Slide10

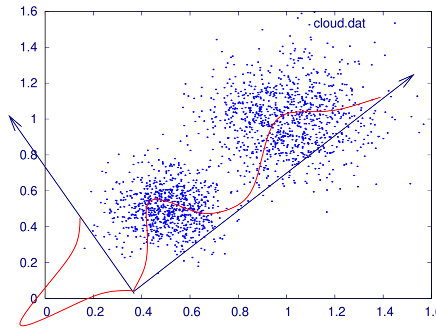
Haykin Chapter 8: Principal Components Analysis

CPSC 636-600

Instructor: Yoonsuck Choe

Spring 2015

Motivation



- How can we project the given data so that the variance in the projected points is maximized?

Principal Component Analysis: Variance Probe

- $\mathbf{X}$: m-dimensional random vector (vector random variable following a certain probability distribution).
- Assume $E[\mathbf{X}] = \mathbf{0}$.
- Projection of a unit vector $\mathbf{q}$ ($(\mathbf{q}\mathbf{q}^T)^{1/2} = 1$) onto $\mathbf{X}$:

$$A = \mathbf{X}^T\mathbf{q} = \mathbf{q}^T\mathbf{X}.$$

- We know $E[A] = E[\mathbf{q}^T\mathbf{X}] = \mathbf{q}^T E[\mathbf{X}] = 0$.
- The variance can also be calculated:

$$\begin{aligned}\sigma^2 = &\quad E[A^2] = E[(\mathbf{q}^T\mathbf{X})(\mathbf{X}^T\mathbf{q})] \\ = &\quad \mathbf{q}^T \underbrace{E[\mathbf{X}\mathbf{X}^T]}_{\text{covariance matrix}} \mathbf{q} \\ = &\quad \mathbf{q}^T\mathbf{R}\mathbf{q}.\end{aligned}$$

Principal Component Analysis: Variance Probe (cont'd)

- This is sort of a *variance probe*: $\psi(\mathbf{q}) = \mathbf{q}^T\mathbf{R}\mathbf{q}$.
- Using different unit vectors $\mathbf{q}$ for the projection of the input data points will result in smaller or larger variance in the projected points.
- With this, we can ask *which vector direction does the variance probe $\psi(\mathbf{q})$ has extermal value*?
- The solution to the question is obtained by finding unit vectors satisfying the following condition:

$$\mathbf{R}\mathbf{q} = \lambda\mathbf{q},$$

where λ is a scaling factor. This is basically an *eigenvalue problem*.

PCA

- With an $m \times m$ covariance matrix $\mathbf{R}$, we can get m eigenvectors and m eigenvalues:

$$\mathbf{R}\mathbf{q}_j = \lambda_j \mathbf{q}_j , j = 1, 2, ..., m$$

- We can sort the eigenvectors/eigenvalues according to the eigenvalues, so that

$$\lambda_1 > \lambda_2 > ... > \lambda_m .$$

and arrange the eigenvectors in a column-wise matrix

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_m].$$

- Then we can write

$$\mathbf{R}\mathbf{Q} = \mathbf{Q}\boldsymbol{\lambda}$$

where $\boldsymbol{\lambda} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m)$.

- $\mathbf{Q}$ is orthogonal, so that $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$. That is, $\mathbf{Q}^{-1} = \mathbf{Q}^T$.

PCA: Summary

- The eigenvectors of the covariance matrix $\mathbf{R}$ of zero-mean random input vector $\mathbf{X}$ define the principal directions $\mathbf{q}_j$ along with the variance of the projected inputs have extremal values.
- The associated eigenvaluess define the extremal values of the variance probe.

PCA: Usage

- Project input $\mathbf{x}$ to the principal directions:

$$\mathbf{a} = \mathbf{Q}^T \mathbf{x}.$$

- We can also recover the input from the projected point $\mathbf{a}$:

$$\mathbf{x} = (\mathbf{Q}^T)^{-1}\mathbf{a} = \mathbf{Q}\mathbf{a}.$$

- Note that we don't need all m principal directions, depending on how much variance is captured in the first few eigenvalues: We can do dimensionality reduction.

