Machine Learning

Volker Roth

Department of Mathematics & Computer Science
University of Basel

Chapter 10: Linear latent variable models

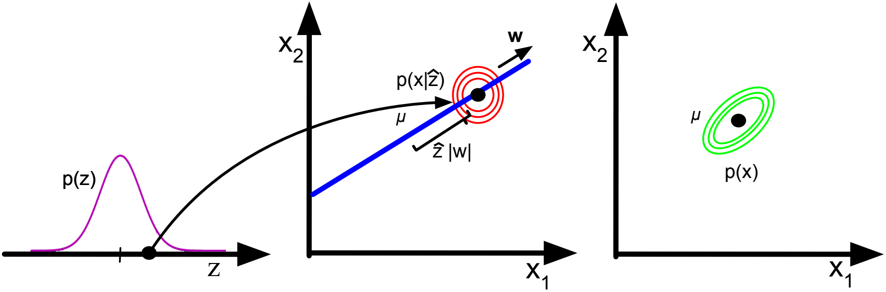
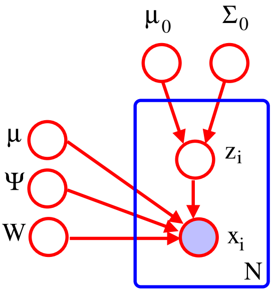


Figure 12.1 in K. Murphy: Machine Learning. MIT Press 2012.

Factor analysis

- One problem with mixture models: **only a single latent variable**. Each observation can only come from one of K prototypes.
- Alternative: $\boldsymbol{z}_i \in \mathbb{R}^k$. Gaussian prior:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i | \boldsymbol{\mu}_0, \Sigma_0)$$



- For observations $\boldsymbol{x}_i \in \mathbb{R}^p$, we may use a **Gaussian likelihood.**
- As in linear regression, we assume the mean is a **linear** function:

$$p(\boldsymbol{x}_i | \boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(W\boldsymbol{z}_i + \boldsymbol{\mu}, \Psi),$$

  W: **factor loading matrix**, and Ψ: **covariance matrix**.
- We take Ψ to be **diagonal**, since the whole point of the model is to "force" z_i to **explain the correlation.**

Factor analysis: generative process

Generative process ($k = 1$, $p = 2$, diagonal Ψ):

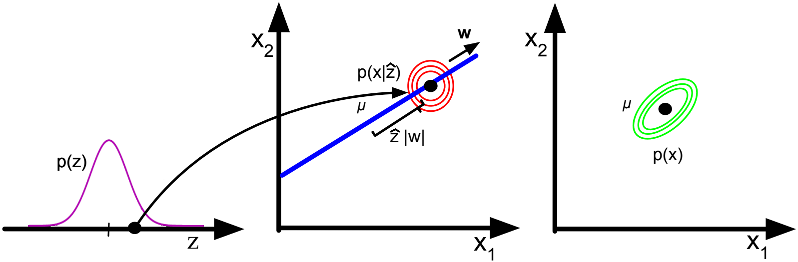


Figure 12.1 in K. Murphy: Machine Learning. MIT Press 2012.

We take an isotropic Gaussian "spray can" and slide it along the 1d line defined by $wz_i + \mu$. This induces a correlated Gaussian in 2d.

FA is a low rank parameterization of an MVN

- The induced marginal distribution $p(\boldsymbol{x}_i|\boldsymbol{\theta})$ is Gaussian:

$$\begin{aligned} p(\boldsymbol{x}_i|\boldsymbol{\theta}) &= \int \mathcal{N}(\boldsymbol{x}_i, W\boldsymbol{z}_i + \boldsymbol{\mu}, \Psi)\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \Sigma_0)\, d\boldsymbol{z}_i, \\ &= \mathcal{N}(\boldsymbol{x}_i|W\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \Psi + W\Sigma_0 W^t) \end{aligned}$$

