Teknillinen Korkeakoulu
Mat-2.177 Operaatiotutkimuksen projektityöseminaari
Loppuraportti, kevät 2002

Puuttuvan datan täyttäminen matriisiin EM-algoritmilla

Palautettu 29.4.2002

Työryhmä:
Ilkka Lehmusvirta 50448D
Jukka Lehmusvirta 48924W
Ilkka Männistö 48474F
Jaakko Saarinen 46338U

Tiivistelmä

Projektin tavoite oli luoda Excel-työkalu puuttuvien havaintojen estimointiin aikasarjadatassa. Tässä käytettiin hyväksi EM-algoritmia ja tulos on tarkoitettu finanssimaailman käyttöön. Projektin toimeksiantajana oli Sampo Pankki Oyj ja projektityö liittyi Teknillisen korkeakoulun kurssiin Mat-2.177 Operaatiotutkimuksen projektityöseminaari.

Projektityön aikana laadittiin osittain alkuperäisiin tavoitteisiin nähden suorituskyvyltään supistettu Excel-rutiini Visual Basic –ohjelmointikielellä. Supistamiseen päädyttiin lähinnä aikaresurssien riittämättömyyden vuoksi kurssin määrittelemissä puitteissa. Rutiini on kykenevä estimoimaan puuttuvat alkiot ainoastaan yhteen aikasarjaan yleisen tapauksen sijasta, jossa puuttuvia havaintoja voi olla useammassa aikasarjassa. Kärsineestä laajuudesta huolimatta ohjelma on käyttökelpoinen puuttuvien havaintojen täydentämisessä monissa ongelmatilanteissa. Työkalun valmistuttua sen pohjalta tehdyissä tulosanalyyseissa kuitenkin havaittiin puutteita menetelmän soveltumisessa todellisiin aikasarjoihin lähinnä niiden epästationaarisen käyttäytymisen vuoksi. Tällaisissa tilanteissa EM-algoritmi ei enää ole kykenevä estimoimaan puuttuvia alkioita luotettavasti. Käytännössä muodostuu tarve muokata aikasarjaa esimerkiksi differoinneilla tai jakamalla se osiin, joten käyttäjältä vaaditaan tarkkaavaisuutta ohjelman soveltamisessa ja tulosten tulkitsemisessa.

Sisällysluettelo

1. Johdanto

1.1. Ongelman tausta

Finanssimaailmassa esimerkiksi rahoitusinstrumenttien tai osakkeiden tuotto-odotuksia tai arvon muutoksia seurataan päivittäin. Toisinaan nämä arvot eivät ole saatavissa esimerkiksi kansallisen pyhäpäivän vuoksi, jolloin rahoitusmarkkinoilla ei käydä kauppaa. On kuitenkin tärkeää, että historiallinen data eri instrumenteista on täydellistä erinäisten tunnuslukujen laskemiseksi.

Nykyisiä puuttuvan informaation estimoimisen keinoja ovat edellisen päivän arvon käyttö tai yksinkertainen interpolointi tai ennustaminen kahden arvon välillä. Ryhmämme tehtävänä onkin olemassa olevan teorian pohjalta rakentaa työkalu, joka on edellisiä tapoja hienostuneempi ja luotettavampi.

1.2. Työn tavoite

Ryhmämme tavoitteena on Sampo Pankin toimeksiannosta luoda työkalu puuttuvan informaation estimoimiseen. Työkalu käyttää hyväksi EM-algoritmin teoriaa.

Työkalun pohjalla oleva teoria kehitettiin vuonna 1977[1]. Expectation–Maximization tai EM-algoritmi on monikäyttöinen ja paljon tutkittu matemaattinen algoritmi, joka soveltuu muun muassa rahoitusmaailmassa puuttuvan informaation tarkkaan estimointiin.

Toimeksiantajan pyynnöstä EM-algoritmia hyväksikäyttäen rakennetaan Excel-funktio Sampo Pankki Oyj:n käyttöön puuttuvan informaation estimoimiseksi. Syötteenään Excel-työkalu saa esimerkiksi tuotto-odotusten ja näiden kovarianssien alkioiltaan puutteellisen matriisin. Matriisissa on sarakkeina rahoitusinstrumentit ja riveinä ajanhetkiä. Tämä jako johtuu Excel-ohjelman rajoituksista, jossa sarakkeita voi olla vain äärellinen määrä ja aikasarjat saattavat olla hyvinkin pitkiä. Seuraavaksi työkalu suorittaa EM-algoritmin ja palauttaa lopulta täydellisen matriisin, johon on siis estimoitu puuttuvat alkiot.

Työkalun ohjelmoimiseen käytetään Excelin omaa kieltä Visual Basic:a (VBA), jonka on alustavasti todettu suoriutuvan myös vaativammista matriisilaskuista.

[1] Dempster, Laird & Rubin: "Maximum Likelihood from incomplete data via the EM algorithm", Journal of the Royal Statistical Society, Series *B*, 39, 1-38.

1.3. Työn mahdolliset rajaukset

Työkalu pystyy suoriutumaan tilanteista, joissa alkioita puuttuu ensimmäisestä sarakkeesta eli yhdestä rahoitusinstrumentista. Käyttäjän tehtäväksi jää järjestää matriisi siten, että informaatioltaan puutteellinen aikasarja sijaitsee ensimmäisessä sarakkeessa.

Tähän rajoitukseen päädyttiin ohjelmointityön edistyessä, kun huomattiin, että alkuperäisen suunnitelman mukaisen yleisen EM-algoritmin ohjelmoiminen osoittautui erittäin työlääksi nykyisellä ohjelmointitekniikalla. Asiasta neuvoteltiin toimeksiantajan kanssa, joka suhtautui asiaan myönteisesti. Lisäksi toimeksiantaja arvioi karkeasti, että rajoituksien kanssa työkalu pystyy suoriutumaan 80 prosentista ongelmatapauksista.

1.4. Työjako ja toteutustapa

Työn suoritusta varten neljän hengen ryhmämme jaettiin kahteen osaan, jotka osittain rakensivat projektia itsenäisesti. Projektipäällikköä ei ryhmämme keskuudesta nimetty. Kaksi osaryhmää tapasi säännöllisesti esittämään tuotoksensa ja keskustelemaan mahdollisista ongelmista. Koska kyse oli ohjelmointiprojektista, tapaamisissa pyrittiin yhdistämään molempien ryhmien osaohjelmia ja sopimaan uusista tavoitteista ja osaohjelmista.

Tämä työnjako osoittautui toimivaksi ja ohjelman eri osa-alueiden yhdistäminen sujui melko ongelmattomasti. Lisäksi ryhmän jakaminen kahteen oli onnistunut ratkaisu, joka mahdollisti vielä joustavamman työnjaon osaryhmien sisällä. Joustavuudesta huolimatta voidaan arvioida ryhmän jäsenten työmäärät yhtä suuriksi.

Osaongelmien asetteleminen noudatti pääpiirteittäin toimeksiantajalta saamaamme EM-algoritmin teoriaa kuvaavaa dokumenttia. Dokumentista pyrittiin jaksoittain tunnistamaan ohjelmointiteknisiä kokonaisuuksia, joita jaetaan ryhmille. Tämänkaltaisia kokonaisuuksia olivat esimerkiksi kovarianssimatriisin muodostaminen. Ohjelmointikielenä koko toteutuksen ajan pysyi alun perin valitsemamme Visual Basic. Tämä valinta osoittautui toimivaksi, kunhan alussa ilmenneet ongelmat oli selvitetty. Näitä ongelmia olivat juuri matriisien tallettamiseen ja käsittelyyn liittyvät operaatiot.

1.5. Raportin rakenne

Tästä eteenpäin projektiraportti noudattaa seuraavaa rakennetta. Luvussa 2 käsitellään kattavammin EM-algoritmin teoriaa sekä teorioita, jotka ovat vaikuttaneet EM:n muodostumiseen. Seuraavassa luvussa 3 käsitellään työn käytännön toteutusta ja luodaan syvempää katsausta käytettyihin VBA:n työkaluihin, ohjelman rakenteeseen

ja käsitellään syvemmin poikkeamia alkuperäisestä suunnitelmasta. Luvussa 4 esitellään alkuperäinen ja toteutunut aikataulu. Toteutuneen aikataulun yhteydessä käy ilmi työn ja ohjelmoinnin etenemisen vaiheet. Luku 5 käsittelee riskejä eli siinä vertaillaan alkuperäistä riskikarttaa toteutuneisiin riskeihin. Luvussa 6 käsitellään projektin tuloksia. Tässä luvussa käsitellään myös ohjelman käytettävyyttä ja luotettavuutta sekä muuta analyysia. Luku 7 sisältää pohdintoja ja kehitysideoita projektista. Viimeisenä raportin osana ovat liitteet, joissa ovat ohjelman käyttöohje sekä ohjelmakoodi kommentoituna.

2. Teoria

Kuvataan seuraavaksi EM-algoritmin teoriaa. Tarkoituksena on siis täyttää puuttuvat havainnot aikasarjoihin mahdollisimman luotettavasti. Lähtöoletuksena on, että data on normaalijakautunutta.

2.1. Puuttuvat havainnot

Oletetaan, että tietyllä ajanhetkellä on käytettävissä K aikasarjaa, joista kussakin on T havaintoa, joista osa voi puuttua. Kootaan aikasarjat matriisiin, joka siten on dimensioiltaan (KxT).

$$\mathbf{Z} = \begin{bmatrix} z_{11} & .. & .. & z_{1T} \\ : & . & & : \\ : & & . & : \\ z_{K1} & .. & .. & z_{KT} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \\ : \\ : \\ \mathbf{z}_K^T \end{bmatrix}, \tag{1}$$

jossa matriisin vaakarivejä eli aikasarjoja on merkitty vektoreilla $\mathbf{z}_t$ (Kx1).

Määritellään seuraavaksi täydellinen datamatriisi **R**, joka sisältää kaikki **Z**:ssa olleet havainnot ja lisäksi puuttuvat havainnot ennustettuina. Merkitään **Z**:a vastaavasti myös **R**:n vaakarivejä pystyvektoreilla $\mathbf{r}_t$.

$$\mathbf{R} = \begin{bmatrix} r_{11} & .. & .. & r_{1T} \\ : & . & & : \\ : & & . & : \\ r_{K1} & .. & .. & r_{KT} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^T \\ : \\ : \\ \mathbf{r}_K^T \end{bmatrix}. \tag{2}$$

Jos puuttuvaa informaatiota jossakin aikasarjassa t ei ole, pätee, että $\mathbf{r}_t = \mathbf{z}_t$.