PCA: Dimensionality Reduction

- **Encoding**: We can use the first l eigenvectors to encode $\mathbf{x}$.

$$[a_1, a_2, ..., a_l]^T = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_l]^T \mathbf{x}.$$

- Note that we only need to calculate l projections $a_1, a_2, ..., a_l$, where $l \leq m$.
- **Decoding**: Once $[a_1, a_2, ..., a_l]^T$ is obtained, we want to reconstruct the full $[x_1, x_2, ..., x_l, ..., x_m]^T$.

$$\mathbf{x} = \mathbf{Q}\mathbf{a} \approx [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_l][a_1, a_2, ..., a_l]^T = \hat{\mathbf{x}}.$$

Or, alternatively

$$\hat{\mathbf{x}} = \mathbf{Q}[a_1, a_2, ..., a_l, \underbrace{0, 0, ..., 0}_{m - l \text{ zeros}}]^T.$$

PCA: Total Variance

- The total variance of th em components of the data vector is

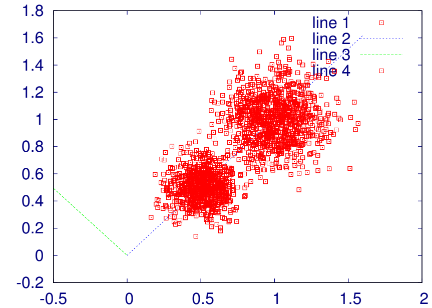
$$\sum_{j=1}^{m} \sigma_j^2 = \sum_{j=1}^{m} \lambda_j.$$

- The truncated version with the first l components have variance

$$\sum_{j=1}^{l} \sigma_j^2 = \sum_{j=1}^{l} \lambda_j.$$

- The larger the variance in the truncated version, i.e., the smaller the variance in the remaining components, the more accurate the dimensionality reduction.

PCA Example




```
inp=[randn(800,2)/9+0.5;randn(1000,2)/6+ones(1000,2)];
```

$$\mathbf{Q} = \begin{bmatrix} 0.70285 & -0.71134 \\ 0.71134 & 0.70285 \end{bmatrix}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} 0.14425 & 0.00000 \\ 0.00000 & 0.02161 \end{bmatrix}$$

PCA's Relation to Neural Networks: Hebbian-Based Maximum Eigenfilter

- How does all the above relate to **neural networks**?
- A remarkable result by Oja (1982) shows that a single linear neuron with Hebbian synapse can evolve into a filter for the first principal component of the input distribution!
  - Activation:

$$y = \sum_{i=1}^{m} w_i x_i$$

  - Learning rule:

$$w_i(n+1) = \frac{w_i(n) + \eta y(n) x_i(n)}{\left(\sum_{i=1}^{m} [w_i(n) + \eta y(n) x_i(n)]^2\right)^{1/2}}$$

Hebbian-Based Maximum Eigenfilter

- Expanding the denominator as a power series, dropping the higher order terms, etc., we get

$$w_i(n+1) = w_i(n) + \eta y(n)[x_i(n) - y(n) w_i(n)] + O(\eta^2),$$

with $O(\eta^2)$ including the second- and higher-order effects of η, which we can ignore for small η.

- Based on that, we get

$$\begin{aligned} w_i(n+1) &= w_i(n) + \eta y(n)[x_i(n) - y(n) w_i(n)] \\ &= w_i(n) + \eta \left(\underbrace{y(n) x_i(n)}_{\text{Hebbian term}} - \underbrace{y(n)^2 w_i(n)}_{\text{Stabilization term}} \right). \end{aligned}$$

Matrix Formulation of the Algorithm

- Activation

$$y(n) = \mathbf{x}^T(n)\mathbf{w}(n) = \mathbf{w}^T(n)\mathbf{x}(n)$$

- Learning

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta y(n)[\mathbf{x}(n) - y(n)\mathbf{w}(n)]$$

- Combining the above,

$$\begin{aligned}\mathbf{w}(n+1) = \quad & \mathbf{w}(n) + \eta[\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) \\ & -\mathbf{w}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)\mathbf{w}(n)]\end{aligned},$$

  represents a nonlinear stochastic difference equation, which is hard to analyze.

