# UNSUPERVISED LEARNING 2011

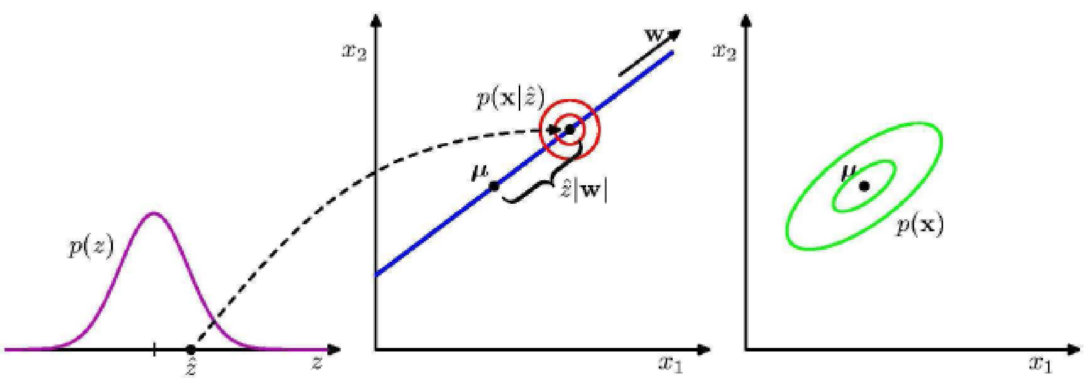
# LECTURE :PPCA

Slides due to Geoffrey Hinton

# Probabilistic PCA

- Probabilistic, generative view of data
- Assumptions:
  - underlying latent variable has a Gaussian distribution
  - linear relationship between latent and observed variables
  - isotropic Gaussian noise in observed dimensions

$$
\begin{aligned}
p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) \\
p(\mathbf{x}|\mathbf{z}) &= \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \mu, \sigma^2\mathbf{I})
\end{aligned}
$$




$$
\mathbf{x} = \mathbf{W}\mathbf{z} + \mu + \epsilon
$$

# Probabilistic PCA: Marginal data density

- Columns of **W** are the *principal components*, $\sigma^2$ is *sensor noise*
- Product of Gaussians is Gaussian: the joint p(**z**,**x**), the marginal data distribution p(**x**) and the posterior p(**z**|**x**) are also Gaussian
- Marginal data density (predictive distribution):

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z})d\mathbf{z} = \mathcal{N}(\mathbf{x}|\mu, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})$$

- Can derive by completing square in exponent, or by just computing mean and covariance given that it is Gaussian:

$$\begin{aligned} E[\mathrm{x}] &= E[\mu + \mathbf{W}\mathbf{z} + \epsilon] = \mu + \mathbf{W}E[\mathbf{z}] + E[\epsilon] \\ &= \mu + \mathbf{W}0 + 0 = \mu \\ \mathbf{C} &= Cov[\mathrm{x}] = E[(\mathbf{z} - \mu)(\mathbf{z} - \mu)^T] \\ &= E[(\mu + \mathbf{W}\mathbf{z} + \epsilon - \mu)(\mu + \mathbf{W}\mathbf{z} + \epsilon - \mu)^T] \\ &= E[(\mathbf{W}\mathbf{z} + \epsilon)(\mathbf{W}\mathbf{z} + \epsilon)^T] \\ &= \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I} \end{aligned}$$

# Probabilistic PCA: Joint distribution

- Joint density for PPCA (**x** is D-dim., **z** is M-dim):

$$p(\begin{bmatrix}\mathbf{z}\\ \mathbf{x}\end{bmatrix}) = \mathcal{N}(\begin{bmatrix}\mathbf{z}\\ \mathbf{x}\end{bmatrix} \mid \begin{bmatrix}0\\ \mu\end{bmatrix}, \begin{bmatrix} I & \mathbf{W}^\top \\ \mathbf{W} & \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\end{bmatrix})$$

  – where cross-covariance terms from:

$$\begin{aligned} Cov[\mathbf{z},\mathbf{x}] &= E[(\mathbf{z}-0)(\mathbf{x}-\mu)^T] = E[\mathbf{z}(\mu + \mathbf{W}\mathbf{z} + \epsilon - \mu)^T] \\ &= E[\mathbf{z}(\mathbf{W}\mathbf{z} + \epsilon)^T] = \mathbf{W}^T \end{aligned}$$

- Note that evaluating predictive distribution involves inverting **C**: reduce O($D^3$) to O($M^3$) by applying *matrix inversion lemma*:

$$\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-2}\mathbf{W}(\mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I})^{-1}\mathbf{W}^T$$

# Probabilistic PCA: Posterior distribution

- Inference in PPCA produces posterior distribution over latent **z**
- Derive by applying Gaussian conditioning formulas (see 2.3 in book) to joint distribution

$$p\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \mid \begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

$$p(\mathbf{x}_1) = \mathcal{N}(\mu_1, \Sigma_{11})$$

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1|\mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2)$$

$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{m}, \mathbf{V})$$

$$\mathbf{m} = \mathbf{W}^T(\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})^{-1}(\mathbf{x} - \mu)$$

$$\mathbf{V} = \mathbf{I} - \mathbf{W}^T(\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})^{-1}\mathbf{W}$$

- Mean of inferred **z** is projection of centered **x** – linear operation
- Posterior variance does not depend on the input **x** at all!