2.2. Suurimman uskottavuuden estimointimenetelmä

Tutkitaan aluksi tilannetta, jossa alkuperäisestä matriisista **Z** ei puutu havaintoja, jolloin **R** = **Z**. Seuraavaksi oletetaan, että millä tahansa ajanjaksolla t = 1,2,...,T

vektori $\mathbf{r}_t$ noudattaa monimuuttujaista normaalijakaumaa keskiarvovektorilla $\boldsymbol{\mu}$ ja kovarianssimatriisilla $\mathbf{S}$. Vektorin $\mathbf{r}_t$ tiheysfunktio on tällöin muotoa

$$p(\mathbf{r}_t) = (2p)^{-k/2}|\mathbf{S}|^{-k/2}\exp\left[-\tfrac{1}{2}(\mathbf{r}_t-\boldsymbol{\mu})^T\mathbf{S}^{-1}(\mathbf{r}_t-\boldsymbol{\mu})\right]. \quad (3)$$

Tehdään oletus, että eri ajanjaksojen välillä ei esiinny tilastollista riippuvuutta, jolloin voidaan kirjoittaa yhteisjakauman tiheysfunktio

$$p(\mathbf{r}_1,...\mathbf{r}_T \mid \boldsymbol{\mu},\mathbf{S}) = \prod_{t=1}^{T} p(\mathbf{r}_t) = (2p)^{-Tk/2}|\mathbf{S}|^{-T/2}\exp\left[-\tfrac{1}{2}\sum_{t=1}^{T}(\mathbf{r}_t-\boldsymbol{\mu})^T\mathbf{S}^{-1}(\mathbf{r}_t-\boldsymbol{\mu})\right]. \quad (4)$$

Tämä kuvaa todennäköisyystiheyttä, kun systeemin määräävät parametrit $\boldsymbol{\mu}$ ja $\mathbf{S}$ tiedetään (eli jakauma ehdolla $\boldsymbol{\mu}$ ja $\mathbf{S}$). Määritellään parametrivektori

$$? = [\boldsymbol{\mu}, \mathbf{S}]. \quad (5)$$

Tarkoituksena on arvioida kyseinen parametrivektori *?* mahdollisimman luotettavasti, kun käytössä on datamatriisi **R**. Tätä varten tarvitaan *?*:n likelihoodfunktio, joka olettaa datan annetuksi ja parametrit satunnaismuuttujiksi. Muuten se on matemaattisesti identtinen yhteisjakauman tiheysfunktion (4) kanssa. Merkitään likelihoodfunktiota

$$L = L(\boldsymbol{\mu},\mathbf{S} \mid \mathbf{r}_1,...\mathbf{r}_T). \quad (6)$$

Kun kyseinen likelihoodfunktio maksimoidaan, saadaan parametriestimaattori $?_{MLE}$, joka kaikkein todennäköisimmin on konstruoinut käytettävissä olevan datamatriisin **R**. Tätä metodia kutsutaan suurimman uskottavuuden menetelmäksi (MLE = Maximum Likelihood Estimation).

Tavoitteena on siis maksimoida likelihoodfunktiota *L*. Likelihoodfunktion sijasta on usein helpompaa maksimoida likelihoodfunktion luonnollista logaritmia, koska tällöin tulomuotoinen funktio saadaan summamuotoiseksi. Lisäksi normaalijakauman tapauksessa esiintyvä eksponenttifunktio eliminoituu samalla ja saadaan

$$\ln L(\boldsymbol{\mu},\mathbf{S} \mid \mathbf{r}_1,...\mathbf{r}_T) = l(\boldsymbol{\mu},\mathbf{S} \mid \mathbf{r}_1,...\mathbf{r}_T) = -\tfrac{1}{2}TK\ln(2p) - \tfrac{T}{2}\ln(|\mathbf{S}|) - \tfrac{1}{2}\sum_{t=1}^{T}(\mathbf{r}_t-\boldsymbol{\mu})^T\mathbf{S}^{-1}(\mathbf{r}_t-\boldsymbol{\mu}) \quad (7)$$

Maksimoidaan tätä *?*:n suhteen. Siten ratkaistavana on yhtälöt

$$\frac{\partial}{\partial\boldsymbol{\mu}} l(\boldsymbol{\mu},\mathbf{S} \mid \mathbf{r}_1,...\mathbf{r}_T) = 0 \quad (8)$$

ja

$$\frac{\partial}{\partial\mathbf{S}} l(\boldsymbol{\mu},\mathbf{S} \mid \mathbf{r}_1,...\mathbf{r}_T) = 0. \quad (9)$$

Ratkaisuna saadut suurimman uskottavuuden estimaattorit ovat

$$\hat{\boldsymbol{\mu}} = [\bar{r}_1, \bar{r}_2, ..., \bar{r}_K]^T, \quad (10)$$

missä $\bar{r}_i$ on otoskeskiarvo ajanjakson (1,*T*) yli ja

$$\hat{\mathbf{S}} = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{r}_t - \hat{\boldsymbol{\mu}})(\mathbf{r}_t - \hat{\boldsymbol{\mu}})^T \ . \tag{11}$$

2.3. Puuttuvien havaintojen estimointi EM-algoritmilla

Suurimman uskottavuuden menetelmää aineiston tunnuslukujen määrittämiseksi ei voida suoraan käyttää, kun matriisista **Z** puuttuu havaintoja. Tähän ongelmaan päästään käsiksi soveltamalla EM-algoritmia.

Kun täydelliset (tai täydennetyt) havainnot $\mathbf{r}_1,\ldots,\mathbf{r}_K$ muodostavat satunnaisotoksen monimuuttujaisesta normaalijakaumasta, algoritmi perustuu täydellisen datan riittävään statistiikkaan (complete data sufficient statistics), jonka määrittelevät suureet

$$T_1 = \sum_{i=1}^{K}\mathbf{r}_i = K\bar{\mathbf{r}} \tag{12}$$

ja

$$T_2 = \sum_{i=1}^{K}\mathbf{r}_i\mathbf{r}_i^T = (K-1)\mathbf{S} + K\bar{\mathbf{r}}\bar{\mathbf{r}}^T \ , \tag{13}$$

missä **S** on otosvarianssien summamatriisi.

EM-algoritmi puuttuvan datan täydentämiseksi matriisin **R** muodostamista varten koostuu kahdesta iteratiivisesta vaiheesta, joille on useampia, hieman toisistaan poikkeavia määrittelyjä. Tässä työssä käytetään viitteessä[2] esitettyä tapaa, jonka mukaan vaiheita kutsutaan ennustamiseksi ja estimoinniksi.

1. **Ennustaminen**

   Jollakin käytettävissä olevalla parametrivektorin estimaatilla $\hat{q}_{EM}^{\,i}$ ennustetaan puuttuvien havaintojen arvot ja täydennetään ne datamatriisiin. Lasketaan tämän avulla uudet arvot riittävää statistiikkaa kuvaaville suureille $\mathbf{T}_1$ ja $\mathbf{T}_2$.

2. **Estimointi**

   Käytetään edellisessä kohdassa uusilla puuttuvien parametrien arvoilla täydennetystä datamatriisista saatuja $\mathbf{T}_1$:ä ja $\mathbf{T}_2$:a uuden parametriestimaatin $\hat{q}_{EM}^{\,i+1}$ määrittämiseen, jonka jälkeen tarkistetaan suppenemiskriteeri. Jos se ei täyty, palataan kohtaan 1.

Ensimmäiseksi tarvitaan estimaatti $\hat{q}_{EM}^{\,0}$. Periaatteessa tämän valinnalle ei ole rajoitteita. On kuitenkin järkevää määritellä parametreille jokin käypä alue O ja valita ensimmäinen estimaatti siten, että

[2] Johnson ja Wichern: "Applied multivariate statistical analysis"; Prentice Hall 1992.

$$\hat{q}_{EM}^{0} \in \Omega. \tag{14}$$

Yleinen menetelmä alkuestimaatin määräämiselle on laskea se olemassaolevasta datasta **Z** käyttämällä kaavoja (10) ja (11), joissa keskiarvot $\bar{r}_i$ korvataan keskiarvoilla $\bar{z}_i$. Puuttuvia havaintoja sisältävien aikasarjojen on kunkin sisällettävä vähintään yksi havainto, jotta estimoiminen olisi mahdollista. Muuten kovarianssimatriisiin tulee nolla-alkioita ja estimoiminen ei onnistu.

Tarkastellaan nyt askelten etenemistä matemaattisesti.

1. Ennustaminen

Merkitään jokaisen täydennetyksi tulevan aikasarjan $\mathbf{r}_i$ puuttuvia alkioita $\mathbf{r}_i^{(1)}$:llä ja käytettävissä olevia alkioita $\mathbf{r}_i^{(2)}$:lla. Siten

$$\mathbf{r}_i^T = \left[\mathbf{r}_i^{(1)T}, \mathbf{r}_i^{(2)T}\right]. \tag{15}$$

Käytettävissä on *i*:s parametriestimaatti $\hat{q}^i = \left[\hat{\boldsymbol{\mu}}^i, \hat{\mathbf{S}}^i\right]$. Sen ja käytettävissä olevan datan avulla puuttuvat havainnot voidaan arvioida

$$\hat{\mathbf{r}}_i^{(1)} = E\left(\mathbf{r}_i^{(1)} \mid \mathbf{r}_i^{(2)}; \hat{\boldsymbol{\mu}}, \hat{\mathbf{S}}\right) = \hat{\boldsymbol{\mu}}^{(1)} + \hat{\mathbf{S}}_{12}\hat{\mathbf{S}}_{22}^{-1}\left(\hat{\mathbf{r}}_i^{(2)} - \hat{\boldsymbol{\mu}}^{(2)}\right). \tag{16}$$

Puuttuvien havaintojen varianssien ja kovarianssien estimaatit voidaan nyt laskea edellisen tuloksen avulla seuraavasti:

$$\mathbf{r}_i^{(1)}\hat{\mathbf{r}}_i^{(1)T} = E\left(\mathbf{r}_i^{(1)}\mathbf{r}_i^{(1)T} \mid \mathbf{r}_i^{(2)}; \hat{\boldsymbol{\mu}}, \hat{\mathbf{S}}\right) = \hat{\mathbf{S}}_{11} - \hat{\mathbf{S}}_{12}\hat{\mathbf{S}}_{22}^{-1}\hat{\mathbf{S}}_{21} + \hat{\mathbf{r}}_i^{(1)}\hat{\mathbf{r}}_i^{(1)T} \tag{17}$$

ja

$$\mathbf{r}_i^{(1)}\hat{\mathbf{r}}_i^{(2)T} = E\left(\mathbf{r}_i^{(1)}\mathbf{r}_i^{(2)T} \mid \mathbf{r}_i^{(2)}; \hat{\boldsymbol{\mu}}, \hat{\mathbf{S}}\right) = \hat{\mathbf{r}}_i^{(1)}\hat{\mathbf{r}}_i^{(2)T}. \tag{18}$$

Tämän jälkeen voidaan laskea tarkistetut arvot $\mathbf{T}_1$:lle ja $\mathbf{T}_2$:lle uusia aikasarjaennusteita $\mathbf{r}_i$ käyttäen kaavoista (12) ja (13).