Asymptotic Stability Theorem

- To ease the analysis, we rewrite the learning rule as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)h(\mathbf{w}(n), \mathbf{x}(n)).$$

- The goal is to associate a *deterministic ordinary differential equation (ODE)* with the stochastic equation.
- Under certain reasonable conditions on η, $\mathbf{h}(\cdot,\cdot)$, and $\mathbf{w}$, we get the **asymptotic stability theorem** stating that

$$\lim_{n\to\infty} \mathbf{w}(n) = \mathbf{q}_1$$

  infinitely often with probability 1.

Conditions for Stability

1. $\eta(n)$ is a decreasing sequence of positive real numbers such that $\sum_{n=1}^{\infty} \eta(n) = \infty$, $\sum_{n=1}^{\infty} \eta^p(n) < \infty$for $p > 1$, $\eta(n) \to 0 \text{ as } n \to \infty$.
2. Sequence of parameter vectors $\mathbf{w}(\cdot)$ is bounded with probability 1.
3. The update function $h(\mathbf{w}, \mathbf{x})$ is continuously differentiable w.r.t. $\mathbf{w}$ and $\mathbf{x}$, and it derivatives are bounded in time.
4. The limit $\bar{h}(\mathbf{w}) = \lim_{n\to\infty} E[h(\mathbf{w}, \mathbf{X})]$ exists for each $\mathbf{w}$, where $\mathbf{X}$ is a random vector.
5. There is a locally asymptotically stable solution to the ODE

$$\frac{d}{dt}\mathbf{w}(t) = \hat{h}(\mathbf{w}(t)).$$

6. Let $\mathbf{q}_1$ denote the solution to the ODE above with a basin of attraction $\mathcal{B}(\mathbf{q})$. The parameter vector $\mathbf{w}(n)$ enters the compact subset $\mathcal{A}$ of $\mathcal{B}(\mathbf{q})$ infinitely often with prob. 1.

Stability Analysis of Maximum Eigenfilter

Set it up to satisfy the conditions of the asymptotic stability theorem:

- Set the learning rate to be $\eta(n) = 1/n$.
- Set $h(\cdot,\cdot)$ to

$$\begin{aligned}h(\mathbf{w}, \mathbf{x}) = \quad & \mathbf{x}(n)y(n) - y^2\mathbf{w}(n) \\ = \quad & \mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) - [\mathbf{w}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)]\mathbf{w}(n)\end{aligned}$$

- Taking expectaion over all $\mathbf{x}$,

$$\begin{aligned}\bar{h} = \quad & \lim_{n\to\infty} E[\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{w}(n) - (\mathbf{w}^T(n)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{w}(n))\mathbf{w}(n)] \\ = \quad & \mathbf{R}\mathbf{w}(\infty) - \left[\mathbf{w}^T(\infty)\mathbf{R}\mathbf{w}(\infty)\right]\mathbf{w}(\infty)\end{aligned}$$

- Substituting $\bar{h}$ into the ODE,

$$\frac{d}{dt}\mathbf{w}(t) = \bar{h}(\mathbf{w}(t)) = \mathbf{R}\mathbf{w}(t) - [\mathbf{w}^T(t)\mathbf{R}\mathbf{w}(t)]\mathbf{w}(t).$$

Stability Analysis of Maximum Eigenfilter

- Expanding $\mathbf{w}(t)$ with the eigenvectors of $\mathbf{R}$,

$$\mathbf{w}(t) = \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k,$$

and using basic definitions

$$\mathbf{R}\mathbf{q}_k = \lambda_k \mathbf{q}, \mathbf{q}_k^T \mathbf{R}\mathbf{q}_k = \lambda_k$$

we get (see next slide for derivation)

$$\sum_{k=1}^{m} \frac{d\theta_k(t)}{dt}\mathbf{q}_k = \sum_{k=1}^{m} \lambda_k \theta_k(t)\mathbf{q}_k - \left[\sum_{l=1}^{m} \lambda_l \theta_l^2(t)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k.$$

Stability Analysis of Maximum Eigenfilter (cont'd)

