CSC412 – Probabilistic Learning & Reasoning Sam Roweis

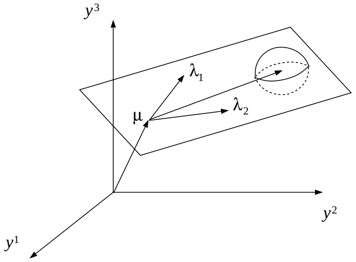
Lecture 10:

Factor Analysis and Principal Component Analysis

February 13, 2006

Continuous Latent Variables

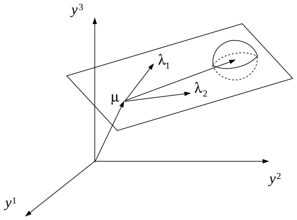
- In many models there are some *underlying causes* of the data.
- Mixture models use a discrete class variable: clustering.
- Sometimes, it is more appropriate to think in terms of continuous *factors* which control the data we observe. Geometrically, this is equivalent to thinking of a data *manifold* or subspace.



- To generate data, first generate a point within the manifold then add noise. Coordinates of point are components of latent variable.

Factor Analysis

- When we assume that the subspace is *linear* and that the underlying latent variable has a Gaussian distribution we get a model known as *factor analysis*:
  — data $\mathbf{y}$ (p-dim);
  — latent variable $\mathbf{x}$ (k-dim)



$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|0, I)$$
$$p(\mathbf{y}|\mathbf{x}, \theta) = \mathcal{N}(\mathbf{y}|\mu + \Lambda\mathbf{x}, \Psi)$$

where μ is the mean vector, Λ is the p by k *factor loading matrix*, and Ψ is the *sensor noise covariance* (ususally diagonal).

- Important: since the product of Gaussians is still Gaussian, the joint distribution $p(\mathbf{x}, \mathbf{y})$, the other marginal $p(\mathbf{y})$ and the conditional $p(\mathbf{x}|\mathbf{y})$ are also Gaussian.

Marginal Data Distribution

- Just as with discrete latent variables, we can compute the marginal density $p(\mathbf{y}|\theta)$ by summing out $\mathbf{x}$. But now the sum is an integral:

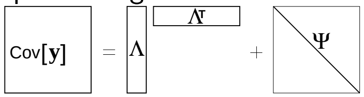
$$p(\mathbf{y}|\theta) = \int_{\mathbf{x}} p(\mathbf{x})p(\mathbf{y}|\mathbf{x}, \theta)d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mu \, , \, \Lambda\Lambda^\top + \Psi)$$

  which can be done by completing the square in the exponent.
- However, since the marginal is Gaussian, we can also just compute its mean and covariance. (Assume noise uncorrelated with data.)

$$\begin{aligned} E[\mathbf{y}] &= E[\mu + \Lambda\mathbf{x} + \text{noise}] = \mu + \Lambda E[\mathbf{x}] + E[\text{noise}] \\ &= \mu + \Lambda \cdot 0 + 0 = \mu \\ \text{Cov}[\mathbf{y}] &= E[(\mathbf{y} - \mu)(\mathbf{y} - \mu)^\top] \\ &= E[(\mu + \Lambda\mathbf{x} + \text{noise} - \mu)(\mu + \Lambda\mathbf{x} + \text{noise} - \mu)^\top] \\ &= E[(\Lambda\mathbf{x} + n)(\Lambda\mathbf{x} + n)^\top] = \Lambda E(\mathbf{x}\mathbf{x}^\top)\Lambda^\top + E(nn^\top) \\ &= \Lambda\Lambda^\top + \Psi \end{aligned}$$

FA = Constrained Covariance Gaussian

- Marginal density for factor analysis ($\mathbf{y}$ is p-dim, $\mathbf{x}$ is k-dim):
$$p(\mathbf{y}|\theta) = \mathcal{N}(\mathbf{y}|\mu\,,\,\Lambda\Lambda^\top + \Psi)$$
- So the effective covariance is the low-rank outer product of two long skinny matrices plus a diagonal matrix:



- In other words, factor analysis is just a constrained Gaussian model. (If Ψ were not diagonal then we could model any Gaussian and it would be pointless.)
- Learning: how should we fit the ML parameters?
- It is easy to find μ: just take the mean of the data.
  From now on assume we have done this and re-centred $\mathbf{y}$.
- What about the other parameters?

