Short course

A vademecum of statistical pattern recognition and machine learning

Dimensionality reduction

Massimo Piccardi
University of Technology, Sydney, Australia

© Massimo Piccardi, UTS

Agenda

- The curse of dimensionality
- Principal component analysis (PCA)
- Probabilistic PCA
- Factor analysis
- Mixtures of linear models
- Linear discriminant analysis (LDA)
- Example papers
- References

© Massimo Piccardi, UTS

The curse of dimensionality

- The “curse of dimensionality” (a colourful expression coined by Bellman in 1961) is the difficulty of learning effective models in spaces of high D dimensionality
- It is difficult to appreciate the vastness of high-dimensional spaces; an illuminating example is that of the ratio between the unit hypersphere and the unit hypercube tending to 0 with D growing
- The number of samples required to maintain the same “density” grows exponentially with D
- The size of parameters grows; for instance, the size of a full covariance matrix grows with the square of D

Remedies

- Use of prior knowledge where available
- Smoothness properties in the data allow for interpolation (at least, locally)
- Lower-dimensional manifolds can be extracted
- Two main approaches to dimensionality reduction:
  - *Feature selection*: select the most informative features in the feature set
  - *Feature extraction*: combine the features of the feature set to obtain an informative reduced set; we focus on this in the following

Feature extraction

- Many techniques are available for feature extraction, often categorised as *linear* and *non-linear*
- Main linear techniques:
  - Principal component analysis (PCA)
  - Probabilistic PCA
  - Factor analysis
  - Independent component analysis (ICA)
  - Linear discriminant analysis (LDA)
- Some non-linear techniques:
  - Kernel PCA
  - Autoassociative neural networks
- Locally linear models can be expressed as *mixtures* of linear techniques

Principal component analysis (PCA)

- Consider a D-variate continuous random variable, x and the following linear transformations:

$$z = W^T(x - \mu)$$
$$\tilde{x} = Wz + \mu$$

- We restrict z to be M-dimensional, with M ≤ D. The size of W is then D x M and that of μ is D x 1
- These transformations can be called *compression* and *reconstruction*. The *projection* is the combination of the two:

$$\tilde{x} = P(x) = WW^T(x - \mu) + \mu$$

Parameter estimation

- We are now given a training set x_i, i=1..N, to optimally determine parameters W, μ
- As objective function, we choose the sum of the *square reconstruction errors*, $\varepsilon_i = x_i - \tilde{x}_i$:

$$\sum_{i=1}^{N} \varepsilon_i^T \varepsilon_i = \sum_{i=1}^{N} (x_i - \tilde{x}_i)^T (x_i - \tilde{x}_i) =$$

$$= \sum_{i=1}^{N} \left(x_i - WW^T(x_i - \mu) + \mu\right)^T \left(x_i - WW^T(x_i - \mu) + \mu\right)$$

- The above can be called a *self-regression* target since it is analogous to linear regression between input and output variables

PCA parameters

It can be proven that:

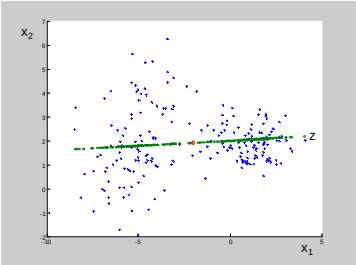
- $$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$
- The M columns of W are the first M *eigenvectors*, u_j, of the sample covariance matrix:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T$$

in descending eigenvalue order. W, μ so chosen are the parameters of PCA

- The above holds for any value of M ≤ D

Example



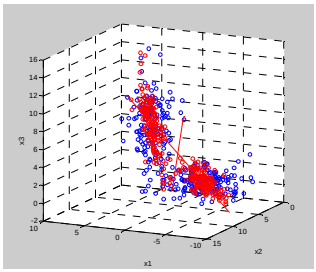
- D = 2, M = 1
- The blue points are the x_i
- The green points are simultaneously two things: the 1-dimensional z_i along the z sub-space; the 2-dimensional $\tilde{x}_i$ in x space
- The red dot is both x = μ and z = 0

