

Korrelaatio Javalla

Tehdään vain yksinkertainen korrelaatio ilman normalisointeja:

$$r_{12}(j) = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n) \cdot x_2(n+j),$$

jossa j on viive. Joskus esiintyy ilman kerrointa $1/N$.

Java-koodi:

```
/*
Janne Koljonen
Vaasan yliopisto
AUTO1030
2.3.2012
Korrelaation laskeminen
*/

import java.text.*;

public class Korrelaatio
{
    // Korrelaatio kaikilla vaihesiirroilla
    public static double[] corr(double[] x1, double[] x2)
    {
        double[] tulos = new double[x1.length+x2.length-1];

        // Kun vaihesiirto j = 0, x1[0] ja x2[0] ovat kohdakkain.
        // Oletetaan, että liu'utetaan x2:ta, jonka pituus on l2.
        // Olkoon x1:n pituus l1.
        // Kun x2 siirtyy vasemmalle l2-1 verran (eli j = +l2-1),
        // ovat x1[0] ja x2[l2-1] (eli x2:n viimeinen näyte) kohdakkain.
        // Vastaavasti kun x2 siirtyy l1-1 verran oikealle (eli j = -l1+1),
        // ovat x1[l1-1] (eli x1:nviimeinen näyte) ja x2[0] kohdakkain.
        // Vaihesiirto j voi siis olla [-l1+1, l2-1]
        for (int j=-x1.length+1; j<x2.length; j++)
        {
            // Korreloitavan näyteosan pituus vaihtelee
            // vaihesiirron mukaisesti. Maksimissaan korreloitavia näytteitä
            // on lyhyemmän signaalin min(l1, l2) verran.
            double[] temp1 = new double[Math.min(x1.length, x2.length)];
            double[] temp2 = new double[Math.min(x1.length, x2.length)];

            // Kopioidaan korreloitavat näytteet temp-taulukoihin.
            // Tarkistetaan, etteivät indeksi mene taulukoiden ulkopuolelle.
            // Lasketaan kuinka monta näytettä (N) korrelaatioon tulee.
            int N = 0;

            // (Kannattaa piirtää, miten indeksoinnit oikein menevät!)
            if (j<0) // x2 siirtyy oikealle
            {
                for (int i=0; i<temp1.length; i++)
                {
                    if (i-j<x1.length)
                    {
                        N++;
                        temp1[i] = x1[i-j];
                        temp2[i] = x2[i];
                    }
                }
            }
            else // x2 siirtyy vasemmalle
            {
                for (int i=0; i<temp1.length; i++)
                {
                    if (i+j < x2.length)
                    {

```

```
        N++;
        temp1[i] = x1[i];
        temp2[i] = x2[i+j];
    }
}

// Vihdoin voidaan laskea pistetulo
tulos[j+x1.length-1] = pistetulo(temp1, temp2)/N;
}
return tulos;
}

// Pistetulo eli yhden korrelaatiopisteen laskenta
// olettaen, että x1 ja x2 ovat yhtä pitkiä
private static double pistetulo(double[] x1, double[] x2)
{
    double temp=0;
    for (int i=0; i<x1.length; i++)
    {
        temp += x1[i]*x2[i];
    }
    return temp;
}

/** double-tilukoiden tulostus */
public static void printVector(double[] v)
{
    NumberFormat nf=NumberFormat.getInstance();
    nf.setMinimumFractionDigits(2);
    nf.setMaximumFractionDigits(2);
    for(int i=0;i<v.length;i++)
    {
        System.out.print(nf.format(v[i])+" ");
    }
    System.out.println();
}

/** double-tilukoiden tulostus */
public static void printVector(double[] v,String otsikko)
{
    System.out.print(otsikko+" ");
    printVector(v);
}

// Testataan korrelaatiota
public static void main(String[] sArgs)
{
    System.out.println("Testi alkaa");

    // Testi 1
    double[] x1 = {1.0, 0.0, -1.0, 0.0, 0.0};
    double[] x2 = {0.0, 0.0, 1.0, 0.0, -1.0, 0.0, 0.0};
    double[] tulos1 = corr(x1, x2);
    printVector(tulos1, "korrelaatiot");
}
}
```

Huomaa, että tulostuksessa ensimmäinen korrelaatio vastaa vaihesiirtoa $j = -4$. Joitain tuloksia: $r(0) = -0,2$, $r(2) = 0,4$. Eli maksimikorrelaatio saavutetaan, kun $j = 2$.