- We can set $\boldsymbol{\mu}_0 = \mathbf{0}$ without loss of generality (absorb $W\boldsymbol{\mu}_0$ into $\boldsymbol{\mu}$).
  Similarly, we can set $\Sigma_0 = I$ using $\tilde{W} = W\Sigma_0^{-1/2}$.
  Covariance structure:

$$cov[\boldsymbol{x}|\boldsymbol{\theta}] = (W\Sigma_0^{-1/2})\Sigma_0(W\Sigma_0^{-1/2})^t + \Psi = WW^t + \Psi.$$

- We thus see that FA approximates the covariance matrix of $\boldsymbol{x}$ using a low-rank decomposition:

$$C := cov[\boldsymbol{x}] = WW^t + \Psi.$$

  Only $O(p \cdot k)$ parameters $\rightsquigarrow$ compromise between full covariance with $O(p^2)$ params, and a diagonal covariance $O(p)$ params.

Inference of the latent factors

- We hope that the latent factors z will reveal something interesting about the data $\rightsquigarrow$ compute posterior over the latent variables:

$$\begin{aligned} p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{m}_i, \Sigma) \\ \Sigma &= (I + W^t \Psi^{-1} W)^{-1} \\ \boldsymbol{m}_i &= \Sigma W^t \Psi^{-1} \boldsymbol{x}_i \end{aligned}$$

- The posterior means $\boldsymbol{m}_i$ are called the **latent scores, or latent factors.**

Example

- Example from (Shalizi 2009). $p = 11$ variables and $n = 387$ cases describing aspects of cars: engine size, $\#$(cylinders), miles per gallon (MPG), price, etc.
- Fit a $p = 2$ dim model. Plot m_i scores as points in $\mathbb{R}^2$.
- To get a better understanding of the "meaning" of the latent factors, project unit vectors $e_1 = (1, 0, \ldots, 0), e_2 = (0, 1, 0, \ldots, 0)$, etc. into the low dimensional space (blue lines)
- Horizontal axis represents price, corresponding to the features labeled "dealer" and "retail", with expensive cars on the right. Vertical axis represents fuel efficiency (measured in terms of MPG) versus size: heavy vehicles are less efficient and are higher up, whereas light vehicles are more efficient and are lower down.
- Verify by finding the closest exemplars in the training set.

Example

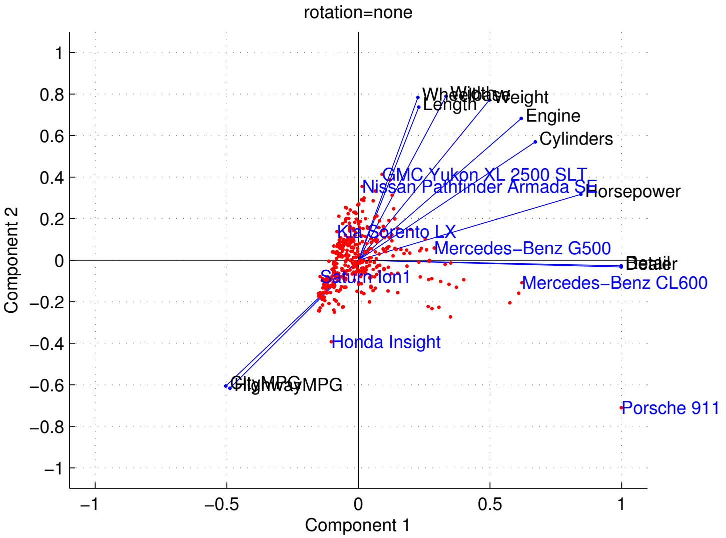
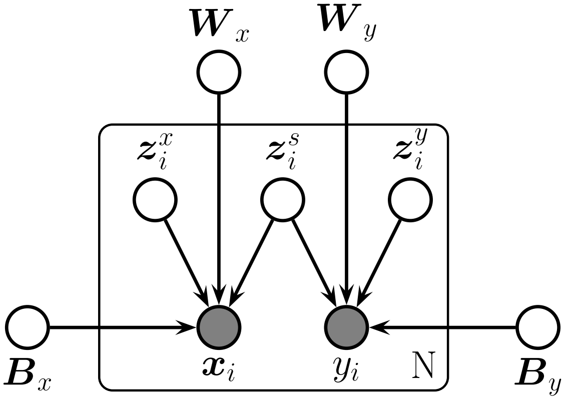


Figure 12.2 in K. Murphy: Machine Learning. MIT Press 2012.

