellyttivät toimintatilojen tunnistamista tai käyttöä ilman ulkoista asiantuntijaa. Luvussa tutustuttiin muutamaan merkittävään koneoppimismenetelmään, joista kukin omasi erilaiset lähtöoletukset ja käyttökohteet; täten menetelmien paremmuudesta ei voida tehdä suurta analyysiä. Asiantuntijain kirjo muunnelmiseen sallii yksinkertaisen veräjäverkon toimintapisteen määrittämiseen. Heikkoutena menetelmä huomioi vain sillä hetkellä saadun näytteen tutkimatta edellistä toimintapistettä. Tämän heikkouden korjasi piilotetun Markovin mallin muunnelma, sisään-ulos piilotettu Markovin malli, joka kouluttamisen jälkeen vaihtoi toimintapistettä vain tietyllä todennäköisyydellä ja silloinkin riippuen sisääntuloarvoista. Lisäksi luvussa 2 laitettiin merkille kirjallisuudesta löytyneet viitteet hippokampuksen ja aivokuoren merkityksestä erilaisten toimintatilojen tunnistajina.

Toimintapisteiden tunnistamista ja oppimista käsiteltiin syvällisemmin luvussa 4. Luvussa esiteltiin neljä merkittävää lähestymistapaa. Ensimmäisessä lähestymistavassa ihmisen osaaminen valjastettiin toimintapisteiden hakuun. Tällöin ihminen määritteli erilaiset toimintapisteet, joita voitaisiin hyödyntää luvussa 2 esitettävillä menetelmillä. Toisessa lähestymistavassa käytettiin Ronald Memisevicin ja Geoffrey Hintonin kehittämää veräjäytystä ehdollista rajoitettua Boltzmannin konetta aineiston muutosten havaitsemiseen. Menetelmä oli suunniteltu ohjaamattomaan kuvamuutosten tunnistamiseen, mutta luvussa havainnollistettiin sen merkitystä myös toimintapisteiden haussa. Menetelmän todettiin kykenevän havaitsemaan kahden

tilan välillä esiintyvät muutokset. Muutokset itsessään indikoivat toimintapisteelle ominaista käyttäytymistä, kun taas muutoksissa esiintyvät muutokset indikoivat toimintapisteen vaihtumista. Kolmas menetelmä oli varsin erilainen suhteessa edelliseen. Siinä käytettiin puolirajoitettuja Boltzmannin koneita syvän uskomusverkon rakenneosina. Syvän uskomusverkon ylimmät neuronit taasen sopivalla koulutuksella saattoivat ilmaista yleisellä tasolla alemman tason mutkikasta toimintaa. Neljäs lähestymistapa koostui ZenRobotics Oy:n patentoimasta menetelmäkirjosta, joka kykeni tunnistamaan toimintapisteen vaihtumisen ja tunnistamaan järjestelmän käyttäytymisen toimintapisteessä.

Ihmisen päättäessä toimintapisteet pysytään varsin kaukana varsinaisesta oikeasta toimintapisteiden automaattisesta oppimisesta. Sen sijaan kolme muuta esiteltyä menetelmää kykenevät tunnistamaan muuttuneet toimintapisteet ja toimintapisteen jälkeiset käyttäytymiset aineistolähtöisesti ilman mitään ennakkotietoa aineistosta. Menetelmien välillä ei työn aikana suoritettu varsinaista vertailua, eikä menetelmien puolueeton vertailu ole todennäköisesti mahdollista; menetelmät sisältävät paljon muokattavia osia, jotka riippuvat käyttötarkoituksesta.

Veräjöity Boltzmannin kone kykenee havaitsemaan toimintapisteen muuttumisen reaaliajassa. Mikäli ennuste ei vastaa odotettua riittävästi, on toimintapiste todennäköisesti muuttunut. Tällä tavoin käytettynä veräjöity Boltzmannin kone nojautuu Gibbs-otantaan, joka on iteratiivinen ja suoritusaikaa vaativa. Valmiin aineistonkin tapauksessa menetelmän käyttö tällä tavoin on hidasta.

Puolirajoitettujen Boltzmannin koneiden ja syvien uskomusverkkojen käyttö mahdollistaa hyvin nopean päättelyn saaduista arvoista. Päättely voidaan suorittaa tilastollisesti todennäköisyyslaskennan keinoin, mikä on suuri etu muihin menetelmiin nähden. Nopea päättely on kuitenkin mahdollista vain järjestelmän kouluttamisen jälkeen. Täten päättelyä varten järjestelmälle tulee syöttää laajasti aineistoa koulutusta varten, mikäli päättelyn haluaa tehdä etukäteen. Mikäli päättelyn haluaa toteuttaa kerätylle aineistolle, saadaan parhaimmillaan järjestelmän eri toimintapisteet esille vain tutkimalla ylintä piilotettua neuronien tasoja.

ZenRobotics Oy:n patentoima menetelmä on varsin kiinnostava poikkeama näistä kahdesta. Menetelmä koostuu useista tunnetuista tilastotieteen menetelmistä, jotka lopulta paljastavat toimintapisteen muutokset.

Voidaan todeta toimintapisteen tunnistamisen ja käytön vaatimien perustyökalujen olevan olemassa. Erityisesti toimintapisteen tunnistamiseen soveltuvilla menetelmillä tuotetut käytännön sovellukset ovat vähäisiä, mutta työssä esiteltyt esimerkit eri käyttötapauksista osoittavat menetelmiin pohjautuvan potentiaalin.