Likelihood Function

- Since the FA data model is Gaussian, likelihood function is simple:
$$\begin{aligned}\ell(\theta;\mathcal{D}) &= -\frac{N}{2}\log|\Lambda\Lambda^\top + \Psi| - \frac{1}{2}\sum_n(\mathbf{y}^n - \mu)^\top(\Lambda\Lambda^\top + \Psi)^{-1}(\mathbf{y}^n - \mu)\\ &= -\frac{N}{2}\log|\mathbf{V}| - \frac{1}{2}\mathrm{trace}\left[\mathbf{V}^{-1}\sum_n(\mathbf{y}^n - \mu)(\mathbf{y}^n - \mu)^\top\right]\\ &= -\frac{N}{2}\log|\mathbf{V}| - \frac{1}{2}\mathrm{trace}\left[\mathbf{V}^{-1}\mathbf{S}\right]\end{aligned}$$
  $\mathbf{V}$ is model covariance; $\mathbf{S}$ is sample data covariance.
- In other words, we are trying to make the constrained model covariance as close as possible to the observed covariance, where "close" means the trace of the ratio.
- Thus, the sufficient statistics are the same as for the Gaussian: mean $\sum_n \mathbf{y}^n$ and covariance $\sum_n(\mathbf{y}^n - \mu)(\mathbf{y}^n - \mu)^\top$.

EM for Factor Analysis

- We will do maximum likelihood learning using (surprise, surprise) the EM algorithm.
  E-step: $q_n^{t+1} = p(\mathbf{x}^n|\mathbf{y}^n, \theta^t)$
  M-step: $\theta^{t+1} = \mathrm{argmax}_\theta \sum_n \int_{\mathbf{x}} q^{t+1}(\mathbf{x}^n|\mathbf{y}^n)\log p(\mathbf{y}^n, \mathbf{x}^n|\theta)d\mathbf{x}^n$
- For E-step we need the conditional distribution (inference)
  For M-step we need the expected log of the complete data.
$$\begin{aligned}\mathbf{E-step}: q_n^{t+1} &= p(\mathbf{x}^n|\mathbf{y}^n, \theta^t) = \mathcal{N}(\mathbf{x}^n|\mathbf{m}^n, \mathbf{V}^n)\\ \mathbf{M-step}: \Lambda^{t+1} &= \mathrm{argmax}_\Lambda \sum_n \langle \ell_c(\mathbf{x}^n, \mathbf{y}^n)\rangle_{q_n^{t+1}}\\ \Psi^{t+1} &= \mathrm{argmax}_\Psi \sum_n \langle \ell_c(\mathbf{x}^n, \mathbf{y}^n)\rangle_{q_n^{t+1}}\end{aligned}$$

From Joint Distribution to Conditional

- To get the conditional $p(\mathbf{x}|\mathbf{y})$ we will start with the joint $p(\mathbf{x},\mathbf{y})$ and apply Bayes rule for Gaussian conditionals.
- Write down the joint distribtion of $\mathbf{x}$ and $\mathbf{y}$:
$$p(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}) = \mathcal{N}(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \mid \begin{bmatrix}0\\ \mu\end{bmatrix}, \begin{bmatrix}I & \Lambda^\top\\ \Lambda & \Lambda\Lambda^\top + \Psi\end{bmatrix})$$
  where the corner elements $\Lambda^\top, \Lambda$ come from $\mathrm{Cov}[\mathbf{x},\mathbf{y}]$:
$$\begin{aligned}\mathrm{Cov}[\mathbf{x},\mathbf{y}] &= E[(\mathbf{x} - 0)(\mathbf{y} - \mu)^\top] = E[\mathbf{x}(\mu + \Lambda\mathbf{x} + noise - \mu)^\top]\\ &= E[\mathbf{x}(\Lambda\mathbf{x} + noise)^\top] = \Lambda^\top\end{aligned}$$
- Assume noise is uncorrelated with data or latent variables.

