

# Slide10

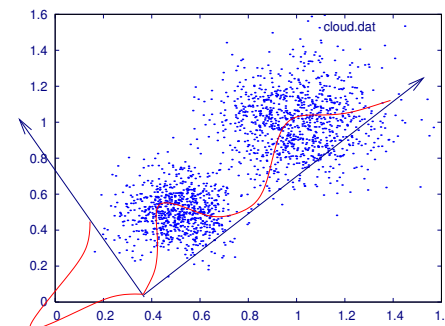
## Haykin Chapter 8: Principal Components Analysis

CPSC 636-600

Instructor: Yoonsuck Choe

Spring 2012

### Motivation



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

1

2

### 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:

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

3

### 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 external value?*
- The solution to the question is obtained by finding unit vectors satisfying the following condition:

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

where  $\lambda$  is a scaling factor. This is basically an *eigenvalue problem*.

4

## PCA

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

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

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

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

and arrange the eigenvectors in a column-wise matrix

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

- Then we can write

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

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

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

5

## 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.

7

## PCA: Summary

- The eigenvectors of the correlation 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 eigenvalues define the extremal values of the variance probe.

6

## PCA: Dimensionality Reduction

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

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

- Note that we only need to calculate  $l$  projections  $a_1, a_2, \dots, a_l$ , where  $l \leq m$ .

- **Decoding:** Once  $[a_1, a_2, \dots, a_l]^T$  is obtained, we want to reconstruct the full  $[x_1, x_2, \dots, x_l, \dots, x_m]^T$ .

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

Or, alternatively

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

8

## PCA: Total Variance

- The total variance of the  $m$  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.

9

## 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:

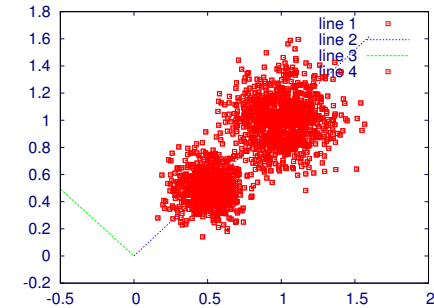
$$y = \sum_{i=1}^m w_i x_i$$

- Learning rule:

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

11

## PCA Example



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

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

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

10

## Hebbian-Based Maximum Eigenfilter

- Expanding the weight update rule using Taylor series, 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  $\eta$ , which we can ignore for small  $\eta$ .

- Based on that, we get

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

12

## 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,

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

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

13

## 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  $p > 1$ ,  $\eta(n) \rightarrow 0$  as  $n \rightarrow \infty$ .
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 its derivatives are bounded in time.
4. The limit  $\bar{h}(\mathbf{w}) = \lim_{n \rightarrow \infty} E[h(\mathbf{w}, \mathbf{X})]$  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  $\mathcal{B}(\mathbf{q}_1)$ . The parameter vector  $\mathbf{w}(n)$  enters the compact subset  $\mathcal{A}$  of  $\mathcal{B}(\mathbf{q}_1)$  infinitely often with prob. 1.

15

## 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  $\eta$ ,  $h(\cdot, \cdot)$ , and  $\mathbf{w}$ , we get the **asymptotic stability theorem** stating that

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

infinitely often with probability 1.

14

## 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

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

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

$$\begin{aligned} \bar{h} &= \lim_{n \rightarrow \infty} E[\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) - (\mathbf{w}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n))\mathbf{w}(n)] \\ &= \mathbf{R}\mathbf{w}(\infty) - [\mathbf{w}^T(\infty)\mathbf{R}\mathbf{w}(\infty)]\mathbf{w}(\infty) \end{aligned}$$

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

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

16

## 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)

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

17

## 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

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

18

## 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  $\mathbf{R}\mathbf{q}_k = \lambda_k \mathbf{q}$ .

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

19

## Stability Analysis of Maximum Eigenfilter (cont'd)

Next, we show

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

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

20

## 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:

- Case I:  $k \neq 1$

In this case,  $\alpha_k(t) = \frac{\theta_k(t)}{\theta_1(t)} \rightarrow 0$  as  $t \rightarrow \infty$ , by using

$$\frac{d\theta_k(t)}{dt} \text{ above to derive } \frac{d\alpha_k(t)}{dt} = -(\underbrace{\lambda_1 - \lambda_k}_{\text{positive!}}) \alpha_k(t).$$

- Case II:  $k = 1$

In this case,  $\theta_1(t) \rightarrow \pm 1$  as  $t \rightarrow \infty$ , from

$$\frac{d\theta_1(t)}{dt} = \lambda_1 \theta_1(t) [1 - \theta_1^2(t)].$$

21

## 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 correlation matrix  $\mathbf{R}$  ( $y(n)$  is the output):

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

- Synaptic weight vector approaches the associated eigenvector

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

with

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

23

## 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

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

where  $\mathbf{q}_1$  is the normalized eigenvector associated with the largest eigenvalue  $\lambda_1$  of the correlation matrix  $\mathbf{R}$ .

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

22

## Generalized Hebbian Algorithm for full PCA

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

- Activation

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

- Learning

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

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

24