

# Neurovõrgud. Praktikum 8.

8. aprill 2005. a.

## 1 Normaaljaotusega klasside eristamine

Eelmises praktikumis vaatlesime kahte normaaljaotusega klassi eristamist. Klassi  $y$  tihedus avaldus siis valemiga:

$$P(\mathbf{x} | y) = \frac{1}{\sqrt{(2\pi)^m \det(\boldsymbol{\Sigma}_y)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{x} - \boldsymbol{\mu}_y)\right)$$

kus  $\boldsymbol{\mu}_y$  ja  $\boldsymbol{\Sigma}_y$  on klassi  $y$  keskvärtus ja kovariatsiooni maatriks ning  $m$  on dimensioonide arv.

Näitasime et juhul kui  $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$  ja klassid on võrdtöenäosed, on Bayesi klassifitseerija nende klasside eristamiseks lineaarne, ning eraldusjoon avaldub valemiga

$$\mathbf{w}^T (\mathbf{x} - \boldsymbol{\mu}_c) = 0 \quad \text{kus} \quad \boldsymbol{\mu}_c = \frac{\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1}{2}, \quad \mathbf{w} = \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

Pange tähele, et valem on suhteliselt intuiitvne: juhul kui  $\boldsymbol{\Sigma}$  on ühikmaatriks ütleb ta et eraldav hüperatasand läbib jaotuste keskvärtuste keskmist ning on perpendikulaarne neid ühendava sirgiga. Juhul kui  $\boldsymbol{\Sigma}$  ei ole ühikmaatriks, sama olukorra võib saavutada teisendates terve ruumi teatud lineaarteisendusega (veendu selles!).

Paljudel juhtudel võib arvuliste tunnuste jaotust aproksimeerida normaaljaotusega, mille keskvärtust ja kovariatsioonimaatriksit võib hinnata valimi abil:

$$\hat{\boldsymbol{\mu}}_y = \frac{1}{n_y} \sum_{\mathbf{x} \text{ klassist } y} \mathbf{x} \quad \hat{\boldsymbol{\Sigma}}_y = \frac{1}{n_y - 1} \sum_{\mathbf{x} \text{ klassist } y} (\mathbf{x} - \hat{\boldsymbol{\mu}}_y)(\mathbf{x} - \hat{\boldsymbol{\mu}}_y)^T$$
$$\hat{\boldsymbol{\Sigma}} = \frac{1}{2} (\hat{\boldsymbol{\Sigma}}_1 + \hat{\boldsymbol{\Sigma}}_2)$$

Niimoodi hinnantud parameetrite abil leitud lineaarset Bayesi klassifitseerijat nimetatakse *Fisheri diskriminandiks*.

**Ülesanne 1 (1p):** Olgu  $\boldsymbol{\mu}_1 = (0, 0)^T$ ,  $\boldsymbol{\mu}_2 = (1, 3)^T$ ,  $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ . Geneereeri igast klassist 1000 esindajat, leia selle valimi järgi parameetrite empiirilised hinnangud, võrdle neid tegelikke parameetritega. Leia hinnatud parameetrite abil lineaarset Bayesi klassifitseerijat ning hinda tema täpsust. Pane graafikule punktid koos klassifitseerija eraldava joonega. Esita kood ja pilt.

Osutub et niimoodi leitud klassifitseerija normaal (ehk kaalude vektor)  $\mathbf{w}$  maksimiseerib *Fisher'i kriteeriumi*:

$$J(\mathbf{w}) = \frac{|\pi(\hat{\boldsymbol{\mu}}_1) - \pi(\hat{\boldsymbol{\mu}}_2)|^2}{\hat{\sigma}_1^2 + \hat{\sigma}_2^2} = \frac{\mathbf{w}^T(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^T \mathbf{w}}{\mathbf{w}^T \hat{\boldsymbol{\Sigma}} \mathbf{w}}$$

kus  $\pi$  tähistab projektsiooni vektorile  $\mathbf{w}$  ning  $\hat{\sigma}_y^2 = \frac{1}{n_y - 1} \sum (\pi(\mathbf{x}) - \pi(\hat{\boldsymbol{\mu}}_y))^2$  on klassi  $y$  projektsiooni dispersioon.

**Ülesanne 2\* (2p):** Näita et  $\mathbf{w}$ , mis maksimiseerib  $J(\mathbf{w})$  avaldub kujul  $\mathbf{w} = \lambda \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)$  kus  $\lambda$  on suvaline reaalarv.

Maksimiseerides Fisher'i kriteeriumi me seega leiame sellist sirgjoont, millele andmeid projitseerides on projektsioonide keskväärtused võimalikult kaugel üksteisest samas dispersioonid võimalikud väikesed. See annab teatud põhjendust sellele, miks võib see klassifitseerija olla hea ka erinevate kovariatsioonidega klasside puhul, erinevate aprioorsete tõenäosustega ning isegi mittenormaalsete jaotustega klasside eristamiseks. Tihti võib temaga saavutada häid tulemusi, kuid leidub jaotusi, mille puhul on Fisheri diskriminant väga halb.