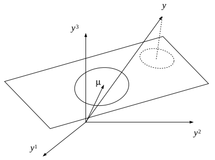
E-step: Inference in Factor Analysis

- Apply the Gaussian conditioning formulas to the joint distribution we derived above. This gives:

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\mathbf{m},\mathbf{V}) \\ \mathbf{V} &= I - \Lambda^\top(\Lambda\Lambda^\top + \Psi)^{-1}\Lambda \\ \mathbf{m} &= \Lambda^\top(\Lambda\Lambda^\top + \Psi)^{-1}(\mathbf{y} - \mu) \end{aligned}$$

- Now apply the matrix inversion lemma to get:

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\mathbf{m},\mathbf{V}) \\ \mathbf{V} &= (I + \Lambda^\top\Psi^{-1}\Lambda)^{-1} \\ \mathbf{m} &= \mathbf{V}\Lambda^\top\Psi^{-1}(\mathbf{y} - \mu) \end{aligned}$$



Inference is Linear

- Note: inference just multiplies $\mathbf{y}$ by a matrix:

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\mathbf{m},\mathbf{V}) \\ \mathbf{V} &= I - \Lambda^\top(\Lambda\Lambda^\top + \Psi)^{-1}\Lambda \\ &= (I + \Lambda^\top\Psi^{-1}\Lambda)^{-1} \\ \mathbf{m} &= \Lambda^\top(\Lambda\Lambda^\top + \Psi)^{-1}(\mathbf{y} - \mu) \\ &= \mathbf{V}\Lambda^\top\Psi^{-1}(\mathbf{y} - \mu) \end{aligned}$$

- Note: inference of the posterior mean is just a linear operation!

$$\mathbf{m} = \beta(\mathbf{y} - \mu)$$

  where β can be computed beforehand given the model parameters.
- Also: posterior covariance does not depend on observed data!

$$\mathrm{cov}[\mathbf{x}|\mathbf{y}] = \mathbf{V} = (I + \Lambda^\top\Psi^{-1}\Lambda)^{-1}$$

Complete Data Likelihood

- We know the optimal μ is the data mean.
  Assume the mean has been subtracted off $\mathbf{y}$ from now on.
- The complete likelihood (ignoring mean):

$$\begin{aligned} \ell_c(\Lambda,\Psi) &= \sum_n \log p(\mathbf{x}^n, \mathbf{y}^n) \\ &= \sum_n \log p(\mathbf{x}^n) + \log p(\mathbf{y}^n|\mathbf{x}^n) \\ &= -\frac{N}{2}\log|\Psi| - \frac{1}{2}\sum_n \mathbf{x}^\top\mathbf{x} - \frac{1}{2}\sum_n(\mathbf{y}^n - \Lambda\mathbf{x}^n)^\top\Psi^{-1}(\mathbf{y}^n - \Lambda\mathbf{x}^n) \\ \ell_c(\Lambda,\Psi) &= -\frac{N}{2}\log|\Psi| - \frac{N}{2}\mathrm{trace}[S\Psi^{-1}] \\ S &= \frac{1}{N}\sum_n(\mathbf{y}^n - \Lambda\mathbf{x}^n)(\mathbf{y}^n - \Lambda\mathbf{x}^n)^\top \end{aligned}$$

M-step: Optimize Parameters

- Take the derivates of the complete log likelihood wrt. parameters:

$$\begin{aligned} \partial\ell_c(\Lambda,\Psi)/\partial\Lambda &= -\Psi^{-1}\sum_n \mathbf{y}_n\mathbf{x}_n^\top + \Psi^{-1}\Lambda\sum_n \mathbf{x}_n\mathbf{x}_n^\top \\ \partial\ell_c(\Lambda,\Psi)/\partial\Psi^{-1} &= +(N/2)\Psi - (N/2)S \end{aligned}$$