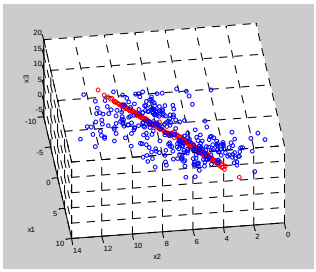
PCA properties

The main properties of PCA are:

- The eigenvectors are orthogonal unit vectors; therefore, matrix W is orthogonal (aka orthonormal) by columns, i.e. $\mathbf{W}^T \mathbf{W} = \mathbf{I}$ (an M x M identity matrix)
- The sample mean in z space, $\boldsymbol{\mu}_z$, **is 0** by construction
- The sample covariance matrix in z space, $\boldsymbol{\Sigma}_z$, **is diagonal** (the M dimensions are uncorrelated) and its elements are the M largest eigenvalues
- Out of all orthogonal D x M matrices, PCA's W maximises Σ_z, in the sense that the first dimension captures as much variance as possible in one dimension, the second as well in the remaining D-1 sub-space etc
- The eigenvalues of Σ are also the square of the *singular values* of the mean-subtracted data matrix, scaled by $(N-1)^{-1/2}$

Example





- D = 3, M = 2

$$W = \begin{bmatrix} -0.371 & 0.818 \\ 0.397 & -0.288 \\ 0.839 & 0.498 \end{bmatrix} \qquad \lambda = \begin{bmatrix} 2.899 & 0 & 0 \\ 0 & 4.389 & 0 \\ 0 & 0 & 28.20 \end{bmatrix}$$

$$\mu_z = \begin{bmatrix} \sim 0 \\ \sim 0 \end{bmatrix} \qquad rank(\Sigma_{\tilde{x}}) = 2$$

$$\Sigma_z = \begin{bmatrix} 4.389 & 0 \\ 0 & 28.20 \end{bmatrix}$$

$$W^T W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Why PCA?

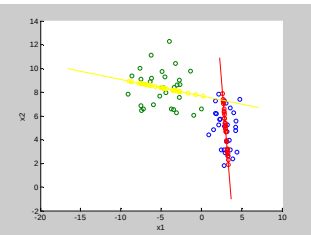
- PCA retains M linearly combined dimensions out of the initial D in the hope of retaining “information” and discarding “noise”
- By retaining maximum variance in the compressed space, PCA “minimises the loss of information”
- In the case of a classification problem, a typical use of PCA is that of learning the model from data from all classes (unlabelled data), and performing classification in the compressed space. Thanks to the dimensionality reduction, patterns may become more evident and classifier training, eased
- Typically, PCA is sensitive to the training set

Whitening

- It is worth mentionining *whitening*, a pre-processing technique directly related to PCA
- One may not want to reduce dimensions, rather decorrelate the data and scale them to a common scale
- Set M = D and simply compute $z_i' = \Sigma_z^{-1/2} z_i$; the z_i' have spherical, unit covariance, $\Sigma_{z'} = \mathbf{I}$
- Whitening the data is a stronger manipulation than simple *standardization* (i.e. for each dimension, subtract the mean and divide by the standard deviation) as it also decorrelates the dimensions

Labelled PCA

- The reconstruction error measures how much a point moves away from the PCA model
- In a classification problem, if labelled training data are provided, one may build one PCA model per class, and use the *minimum reconstruction error* to assign unseen data to classes



- However, there are two issues: a) the models are unbounded along the PCA sub-space; b) the reconstruction errors have the same weights for all classes

Probabilistic PCA

- Standard PCA lacks a precise statistical characterization
- **Probabilistic PCA** provides a statistical model for PCA, bounding it also along the PCA subspace and introducing a weight on the reconstruction error
- Probabilistic PCA was independently proposed by Tipping and Bishop and Sam Roweis (under the name of *sensible PCA*)