Special Cases: PCA and CCA

- Covariance matrix $\Psi = \sigma^2 I \rightsquigarrow$ (probabilistic) **PCA**.
- Two-view version involving $\boldsymbol{x}$ and $\boldsymbol{y} \rightsquigarrow$ **CCA**.



From figure 12.19 in K. Murphy: Machine Learning. MIT Press 2012.

PCA and dimensionality reduction

Given n data points in p dimensions:

$$X = \begin{bmatrix} - & \boldsymbol{x}_1 & - \\ - & \boldsymbol{x}_2 & - \\ - & \vdots & - \\ - & \boldsymbol{x}_n & - \end{bmatrix} \in \mathbb{R}^{n \times p}$$

Want to reduce dimensionality from p to k. Choose k directions $\boldsymbol{w}_1, \dots, \boldsymbol{w}_k$, arrange them as columns in matrix W:

$$W = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_k \end{bmatrix} \in \mathbb{R}^{p \times k}$$

For each $\boldsymbol{w}_j$, compute **similarity** $z_j = \boldsymbol{w}_j^t \boldsymbol{x}, \; j = 1 \dots k$.
Project $\boldsymbol{x}$ down to $\boldsymbol{z} = (z_1, \dots, z_k)^t = W^t \boldsymbol{x}$. How to choose W?

Encoding–decoding model

The projection matrix W serves two functions:

- **Encode:** $\mathbf{z} = W^t\mathbf{x}, \;\; \mathbf{z} \in \mathbb{R}^k, \;\; z_j = \mathbf{w}_j^t\mathbf{x}$.
  - The vectors $\mathbf{w}_j$ form a basis of the projected space.
  - We will require that this basis is orthonormal, i.e. $W^tW = I$.
- **Decode:** $\tilde{\mathbf{x}} = W\mathbf{z} = \sum_{j=1}^k z_j\mathbf{w}_j, \;\; \tilde{\mathbf{x}} \in \mathbb{R}^p$.
  - If $k = p$, the above orthonormality condition implies $W^t = W^{-1}$, and encoding can be undone without loss of information.
  - If $k < p$, we use the pseudo-inverse $\rightsquigarrow$ the reconstruction error will be nonzero.
- Above we assumed that the origin of the coordinate system is in the sample mean, i.e. $\sum_i \mathbf{x}_i = 0$.

Principal Component Analysis (PCA)

In the general case, we want the reconstruction error $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$ to be small.
Objective: minimize $\min_{W \in \mathbb{R}^{p \times k}:\, W^t W = I} \sum_{i=1}^{n} \|\boldsymbol{x}_i - W W^t \boldsymbol{x}_i\|^2$

Finding the principal components

Projection vectors are orthogonal $\rightsquigarrow$ can treat them separately:

$$\min_{\boldsymbol{w}:\, \|\boldsymbol{w}\|=1} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^t \boldsymbol{x}_i\|^2$$

$$\begin{aligned}
\sum_i \|\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^t \boldsymbol{x}_i\|^2 &= \sum_{i=1}^{n} [\boldsymbol{x}_i^t \boldsymbol{x}_i - 2\boldsymbol{x}_i^t \boldsymbol{w}\boldsymbol{w}^t \boldsymbol{x}_i + \boldsymbol{x}_i^t \boldsymbol{w}\underbrace{\boldsymbol{w}^t \boldsymbol{w}}_{=1}\boldsymbol{w}^t \boldsymbol{x}_i] \\
&= \sum_i [\boldsymbol{x}_i^t \boldsymbol{x}_i - \boldsymbol{x}_i^t \boldsymbol{w}\boldsymbol{w}^t \boldsymbol{x}_i] \\
&= \sum_i \boldsymbol{x}_i^t \boldsymbol{x}_i - \boldsymbol{w}^t \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^t \boldsymbol{w} \\
&= \underbrace{\sum_i \boldsymbol{x}_i^t \boldsymbol{x}_i}_{\text{const.}} - \boldsymbol{w}^t X^t X \boldsymbol{w}.
\end{aligned}$$