Equating the RHS's of the following

$$\frac{d\mathbf{w}(t)}{dt} = \frac{d}{dt}\left(\sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k\right),$$

$$\frac{d}{dt}\mathbf{w}(t) = \bar{h}(\mathbf{w}(t)) = \mathbf{R}\mathbf{w}(t) - [\mathbf{w}^T(t)\mathbf{R}\mathbf{w}(t)]\mathbf{w}(t).$$

we get

$$\sum_{k=1}^{m} \frac{d\theta_k(t)}{dt}\mathbf{q}_k = \sum_{k=1}^{m} \lambda_k \theta_k(t)\mathbf{q}_k - \left[\sum_{l=1}^{m} \lambda_l \theta_l^2(t)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k.$$

Stability Analysis of Maximum Eigenfilter (cont'd)

First, we show $\mathbf{R}\mathbf{w}(t) = \sum_{k=1}^{m} \lambda_k \theta_k(t)\mathbf{q}_k$, using $\mathbf{R}\mathbf{q}_k = \lambda_k \mathbf{q}$.

$$\begin{aligned}
\mathbf{R}\mathbf{w}(t) &= \mathbf{R}\sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \sum_{k=1}^{m} \theta_k(t)\mathbf{R}\mathbf{q}_k \\
&= \sum_{k=1}^{m} \lambda_k \theta_k(t)\mathbf{q}_k
\end{aligned}$$

Stability Analysis of Maximum Eigenfilter (cont'd)

Next, we show
$[\mathbf{w}^T(t)\mathbf{R}\mathbf{w}(t)]\mathbf{w}(t) = \left[\sum_{l=1}^{m} \lambda_l \theta_l^2(t)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k.$

$$\begin{aligned}
&[\mathbf{w}^T(t)\mathbf{R}\mathbf{w}(t)]\mathbf{w}(t) \\
&= [\mathbf{w}^T(t)\mathbf{R}\mathbf{w}(t)] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \left[\left(\sum_{l=1}^{m} \theta_l(t)\mathbf{q}_l^T\right) \mathbf{R} \left(\sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k\right)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \left[\sum_{l=1}^{m} \left(\theta_l(t)\mathbf{q}_l^T \mathbf{R} \left(\sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k\right)\right)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \left[\sum_{l=1}^{m} \left(\sum_{k=1}^{m} \theta_l(t)\mathbf{q}_l^T \mathbf{R}\theta_k(t)\mathbf{q}_k\right)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \left[\sum_{l=1}^{m} \left(\sum_{k=1}^{m} \theta_l(t)\theta_k(t)\mathbf{q}_l^T \mathbf{R}\mathbf{q}_k\right)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \left[\sum_{l=1}^{m} \left(\sum_{k=1}^{m} \theta_l(t)\theta_k(t)\mathbf{q}_l^T (\lambda_k \mathbf{q}_k)\right)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \left[\sum_{l=1}^{m} \left(\sum_{k=1}^{m} \theta_l(t)\theta_k(t)\lambda_k \mathbf{q}_l^T \mathbf{q}_k)\right)\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&\quad \{\text{ Inner sum disappears since } \mathbf{q}_l^T \mathbf{q}_k = 0 \text{ for } l \neq k \text{ and } = 1 \text{ for } l = k\} \\
&= \left[\sum_{l=1}^{m} \theta_l(t)\theta_l(t)\lambda_l\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k \\
&= \left[\sum_{l=1}^{m} \theta_l^2(t)\lambda_l\right] \sum_{k=1}^{m} \theta_k(t)\mathbf{q}_k
\end{aligned}$$

Stability Analysis of Maximum Eigenfilter (cont'd)

- Factoring out $\mathbf{q}_k$, we get

$$\frac{d\theta_k(t)}{dt} = \lambda_k\theta_k(t) - \left[\sum_{l=1}^{m}\lambda_l\theta_l^2(t)\right]\theta_k(t).$$

- We can analyze the above in two cases (details in following slides):
  - Case I: $k \neq 1$
    In this case, $\alpha_k(t) = \frac{\theta_k(t)}{\theta_1(t)} \to 0$ as $t \to \infty$, by using $\frac{d\theta_k(t)}{dt}$ above to derive $\frac{d\alpha_k(t)}{dt} = -\underbrace{(\lambda_1 - \lambda_k)}_{\text{positive!}}\alpha_k(t)$.
  - Case II: $k = 1$
    In this case, $\theta_1(t) \to \pm 1$ as $t \to \infty$, from $\frac{d\theta_1(t)}{dt} = \lambda_1\theta_1(t)\left[1 - \theta_1^2(t)\right]$.