Kirjallisuutta

- [1] Alpaydin, E., Introduction to Machine Learning, *The MIT Press*, ISBN 0-262-01211-1 2004
- [2] Jacobs, R. A.; Jordan, M. I.; Nowlan S. J. ja Hinton, G. E. Adaptive Mixtures of Local Expert, *Neural Computation*, Vol. 3, sivut 79-87, 1991
- [3] Jordan, M. I. ja Jacobs, R. A., Hierarchical Mixtures of Experts and EM Algorithm, *Neural Computation*, Vol 6., sivut. 181-214 1994
- [4] Bishop, C. M., Pattern Recognition and Machine Learning *Springer*, ISBN-13: 978-0387-31073-2 2006
- [5] Bengio, Y. ja Frasconi, P., An input output HMM architecture, *Advances in Neural Information Processing Systems*, 1995
- [6] Hiltunen, E.; Holmberh, P.; Kaikkonen, M.; Lindblom-Ylänne, S.; Nienstedt, W. ja Wähälä, K., Galenos, Ihmiselimistö kohtaa ympäristön, *Werner Söderström Osakeyhtiö*, ISBN 951-0-30252-X, 6.-7. painos, 2006
- [7] Kohlmorgen, J.; Mueller, K. R. ja K. Pawelzik, Analysis of drifting dynamics with neural network hidden Markov models, *Advances in Neural Information Processing Systems*, Vol. 7 (1995), sivut 427-434., 1998
- [8] Taylor, G., W. ja Hinton, G. E., Factored Conditional Restricted Boltzmann Machines for Modeling Motion Style, *International Conference on Machine Learning*, 2009
- [9] Rolls, E. T., An attractor network in the hippocampus: Theory and neurophysiology, *Learning & Memory* 14:714-731, 2007
- [10] Shastri, L., Episodic Memory and Cortico-Hippocampal Interactions *Trends in Cognitive Sciences*, Vol. 6, numero 4, Huhtikuu 2002
- [11] O'Keefe, J. ja Dostrovsky, J., The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat, *Brain Research*, Vol. 34, sivut 171-175, 1971

- [12] Taube, J. S.; Muller, R. U. ja Ranck J. B., Jr, Head-Direction Cells Recorded from the Postsubiculum in Freely Moving Rats. I. Description and Quantitative Analysis, *The Journal of Neuroscience*, Helmikuu 1990
- [13] William E. Skaggs ja Bruce L. McNaughton, Spatial Firing Properties of Hippocampal CA1 Populations in an Environment Containing Two Visually Identical Regions, *The Journal of Neuroscience*, 18(20):8455-8466, Lokakuu, 1998
- [14] Robert Desimone, Visual attention mediated by biased competition in extrastriate visual cortex, *Neuron*, Vol. 22, tapaus 4, sivu 751 1999
- [15] Szabo, M.; Almeida, R.; Deco G. ja Stetter M., Cooperation and biased competition model can explain attentional filtering in the prefrontal cortex, *European Journal of Neuroscience*, Vol, 19, sivut 1969-1977, 2004
- [16] Buehlmann A. ja Deco G., The Neuronal Basis of Attention: Rate versus Synchronization Modulation, *The Journal of Neuroscience*, 28(30), sivut 7679-7686, Heinäkuu 2008
- [17] Silvanto, J.; Cowey, A.; Lavie, N. ja Walsh V., Striate cortex (V1) activity gates awareness of motion *Nature Neuroscience*, Vol 8, Numero 2, Tammikuu 2005
- [18] Hotelling, H., Relations Between Two Sets of Variates, *Biometrika*, Vol. 28, Numero 3/4, sivut. 321-377, Joulukuu 1936
- [19] Borga, M., Canonical Correlation - a Tutorial, *Department of Biomedical Engineering*, Linköping University, Saatavilla: <http://www.imt.liu.se/magnus/cca/> Haettu 5.4.2010, Tammikuu 2001
- [20] Valpola, H., An Efficient Method for Learning Latent Variables that Describe Relations of Input Variables, *Patentihakemus*, 2010
- [21] Vía, J.; Santamaria, I. ja Pérez, J., Canonical Correlation Analysis (CCA) Algorithms for Multiple Data Sets: Application to Blind Source Equalization, *European Signal Processing Conference*, Syyskuu 2005
- [22] Vía, J.; Santamaria, I. ja Pérez, J., A Robust RLS Algorithm for Adaptive Canonical Correlation Analysis, *Proc. of 2005 IEEE International Conference on Acoustics, Speech and Signal Processing*, Maaliskuu 2005
- [23] Memisevic, R. ja Hinton, G. E., Unsupervised Learning of Image Transformations, *Department of Computer Science, University of Toronto*, 2006
- [24] Haykin S., Neural Networks and Learning Machines, *Prentice Hall*, ISBN-10 0-13-129376-1, 2009
- [25] Osindero, S. ja Hinton, G. E., Modeling image patches with a directed hierarchy of Markov Random Fields *Department of Computer Science, University of Toronto*, 2008

- [26] Hinton, G. E., Learning Multiple Layers of Representation, *Trends in Cognitive Sciences*, Vol. 11, sivut 428-434, 2007
- [27] Hinton, G. E.; Osindero, S.; Teh, Y-W, A Fast Learning Algorithm for Deep Belief Nets *Neural Computation*, Vol. 18, sivut 1527-1554, 2006
- [28] Wiskott, L. ja Sejnowski, T.J., Slow feature analysis: Unsupervised learning of invariances, *Neural Computation*, Vol. 14, sivut 715-770, 2002