For today

- Multivariate Gaussians
- Covariance and Precision matrix
- Factor analysis

Multivariate probabilities refresher

In general, regardless how $\mathbf{x}$ is distributed

$$\begin{aligned} \mathrm{E}[\mathbf{A}\mathbf{x} + \mathbf{y}] &= \mathbf{A}(\mathrm{E}[\mathbf{x}]) + \mathbf{y} \\ \mathrm{Cov}[\mathbf{A}\mathbf{x} + \mathbf{y}] &= \mathbf{A}(\mathrm{Cov}[\mathbf{x}])\mathbf{A}^T \end{aligned}$$

Gaussian

$$\begin{aligned} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\} \\ \mathbf{x} &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{aligned}$$

Moments of Gaussian distributed random variables $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then

$$\begin{aligned} \mathrm{E}[\mathbf{x}] &= \int p(\mathbf{x})\mathbf{x}d\mathbf{x} = \boldsymbol{\mu} \\ \mathrm{E}[\mathbf{x}\mathbf{x}^T] &= \int p(\mathbf{x})\mathbf{x}\mathbf{x}^T d\mathbf{x} = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma} \end{aligned}$$

Drawing from Multivariate Gaussians

How to sample multivariate Gaussians

$\mathbf{x} \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$	`x = randn(p,1)`
$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_p)$	`x = randn(p,1) + mu`
$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}\mathbf{A}^T)$	`x = A*randn(p,1) + mu`

We know how to sample $\mathcal{N}(\boldsymbol{\mu}, \mathbf{A}\mathbf{A}^T)$ but what about sampling $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$? Put differently how do we get factorization $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^T$.

Various options available

- Cholesky decomposition (`chol`) of $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$ where $\mathbf{L}$ is lower diagonal, use $\mathbf{A} = \mathbf{L}$
- Eigen-decomposition (`eig`) of $\boldsymbol{\Sigma} = \mathbf{V}\mathbf{D}\mathbf{V}^T$, where $\mathbf{V}$ is orthonormal and $\mathbf{D}$ is diagonal, use $\mathbf{A} = \mathbf{V}\mathbf{D}^{1/2}$ †

† $\mathbf{D}^{1/2}$ here is a matrix square root but since matrix is diagonal, this is equivalent to entrywise square root, so you can do `diag(sqrt(diag(D)))`

Multivariate Gaussians equalities

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathbf{T}} & \mathbf{B} \end{bmatrix} \right)$$

Marginals

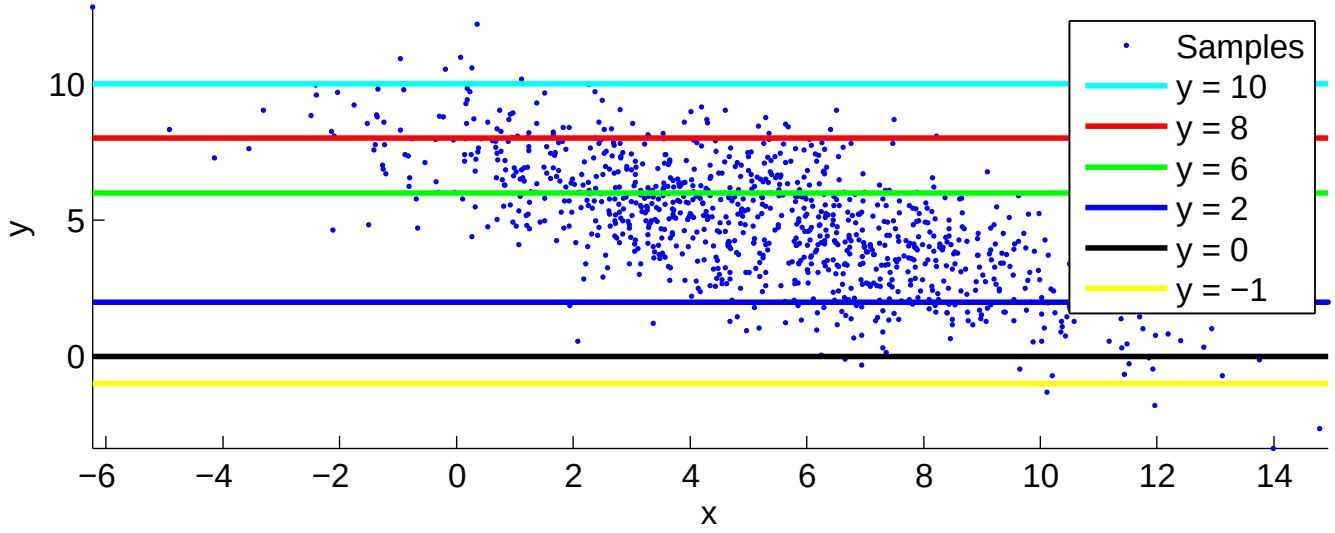
$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}(\mathbf{a}, \mathbf{A}) \\ \mathbf{y} &\sim \mathcal{N}(\mathbf{b}, \mathbf{B}) \end{aligned}$$