Stability Analysis of Maximum Eigenfilter (cont'd)

Case I (in detail): $k \neq 1$

- Given

$$\frac{d\theta_k(t)}{dt} = \lambda_k\theta_k(t) - \left[\sum_{l=1}^{m}\lambda_l\theta_l^2(t)\right]\theta_k(t).(1)$$

- Define $\alpha_k(t) = \frac{\theta_k(t)}{\theta_1(t)}$.
- Derive

$$\frac{d\alpha_k(t)}{dt} = \frac{1}{\theta_1(t)}\frac{d\theta_k(t)}{dt} - \frac{\theta_k(t)}{\theta_1^2(t)}\frac{d\theta_1(t)}{dt}(2)$$

- Plug in (1) above into (2). (Both $d\theta_k(t)/dt$ and $d\theta_1(k)/dt$.)
- Finally, we get: $\frac{d\alpha_k(t)}{dt} = -(\lambda_1 - \lambda_k)\alpha_k(t)$, so $\alpha_k(t) \to 0$ as $t \to \infty$.

Stability Analysis of Maximum Eigenfilter (cont'd)

Case II: $k = 1$

$$\begin{aligned}
\frac{d\theta_1(t)}{dt} &= \lambda_1\theta_1(t) - \left[\sum_{l=1}^{m}\lambda_l\theta_l^2(t)\right]\theta_k(t) \\
&= \lambda_1\theta_1(t) - \lambda_1\theta_1^3(t) - \theta_1(t)\sum_{l=2}^{m}\lambda_l\theta_l^2(t) \\
&= \lambda_1\theta_1(t) - \lambda_1\theta_1^3(t) - \theta_1^3(t)\sum_{l=2}^{m}\lambda_l\alpha_l^2(t)
\end{aligned}$$

Using results from Case I ($\alpha_l \to 0$ for $l \neq 1$ and $t \to \infty$), $\theta_1(t) \to \pm 1$ as $t \to \infty$, from $\frac{d\theta_1(t)}{dt} = \lambda_1\theta_1(t)\left[1 - \theta_1^2(t)\right]$.

Stability Analysis of Maximum Eigenfilter (cont'd)

- Recalling the original expansion

$$\mathbf{w}(t) = \sum_{k=1}^{m}\theta_k(t)\mathbf{q}_k,$$

  we can conclude that

$$\mathbf{w}(t) \to \mathbf{q}_1, \ \text{as } t \to \infty.$$

  where $\mathbf{q}_1$ is the normalized eigenvector associated with the largest eigenvalue λ_1 of the covariance matrix $\mathbf{R}$.

- Other conditions of stability can also be shown to hold (see the textbook).

Summary of Hebbian-Based Maximum Eigenfilter

Hebbian-based linear neuron converges with probability 1 to a fixed point, which is characterized as follows:

- Variance of output approaches the largest eigenvalue of the covariance matrix $\mathbf{R}$ ($y(n)$ is the output):

$$\lim_{n\to\infty} \sigma^2(n) = \lim_{n\to\infty} E[Y^2(n)] = \lambda_1$$

- Synaptic weight vector approaches the associated eigenvector

$$\lim_{n\to\infty} \mathbf{w}(n) = \mathbf{q}_1$$

with

$$\lim_{n\to\infty} \|\mathbf{w}(n)\| = 1.$$

Generalized Hebbian Algorithm for full PCA

- Sanger (1989) showed how to construct a feedfoward network to learn all the eigenvectors of $\mathbf{R}$.
- Activation

$$y_j(n) = \sum_{i=1}^{m} w_{ji}(n)x_i(n), j = 1, 2, ..., l$$

- Learning

$$\Delta w_{ji}(n) = \eta \left[y_j(n)x_i(n) - y_j(n) \sum_{k=1}^{j} w_{ki}(n)y_k(n) \right],$$

$$i = 1, 2, ..., m, \quad j = 1, 2, ..., l.$$