2. Estimointi

Lasketaan uusi estimaatti $\hat{q}^{i+1}$ parametrivektorille edellisessä vaiheessa saaduista täydellisen datan riittävän statistiikan suureista.

$$\hat{\boldsymbol{\mu}}^{i+1} = \frac{\hat{\mathbf{T}}_1^i}{K} \tag{19}$$

ja

$$\hat{\mathbf{S}}^{i+1} = \frac{\hat{\mathbf{T}}_2^i}{K} - \hat{\boldsymbol{\mu}}^{(i+1)}\hat{\boldsymbol{\mu}}^{(i+1)T}. \tag{20}$$

Tämän jälkeen tarkistetaan suppenemiskriteeri. Sitä voidaan tutkia parametrivektorista monilla eri tavoilla. Yleisesti voidaan sanoa, että suppeneminen on ollut riittävää, kun peräkkäiset parametriestimaatit $\hat{q}^i$ ja $\hat{q}^{i+1}$ ovat riittävän lähellä

toisiaan. Tavallisinta lienee verrata *i*:ttä ja *i*+1:ttä keskiarvoa toisiinsa tai esimerkiksi viittä edellistä keskiarvoestimaattia uusimpaan, jolloin vältetään iterointien liian aikainen lopettaminen, jos suppenemisen värähtely jatkuu pitkään. Rajana voi myös olla esimerkiksi iteraatiokierrosten lukumäärä.

2.4. EM-algoritmin ominaisuuksia

Teoriaa muodostettaessa pohjaoletuksena oli analysoitavat datasarjat generoineen jakauman normaalijakautuneisuus. Tämä ei yleistettäessä ole ongelma, sillä on osoitettu, että vaikka todellinen jakauma ei olisikaan normaali, parametriestimaatit ovat silti asymptoottisesti konsistentteja, vaikka eivät välttämättä olekaan asymptoottisesti tehokkaita[3]. Sen sijaan aikasarjojen epästationaarisuus, eli parametrien *?* aikariippuva käyttäytyminen aiheuttaa ongelmia estimaattien luotettavuudessa.

EM-algoritmin keskeisiä ominaisuuksia ovat:

1. Menetelmä on stabiili.
2. Se takaa likelihoodfunktion arvon monotonisen kasvun.
3. Siten likelihoodfunktion maksimi on EM-algoritmille stabiili piste ja jos se on ylhäältä rajoitettu, likelihoodfunktiotulosten sarjalla on äärellinen yläraja *L**.
4. Kyseinen *L** voi olla globaali tai paikallinen maksimi, stationaarinen arvo tai jokin iteroinnin aikana muodostuva vakaa arvo.
5. Suurin haittatekijä on melko hidas suppeneminen.

Kohta 4. ei muodostune ongelmaksi, kun parametrin alkuestimaatti valitaan järkevästi kaavan (14) mukaisesti.

3. Toteutus

3.1. Käytetyt työkalut

Projektin päätavoitteena ollut ohjelma toteutettiin Microsoft Excelissa toimivana VBA-sovelluksena. Koska lopullinen ohjelma haluttiin toimimaan Excelissä, oli Visual Basic luonnollinen valinta ohjelmointikieleksi. Excelin sisäänrakennettuja funktioita voidaan kutsua suoraan VBA-koodista, joten tarvittavien perusmatriisilaskujen suorittamiseen on hyvät työkalut jo valmiiksi. On vain huomioitava, että halutut matriisit tulee tallentaa Variant-tietotyyppiin, jotta niitä

[3] RiskMetrics™ –Technical Document, 4th edition.

voidaan käsitellä Excelin funktioilla. Matriiseja käsitellään käytännössä Excelin taulukoina. Excelin funktioiden kutsuminen VBA-koodista tapahtuu kutsulla:

Application.WorksheetFunction.Function(Arguments).

Ohjelmassa on yleisesti käytetty lukujen tiedostotyyppinä Double-tyyppiä, jonka tarkkuus riittää varmasti ohjelman sovellusalueella esiintyvissä ongelmissa. Kokonaisluvut ovat luonnollisesti Integer-tyyppiä. Suuri osa muuttujista on toteutettu Variant-datatyyppinä helpottamaan Excelin omien funktioiden kutsumista, vaikka tämä johtaakin hieman epäselvempään koodiin. Jotta ohjelma suoriutuisi suuristakin matriiseista, on muuttujien määrittely toteutettu dynaamisesti; muuttujien koko määrätään vasta, kun matriisin koko on tiedossa. Periaatteessa ei siis ole rajoituksia sille, kuinka suuria aikasarjoja käsitellään. Käyttäjän on kuitenkin huomioitava ohjelman käytännön rajoitukset, eli suurin alkioiden määrä, jonka Excel voi vielä käsitellä, ja se, että tulokset eivät välttämättä parane, vaikka lisää havaintoaineistoa otettaisiinkin iteraatioprosessiin mukaan. Näihin seikkoihin palataan myöhemmin.

3.2. Käyttövaatimukset

Jotta ohjelma olisi mahdollisimman käyttökelpoinen Sampo Oyj:lle, sen on hyvä olla käytettävissä Microsoft Excelissä, joka on yksi finanssimaailman perustyökaluista. Tämä ei ole ongelma, koska ohjelma toteutetaan Microsoft Exceliin liitettävänä VBA-ohjelmana. Data, jota ohjelmalla käsitellään, koostuu esimerkiksi osake- tai valuuttakursseja sisältävistä aikasarjoista. Normaalimuodossa Sampo Oyj:ssa aikasarjat on tallennettu Excelillä luettaviin arkistoihin, joissa aikasarjat ovat valmiiksi tallennettuina ylhäältä alas. Siten yhden valuutta- tai osakekurssin tarvitsema tila on yksi pystysarake Excelissä. Ohjelma on siis toteutettu siten, että se lukee aikasarjat ylhäältä alas. Käytännössä ohjelma lukee matriisimuodossa käyttäjän haluamat aikasarjat muuttujiin, ja sen jälkeen iteroi EM-menetelmällä puuttuvat arvot yhteen aikasarjaan. Sinänsä käytetty algoritmi huomioi syötettyjen aikasarjojen korreloinnin, joten periaatteessa käyttäjän ei tarvitse huolehtia siitä, sisältävätkö syötetyt aikasarjat informaatiota puuttuvista arvoista. Käyttövaatimuksena on myös, että ohjelmalla pystyttäisiin käsittelemään hyvinkin suuria matriiseja; aikasarjat saattavat olla finanssikäytössä suuruudeltaan satoja havaintoja. Ohjelman datatyyppeinä onkin käytetty dynaamisia taulukoita käyttämällä Visual Basicin ReDim-käskyä alkuperäistä datamatriisia luettaessa.

EM-algoritmin perustuessa iteratiiviseen prosessiin ohjelman on tunnistettava myös tilanteet, joissa iteraatio ei suppene toivotulla tavalla. Ohjelmaan voidaan siksi asettaa maksimimäärä suoritettaville iteraatioille, jotta ohjelma ei jää loputtomaan silmukkaan. Käyttäjä määrittää sekä iteraatioiden maksimimäärän että

suppenemiskriteerin. Käyttäjän on myös mahdollista määrittää maksimiaika, jonka ohjelma käyttää iterointiin. Käytännön kokeissa on kuitenkin havaittu ohjelman tarvitsevan suurillakin matriiseilla maksimissaan vain kymmeniä sekunteja iterointiin.

3.3. Algoritmin toteutus

Toteutetussa ohjelmassa algoritmin iteraatio-osasta vastaa Iterate-niminen funktio. Iteraatiosilmukka on toteutettu tämän funktion sisään; funktio pitää itse huolen prosessin suppenemisesta ja lopettamisesta. Funktio itsessään on kirjoitettu Microsoft Excelin moduliin (Module 1). Funktio on muotoa:

```
Sub Iterate(initialCov, initialMean, originalTimeSerie, rowNum, columnNum, miss, missNumber, MaxIterations, convergenceLimit)
```

Seuraavassa on selvitetty funktion argumentit:

initialCov	alkuperäinen kovarianssimatriisi
initialMean	alkuperäinen keskiarvovektori
originalTimeSerie	käsiteltävät aikasarjat matriisina
rowNum	datamatriisissa olevien vaakarivien määrä
columnNum	datamatriisissa olevien pystyrivien määrä
miss	vektori, jonka alkioina on puuttuvien arvojen sijainnit
missNumber	puuttuvien arvojen lukumäärä
MaxIterations	iteraatiokierrosten lukumäärä
convergenceLimit	suhteellisen virheen muutosraja.

Koodi itsessään noudattaa hyvin suoraviivaisesti EM-algoritmin matemaattista periaatetta. Jokainen iteraatiokierros koostuu kahdesta vaiheesta: Puuttuvien arvojen uusien estimaattien laskeminen ja tämän jälkeen parametrien *?* estimointi uudesta matriisista. Puuttuvien arvojen estimaattien päivitys on näistä huomattavasti yksinkertaisempi toimenpide. Seuraavassa on esitetty algoritmin for-silmukka.

```
For i = 1 To MaxIteration
        Jaetaan kovarianssimatriisi osiin
        For j = 1 To MissNumber
                Käydään läpi datamatriisi ja täydennetään siihen uudet
estimaatit
        Next i
        For k = 1 To Rows
                For m = 1 To Rows
                        Muodostetaan uusi kovarianssimatriisi EM–algoritmin
                        mukaan
                Next m
        Next k

        Tarkistetaan konvergenssi; terminoidaan silmukka tarvittaessa

Next i
```

Monimutkaisin osa ohjelmaa on uuden kovarianssimatriisin laskeminen. Uusia parametreja ei voida laskea suoraan uusilla estimaateilla täydennetystä datamatriisista, vaan ne lasketaan suurimman uskottavuuden estimointiperiaatteen mukaan. Ohjelman täytyy siis huomioida $\mathbf{T}_1$-vektoria ja $\mathbf{T}_2$-matriisia (kaavat 12 ja 13) muodostettaessa, käytetäänkö $\mathbf{r}\mathbf{r}^T$:n vai $\mathbf{r}$:n estimaattia. Tämä on otettu huomioon koodissa useiden if-lauseiden avulla.