Probabilistic PCA

- We have defined the reconstruction error as $\varepsilon = x - \tilde{x}$. Therefore:

$$x = \tilde{x} + \varepsilon = Wz + \mu + \varepsilon$$

- We now assume that ε is a random variable distributed according to a spherical Gaussian distribution with zero mean, $\boldsymbol{\varepsilon \sim N(\varepsilon \mid 0, \sigma^2 I)}$
- The above assumption determines the following conditional distribution:

$$p(x \mid z) = N(x \mid Wz + \mu, \sigma^2 \mathrm{I})$$

Probabilistic PCA

- We now complete the model with an assumption over p(z):

$$p(z) = N(z \mid 0, \mathrm{I})$$

  It can be proven that this assumption is not limiting compared to a general mean and covariance

- The joint model, p(x,z), can then be simply obtained by Bayes' theorem:

$$p(x, z) = p(x \mid z)p(z) = N(x \mid Wz + \mu, \sigma^2 \mathrm{I})N(z \mid 0, \mathrm{I})$$

- The theory of *linear Gaussian models* (see Appendix) provides us with an immediate way to compute marginal **p(x)** and conditional **p(z|x)**

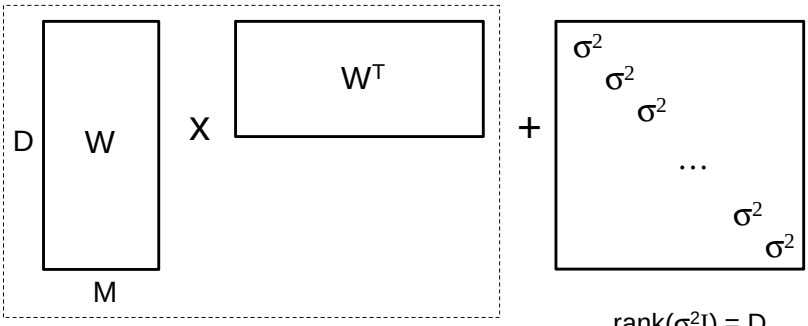
Probabilistic PCA

- We obtain:

$$p(x) = N(x \mid \mu, WW^T + \sigma^2 \mathrm{I})$$

- The above is the distribution of r.v. x, under the PPCA constraint. The mean, μ, is an ordinary D x 1 vector, but the D x D covariance is constrained to decompose as the sum of a rank-deficient square matrix, WW^T, (max rank: $M \leq D$), and a spherical covariance, $\sigma^2\mathrm{I}$
- The low-dimensional variable, z, does not need to be computed explicitly at any stage (it is marginalised)
- Parameters are μ, W and σ^2

PPCA covariance matrix

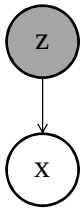


rank(WW^T) = M at most

rank($\sigma^2 I$) = D

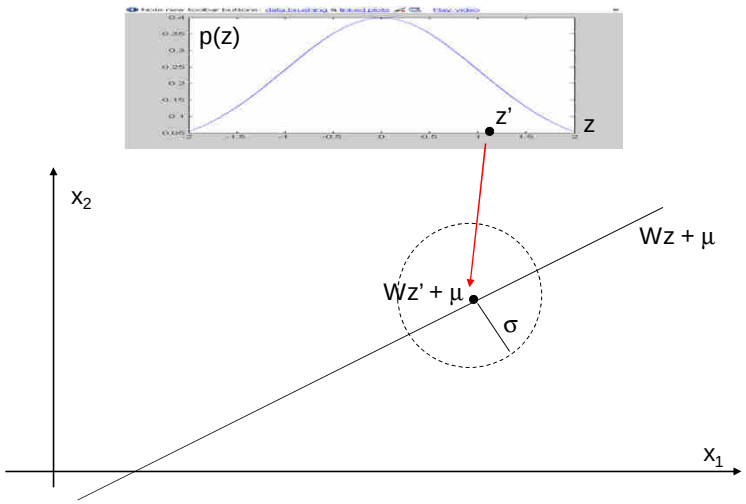
Generative model

- Joint probability p(x,z) = p(x|z)p(z) is a simple *Bayesian network* with x observed and z latent:



- It can generate samples $\{x_i,z_i\} \sim p(x,z)$ by first sampling $z_i \sim p(z)$ and then sampling $x_i \sim p(x|\ z_i)$ (*generative model*). Such samples are also samples of marginal p(x)

Generative model



ML solution

- Given a set of x_i, i=1..N, in D dimensions, we can define their log-likelihood in the usual way:

$$LL(\theta) = \sum_{i=1}^{N} \ln p(x_i \mid \theta) = \sum_{i=1}^{N} \ln N\left(x_i \mid \mu, WW^T + \sigma^2 \mathrm{I}\right)$$

- [Tipping and Bishop 99a] presents a closed-form solution for the maximum of the log-likelihood:

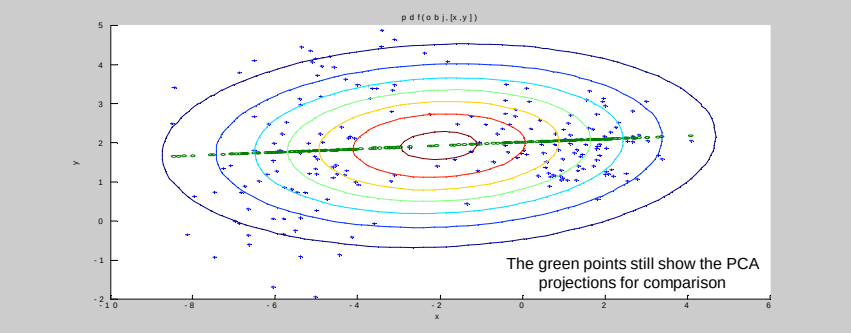
$$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \sigma^2_{ML} = \frac{1}{D-M}\sum_{k=M+1}^{D} \lambda_k$$

$$W_{ML} = U_M \left(L_M - \sigma^2_{ML} I\right)^{1/2}$$

ML solution

- U_M is a D x M matrix whose columns are the 'largest' eigenvectors of the data covariance matrix (like in PCA)
- L_M is a M x M diagonal matrix with the corresponding eigenvalues; thus, $W = U_M L_M^{1/2}$ gives the data their original scale (remember that $z \sim N(0,I)$)
- σ^2_{ML} is given by the average of the discarded eigenvalues, λ_k, k = M+1,…,D
- In addition, W_{ML} can be rotated by an arbitrary M x M rotation matrix, R, and remain an equivalent solution: $(WR)(WR)^T = WRR^TW^T = WW^T$

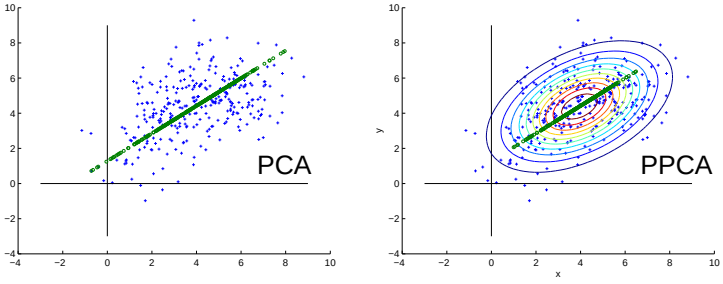
Example



- $\mu = [-2.0241\ 1.9208]^T$
- $\lambda_1 = 11.3778$; $\lambda_2 = 1.6852$
- $W = [-3.1105\ \ -0.1320]^T$; $\sigma^2 = 1.6852$
- $WW^T + \sigma^2 I = [11.3604\ \ 0.4104;\ \ 0.4104\ \ 1.7026]$