- Take the expectation with respect to q^t from E-step:

$$\begin{aligned} < \ell'_\Lambda > &= -\Psi^{-1}\sum_n \mathbf{y}_n\mathbf{m}_n^\top + \Psi^{-1}\Lambda\sum_n \mathbf{V}_n \\ < \ell'_{\Psi^{-1}} > &= +(N/2)\Psi - (N/2) < S > \end{aligned}$$

- Finally, set the derivatives to zero to solve for optimal parameters:

$$\begin{aligned} \Lambda^{t+1} &= \left(\sum_n \mathbf{y}^n\mathbf{m}^{n\top}\right)\left(\sum_n \mathbf{V}^n\right)^{-1} \\ \Psi^{t+1} &= \frac{1}{N}\mathrm{diag}\left[\sum_n \mathbf{y}^n\mathbf{y}^{n\top} + \Lambda^{t+1}\sum_n \mathbf{m}^n\mathbf{y}^{n\top}\right] \end{aligned}$$

Final Algorithm: EM for Factor Analysis

- First, set μ equal to the sample mean $(1/N)\sum_n \mathbf{y}_n$, and subtract this mean from all the data.
- Now run the following iterations:

$$\begin{aligned}
\mathbf{E-step}: q^{t+1} &= p(\mathbf{x}|\mathbf{y},\theta^t) = \mathcal{N}(\mathbf{x}^n|\mathbf{m}^n,\mathbf{V}^n)\\
\mathbf{V}^n &= (I+\Lambda^\top\Psi^{-1}\Lambda)^{-1}\\
\mathbf{m}^n &= \mathbf{V}^n\Lambda^\top\Psi^{-1}(\mathbf{y}-\mu)\\
\mathbf{M-step}: \Lambda^{t+1} &= \left(\sum_n \mathbf{y}^n\mathbf{m}^{n\top}\right)\left(\sum_n \mathbf{V}^n\right)^{-1}\\
\Psi^{t+1} &= \frac{1}{N}\mathrm{diag}\left[\sum_n \mathbf{y}^n\mathbf{y}^{n\top} + \Lambda^{t+1}\sum_n \mathbf{m}^n\mathbf{y}^{n\top}\right]
\end{aligned}$$

Principal Component Analysis

- In Factor Analysis, we can write the marginal density explicitly:

$$p(\mathbf{y}|\theta) = \int_{\mathbf{x}} p(\mathbf{x})p(\mathbf{y}|\mathbf{x},\theta)d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mu\ ,\ \Lambda\Lambda^\top + \Psi)$$

- Noise Ψ mut be restricted for model to be interesting. (Why?)
- In Factor Analysis the restriction is that Ψ is *diagonal* (axis-aligned).
- What if we further restrict $\Psi = \sigma^2 I$ (ie *spherical*)?
- We get the Probabilistic Principal Component Analysis (PPCA) model:

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

  where μ is the mean vector,
  columns of Λ are the *principal components* (usually orthogonal),
  and σ^2 is the *global sensor noise*.

Likelihood Function

- As with FA, the PPCA data model is Gaussian.
  Thus, the likelihood function is simple:

$$\begin{aligned}
\ell(\theta;\mathcal{D}) &= -\frac{N}{2}\log|\Lambda\Lambda^\top + \Psi| - \frac{1}{2}\sum_n(\mathbf{y}^n-\mu)^\top(\Lambda\Lambda^\top+\Psi)^{-1}(\mathbf{y}^n-\mu)\\
&= -\frac{N}{2}\log|\mathbf{V}| - \frac{1}{2}\mathrm{trace}\left[\mathbf{V}^{-1}\sum_n(\mathbf{y}^n-\mu)(\mathbf{y}^n-\mu)^\top\right]\\
&= -\frac{N}{2}\log|\mathbf{V}| - \frac{1}{2}\mathrm{trace}\left[\mathbf{V}^{-1}\mathbf{S}\right]
\end{aligned}$$

  $\mathbf{V}$ is model covariance; $\mathbf{S}$ is sample data covariance.
- In other words, we are trying to make the constrained model covariance as close as possible to the observed covariance, where "close" means the trace of the ratio.
- Thus, the sufficient statistics are the same as for the Gaussian: mean $\sum_n \mathbf{y}^n$ and covariance $\sum_n(\mathbf{y}^n-\mu)(\mathbf{y}^n-\mu)^\top$.