Konvergointi testataan vertaamalla parametrien suhteellista muutosta kahden kierroksen välillä. Testisuure on siis muotoa:

$$Error = \left| \frac{m(k-1)}{m(k)} - 1 \right| \tag{21}$$

Jos tämä testisuure on pienempi kuin käyttäjän määrittelemä konvergointiraja, silmukka päätetään. Kaavassa $\mu(k)$ on kierroksen k keskiarvo.

Koodi kokonaisuudessaan on liitteessä 2.

3.4. Poikkeamat suunnitelmasta

Projektityön alkuperäinen tarkoitus oli luoda ohjelma, joka tiettyjen reunaehtojen vallitessa löytää suurimman uskottavuuden estimaatit puuttuville data-alkioille. Tällaisenaan ohjelman toteutuksesta olisi kuitenkin tullut liian raskas kurssin

puitteissa suoritettavaksi. Sampo Oyj:n edustajien kanssa käydyissä neuvotteluissa päädyttiin vaatimusten helpottamiseen. Ohjelman käyttötarkoitusta muutettiin siten, että sillä pystyttäisiin paikkaamaan yksi kokonainen aikasarja. Tämä olisi hyödyllistä finanssikäytössä, kun tehdään analyysia esimerkiksi osakekursseista, jolloin tarvitaan tiedot koko osakkeen historian ajalta. Paikattava aikasarja voi periaatteessa olla vaikka kuinka puutteellinen, mutta referenssiaikasarjoista ei saa puuttua dataa. Matemaattisesti tarkasteltuna ei myöskään välttämättä päädytä kovin luotettaviin tuloksiin, jos paikataan yksi aikasarja kokonaiseksi, ja sitten käytetään tätä aikasarjaa toisen aikasarjan paikkaamiseen. Ohjelma ei tällaisenaan myöskään tunnista, mistä aikasarjasta dataa puuttuu; ohjelma olettaa käyttäjän laittavan paikattavan aikasarjan ensimmäiseksi, ruudun vasempaan laitaan.

Muilta osin ohjelma on toteutettu suunnitelman mukaan.

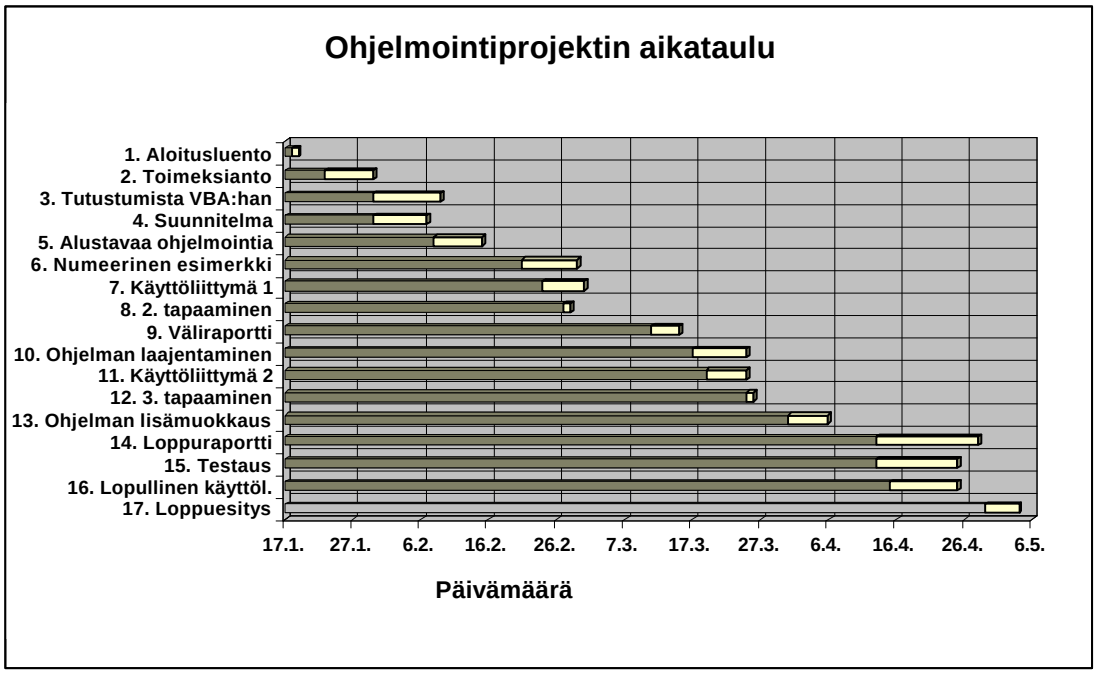
4. Projektin aikataulu ja eteneminen

Projektisuunnitelmassa luotu melko yleinen aikataulu on esitetty alla (Taulukko 1).

Taulukko 1. Projektin alkuperäinen aikataulu

Aika	Tavoite tai päämäärä
27.2.2002	Projektisuunnitelman valmistuminen. Lisäksi tapaaminen Sammossa, jossa käsitellään alustavassa tutustumisessa esiin tulleita ongelmia ja kysymyksiä.
15.3.2002	Projektista tuotetaan väliraportti, joka esitellään koululla ja mahdollisesti myöhemmin myös Sammossa. Lisäksi molemmissa tilaisuuksissa pyritään esittelemään alustava toimiva versio projektityön tavoitteena olevasta Excel-työkalusta.
1.4.2002	Seuraava versio työkalusta on valmis. Mahdollinen lisätapaaminen Sammossa, jos siihen on tarvetta.
26.4.2002	Projektityö on valmis, eli projektin tavoite on saavutettu ja loppuraportti jätetään koululle. Tässä vielä saattaa olla tarvetta käyttöliittymän hiomiseen.
3.5.2002	Projektin kulku ja tuotos esitellään koululla seminaaritilaisuudessa. Tällöin käsitellään tarkemmin työkalun toimintaa ja käytettyjä ratkaisuja.
3.5.2002-1.6.2002	Projekti esitellään ja tuotos palautetaan Sampoon, ellei tätä ole jo tehty koulussa järjestetyssä seminaaritilaisuudessa.

Alkuperäiseen aikatauluun tuli luonnollisesti muutoksia projektin edetessä. Lisäksi aikataulu tarkentui ja muodostui kokonaisuuksiltaan pienemmäksi. Seuraavassa (Kuva 1) on esitelty projektin toteutunut aikataulu osaohjelmien, raporttien ja tapaamisten suhteen. Aikataulun muodostamisessa on käytetty apuna Excel-ohjelmaa. Aikataulun esittämistä seuraa tarkempi kommentointi kustakin työvaiheesta.



Kuva 1. Projektin aikataulu ja työvaiheet

1. Aloitusluento: Kurssin aloitusluennolla professori Ahti Salo esitteli kurssin tavoitteet ja alustavan aikataulun sekä kertoi yleisesti projektien piirteistä. Tällöin mainittiin muutamia toimeksiantajayrityksiä, jotka olivat alustavasti sitoutuneet tarjoamaan projektityön aiheita oppilasryhmille.

2. Toimeksianto. Toimeksiantovaihe alkoi eri yrityksien edustajien esitellessä projektityön aiheita. Sammon edustaja Petri Viertiö kertoi Sammon toimeksiannosta, jonka ryhmä Sampo 1 toteutti. Tällä luennolla muodostettiin projektiryhmä, jolle Viertiö lupasi rajata projektityötä varten aiheen. Toimeksiantovaihe päättyi Sammossa järjestettyyn palaveriin, jossa olivat läsnä molemmat Sammon projektiryhmät. Tällöin projektiryhmä sai toimeksiannon aiheen ja lisäksi mietittiin alustavaa lähestymistapaa ongelman ratkaisemiseksi.

3. Tutustumista VBA:han: Koska ryhmämme osaaminen Visual Basic ohjelmointikielellä oli vähäistä, aloitettiin samaan aikaan suunnitelman miettimisen kanssa tutustuminen ohjelmointikielen ominaisuuksiin. Ohjelmointikielen perusteltu valinta tässä vaiheessa oli erittäin tärkeää, koska myöhempi vaihtaminen olisi ollut hyvin vaivalloista.

4. Suunnitelma: Projektin suunnitteluvaihe piti sisällään alustavaa pohdintaa työn toteutustavasta, työnjaosta, aikataulusta sekä tietenkin tavoitteesta. Tämä vaihe päättyi projektisuunnitelman kirjoittamiseen, joka muodostettiin kurssin tarjoamien puitteiden mukaan.

5. Alustavaa ohjelmointia: Ongelmaa lähdettiin lähestymään pyrkimällä tekemään yksinkertaisia ohjelmia, jotta opittaisiin VBA:n toimintaa ja piirteitä. Tällöin pyrittiin etsimään mahdollisia valmiita rakenteita, jotka voisivat olla hyödyllisiä lopullisen tavoitteen kannalta.

6. Numeerinen esimerkki: Toimeksiantajalta saamassa EM-algoritmin teoriaa kuvaavassa artikkelista löytyi viittaus numeeriseen esimerkkiin, jonka jäljitimme käyttöömme Helsingin Yliopiston valtiotieteellisen tiedekunnan tilastotieteen kirjastosta. Numeerisen esimerkin pohjalta lähdettiin työstämään ensimmäistä ohjelman versiota, joka pystyisi laskemaan saman esimerkkitehtävän hieman yksinkertaistettuna. Tätä perusteltiin mahdollisuudella tarkistaa ohjelman laskemia tuloksia vertailemalla niitä esimerkissä annettuihin. Lisäksi esimerkki selvensi EM-algoritmin teoriaa ja vaiheita. Numeerisen esimerkin ratkaiseva ohjelma koodattiin ja tulokset olivat hyviä. Kuitenkin esille nousi ongelmia lopullista tavoitetta ajatellen.