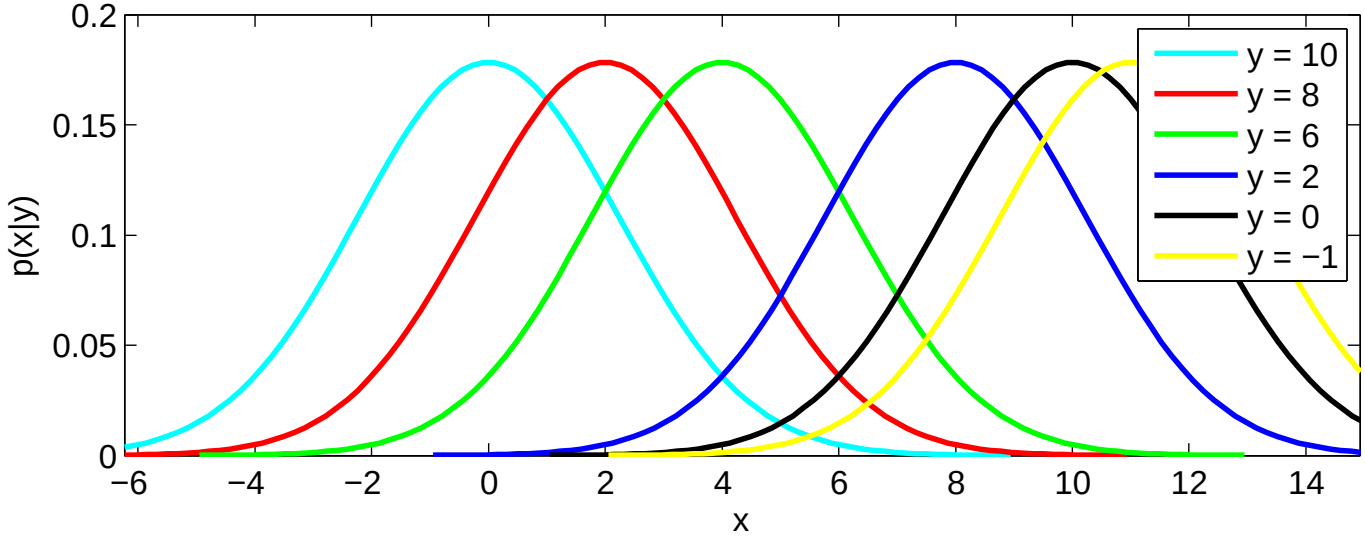
From a joint to conditionals

$$\begin{aligned} \mathbf{x}|\mathbf{y} &\sim \mathcal{N}\left(\mathbf{a} + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^{T}\right) \\ \mathbf{y}|\mathbf{x} &\sim \mathcal{N}\left(\mathbf{b} + \mathbf{C}^{T}\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^{T}\mathbf{A}^{-1}\mathbf{C}\right) \end{aligned}$$

$\mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^{T}$ is Schur complement of the joint covariance matrix.