Finding the principal components

- Want to maximize $\boldsymbol{w}^t X^t X \boldsymbol{w}$ under the constraint $\|\boldsymbol{w}\| = 1$
- Can also maximize the ratio $J(\boldsymbol{w}) = \frac{\boldsymbol{w}^t X^t X \boldsymbol{w}}{\boldsymbol{w}^t \boldsymbol{w}}$ and rescale $\hat{\boldsymbol{w}}$.
- Optimal projection u is the eigenvector of $X^t X$ with largest eigenvalue (J is a Rayleigh quotient).
- Note that we assumed that $\sum_i \boldsymbol{x}_i = 0$. Thus, the columns of X are assumed to sum to zero.
  $\rightsquigarrow$ compute SVD of "centered" matrix X
  $\rightsquigarrow$ column vectors in W are eigenvectors of $X^t X$
  $\rightsquigarrow$ they are the principal components.

Eigen-faces [Turk and Pentland, 1991]

- p = number of pixels
- Each $\boldsymbol{x}_i \in \mathbb{R}^p$ is a face image
- x_{ji} = intensity of the j-th pixel in image i

$$(X^t)_{p\times n} \approx W_{p\times k} \quad (Z^t)_{k\times n}$$

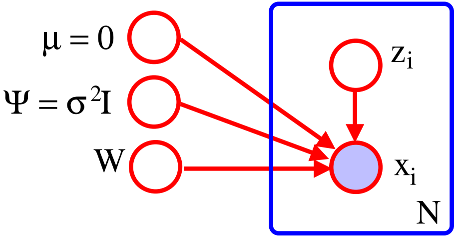
$\Big($  $\Big) \approx \Big($  $\Big)$ $\begin{bmatrix} | & & | \\ \boldsymbol{z}_1 & \dots & \boldsymbol{z}_n \\ | & & | \end{bmatrix}$

Idea: $\boldsymbol{z}_i$ more 'meaningful' representation of i-th face than $\boldsymbol{x}_i$

Can use $\boldsymbol{z}_i$ for nearest-neighbor classification

Much faster when $p \gg k$.

Probabilistic PCA



- Assuming $\Psi = \sigma^2 I$ and centered data in the FA model
  $\rightsquigarrow$ likelihood and marginal likelihood

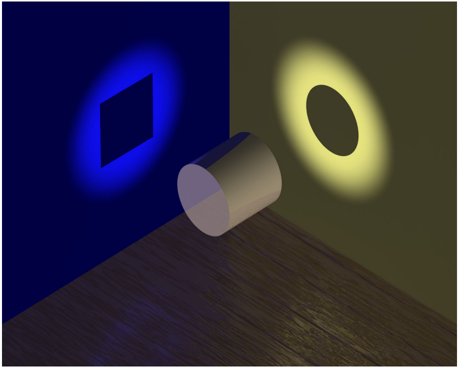
$$
\begin{aligned}
p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) &= \mathcal{N}(W\boldsymbol{z}_i, \sigma^2 I), \\
p(\boldsymbol{x}_i|\boldsymbol{\theta}) &= \int \mathcal{N}(\boldsymbol{x}_i, W\boldsymbol{z}_i, \sigma^2 I)\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, I)\, d\boldsymbol{z}_i, \\
&= \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{0}, \sigma^2 I + WW^t)
\end{aligned}
$$

Probabilistic PCA

- (Tipping & Bishop 1999): Maxima of the marg. likelihood given by
  $$\hat{W} = V(\Lambda - \sigma^2 I)^{\frac{1}{2}} R,$$
  where R is an arbitrary orthogonal matrix,
  columns of V: first k eigenvectors of $S = \frac{1}{n} X^t X$,
  Λ: diagonal matrix of eigenvalues.
- As $\sigma^2 \to 0$, we have $\hat{W} \to V$, **as in classical PCA** (for $R = \Lambda^{-\frac{1}{2}}$).
- Projections $\boldsymbol{z}_i$: Posterior over the latent factors:
  $$\begin{aligned} p(\boldsymbol{z}_i | \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}) &= \mathcal{N}(\boldsymbol{z}_i | \hat{\boldsymbol{m}}_i, \sigma^2 \hat{F}^{-1}) \\ \hat{F} &= \sigma^2 I + \hat{W}^t \hat{W} \\ \boldsymbol{m}_i &= \hat{F}^{-1} \hat{W}^t \boldsymbol{x}_i \end{aligned}$$
  For $\sigma^2 \to 0$, $\boldsymbol{z}_i \to \boldsymbol{m}_i$ and $\boldsymbol{m}_i \to V^t \boldsymbol{x}_i \rightsquigarrow$ orthogonal projection of the data onto the column space of V, **as in classical PCA.**