7. Käyttöliittymä 1: Samaan aikaan numeerisen esimerkin kanssa lähdimme ohjelmoimaan ensimmäistä luonnosta käyttöliittymästä. Tavoitteena oli luoda käyttäjälle helppotajuinen ja selkeä pohja, joka olisi mahdollisimman informatiivinen. Tässä vaiheessa käyttöliittymä oli melko yksinkertainen, mutta toimiva.

8. 2. tapaaminen: Tämän tapaamisen tarkoituksena oli keskustella projektin etenemisestä ja mahdollisesti esille tulleista kysymyksistä. Tässä tapaamisessa otettiinkin esille numeerisen esimerkin ohjelmoinnissa ilmenneitä ongelmia. Ongelmat koskivat informaatiomatriisin puuttuvien alkioiden sijaintia ja näistä aiheutuvia ohjelmointiteknillisiä vaikeuksia ja niiden ratkaisuja. Tässä tapaamisessa sovittiin projektin nykyisistä rajauksista.

9. Väliraportti: Kurssin vaatimuksiin kuului väliraportin laatiminen ja sen esittäminen kurssin henkilökunnalle ja muille ryhmille. Tämä oli hyödyllistä, jolloin kokonaiskuva projektista tarkentui ja tavoitteet sekä keinot niiden saavuttamiseksi selkiytyivät. Samalla kuultiin muiden ryhmien projektin edistymisestä.

10. Ohjelman laajentaminen: Tässä projektin työläimmässä vaiheessa ohjelmoitiin peruskoodi algoritmin toimintaan annettujen rajausten mukaisesti. Työvaiheeseen liittyi paljon testaamista erikokoisten matriisien kanssa

ohjelman toimimisen varmistamiseksi. Osa-alueina ohjelmoinnissa oli pohja-algoritmin luominen ja iteraatiokierrosten sekä iteroinnin parametrien käytön pohtiminen.

11. Käyttöliittymä 2: Luotiin uusi enemmän ominaisuuksia sisältävä käyttöliittymä, jossa käyttäjä pystyy määrittämään iteroinnin parametreja sekä tarkistelemaan ohjelman ulostuloa.

12. 3. tapaaminen: Esiteltiin Sammossa tämänhetkiset aikaansaannokset (laajennettu ohjelma ja käyttöliittymä) ja kysyttiin niistä mielipiteitä. Tapaamisessa käydyssä keskustelusta saatiin toivomukset lopullisen käyttöliittymän piirteistä sekä kommentointia ohjelmasta ja sen soveltuvuudesta. Tässä tapaamisessa toimeksiantaja ilmaisi tyytyväisyytensä jo nykyiseen ohjelmaan ja neuvoi ryhmää laajentamaan ohjelmaa, jos tämä työmäärä pysyisi kurssin laajuuden puitteissa. Lisäksi tapaamisessa pyydettiin oikeita informaatiomatriiseja ohjelman testaukseen

13. Ohjelman lisämuokkaus: Kolmannen tapaamisen palautteen perusteella muokattiin ohjelmaa sopivammaksi, esimerkiksi käsiteltävät matriisit transponoitiin Excelin sarakelukumäärän rajoitusten johdosta. Lisäksi lisättiin kattavaa kommentointia ohjelmaan koodin ymmärtämisen helpottamiseksi. Merkittävin osa tätä vaihetta olivat yrityksen parantaa ohjelman yleisyyttä eli pyrkimystä muuttaa ohjelmaa käsittelemään koko matriisia ensimmäisen sarakkeen sijaan. Keinot tähän tavoitteeseen pääsemiseksi tunnistettiin, mutta työmäärän huomattiin olevan liian suuri kurssin puitteisiin.

14. Loppuraportti: Projektin loppuraportin kirjoittaminen aloitettiin luomalla sisällysluettelo eli raportin käsittelemät asiat, jotka valittiin kurssin vaatimuksien ja aikaisempien kokemuksien mukaan. Tämä työvaihe päättyi raportin palauttamiseen koululle.

15. Testaus: Testaus käsitteli ohjelman ja EM-algoritmin toimivuutta sekä rajoituksia. Tässä työnvaiheessa siis selvitettiin kokonaisuuden luotettavuutta. Toisaalta haluttiin selvittää, toimiiko ohjelma EM-algoritmin mukaisesti, ja toisaalta selvitettiin, miten hyvin EM-algoritmin teoria täytti sille annetun tehtävän eli puuttuvan informaation ennustamisen tarkasti. Testaus todettiin tärkeäksi osaksi projektia, sillä tämä vähentää käyttäjän pohdiskelua tulosten oikeellisuudesta ja käytettävyydestä.

16. Lopullinen käyttöliittymä: Lopullinen käyttöliittymä muodostettiin toimeksiantajan toiveiden mukaan ja ohjelma kasattiin viimeiseen toimeksiantajalle palautettavaan muotoon.

17. Loppuesitys: Tämä projektin vaihe sisältää projektin esityksen valmistamisen, esittelyn aluksi Sammossa ja lopuksi koulussa. Lisäksi projektin tuotos palautetaan Sampoon. Tämä vaihe päättää projektin.

Seuraavassa arvioidaan ryhmämme ajallista panostusta eri projektin vaiheisiin. Tämä on tarpeen projektin kannalta, koska ajalliset resurssit ovat ainoa mittari määriteltäessä projektin kustannuksia tai vaadittuja panoksia. Kunkin työvaiheen vaatimat koko ryhmän keskimääräiset ajalliset panostukset on esitetty seuraavassa (Taulukko 2).

Taulukko 2. Projektin työvaiheiden ajallinen panostus

Työvaihe	Tunti-määrä	Työvaihe	Tunti-määrä	Työvaihe	Tunti-määrä
Aloitusluento	2	Käyttöliittymä 1	5	Ohjelman lisämuokkaus	8
Toimeksianto	4	2. tapaaminen	2	Loppuraportti	12
Tutustumista VBA:han	6	Väliraportti	4	Testaus	10
Suunnitelma	3	Ohjelman laajentaminen	21	Lopullinen käyttöliittymä	6
Alustavaa ohjelmointia	5	Käyttöliittymä 2	5	Loppuesitys	6
Numeerinen esimerkki	8	3. tapaaminen	2	**Yhteensä:**	**109**

Projektin koko työmäärä jää alle kurssin opintoviikkomäärän vaatiman tuntityövaatimuksen (= n. 120 h), mutta toisaalta vaihtoehtoinen työmäärä olisi ollut huomattavasti suurempi, jos ohjelmaa oltaisiin lähdetty edelleen laajentamaan.

5. Riskit

Kappaleessa verrataan alkuperäistä riskikuvaa sekä lopulta toteutuneita riskejä. Viimeksi mainittujen joukossa on ryhmämme toiminnasta riippumattomia riskejä, joiden olemassaolo ja toteutuneisuus huomattiin ohjelman testausvaiheessa.

5.1. Alkuperäinen riskikuva

Suunnitteluvaiheessa tunnistettiin seuraavat riskit:

- Visual Basic ohjelmointikieli. VBA on melko yksinkertainen ja suhteellisen tehoton. Valmis matriisioperaatiovalikoima on vähäinen ja paikkatunnistaminen ja sijoittaminen on ongelmallista. Projekti saattaa osoittautua VBA:lla toteutettuna siis liian työlääksi.
- Aikatauluriski. Jos VBA huomataan toimimattomaksi liian myöhään, edessä saattaa olla jo tehtyjen työvaiheiden toistaminen jollain toisella ohjelmointikielellä. Tällöin projektin aikataulu venyy. Aikatauluriski saattaa syntyä myös muutenkin, jos projektin toteutus osoittautuu erittäin työlääksi.

Näitä riskejä tunnistettaessa projekti oli vielä alkutekijöissään ja paremman kokemuksen puutteessa riskejä oli vaikea arvioida. Tosiasia jo tässä vaiheessa kuitenkin oli, että projektin pieni sidosryhmien määrä vähentää riskien lukumäärää. Uhkien arvioitiin tulevan joko projektiryhmästä tai Visual Basic ohjelmointikielestä. Osittain tämä piti paikkaansa ja osittain ei. Seuraavassa käsitellään toteutuneita riskejä ja vertaillaan näitä alkuperäisiin.

5.2. Riskien toteutuminen

Alla käsitellään projektin aikana toteutuneita riskejä sekä niiden vaikutusta projektiin. Alkuperäisiin riskilähteisiin tuli tässä vaiheessa mukaan myös itse EM-algoritmin, kun projektin lopullisena tavoitteena oli työkalun luominen puuttuvan datan estimointiin.

- Visual Basicilla toteutettuna ohjelmointi alkuperäisen tavoitteen mukaan osoittautui liian työlääksi. Varmasti ei kuitenkaan voida sanoa olisiko muun ohjelmointikielen valinta vaikuttanut työmäärään. Kuitenkin Visual Basicilla onnistuttiin luomaan työkalu, jolla on soveltuvuuskohteita ja käyttöä. Tämä työkalu noudatti uusia projektin edetessä saatuja rajoituksia, mutta alkuperäiseen tavoitteeseen nähden riski siis toteutui. Riskin vaikutus jäi suunnitelmassa mietittyä uhkakuvaa pienemmäksi juuri tavoitteiden rajausten johdosta.
- Projektin aikataulussa ei tapahtunut huomattavaa venymistä ja kokonaisuudessaan projekti saatiin päätökseen aikataulun puitteissa. Jonkin verran tapahtui myöhästymistä suunnitelmassa ja väliraportissa esitetyistä osatavoitteista, mutta nämä pienet venymiset eivät aiheuttaneet riskiä koko projektin myöhästymiselle. Alkuperäisessä suunnitelmassa yhtenä aikatauluriskin osana oli myös mahdollisuus, että projekti osoittautuu liian työlääksi nykyisillä resursseilla. Vaikka tämä riski toteutuikin, tavoitteiden rajauksilla aikataulun venymistä ei tapahtunut.