Example of conditional distributions





Multivariate Gaussians

Forming a joint from marginal and conditional

$$
\begin{aligned}
\mathbf{x} &\sim \mathcal{N}(\mathbf{a}, \mathbf{A}) \\
\mathbf{y}|\mathbf{x} &\sim \mathcal{N}(\mathbf{b} + \mathbf{C}\mathbf{x}, \mathbf{B}) \\
\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} + \mathbf{C}\mathbf{a} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{A}^\mathsf{T}\mathbf{C}^\mathsf{T} \\ \mathbf{C}\mathbf{A} & \mathbf{B} + \mathbf{C}\mathbf{A}^\mathsf{T}\mathbf{C}^\mathsf{T} \end{bmatrix} \right)
\end{aligned}
$$

Sum of Gaussian distributed random variables

Given two Gaussian distributed random variables $\mathbf{x}$ and $\mathbf{y}$

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \mathbf{y} &\sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \end{aligned}$$

their sum $\mathbf{z} = \mathbf{x} + \mathbf{y}$ is also Gaussian distributed

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)$$

Product of Gaussian densities

Product of two $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ is an *unnormalized* Gaussian density

$$\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \propto \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where

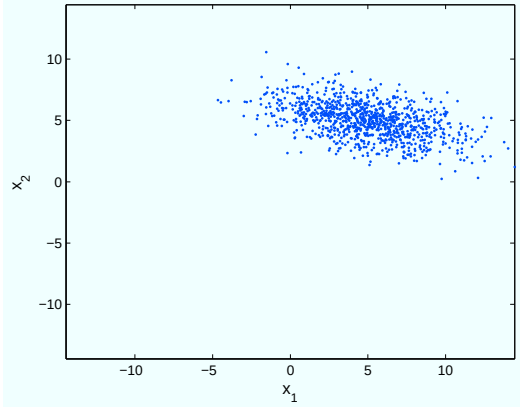
$$\begin{aligned} \boldsymbol{\Sigma} &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} \\ \boldsymbol{\mu} &= \boldsymbol{\Sigma}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 \end{aligned}$$

Whitening

Any multivariate Gaussian distribution can be seen as a rotated, scaled and shifted version of $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

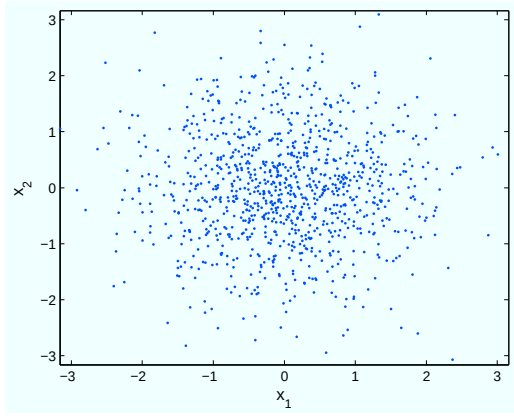
Procedure that takes data and transforms it so that its empirical covariance matrix is identity matrix is called *whitening*.

original



coordinates dependent

whitened



coordinates independent

Whitening

Given some data matrix $\mathbf{Z}$ we want to find a matrix $\mathbf{A}$ such that

$$\mathbf{I} = \mathrm{Cov}(\mathbf{AZ}) = \mathbf{A}\,\mathrm{Cov}(\mathbf{Z})\mathbf{A}^T$$

We can perform eigenvalue decomposition of $\mathrm{Cov}(\mathbf{Z})$ to obtain

$$\mathrm{Cov}(\mathbf{Z}) = \mathbf{VDV}^T = \mathbf{V}\mathbf{D}^{\frac{1}{2}}(\mathbf{D}^{\frac{1}{2}})^T\mathbf{V}^T$$

Hence

$$\mathbf{A} = \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T$$

Note that to produce whitened matrix you need to shift the data matrix $\mathbf{Z}$ so that it has zero mean, and then multiply it with $\mathbf{A}$

Gaussian MRFs

One of the tractable versions of MRFs that can have an arbitrarily complicated graph.