Multiple Views: CCA

- Consider paired samples from different views.
- What is the dependency structure between the views ?
- Standard approach: global linear dependency detected by CCA.



Canonical Correlation Analysis [Hotelling, 1936]

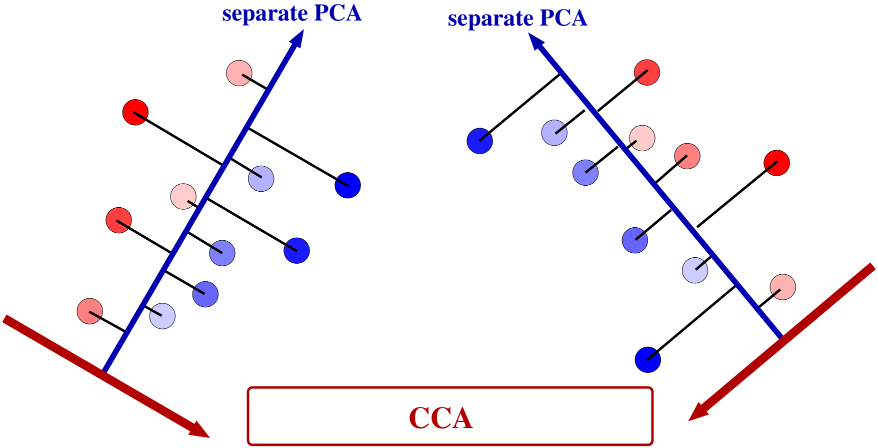
Often, each data point consists of **two views**:

- Image retrieval: for each image, have the following:
  - X: Pixels (or other visual features) Y: Text around the image
- Time series:
  - X: Signal at time t
  - Y: Signal at time $t + 1$
- Two-view learning: divide features into two sets
  - X: Features of a word/object, etc.
  - Y: Features of the context in which it appears

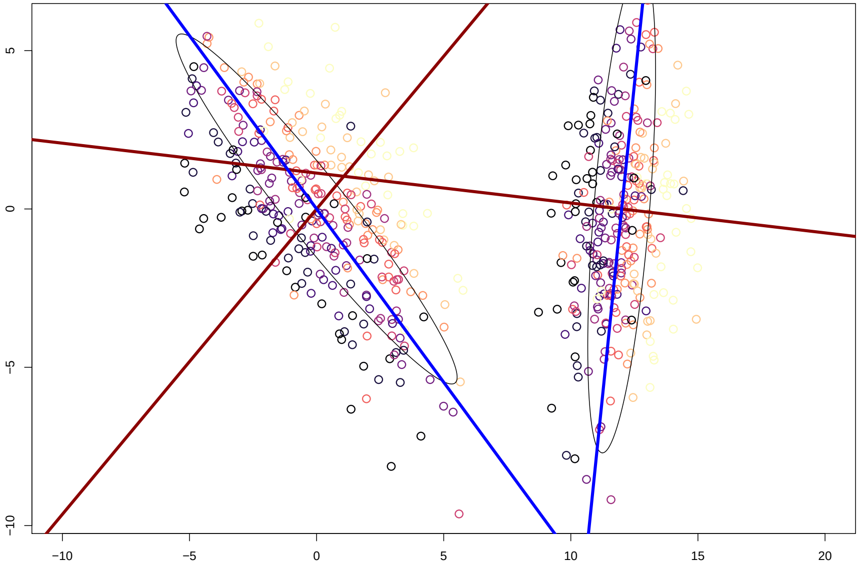
Goal: reduce the dimensionality of the two views **jointly**.