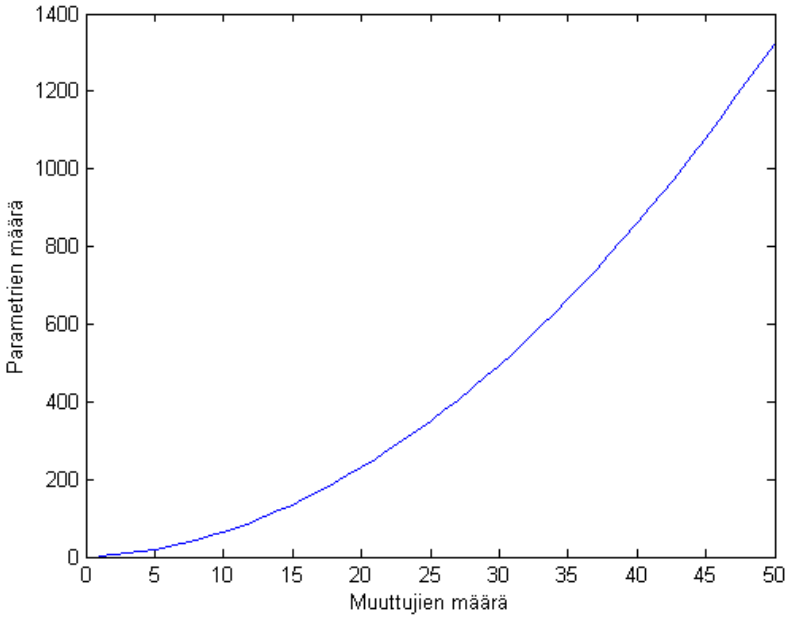
- EM-algoritmi ei tietyissä tapauksissa toimi kuten pitäisi, vaan algoritmin ominaisuudet saattavat aiheuttaa vääristyneitä tai huonoja tuloksia. Tätä riskiä ei osattu suunnitteluvaiheessa tunnistaa, vaan se ilmeni ohjelman testauksessa. Riskin toteutuminen ei siis ollut kiinni projektiryhmästä tai Visual Basicista. Riskin luonnetta on käsitelty tarkemmin luvussa 6 sekä pohdintojen ja kehitysehdotusten yhteydessä luvussa 7.

6. Tulokset

6.1. Ohjelman käytettävyys

EM-algoritmin soveltamisessa on hyvin tärkeää arvioitavan datan valinta. Tämän ohjelman yhteydessä tällä ymmärretään vertailtavien aikasarjojen valintaa sekä tutkittavien ajanhetkien määrää, eli aikasarjojen pituutta. Sekä tilastollisesta että käytännöllisestä perspektiivistä katsottuna ei ole hyödyllistä ajaa koko käytettävissä olevaa aikasarjadataa algoritmin läpi. On kannattavampaa jakaa datamatriisi osiin, ja sitten näistä estimoida puuttuvat arvot. Päädytään parempiin tuloksiin, jos esimerkiksi 500 ajanhetkeltä dataa sisältävä matriisi jaetaan kymmeneen osaan, jotka sitten estimoidaan yksitellen. Seuraavassa käsitellään syitä tähän hieman tarkemmin.

EM-algoritmin estimoimien parametrien määrä kasvaa hyvin suureksi, jos estimoitavia aikasarjoja on useita. Jos alkuperäisen datamatriisin (matriisi, jossa on siis kaikki aikasarjat) koko on T x K, jossa T on ajanhetkien lukumäärä, joilta havaintoja on, ja K aikasarjojen lukumäärä, on estimoitavien parametrien lukumäärä $K + K(K+1)/2$. Jotta estimointi pysyisi järkevänä, ei K siis voi olla liian suuri. Estimoitavien parametrien lukumäärää havainnollistaa seuraava kuva. Parametrien määrä kasvaa siis voimakkaasti K:n funktiona. Parametreilla tarkoitetaan keskiarvovektoria sekä kovarianssimatriisia. Toisaalta parametrien määrä ei käytännössä muodostu ongelmaksi, koska aikasarjoja on usein tarpeen ositella.



Kuva 2 Estimoitavien parametrien määrä

Aikasarjojen valinta täytyy suorittaa siten, että EM-algoritmissa käytettävät aikasarjat korreloivat hyvin keskenään. Jos käytetyt aikasarjat valitaan tällä periaatteella, ei algoritmin tehokkuus kärsi, vaikka informaatiota onkin vähemmän käytettävissä. Tämä johtuu yksinkertaisesti siitä, että aikasarjojen, jotka eivät korreloi paikattavien aikasarjojen kanssa, lisääminen ei paranna parametrien estimaatteja.

Käytännön rajoituksiin liittyy myös Excelin funktioiden oma rajoitus, joka havaittiin hyvin suuria matriiseja estimoitaessa. Excelin sisäänrakennetut funktiot eivät voi käsitellä matriiseja, joiden dimensioiden tulo *T*x*K* ylittää noin 5450 alkiota. Syytä tähän rajoitukseen ei ole saatu selville, mutta kauhean vakavasta rajoitteesta ei ole kyse. Joka tapauksessa näin suuria matriiseja ei ole järkevää käyttää estimointiin.

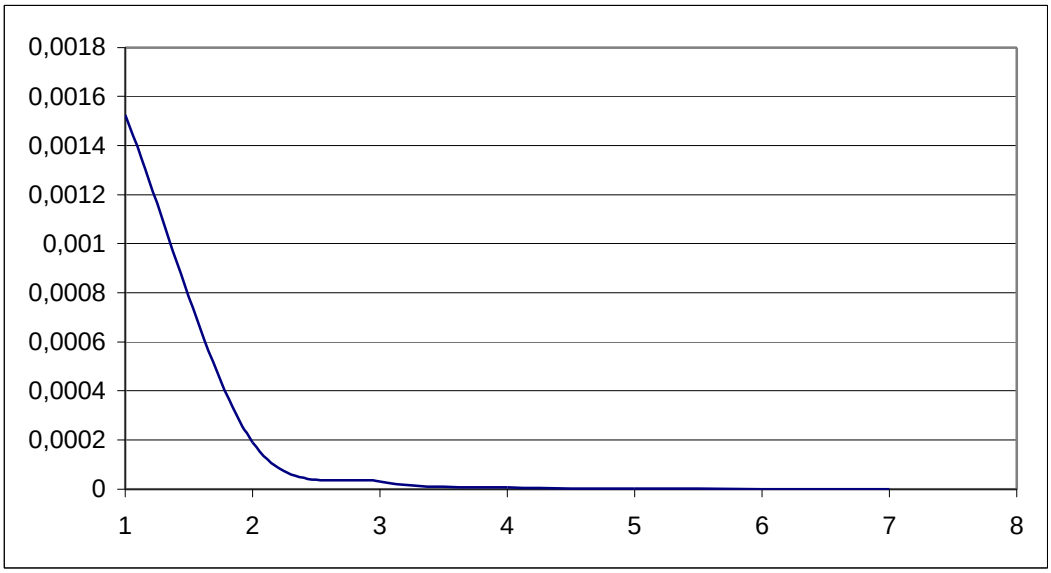
Toteutettua ohjelmaa on tarkoitus käyttää finanssimaailmassa osakkeiden ja valuuttakurssien puuttuvien arvojen estimointiin. Tämä käyttötarkoitus on hieman ristiriidassa EM-algoritmin oletusten kanssa. Aikasarjat eivät ole stationaarisia, vaan niissä voi esiintyä trendejä. EM-algoritmia voidaan kuitenkin käyttää näidenkin aikasarjojen estimointiin. Käyttäjän tehtäväksi jää kuitenkin pitää huoli siitä, että estimointi tapahtuu järkevän pituisissa ajanjaksoissa. Lyhyellä aikavälillä mahdollinen trendi ei pääse vaikuttamaan iteroinnin lopputuloksiin kovin voimakkaasti.

6.2. Tulosten luotettavuus

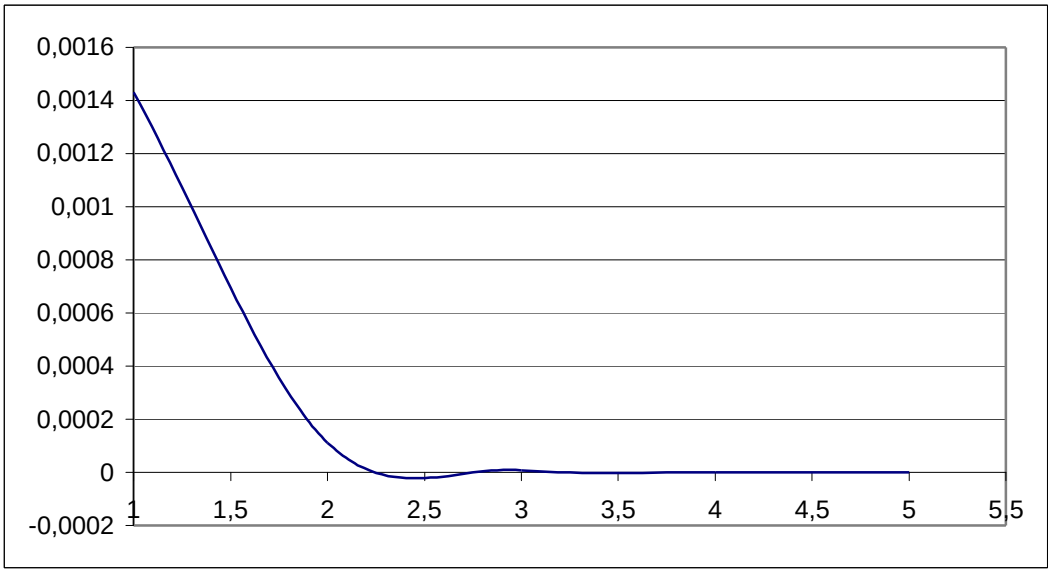
Ohjelman testauksessa havaittiin että puuttuville data-alkioille saatujen estimaattien oikeellisuus riippuu hyvinkin paljon käytetystä datasta. Parhaimpiin tuloksiin päästään

stationaarisella datalla. Ohjelman käyttöalue kuitenkin suuntautuu osakekursseihin sekä valuuttakursseihin, jotka eivät ole stationaarisia.

