Übung zur Vorlesung

Maschinelles Lernen 1

Wintersemester 2009/2010

Abteilung Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Blatt 4

Abgabe bis Donnerstag, den 12. November 2009 in der Übung.

Aufgaben

Independent Component Analysis/TDsep

Während die Hauptkomponentenanalyse (PCA) einen multivariaten Datensatz in voneinander unkorrelierte Komponenten zerlegt, geht die *Independent Component Analysis* (ICA) einen Schritt weiter: eine Zerlegung in unabhängige Komponenten. Wie die PCA berechnet auch die ICA eine lineare Transformation, diese ist allerdings im allgemeinen **nicht** orthogonal.

Das generische Modell der Datenmatrix $X \in \mathbb{R}^{n\times T}$ ist eine instantane und lineare Überlagerung von unabhängigen Quellen $S \in \mathbb{R}^{m\times T}$ gemäß

$$X = AS$$

wobei A als Mischungsmatrix bezeichnet wird. In diesem Zusammenhang spricht man auch von *blinder Quellentrennung* (blind source separation, BSS), da hier das Ziel ist, die Mischung wieder rückgängig zu machen. Eine solche Mischung wäre etwa gegeben durch eine Mehrspuraufnahme von m Schallquellen durch n im Raum verteilte Mikrofone oder durch die Aufnahme von m neuronalen Signalen durch n auf der Kopfhaut angebrachte EEG-Elektroden. Jede Quelle entspräche dann einer Zeile von S, jede Aufnahme einer Zeile von X, während jede Spalte einen Zeitpukt definiert. Vereinfachend nehmen wir im folgenden $n = m$ an, A ist dann eine quadratische Matrix.

Ziel der blinden Quellenrennung ist es, aus den gemessenen Daten X eine Schätzung Y für die unabhängigen Quellsignale S zu berechnen:

$$Y = BX.$$

Die geschätzten Signale (Zeilen von Y) werden o.B.d.A. auf 1 normiert.

1. Besitzen die gesuchten Quellsignale eine zeitliche Struktur (nichtverschwindende Autokorrelationen) und definiert man *zeitversetzte Kovarianzmatrizen* für X und Y als

   $$\begin{aligned} C^x(\tau) &= E(x(t)x^T(t-\tau)) \\ C^y(\tau) &= E(y(t)y^T(t-\tau)) \end{aligned}$$

   (wobei $x(t)$ und $y(t)$ die t-te Spalte von X bzw. Y bezeichnen), so lässt sich das Quellentrennungsproblem lösen durch Bestimmung einer Matrix B, für die (bei geeignet gewähltem τ) gilt

   $$\begin{aligned} &BC^x(0)B^T = I \\ &BC^x(\tau)B^T \text{ ist eine Diagonalmatrix} \end{aligned}$$

   Warum würde so eine Zerlegung bedeuten, dass man die Quellen tatsächlich entmischt hat? **(8 Punkte)**

2. Für zwei $n\times n$-Matrizen A, B heisst λ *verallgemeinerter Eigenwert zum Eigenvektor* v, wenn

   $$Av = \lambda Bv.$$

   Angenommen, man erhält einen vollen Satz $v_1, \dots, v_n$ solcher Vektoren und Eigenwerten $\lambda_1, \dots, \lambda_n$, und man bildet die Matrizen

   $$V = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix}, \qquad L = \operatorname{diag}(\lambda_1, \dots, \lambda_n),$$

dann kann man die obige Gleichung auch als $AV = BVL$ schreiben.

Zeige, dass sowohl V^TAV als auch V^TBV diagonal sind, wenn A und B symmetrisch sind. (*Hinweis:* Der ij-te Eintrag von V^TAV lautet $v_i^TAv_j$. Zeige, dass die Einträge 0 sind, wenn $i \neq j$.) **(7 Punkte)**

3. Ergänze das Programmskelett `sheet04.m` um die Funktionen `tdcov` und `tdsep`, um ICA mittels TDSEP zu implementieren. **(15 Punkte)**
   - `tdcov` schätzt die Kovarianzamatrix $C^x(\tau) = E(x(t)x^T(t-\tau))$ aus einer gegebenen Zeitreihe.
   - `tdsep` berechnet die Entmischungsmatrix B mittels der verallgemeinerten Eigenzerlegung der Matrizen $C^y(0)$ unad $C^y(\tau)$. Theoretisch ist $C^y(\tau)$ bei unabhängigen Quellen symmetrisch, praktisch und numerisch ist das nicht immer der Fall. Die Matrix sollte deswegen vor dem Diagonlisieren durch

$$C^y(\tau) \leftarrow \frac{1}{2}(C^y(\tau) + C^y(\tau)^T)$$

symmetrisiert werden.

```
function sheet04

% generate some data
T = linspace(0, 10, 1000);
X1 = sin(pi*T);
X2 = 2*(T - floor(T)) - 1;
X3 = 0.1*randn(1, 1000);

X = [X1; X2; X3];

% plot the sources
figure(1)
plotsources(T, X)

% generate a random mixing matrix
A = randn(3, 3);
Y = A * X;

% plot the mixed sources
figure(2)
plotsources(T, Y);

% compute time-lagged
B = tdsep(Y, 5);

figure(3)
plotsources(T, B'*Y)

function plotsources(T, X)
N = size(X, 1);
for I = 1:N
  subplot(N, 1, I)
  plot(T, X(I, :))
end

% Compute the TDSEP estimate by diagonalizing both matrices
% simultaneously. Use matlabs EIG function for two matrices.
function B = tdsep(Y, T)
C0 = tdcov(Y, 0);
CT = tdcov(Y, T);
% ...
```

```
% Compute the time-lagged covariance matrices. Extract a "normal" and a
% lagged version of the signal and estimate the covariance matrix for
% both parts.
function C = tdcov(X, T)
% ...
```

Für Fragen zum Übungsblatte bitte in der Google Group http://groups.google.com/group/mikiobraun-lehre registrieren und die Frage an die Mailingliste stellen.