Find projections such that projected views are **maximally correlated.**

CCA vs PCA



CCA vs PCA



CCA: Setting

- Let $\boldsymbol{X}$ be a random vector $\in \mathbb{R}^{p_x}$ and $\boldsymbol{Y}$ be a random vector $\in \mathbb{R}^{p_y}$ Consider the combined $(p := p_x + p_y)$-dimensional random vector $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})^t$. Let its $(p \times p)$ covariance matrix be partitioned into blocks according to:

$$\boldsymbol{Z} = \begin{bmatrix} \Sigma_{XX} \in \mathbb{R}^{p_x \times p_x} & | & \Sigma_{XY} \in \mathbb{R}^{p_x \times p_y} \\ \Sigma_{YX} \in \mathbb{R}^{p_y \times p_x} & | & \Sigma_{YY} \in \mathbb{R}^{p_y \times p_y} \end{bmatrix}$$

- Assuming centered data, the blocks in the covariance matrix can be estimated from observed data sets $X \in \mathbb{R}^{n \times p_x}$, $Y \in \mathbb{R}^{n \times p_y}$:

$$\mathrm{Z} \approx \frac{1}{n} \begin{bmatrix} X^t X & | & X^t Y \\ Y^t X & | & Y^t Y \end{bmatrix}$$

CCA: Setting

- Correlation$(x, y) = \frac{\text{covariance}(x,y)}{\text{standard deviation}(x)\cdot \text{ standard deviation}(y)}$

$$\rho = cor(x, y) = \frac{cov(x, y)}{\sigma(x)\sigma(y)}.$$

- Sample correlation:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})^t}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \overset{\text{centered observations}}{=} \frac{\boldsymbol{x}^t \boldsymbol{y}}{\sqrt{\boldsymbol{x}^t \boldsymbol{x}}\sqrt{\boldsymbol{y}^t \boldsymbol{y}}}.$$

- Want to find maximally correlated 1D-projections $\boldsymbol{x}^t \boldsymbol{a}$ and $\boldsymbol{y}^t \boldsymbol{b}$.
- Projected covariance: $cov(\boldsymbol{x}^t \boldsymbol{a}, \boldsymbol{y}^t \boldsymbol{b}) \overset{\text{zero means}}{=} \boldsymbol{a}^t \Sigma_{XY} \boldsymbol{b}$.
- Define $\boldsymbol{c} = \Sigma_{XX}^{\frac{1}{2}} \boldsymbol{a}$, $\boldsymbol{d} = \Sigma_{YY}^{\frac{1}{2}} \boldsymbol{b}$.
- Thus, the projected correlation coefficient is: $\rho = \frac{\boldsymbol{c}^t \Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XY} \Sigma_{YY}^{-\frac{1}{2}} \boldsymbol{d}}{\sqrt{\boldsymbol{c}^t \boldsymbol{c}}\sqrt{\boldsymbol{d}^t \boldsymbol{d}}}$.

CCA: Setting

- By the Cauchy-Schwarz inequality ($x^t y \leq \|x\| \cdot \|y\|$), we have

$$\Big(c^t \underbrace{\Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XY} \Sigma_{YY}^{-\frac{1}{2}}}_{H}\Big) d \leq \Big(c^t \underbrace{\Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-\frac{1}{2}}}_{G := HH^t} c\Big)^{\frac{1}{2}} \left(d^t d\right)^{\frac{1}{2}},$$

$$\rho \leq \frac{(c^t G c)^{\frac{1}{2}}}{(c^t c)^{\frac{1}{2}}},$$

$$\rho^2 \leq \frac{c^t G c}{c^t c}.$$

- Equality: vectors d and $\Sigma_{YY}^{-\frac{1}{2}} \Sigma_{YX} \Sigma_{XX}^{-\frac{1}{2}} c$ are collinear.
- Maximum: c is the eigenvector with the maximum eigenvalue of $G := \Sigma_{XX}^{-\frac{1}{2}} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-\frac{1}{2}}$.
  Subsequent pairs $\rightsquigarrow$ using eigenvalues of decreasing magnitudes.
- Collinearity: $d \propto \Sigma_{YY}^{-\frac{1}{2}} \Sigma_{YX} \Sigma_{XX}^{-\frac{1}{2}} c$
- Transform back to original variables $a = \Sigma_{XX}^{-\frac{1}{2}} c$, $b = \Sigma_{YY}^{-\frac{1}{2}} d$.