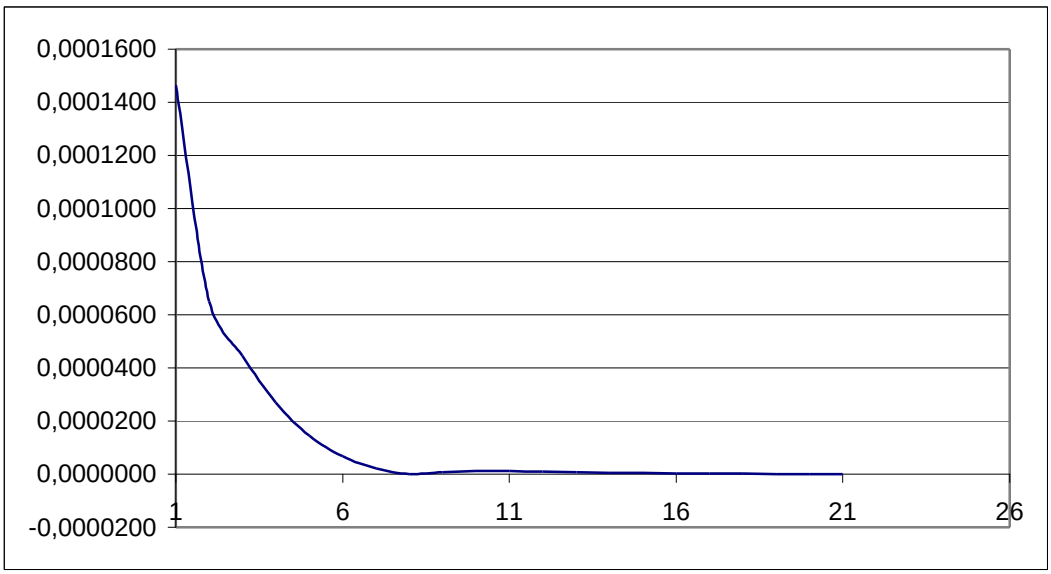
EM-algoritmi toimi testeissämme luotettavasti siinä mielessä, että iteraatioprosessi suppeni jokaisella kerralla ja usein varsin tehokkaasti. Seuraavassa on esitetty kuvaajia konvergoimisprosesseista erilaisilla T, K, N arvoilla. T on aikasarjan pituus, K käytettyjen aikasarjojen määrä ja N puuttuvien data-alkioiden määrä. Suppenemiskriteerinä on, että kaavasta (21) laskettu estimaattien suhteellinen erotus on alle 10^{-7}. Tämä suure on kuvien pystyakselina ja vaaka-akselina iteraatioiden lukumäärä. EM-algoritmi on siis hyvin luotettava tässä mielessä.



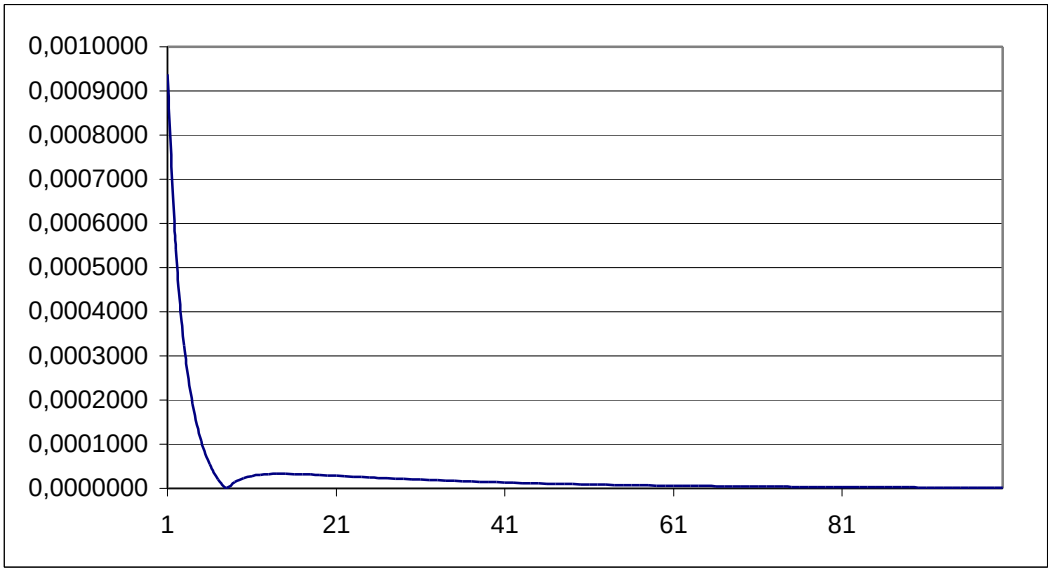
Kuva 3 T = 300, K = 9, N = 16



Kuva 4 T = 300, K = 9, N = 7



Kuva 5 T= 28, K = 9, N = 7



Kuva 6 T = 28, K = 18, N = 7

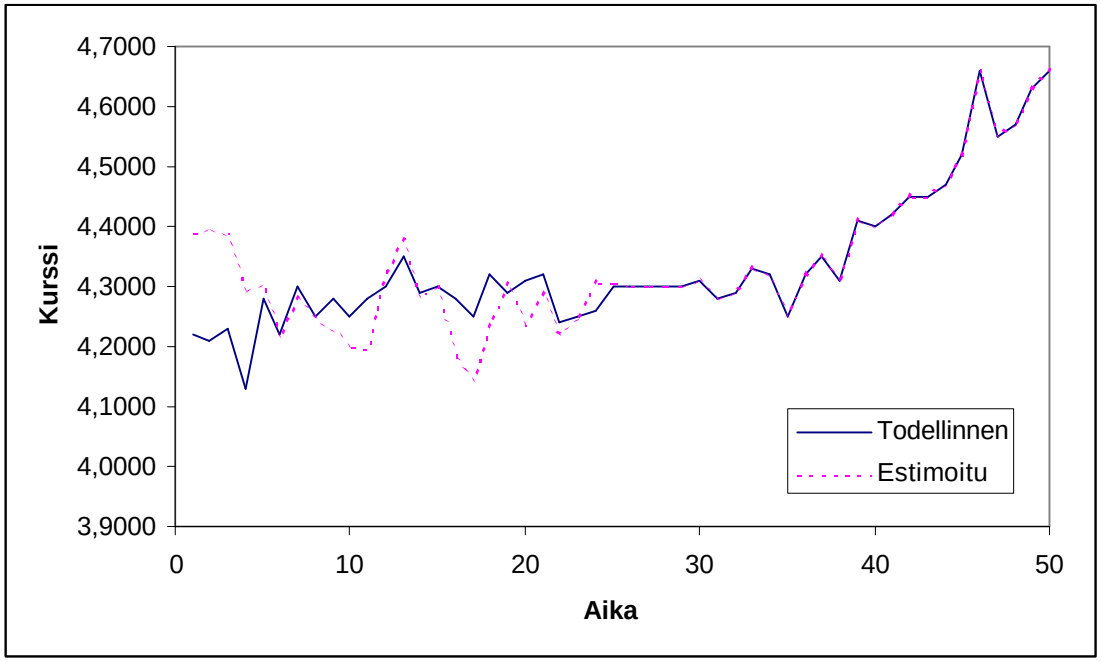
Vaikka suppeneminen onkin tehokasta, kuvista voidaan havaita, että konvergoitumisessa saattaa olla kehitystä välillä huonompaankin suuntaan ennen lopullista suppenemista.

Lisättäköön kuitenkin vielä, että lopullinen vastuu on tässäkin ohjelmassa käyttäjän harteilla. Kun puuttuva data on estimoitu aikasarjaan, on syytä tarkistaa lopputulos siltä varalta, että käytetty data on hyvin epästationaarista. Jos lopputulos on huono, voi estimointia kokeilla suorittaa lyhyemmissä osissa, jolloin epästationaarisuuden vaikutus pienenee. Myös huonosti korreloivin aikasarjojen poistaminen parantaa ohjelman toimintaa. Se, milloin epästationaarisuuden vaikutus ei enää ole merkittävää, on hyvin vaikea sanoa.

6.3. Analyysia

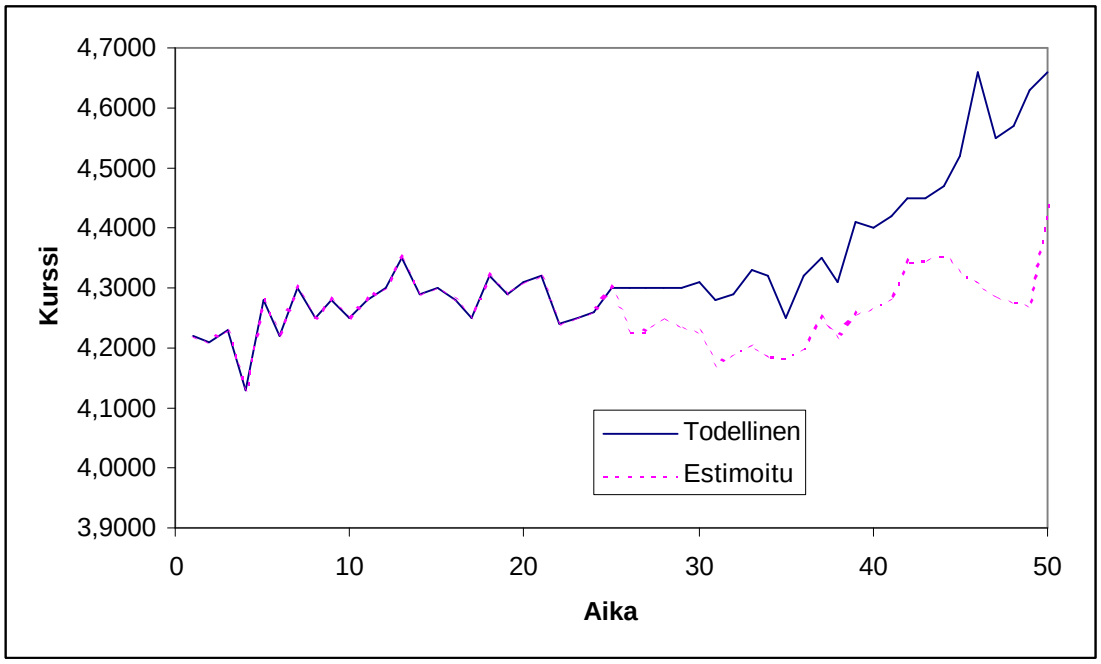
Ohjelmaa on testattu erilaisilla datamatriiseilla ja vertailtu saatuja estimaatteja alkuperäiseen aikasarjaan. Eräs ohjelman käyttötarkoitus on estimoida jollekin aikasarjalle menneisyys. Ennustaminen on kuitenkin havainnollista, kun tutkitaan kykeneekö algoritmi estimoimaan nousevan vai laskevan trendin.

Testasimme ohjelmaa neljässä eri tilanteessa. Kaikissa käytimme 18 aikasarjaa, joissa jokaisessa oli 50 havaintoa. Ensimmäisestä (kuva 7) aikasarjasta puuttuu 25 ensimmäistä havaintoa.