Instead of working with a covariance matrix, we will work with the inverse covariance matrix, also called a **precision** matrix.

$$p(\mathbf{x}) = (2\pi)^{-d/2}(\det \mathbf{P})^{1/2} \exp\left\{-(1/2)(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{P}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Precision matrix and conditional independencies

It can be shown that

$$x_i \perp x_j | x_{\{k|k\neq i, k\neq j\}} \Leftrightarrow p_{ij} = 0.$$

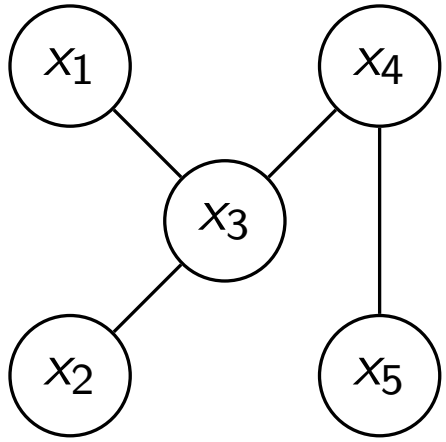
And as a consequence

$$x_i \perp x_j | x_{B(i)}, \forall j \notin B(i)$$

where $B(i) = \{k | p_{ik} \neq 0\}$.

For example gene i's expression is conditionally independent from all other genes given genes in $B(i)$.

Precision matrix and GMRF structure



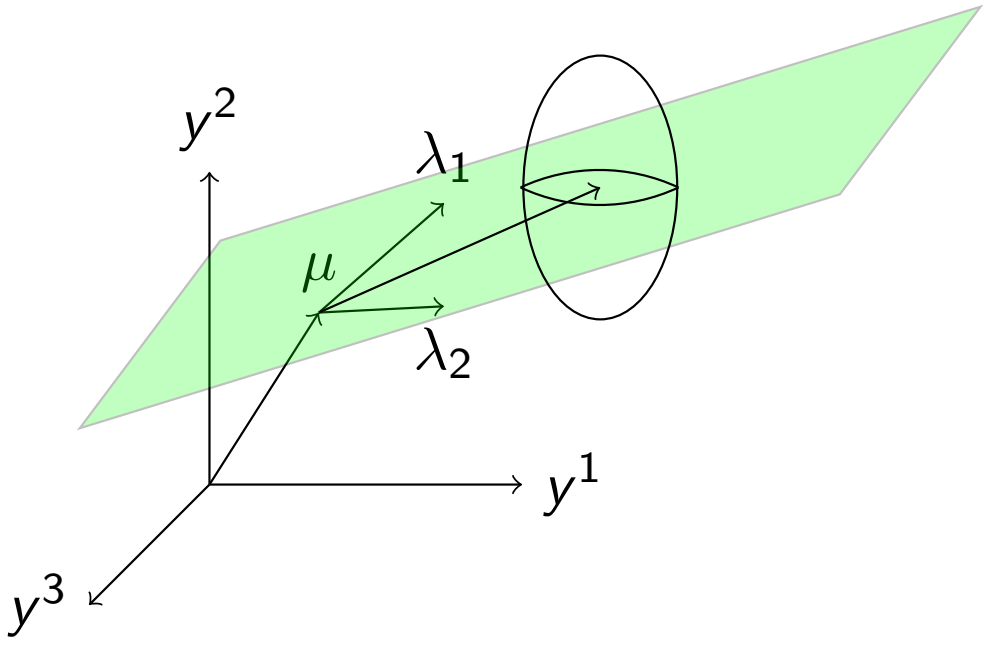
$$P = \begin{bmatrix} \bullet & & \bullet & & \\ & \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \bullet & \\ & & \bullet & \bullet & \bullet \\ & & & \bullet & \bullet \end{bmatrix}$$

Factor analysis generative model

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

$\mathbf{x}$ Factors

$\boldsymbol{\Lambda}$ Factor loading matrix



$$y = \mu + x_1\lambda_1 + x_2\lambda_2 + \epsilon$$

Marginal distribution of data

We can compute joint distribution over data and hidden variables using equalities from the initial slides

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

and Gaussian identities to compute a joint

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \mathbf{I} & \boldsymbol{\Lambda}^{\mathbf{T}} \\ \boldsymbol{\Lambda} & \boldsymbol{\Lambda}\boldsymbol{\Lambda}^{\mathbf{T}} + \boldsymbol{\Psi} \end{bmatrix}\right).$$

From this joint we can read out the marginal probability of the data

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}).$$

Dimensions

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

Factor analysis explains high-dimensional data (dim d) in a low-dimensional space (dim k).