Fitting the PPCA model

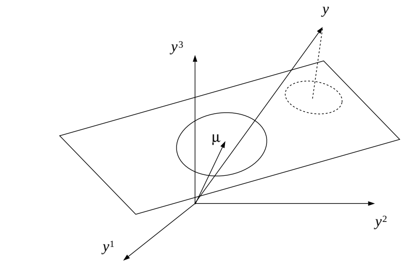
- The standard EM algorithm applies to PPCA also:
  E-step: $q^{t+1} = p(\mathbf{x}^n|\mathbf{y}^n,\theta^t)$
  M-step: $\theta^{t+1} = \mathrm{argmax}_\theta \sum_n \int_{\mathbf{x}} q^{t+1}(\mathbf{x}^n|\mathbf{y}^n)\log p(\mathbf{y}^n,\mathbf{x}^n|\theta)d\mathbf{x}^n$
- For this we need the conditional distribution (inference) and the expected log of the complete data. Results:

$$\begin{aligned}
\mathbf{E-step}: q^{t+1} &= p(\mathbf{x}|\mathbf{y},\theta^t) = \mathcal{N}(\mathbf{x}^n|\mathbf{m}^n,\mathbf{V}^n)\\
\mathbf{V}^n &= (I+\sigma^{-2}\Lambda^\top\Lambda)^{-1}\\
\mathbf{m}^n &= \sigma^{-2}\mathbf{V}^n\Lambda^\top(\mathbf{y}-\mu)\\
\mathbf{M-step}: \Lambda^{t+1} &= \left(\sum_n \mathbf{y}^n\mathbf{m}^{n\top}\right)\left(\sum_n \mathbf{V}^n\right)^{-1}\\
\sigma^{2t+1} &= \frac{1}{DN}\sum_i\left[\sum_n \mathbf{y}^n\mathbf{y}^{n\top} + \Lambda^{t+1}\sum_n \mathbf{m}^n\mathbf{y}^{n\top}\right]_{ii}
\end{aligned}$$

PCA: Zero Noise Limit

- The traditional PCA model is actually a limit as $\sigma^2 \to 0$.
  The model we saw is actually called "probabilistic PCA".
- However, the ML parameters Λ^* are the same.
  The only difference is the global sensor noise σ^2.
- In the zero noise limit inference is easier: orthogonal projection.

$$\lim_{\sigma^2 \to 0} \Lambda^\top(\Lambda\Lambda^\top + \sigma^2 I)^{-1} = (\Lambda^\top\Lambda)^{-1}\Lambda^\top$$



Direct Fitting

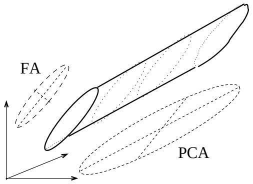
- For FA the parameters are coupled in a way that makes it impossible to solve for the ML params directly.
  We must use EM or other nonlinear optimization techniques.
- But for (P)PCA, the ML params can be solved for directly:
  The k^{th} column of Λ is the k^{th} largest eigenvalue of the sample covariance $\mathbf{S}$ times the associated eigenvector.
- The global sensor noise σ^2 is the sum of all the eigenvalues smaller than the k^{th} one.
- This technique is good for initializing FA also.
- Actually PCA is the limit as the ratio of the noise variance on the output to the prior variance on the latent variables goes to zero.
  We can either achieve this with zero noise or with infinite variance priors.

Scale Invariance in Factor Analysis

- In FA the *scale* of the data is unimportant: we can multiply $\mathbf{y}_i$ by α_i without changing anything:

$$\begin{aligned}\mu_i &\leftarrow \alpha_i \mu_i \\ \Lambda_{ij} &\leftarrow \alpha_i \Lambda_{ij} \quad \forall j \\ \Psi_i &\leftarrow \alpha_i^2 \Psi_i\end{aligned}$$

- However, the *rotation* of the data *is* important.
- FA looks for directions of large correlation in the data, so it is not fooled by large variance noise.