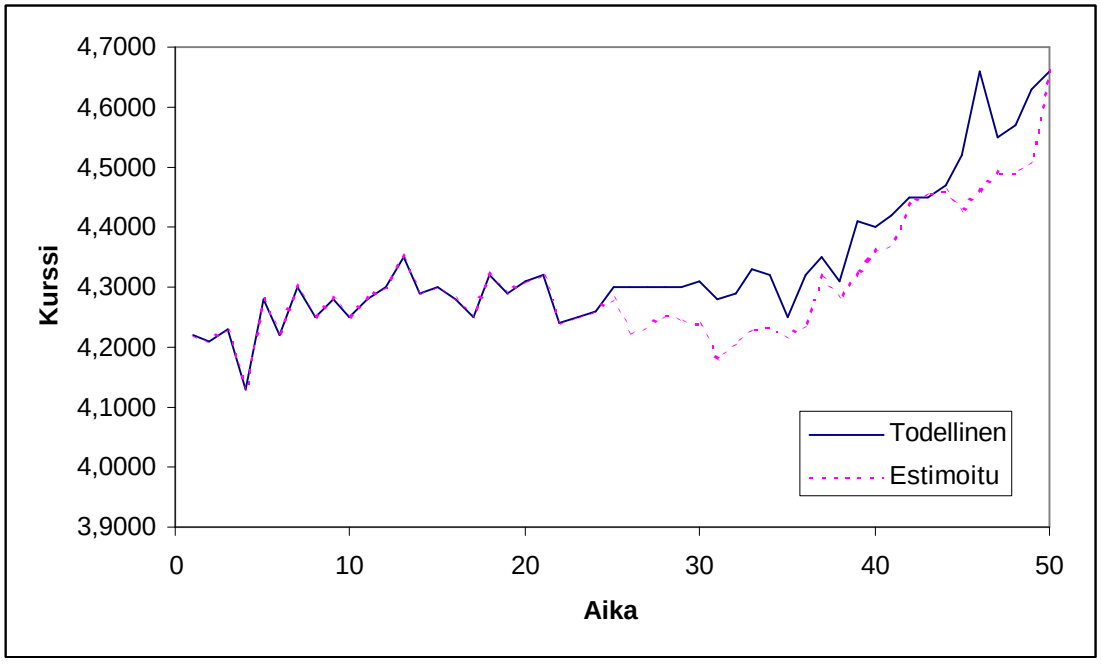
Kuva 7. EM-algoritmin tuloksia, kun puuttuu 25 ensimmäistä havaintoa

Kuvasta havaitaan, että algoritmi toimii suhteellisen hyvin, kun puuttuvan datan kohdalla ei ole vahvaa trendiä havaittavissa. Jos dataa kuitenkin puuttuu juuri trendin kohdalta, algoritmi ei suoriudu niin hyvin. Tämä voidaan havaita seuraavasta kuvasta.



Kuva 8. EM-algoritmin toiminta, kun puuttuu 25 havaintoa trendin kohdalta

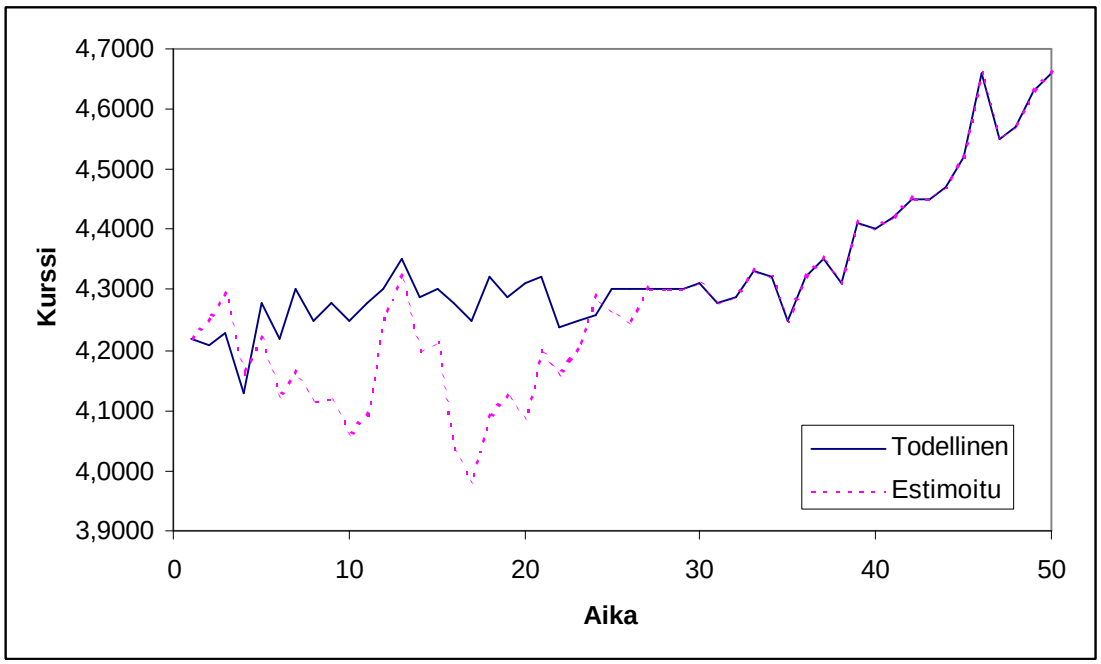
Koska EM-algoritmi perustuu oletukseen aikasarjan stationaarisuudesta, se ei suoriudu hyvin, jos aikasarjassa on trendi. Yllä olevassa kuvassa loppupuolella olevaa trendiä algoritmi ei huomioi, ja tulokset jäävät huonoksi. Parempiin tuloksiin päästään, jos trendin kummaltakin puolelta on käytössä dataa alkuperäisestä aikasarjassa.



Kuva 9. EM-algoritmin toiminta, kun puuttuu 25 havaintoa trendin kohdalta

Kuvasta 9 huomataan, että jos käytettävissä on piste-estimaatti tulevaisuudesta, EM-algoritmilla saadaan estimoiduksi aikasarja, joka seuraa suhteellisen hyvin todellista ja nousevan trendin sisältävää aikasarjaa. Viimeiseksi tutkitaan tilanne, jossa piste-

estimaatti on käytettävissä aivan aikasarjan alkupäästä ja sitä seuraavat 25 arvoa puuttuvat. Tilanne on esitetty kuvassa 10.



Kuva 10. EM-algoritmin tuloksia, kun puuttuu 25 havaintoa piste-estimaatin jälkeen

Kuvasta 10 havaitaan kuitenkin, että tulosten laatu voi vaihdella merkittävästi, ja vastuu on käyttäjällä.

7. Pohdinnat ja kehitysideoita

Tämä kappale sisältää pohdiskelua projektin onnistumisesta ja pohdiskelua siitä, miten projektin tulosta oltaisiin voitu parantaa. Alkuperäiseen tavoitteiseen nähden projekti onnistui vain osittain. Tulos ei ollut kokonaisia matriiseita täydentävä Excel-työkalu, vaan tavoitetta jouduttiin projektin edetessä supistamaan. Tämä rajoitus tehtiin, kun huomattiin, että projektille osoitetut ajalliset resurssit ylittäisivät projektin vaatimat. Kuitenkin nykyinen projektin tulos on testauksissa osoittautunut toimivaksi, mutta käyttäjälle jää vastuu tässä raportissa esitettyjen analyysien mukaisesti.

EM-algoritmin validiteetti on tietyissä tapauksissa huono. Käytännössä suuri osa todellisuudessa muun muassa talouden alalla esiintyvistä aikasarjoista on epästationaarisia esimerkiksi suhdannevaihteluiden ym. vuoksi. Erityisesti pitkien aikasarjojen tapauksessa tämä aiheuttaa suuria ongelmia EM-algoritmin tulosten luotettavuuteen jo yksittäisten välistä puuttuvien havaintojen estimoinnissa. Jos aikasarjalla on ennustettavissa olevaa trendikäyttäytymistä tai jaksollisuutta, epästationaarisuudesta saatetaan päästä eroon differoimalla aikasarjaa. Sen sijaan, jos epästationaarista käytöstä ei voida ennustaa ja poistaa, EM-algoritmia ei välttämättä enää voida pitää optimaalisena työkaluna puuttuvien havaintojen estimoimiseksi.

Tällaisissa tilanteissa saatetaan hyvin todennäköisesti päästä parempiin tuloksiin esimerkiksi interpoloimalla puuttuvat havainnot olemassa olevasta datasta. Stationaarisuusongelmien huomioiminen on käytännössä käyttäjän kontolla, kuten myös puutteellisen aikasarjan referenssiksi valittavien riittävästi korreloituneiden aikasarjojen valinta, mikä on myös hyvin olennaista. Luotettavuutta voidaan parantaa pilkkomalla aikasarjoja lyhyemmiksi ja mahdollisesti vielä differoimalla näitä erikseen trendien merkityksen vähentämiseksi ja tämän jälkeen sovelletaan EM-algoritmia.

Työryhmän laatima ohjelma kykenee analysoimaan puuttuvat alkiot yhdessä aikasarjassa. Ensimmäinen kehityskohde ohjelman laajentamisen suhteen voisi olla vaakarivien analysoimismahdollisuuden lisääminen. Seuraavaksi ohjelman voisi rakentaa kykeneväksi analysoimaan yleistetyn tapauksen, jossa havaintoja saisi puuttua useammasta aikasarjasta.

Tulosta oltaisiin voitu parantaa lisäämällä ohjelmoinnille osoitettuja ajallisia resursseja. Lisäys olisi onnistunut kahdella tavalla eli joko absoluuttisella resurssimäärän kasvattamisella tai vapauttamalla resursseja muilta projektin osa-alueilta. Resurssien vapauttaminen tai niiden parempi kohdistaminen olisi onnistunut, jos projektiryhmällä olisi ollut ennalta kattavampaa tietämystä Visual Basicista tai EM-algoritmista. Tällöin tutustumisvaiheeseen käytetty aika oltaisiin voitu käyttää itse ohjelmointiin.

Ryhmän työjaon merkitystä projektin onnistumisen on vaikeaa arvioida. Työjako toimi hyvin, mutta toisaalta ei kokeiltu vaihtoehtoisia työtapoja. Koko ryhmän osaamisen paremmalla yhdistämisellä, tarkoittaen työntekoa koko ryhmän ollessa koossa, oltaisiin voitu saavuttaa tuloksia tehokkaammin.

8. Liitteet

1. Käyttöohje
2. Ohjelmakoodi