- each data instance $\mathbf{y}_i$ is of dimension d ($d \times 1$)
- $\boldsymbol{\mu}$ is the same dimension as a data instance, $d \times 1$
- $\mathbf{x}_i$ are the hidden factors (low dimensional explanations) and of size $k \times 1$
- $\boldsymbol{\Lambda}$ maps a k-dimensional space into d-dimensional space, a $d \times k$ matrix not required to be orthogonal
- $\boldsymbol{\Psi}$ is a diagonal matrix of size $d \times d$

Likelihood function for Factor Analysis

$$\begin{aligned}\mathbf{L}(\mathbf{\Lambda}, \mathbf{\Psi}, \mu) &= \sum_i \log \int_{\mathbf{x}_i} p(\mathbf{y}_i|\mathbf{x}_i) p(\mathbf{x}_i) d\mathbf{x}_i \\ &= \sum_i \log \int_{\mathbf{x}_i} \mathcal{N}(\mathbf{y}_i|\boldsymbol{\mu} + \mathbf{\Lambda}\mathbf{x}_i, \mathbf{\Psi}) \mathcal{N}(\mathbf{x}_i|\mathbf{0}, \mathbf{I}) d\mathbf{x}_i\end{aligned}$$

Hidden variables are $\mathbf{x}$ (the coordinates in $\mathbf{\Lambda}$ coordinate system)

$\boldsymbol{\mu}, \mathbf{\Lambda}$ and $\mathbf{\Psi}$ are parameters

EM algorithm for factor analysis

As before we need to iterate steps

$$\begin{array}{lrcl} \text{E:} & q^{\text{new}}(\mathbf{x}_i) & = & p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta}) \\ \text{M:} & \boldsymbol{\theta}^{\text{new}} & = & \text{argmax}_{\boldsymbol{\theta}} \sum_i \int_{\mathbf{x}_i} q^{\text{new}}(\mathbf{x}_i) \log p(\mathbf{y}_i, \mathbf{x}_i|\boldsymbol{\theta}) d\mathbf{x}_i \end{array}$$

where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi})$

Factor analyis E-step

We first need to write out the joint probability over $(\mathbf{x}, \mathbf{y})$
We use the factor analysis model specification

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

and identities we just recalled some slides back to form a joint

$$p(\mathbf{x}, \mathbf{y}|\theta) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} | \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \mathbf{I} & \boldsymbol{\Lambda}^{\mathsf{T}} \\ \boldsymbol{\Lambda} & \boldsymbol{\Lambda}\boldsymbol{\Lambda}^{\mathsf{T}} + \boldsymbol{\Psi} \end{bmatrix}\right)$$

Obtaining posterior distribution $\mathbf{x}|\mathbf{y}$

Now we use joint to conditionals to get $\mathbf{x}|\mathbf{y}$

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

where

$$\begin{aligned} \mathbf{m} &= \mathbf{\Lambda}^T(\mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})^{-1}(\mathbf{y} - \mu) \\ \mathbf{V} &= I - \mathbf{\Lambda}^T(\mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})^{-1}\mathbf{\Lambda} \end{aligned}$$

There are additional tricks for simplifying this – Sherman-Morrison formula[2]

[2] btw. if you have not yet visited the Matrix Reference Manual link from the course page now would be a good time to do it

Matrix inversion lemma

Matrix inversion lemma or Sherman-Morrison-Woodbury formula enables us to take advantage of the rank of a matrix to avoid unnecessary computation.

Suppose you need to compute an inverse of a matrix $\mathbf{A} + \mathbf{UCV}$ where matrix sizes are $\mathbf{A} : n \times n, \mathbf{U} : n \times k, \mathbf{C} : k \times k, \mathbf{V} : k \times n$, $k < n$, and you already have access to inverse of $\mathbf{A}$.

Then you can save computational effort by using the Sherman-Morrison-Woodbury formula is

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\underbrace{(\overbrace{\mathbf{C}^{-1}}^{\text{inv}} + \mathbf{VA}^{-1}\mathbf{U})^{-1}}_{\text{inv}}\mathbf{VA}^{-1}$$

The point here is that if $\mathbf{A}^{-1}$ is known you only need to invert matrices of size $k \times k$.