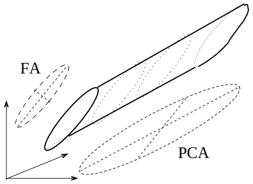


Rotational Invariance in PCA

- In PCA the *rotation* of the data is unimportant: we can multiply the data $\mathbf{y}$ by and rotation $\mathbf{Q}$ without changing anything:

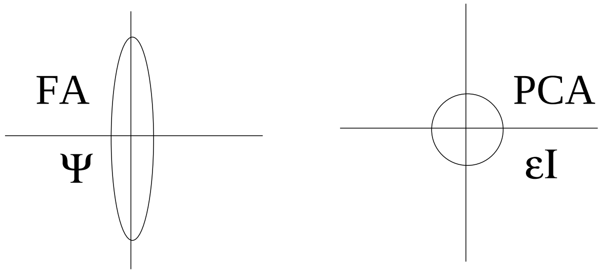
$$\begin{aligned}\mu &\leftarrow \mathbf{Q}\mu \\ \Lambda &\leftarrow \mathbf{Q}\Lambda \\ \Psi &\leftarrow \text{unchanged}\end{aligned}$$

- However, the *scale* of the data *is* important.
- PCA looks for directions of large variance, so it will chase big noise directions.



Gaussians are Footballs in High-D

- Recall the intuition that Gaussians are hyperellipsoids.
- Mean == centre of football
  Eigenvectors of covariance matrix == axes of football
  Eigenvalues == lengths of axes
- In FA our football is an axis aligned cigar.
  In PPCA our football is a sphere of radius σ^2.



Review: Gaussian Conditioning

- Remember the formulas for condtional Gaussian distributions:

$$
\begin{aligned}
p(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}) &= \mathcal{N}(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \mid \begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix})\\
p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1|\mathbf{m}_{1|2}, \mathbf{V}_{1|2})\\
\mathbf{m}_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2)\\
\mathbf{V}_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}
\end{aligned}
$$

Review: Matrix Calculations

- You often need these tricks to compute the M-step:

$$
\begin{aligned}
\frac{\partial}{\partial A}\log|A| &= (A^{-1})^\top\\
\frac{\partial}{\partial A}\mathrm{trace}[B^\top A] &= B\\
\frac{\partial}{\partial A}\mathrm{trace}[BA^\top CA] &= 2CAB
\end{aligned}
$$

- There is a good trick for inverting matrices when they can be decomposed into the sum of an easily inverted matrix (D) and a low rank outer product. It is called the *matrix inversion lemma*.

$$
(D - AB^{-1}A^\top)^{-1} = D^{-1} + D^{-1}A(B - A^\top D^{-1}A)^{-1}A^\top D^{-1}
$$

Review: Means, Variances and Covariances

- Remember the definition of the mean and covariance of a vector random variable:

$$
E[x] = \int_{\mathbf{x}} \mathbf{x}p(\mathbf{x})d\mathbf{x} = \mathbf{m}
$$

$$
\mathrm{Cov}[x] = E[(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^\top] = \int_x (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^\top p(\mathbf{x})d\mathbf{x} = \mathbf{V}
$$

  which is the expected value of the outer product of the variable with itself, after subtracting the mean.
- Also, the covariance between two variables:

$$
\begin{aligned}
\mathrm{Cov}[\mathbf{x},\mathbf{y}] &= E[(\mathbf{x}-\mathbf{m}_\mathbf{x})(\mathbf{y}-\mathbf{m}_\mathbf{y})^\top] = \mathbf{C}\\
&= \int_{\mathbf{xy}} (\mathbf{x}-\mathbf{m}_\mathbf{x})(\mathbf{y}-\mathbf{m}_\mathbf{y})^\top p(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} = \mathbf{C}
\end{aligned}
$$

  which is the expected value of the outer product of one variable with another, after subtracting their means.
  Note: $\mathbf{C}$ is not symmetric.