# Standard PCA: Zero-noise limit of PPCA

- Can derive standard PCA as limit of Probabilistic PCA (PPCA) as $\sigma^2 \rightarrow 0$.
- ML parameters $\mathbf{W}^*$ are the same
- Inference is easier: orthogonal projection

$$\lim_{\sigma^2 \rightarrow 0} \mathbf{W}^T(\mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{W}^T)^{-1} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T$$

- Posterior covariance is zero

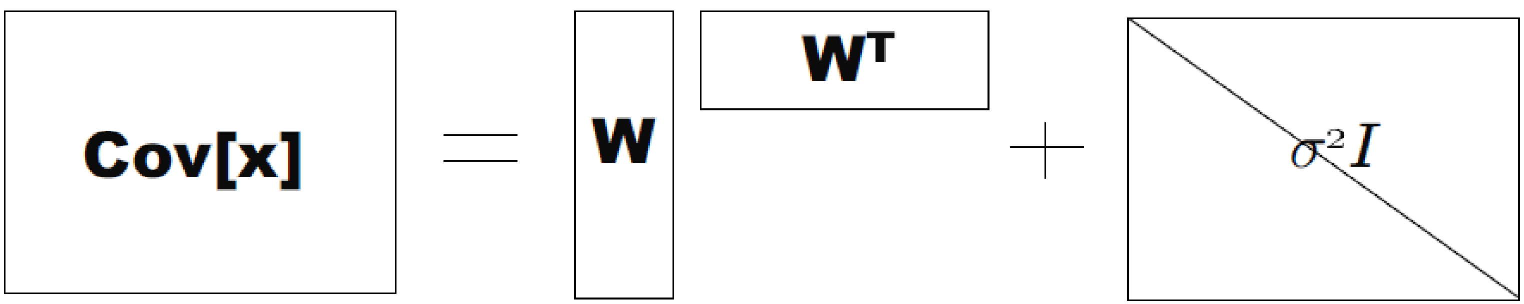
# Probabilistic PCA: Constrained covariance

- Marginal density for PPCA (**x** is D-dim., **z** is M-dim):

$$p(\mathbf{x}|\theta) = \mathcal{N}(\mathbf{x}|\mu, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})$$

  – where $\theta = \mathbf{W}, \mu, \sigma$

- Effective covariance is low-rank outer product of two long skinny matrices plus a constant diagonal matrix




- So PPCA is just a constrained Gaussian model:
  – Standard Gaussian has D + D(D+1)/2 effective parameters
  – Diagonal-covariance Gaussian has D+D, but cannot capture correlations
  – PPCA: DM + 1 – M(M-1)/2, can represent M most significant correlations

# Probabilistic PCA: EM

- Rather than solving directly, can apply EM
- Need complete-data log likelihood

$$\log p(\mathbf{X}, \mathbf{Z} | \mu, \mathbf{W}, \sigma^2) = \sum_n [\log p(\mathbf{x}_n | \mathbf{z}_n) + \log p(\mathbf{z}_n)]$$

- E step: compute expectation of complete log likelihood with respect to posterior of latent variables z, using current parameters – can derive $E[\mathbf{z}_n]$ and $E[\mathbf{z}_n \mathbf{z}_n^T]$ from posterior $p(\mathbf{z}|\mathbf{x})$
- M step: maximize with respect to parameters W and $\sigma^2$
- Iterative solution, updating parameters given current expectations, expectations give current parameters
- Nice property – avoids direct $O(ND^2)$ construction of covariance matrix, instead involves sums over data cases: $O(NDM)$; can be implemented online, without storing data

# Probabilistic PCA: Why bother?

- Seems like a lot of formulas, algebra to get to similar model to standard PCA, but…
- Leads to understanding of underlying data model, assumptions (e.g., vs. standard Gaussian, other constrained forms)
- Derive EM version of inference/learning: more efficient
- Can understand other models as generalizations, modifications
- More readily extend to mixtures of PPCA models
- Principled method of handling missing values in data
- Can generate samples from data distribution