Factor Analysis E-step

So in order to compute $q(\mathbf{x}_i) = p(\mathbf{x}_i|\mathbf{y}_i)$ we simply have to obtain

$$\begin{aligned} \mathbf{m}_i &= \boldsymbol{\Lambda}^T(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi})^{-1}(\mathbf{y}_i - \boldsymbol{\mu}) \\ \mathbf{V} &= I - \boldsymbol{\Lambda}^T(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi})^{-1}\boldsymbol{\Lambda} \end{aligned}$$

These two parameters define a Gaussian distribution $q(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i; \mathbf{m}_i, \mathbf{V})$

We have one of these distributions for each data instance.

But we only need to store the $\mathbf{m}_i$ since $\mathbf{V}$ is constant across data instances.

You apply Sherman-Morrison-Woodbury formula to computation of $(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi})^{-1}$, since $\boldsymbol{\Lambda}$ is of size $d \times k$. Note also that $\boldsymbol{\Psi}$ is diagonal, so also very cheap to invert and multiply with.

Factor Analysis M-step μ

This one is easy. Since μ is independent of $\mathbf{x}$, we can simply set

$$\boldsymbol{\mu} = \frac{1}{N} \sum_i \mathbf{y}_i$$

This really amounts to simply centering our data and it is not a step that needs to be repeated. We assume that the data is centered from now on.

Factor Analysis M-step $\mathbf{\Lambda}$ and $\mathbf{\Psi}$

We just figured out how to compute $q^{\text{new}}(\mathbf{x}_i)$; now let's look at the part of our bound relevant for the M-step

$$\sum_i \int_{\mathbf{x}_i} q(\mathbf{x}_i) \log p(\mathbf{y}_i, \mathbf{x}_i | \theta) d\mathbf{x}_i$$

or after losing terms that are not relevant to $\mathbf{\Lambda}$ and $\mathbf{\Psi}$

$$\sum_i \mathrm{E}_{q(\mathbf{x}_i)} \left[-\frac{1}{2} \log |\mathbf{\Psi}| - \frac{1}{2} (\mathbf{y}_i - \mathbf{\Lambda}\mathbf{x}_i)^T \mathbf{\Psi}^{-1} (\mathbf{y}_i - \mathbf{\Lambda}\mathbf{x}_i) \right]$$

we can use the fact that $q(\mathbf{x}_i)$ is a Gaussian Distribution and use identities for expectations with respect to Gaussians.[3]

[3] $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ then $\mathrm{E}[\mathbf{x}] = \boldsymbol{\mu}$ and $\mathrm{E}[\mathbf{x}\mathbf{x}^T] = \boldsymbol{\mu}\boldsymbol{\mu}^T + \mathbf{\Sigma}$

Factor Analysis M-step $\mathbf{\Lambda}$

We can now compute partial derivative and expectation

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{\Lambda}} \sum_i \mathrm{E}_{q(\mathbf{x}_i)} \left[-\frac{1}{2} \log |\mathbf{\Psi}| - \frac{1}{2} (\mathbf{y}_i - \mathbf{\Lambda}\mathbf{x}_i)^T \mathbf{\Psi}^{-1} (\mathbf{y}_i - \mathbf{\Lambda}\mathbf{x}_i) \right] \\ = \quad & -\mathbf{\Psi}^{-1} \sum_i \mathbf{y}_i \mathbf{m}_i^T + \mathbf{\Psi}^{-1} \mathbf{\Lambda} (\mathbf{V} + \mathbf{m}_i \mathbf{m}_i^T) \end{aligned}$$

equating to zero and solving for $\mathbf{\Lambda}$ yields update

$$\mathbf{\Lambda}^{\mathrm{new}} = \left(\sum_i \mathbf{y}_i \mathbf{m}_i^T \right) \left(\sum_i \mathbf{V} + \mathbf{m}_i \mathbf{m}_i^T \right)^{-1}$$

Little bit of matrix calculus

Trace and quadratic form

$$\begin{aligned}
\sum_i (\mathbf{y}_i - \boldsymbol{\mu})^T \mathbf{P} (\mathbf{y}_i - \boldsymbol{\mu}) &= \sum_{j,k} \left[\overbrace{\sum_i (\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})}^{\mathbf{S}} \right]_{j,k} p_{j,k} \\
&= \operatorname{Trace}(\mathbf{SP}) \\
&= \langle \mathbf{S}, \mathbf{P} \rangle
\end{aligned}$$