Pixels That Sound [Kidron, Schechner, Elad, 2005]

“People and animals fuse auditory and visual information to obtain robust perception. A particular benefit of such cross-modal analysis is the ability to localize visual events associated with sound sources. We aim to achieve this using computer-vision aided by a single microphone”.

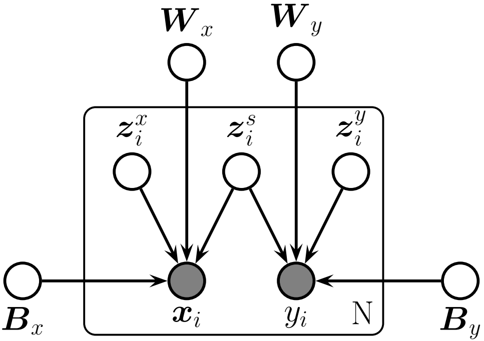
https://webee.technion.ac.il/ yoav/research/pixels-that-sound.html

Probabilistic CCA

(Bach and Jordan 2005): With Gaussian priors

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}^s|\mathbf{0}, I)\mathcal{N}(\boldsymbol{z}^x|\mathbf{0}, I)\mathcal{N}(\boldsymbol{z}^y|\mathbf{0}, I),$$

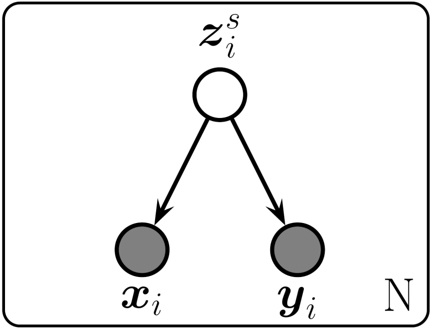
the MLE in the two-view FA model is equivalent to classical CCA (up to rotation and scaling).



From figure 12.19 in K. Murphy: Machine Learning. MIT Press 2012.

Probabilistic CCA: Essential Model Structure

Shared $\boldsymbol{z}^S$ decorrelates the two views $(\boldsymbol{x}, \boldsymbol{y})$:



$$
\begin{aligned}
\boldsymbol{x} &\in \mathbb{R}^p,\ \boldsymbol{y} \in \mathbb{R}^q \\
\boldsymbol{z}^S &\sim \mathcal{N}(\boldsymbol{z}^s | \mathbf{0}, I) \\
(\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{z} &\sim \mathcal{N}_{p+q}(\boldsymbol{\mu}_z, \Sigma) \\
\Sigma &= \begin{pmatrix} \Sigma_x & 0 \\ 0 & \Sigma_y \end{pmatrix}
\end{aligned}
$$

Further connections

- If y is a **discrete class label** $\rightsquigarrow$ CCA is (essentially) equivalent to Linear Discriminant Analysis (LDA), see (Hastie et al. 1994).
- Arbitrary $\mathbf{y}$ $\rightsquigarrow$ CCA is (essentially) equivalent to the **Gaussian Information Bottleneck** (Chechik et al. 2005)
  - Basic idea: **compress $\mathbf{x}$** into compact latent representation while **preserving information about $\mathbf{y}$.**
  - Information theoretic motivation:
    Find encoding distribution $p(\mathbf{z}|\mathbf{x})$ by minimizing
    $$I(\mathbf{x};\mathbf{z}) - \beta I(\mathbf{z};\mathbf{y})$$
    where $\beta \geq 0$ is some parameter controlling the trade-off between compression and predictive accuracy.
- Arbitrary $\mathbf{y}$, **discrete shared latent $\mathbf{z}^s$**
  $\rightsquigarrow$ **dependency-seeking clustering** (Klami and Kaski 2008): find clusters that "explain" the dependency between the two views.