PCA vs PPCA

- Unlike PCA, PPCA normally does not compute the z_i explicitly – it doesn't need them
- However, it is possible to compute the z_i that corresponds to an x_i in the form of an expected value. Posing $M = W^TW + \sigma^2 I$, we infer $p(z|x) = N(z \mid \mathbf{M^{-1} W^T(x - \mu)}, \sigma^2 M^{-1})$ (note that $W^TW \neq I$)



© Massimo Piccardi, UTS

EM solution

- [Roweis 98] presents an EM algorithm for PPCA
- Despite being iterative, it may exhibit lower computational complexity (O(NDM) x N_iter) than PCA ($O(MD^2)$) in high dimensions (M << D)
- Log-likelihood surface is convex, should converge to a global maximum irrespectively of initialisation
- Can also deal with missing data, if any
- For $\sigma^2 \rightarrow 0$, it provides an EM algorithm for standard PCA
- It provides a solution scheme also for other models such as factor analysis and robust PCA

© Massimo Piccardi, UTS

EM solution

- The EM approach for PPCA uses z as the latent variable
- We need to build densities ln p(X,Z|θ) and p(Z|X,θ). For the former, we have:

$$\ln p(X,Z\mid\theta)=\sum_{i=1}^{N}\ln p(x_i,z_i\mid\theta)=\sum_{i=1}^{N}\ln\left[N(x_i\mid Wz_i+\mu,\sigma^2\mathrm{I})N(z_i\mid 0,\mathrm{I})\right]$$

- When multiplied by p(Z|X,θ) and integrated over Z, the terms in z_i and $z_i z_i^T$ just become expected values:

$$\int_Z z_i\ p(Z\mid X,\theta)\,dZ=\int_Z z_i\ p(z_1\ldots z_i\ldots z_N\mid X,\theta)\,dz_1\ldots dz_i\ldots dz_N=$$
$$=\int_{z_i} z_i\ p(z_i\mid X,\theta)\,dz_i=E[z_i\mid X]$$

EM solution

- E step:

$$E[z_i\mid X]=M^{-1}W^T(x_i-\mu)\quad E[z_i z_i^T\mid X]=\sigma^2M^{-1}+E[z_i\mid X]E[z_i\mid X]^T$$

- The M step returns new estimates for μ, W and σ^2. However, for linear-Gaussian models it is common to use μ_{ML} rather than μ from Q and compute only W and σ^2:

$$W_{new}=\sum_{i=1}^{N}(x_i-\mu_{ML})E[z_i\mid X]^T\left[\sum_{i=1}^{N}E[z_i z_i^T\mid X]\right]^{-1}$$

$$\sigma^2_{new}=\frac{1}{ND}\sum_{i=1}^{N}\left[(x_i-\mu_{ML})^T(x_i-\mu_{ML})-2E[z_i\mid X]^TW_{new}^T(x_i-\mu_{ML})\right.$$
$$\left.+tr\left(E[z_i z_i^T\mid X]W_{new}^TW_{new}\right)\right]$$

[Tipping and Bishop 99a, Appendix B]

Factor analysis

- Factor analysis uses the same model of PCA and PPCA:

$$x = Wz + \mu + \varepsilon$$

- The only difference is that ε, the reconstruction error, is ~ N(0,ψ), with ψ **a diagonal matrix**
- No closed-form solutions; [Rubin and Thayer 82] provides an EM solution
- Unlike PCA and PPCA, the FA solution for M + 1 dimensions is not an increment of that for M; controversial interpretation

EM solution for FA

- E step:

$$G = \mathrm{I} + W^T \Psi^{-1} W$$

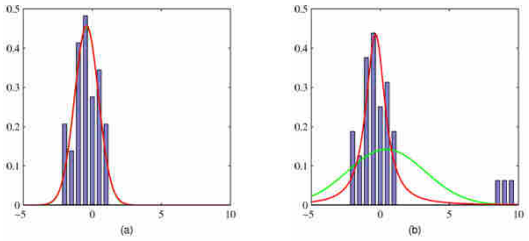
$$E[z_i \mid X] = G^{-1} W^T \Psi^{-1} (x_i - \mu) \quad E[z_i z_i^T \mid X] = G^{-1} + E[z_i \mid X] E[z_i \mid X]^T$$