**Ülesanne 3 (1p):** Fisheri diskriminant on ideaalne kui klasside kovariatsioonimaatriksid on võrdsed, mis praktikas ei pruugi kehtida. Kas seda saaks äkki mingi eeltötluse abil parandada, näiteks normeerides ja tsentreerides muutujaid vms? Miks/Kuidas?

Lõpuks paneme tähele sama kovariatsioonimaatriksiga normaalsete klasside veel ühte omadust:

**Ülesanne 4 (1p):** Olgu kaks klassi jaotatud normaalselt sama kovariatsioonimaatriksiga (ning nad ei pea olema võrdtõenäosed, s.t. mitte tingimata  $P(Y = 1) = P(Y = 2)$ ). Näita, et iga klassi *a-posteriori* tõenäosus avaldub logistilise funktsioonina, s.t.

$$P(Y = 1 | \mathbf{x}_0) = \frac{1}{1 + \exp(\mathbf{w}^T \mathbf{x}_0 + b)}$$

See annab veel ühte põhjendust miks ühe neuroni aktivatsioonifunktsiooniks tihti valitakse just logistilist funktsiooni.

## 2 K-lähima naabri klassifitseerimisreegel

Üks lihtsamatest, kuid ka üks parimatest klassifitseerijatest on  $k$ -lähima naabri klassifitseerimisreegel (*k-nearest neighbors rule, k-nn*): punkti  $\mathbf{x}$  klassifitseerimisel leitakse treeningvalimis talle  $k$  lähimat naabrit, ning punkti klassiks loetakse seda, mille esindajaid on naabrite seas rohkem. Katsetame seda klassifitseerijat meile esimest praktikumist tuntud kasvajakude andmestiku peal.

**Ülesanne 5 (1p):** Realiseeri funktsiooni `function y = knn(x, X, d, k)` mis võtab sisendina klassifitseerimiseks kuuluva vektori  $\mathbf{x}$ , treeningandmete maatriksi  $\mathbf{X}$  koos õigete klassimärgenditega  $\mathbf{d}$ , arvu  $k$  ning tagastab  $k$ -nn algoritmiga leitud klassimärgendi  $y$ . Katseta seda kasvajakude andmestiku peal:

- jaga andmestiku treening- ja testhulkadeks (400/100)
- iga  $k$  korral 1-st 400-ni hinda klassifitseerija täpsus testhulgal. Pane graafikule täpsuse sõltuvus  $k$ -st. Kas saadud graafik on loomulik ning peab alati niimoodi välja nägema?

Esimeses praktikumis saavutas lineaarne klassifitseerija täpsuse kuni 85%. Kas  $k$ -nn on parem? Esita kood ja pilt.

**Ülesanne 5 $\frac{1}{2}$  (1p):** <sup>1</sup> Nagu näha on, ei nõua  $k$ -nn klassifitseerija mingit „treenimist” — on vaja lihtsalt salvestada treeningandmeid. See eest on iga näite klassifitseerimine keerukas: tuleb vaadata kogu treeninghulga läbi ja otsida sealt naabreid. Kas saaks treenimisfaasis andmeid indekseerida selleks et pärast oleks võimalik leida lähimaid naabreid kiiremini? Miks/Kuidas?

## 3 Parzeni klassifitseerija

Lähima naabri reegli peale võib vaadata ka veidi teistmoodi. Tähistame klasse arvudega  $+1$  ja  $-1$ , ning olgu  $K_i(\mathbf{x}) = 1$  kui treeningnäide  $\mathbf{x}_i$  on üks  $k$ -st lähimatest naabritest  $\mathbf{x}$  jaoks, ning  $0$  vastasel juhul. Siis saab naabrireegli panna kirja kujul:

$$f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^n y_i K_i(\mathbf{x}) \right)$$

asendame nüüd funktsioonid  $K_i(\mathbf{x})$  (mis sõltuvad valimist) ühe *kernel funktsiooniga*  $K(\mathbf{x}, \mathbf{x}_i)$ , mille väärtus on lähedal 1-le kui  $\mathbf{x}$  ja  $\mathbf{x}_i$  on lähedal, ning

<sup>1</sup>Praktikumis jagatud väljatrükitud materjalide seas seda ülesannet polnud.

0 vastasel juhul. Tüüpiline valik  $K$  jaoks on gaussiaan keskpunktiga  $\mathbf{x}_i$ -s. Kokkuvõttes saame klassifitseerimisreegli

$$f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^n y_i K(\mathbf{x}, \mathbf{x}_i) \right)$$

mida nimetatakse *Parzen density estimator classifier*.

**Ülesanne 6 (1p):** Selline klassifitseerija on tegelikult Bayesi klassifitseerija empiiriline versioon. Seleta miks. Kas see klassifitseerija on sama mis RBF võrk?

**Ülesanne 7 (1p):** Realiseeri Parzeni klassifitseerija, kus kernel-funktsiooni rollis on  $K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}\right)$ . Realiseeri seda funktsioonina `parzen(x, X, d, s)`, ning samamoodi nagu ülesandes **5** katseta seda rakude andmes- tikul erinevate  $\sigma$  väärtuste korral (proovi väärtused 1, 5, 10, 50, 100, 500, 1000). Esida kood ja  $\sigma$ -vs-täpsuse graafik.