- M step: W_{new} is as in PPCA; ψ_{new} is:

$$\Psi_{new} = diag\left(\frac{1}{N} \sum_{i=1}^{N} \left[(x_i - \mu_{ML})^T (x_i - \mu_{ML}) - 2 W_{new} E[z_i \mid X] (x_i - \mu_{ML})^T + W_{new} E[z_i z_i^T \mid X] W_{new}^T \right] \right)$$

Robust PPCA

- Various PCA variants have been proposed to mollify the effects of outliers during parameter estimation
- **Robust PPCA** is analogous to PPCA, but uses the heavy-tailed Student's *t* distribution in place of the Gaussians



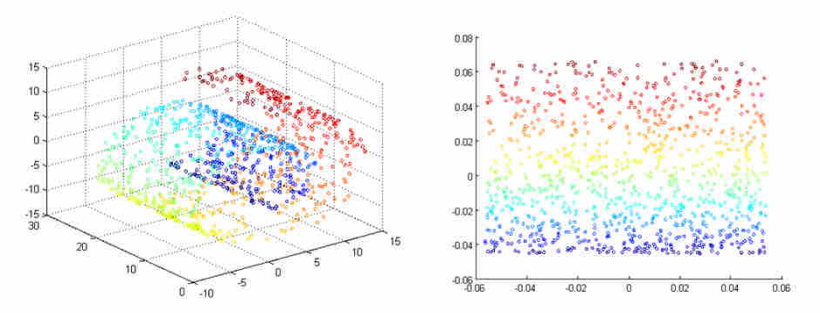
courtesy of C. Bishop, PRML, 2006

- [Liu and Rubin 1995]: basic EM for the Student's *t*
- Various references: [Svensen and Bishop 2004], [de Ridder and Franc 2003], [Khan and Dellaert 2004], [Archambeau et al. 2008]

Non-linear manifolds

- All the techniques presented so far are linear in the latent variable, z, and can be represented by a single transformation matrix, W
- Any new sample can be mapped to the latent space through the transformation W
- Other techniques are non linear, x = f(z), and the dimensionality reduction (*embedding*) of a sample set cannot be represented by a transformation matrix
- To name a famous few: multidimensional scaling, kernel PCA, local linear embedding, ISOMAP, Laplacian eigenmaps
- Out-of-sample extensions: [Yoshua Bengio et al. 2004]
- An intermediate way to represent non-linear embeddings is as a mixture of locally-linear sub-spaces

Example

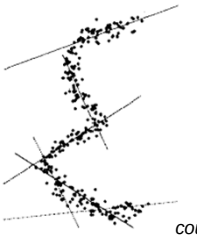


- Unrolling the Swiss Roll with ISOMAP

courtesy of (or unashamedly stolen from) Nik Melchior,
http://www.cs.cmu.edu/~efros/courses/AP06/presentations/melchior_isomap_demo.pdf

Mixture models

- Mixture models are a powerful tool to extend dimensionality reduction to non-linear manifolds which are locally linear



courtesy of Hinton, Dayan, Revow [3]

- Extensions to the previous models are known as mixture of PCA [3], mixture of probabilistic principal component analysers [4], mixture of factor analyzers [5], mixture of robust probabilistic principal component analysers [6]

Mixture models

- Mixture models extend single-component re-estimation formulas through the *responsibility*, $p(l \mid x_i, \theta)$, the fractional membership of sample x_i to component l. Extension is usually straightforward
- The re-estimation formula for the α_l remains the same as for any other mixture
- Formulas such as $\mu = \Sigma x_i / \Sigma 1$, $\Sigma = \Sigma(x_i - \mu)(x_i - \mu)^T / \Sigma 1$ and others, naturally extend to responsibility-weighted samples. For MPPCA, eigenvalue decomposition still applies; yet more efficient algorithms are also possible ([Tipping and Bishop 99b])

Discriminative dimensionality reduction

- So far, we have presented:
  - PCA, as an *unlabelled* technique which compresses all the data in the same way, irrespectively of possible class labels
  - Labelled PCA and various probabilistic extensions (PPCA, FA etc) which can reasonably be used for classification problems. Labelled data are required for training and one model is built for each class from samples of that class only
- It could then be possible to improve the model's estimation by using labelled data from all classes *jointly*: discriminative techniques
- Linear discriminant analysis, supervised PPCA etc

Linear discriminant analysis

- The objective of linear discriminant analysis (LDA) is to perform linear dimensionality reduction while preserving as much class discrimination as possible
- To illustrate LDA, let us consider a problem with only two classes, C_1 and C_2, M = 1, and define these auxiliary quantities:
  - the samples' means, $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$, and covariances, $\mathbf{S_1}$ and $\mathbf{S_2}$, for each class
  - the projected data's means, $\mathbf{m_1}$ and $\mathbf{m_2}$, and variances, $\mathbf{s_1^2}$ and $\mathbf{s_2^2}$, for each class
  - $\mathbf{S_w} = S_1 + S_2$, the **within-class** scatter matrix
  - $\mathbf{S_B} = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$, the **between-class** scatter matrix

LDA: quantities

$$\mu_1 = \frac{1}{N_1}\sum_{x\in C_1} x \quad \mu_2 = \frac{1}{N_2}\sum_{x\in C_2} x$$

$$S_1 = \frac{1}{N_1}\sum_{x\in C_1}(x-\mu_1)(x-\mu_1)^T \quad S_2 = \frac{1}{N_2}\sum_{x\in C_2}(x-\mu_2)(x-\mu_2)^T$$

$$m_1 = \frac{1}{N_1}\sum_{z\in C_1} z = W^T\mu_1 \quad m_2 = \frac{1}{N_2}\sum_{z\in C_2} z = W^T\mu_2$$

$$s_1^2 = \frac{1}{N_1}\sum_{z\in C_1}(z-m_1)^2 = W^T S_1 W \quad s_2^2 = \frac{1}{N_2}\sum_{z\in C_2}(z-m_2)^2 = W^T S_2 W$$

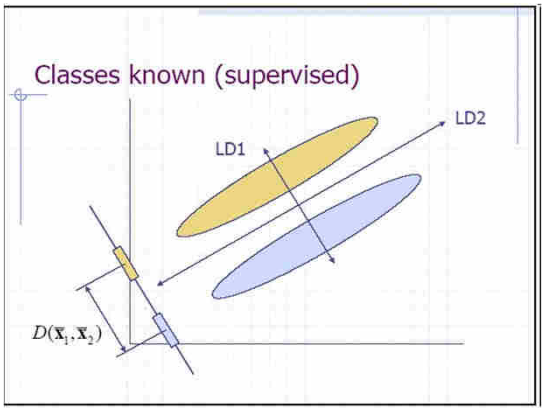
LDA: criterion function

- The criterion function of LDA is:

$$J(W) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} = \frac{W^T S_B W}{W^T S_W W}$$

- J(W) is the ratio between the between-class and within class scatter matrices of the projected data; it is expressed as a function of W and sample scatter matrices S_B, S_W
- Our aim is to find W that maximises it

LDA: example



courtesy of Randy Julian

LDA: solution

- The maximum of the LDA criterion function is given by:

$$W^* = \arg\max_W \{J(W)\} = S_W^{-1}(\mu_1 - \mu_2)$$

- The solution is known as the *Fisher's linear discriminant*; it is, actually, a direction

LDA: C classes

- Criterion function:

$$J(W) = \frac{\det\left(W^T S_B W\right)}{\det\left(W^T S_W W\right)}$$

where S_B, S_W are extended as:

$$S_W = \sum_{j=1}^{C} S_{W_j} \quad S_B = \sum_{j=1}^{C} N_j (\mu_j - \mu)(\mu_j - \mu)^T$$ (μ: overall mean)

- Solution: "largest" eigenvectors of $S_W^{-1}S_B$

LDA: limitations

- M is C-1 at maximum
- Underlying Gaussian model with equal covariance for each class
- Linear model
- Variations of LDA address the above limitations:
  - *Orthonormal LDA* projects to more than C-1 dimensions
  - *Non-parametric LDA* removes the Gaussian assumption
  - The *multilayer perceptron* removes the linearity
  - *Generalized LDA* introduces a Bayes Risk cost function

Example papers

- Application: *face recognition*

  Recognising faces in images
- PCA:

  Turk, M.A.; Pentland, A.P.; Face recognition using eigenfaces, Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on 3-6 June 1991, Page(s):586 – 591
- LDA:

  Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J.; Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 19, Issue 7, July 1997 Page(s):711 - 720

References

1. M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 61, no. 3, pp. 611–622, 1999
2. S. Roweis, “EM algorithms for PCA and SPCA,” in Advances in neural information processing systems, vol. 10. Colorado, United States: The MIT Press, pp. 626-632, 1998
3. G. Hinton, P. Dayan, M. Revow, “Modeling the Manifolds of Images of Handwritten Digits”, IEEE Trans. on Neural Networks, vol. 8, n. 1, pp. 65-74, 1997
4. M. E. Tipping and C. M. Bishop, “Mixtures of probabilistic principal component analyzers,” Neural Computation, 11(2): 443-482, 1999
5. Z. Ghahramani and G. Hinton, “The EM algorithm for mixtures of factor analyzers,” University of Toronto, Tech. Rep. CRG-TR-96-1, 1997
6. C. Archambeau, N. Delannay, M. Verleysen, “Mixtures of robust probabilistic principal component analyzers,” Neurocomputing 71(7-9): 1274-1282 (2008)

Appendix – partitioned Gaussians

- Given a multivariate Gaussian distribution, let’s group its dimensions in two groups of arbitrary size, called x and y for convenience
- Let us use the following notations:

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \quad \Lambda = \Sigma^{-1} = \begin{bmatrix} \Lambda_{xx} & \Lambda_{xy} \\ \Lambda_{yx} & \Lambda_{yy} \end{bmatrix}$$

- We are given the **joint probability**, p(x,y) = N(x,y | μ,Σ). We can immediately derive **both marginals and conditionals**:

$p(x) = N(x \mid \mu_x, \Sigma_{xx})$ expressed more easily in terms of Σ; p(y) is analogous

$p(x \mid y) = N\left(x \mid \mu_x - \Lambda_{xx}^{-1}\Lambda_{xy}(y - \mu_y), \Lambda_{xx}^{-1}\right)$ expressed more easily in terms of Λ; p(y|x) is analogous

Appendix – linear Gaussian models

- In *linear Gaussian models*, we are instead given one conditional and a corresponding marginal, and we can immediately derive **the joint probability, the other marginal and the other conditional**:
- Let us use the same notations as Bishop. We have:

$$p(y \mid x) = N\left(y \mid Ax + b, L^{-1}\right) \qquad p(x) = N\left(x \mid \mu, \Lambda^{-1}\right)$$

- We derive:

$$p(y,x) = N\left(y,x \mid M, R^{-1}\right), \quad M = \begin{bmatrix} \mu \\ A\mu + b \end{bmatrix}, \quad R^{-1} = \begin{bmatrix} \Lambda^{-1} & \Lambda^{-1}A^T \\ A\Lambda^{-1} & L^{-1} + A\Lambda^{-1}A^T \end{bmatrix}$$

$$p(y) = N\left(y \mid A\mu + b, L^{-1} + A\Lambda^{-1}A^T\right)$$

$$p(x \mid y) = N\left(x \mid \left(\Lambda + A^T LA\right)^{-1} A^T L(y - b) + \Lambda\mu, \left(\Lambda + A^T LA\right)^{-1}\right)$$