Note that here $\mathbf{S} = N \operatorname{Cov}(\mathbf{y})$

Little bit of matrix calculus

Derivatives with respect to matrices

$$\begin{aligned} \frac{\partial}{\partial \mathbf{A}} \log\det &= \mathbf{A}^{-T} \\ \frac{\partial}{\partial \mathbf{A}} \mathrm{Trace}(\mathbf{B}^T \mathbf{A}) &= \mathbf{B} \end{aligned}$$

Factor Analysis M-step $\boldsymbol{\Psi}$

Now compute partial derivative and expectation

$$\begin{aligned}
& \frac{\partial}{\partial \boldsymbol{\Psi}^{-1}} \sum_i \mathrm{E}_{q(\mathbf{x}_i)} \left[-\frac{1}{2} \log |\boldsymbol{\Psi}| - \frac{1}{2} (\mathbf{y}_i - \boldsymbol{\Lambda}\mathbf{x}_i)^T \boldsymbol{\Psi}^{-1} (\mathbf{y}_i - \boldsymbol{\Lambda}\mathbf{x}_i) \right] \\
= & \frac{\partial}{\partial \boldsymbol{\Psi}^{-1}} \sum_i \mathrm{E}_{q(\mathbf{x}_i)} \left[\frac{1}{2} \log |\boldsymbol{\Psi}^{-1}| - \frac{1}{2} \mathrm{Trace} \left((\mathbf{y}_i - \boldsymbol{\Lambda}\mathbf{x}_i)(\mathbf{y}_i - \boldsymbol{\Lambda}\mathbf{x}_i)^T \boldsymbol{\Psi}^{-1} \right) \right] \\
= & \frac{N}{2} \boldsymbol{\Psi} - \frac{1}{2} \sum_i \mathrm{E}_{q(\mathbf{x}_i)} \left[(\mathbf{y}_i - \boldsymbol{\Lambda}\mathbf{x}_i)(\mathbf{y}_i - \boldsymbol{\Lambda}\mathbf{x}_i)^T \right]
\end{aligned}$$

equating to zero and solving for $\boldsymbol{\Psi}$ yields update

$$\boldsymbol{\Psi}^{\mathrm{new}} = \frac{1}{N} \mathbf{diag} \left[\sum_i \mathbf{y}_i \mathbf{y}_i^T + \boldsymbol{\Lambda}^{\mathrm{new}} \sum_i \mathbf{m}_i \mathbf{y}_i^T \right]$$

Full EM algorithms for Factor Analysis

$$
\begin{array}{lrcl}
\text{E:} & \mathbf{m}_i & = & \boldsymbol{\Lambda}^T(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi})^{-1}\mathbf{y}_i \\
 & \mathbf{V} & = & I - \boldsymbol{\Lambda}^T(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi})^{-1}\boldsymbol{\Lambda} \\
\text{M:} & \boldsymbol{\Lambda}^{\text{new}} & = & \left(\sum_i \mathbf{y}_i\mathbf{m}_i^T\right)\left(\sum_i \mathbf{V}\right)^{-1} \\
 & \boldsymbol{\Psi}^{\text{new}} & = & \frac{1}{N}\,\mathbf{diag}\left[\sum_i \mathbf{y}_i\mathbf{y}_i^T + \boldsymbol{\Lambda}^{\text{new}}\sum_i \mathbf{m}_i\mathbf{y}_i^T\right]
\end{array}
$$

Note that we assume that data is centered (mean μ has been subtracted from each $\mathbf{y}_i$).

Cautionary remark

The marginal probability of the data is

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}).$$

Hence the way that $\boldsymbol{\Lambda}$ participates in the model of the observed data is as part of the term $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T$.

For any $\mathbf{R}$ rotation matrix recall that $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ and so

$$\boldsymbol{\Lambda}\mathbf{R}(\boldsymbol{\Lambda}\mathbf{R}^T) = \boldsymbol{\Lambda}\mathbf{R}\mathbf{R}^T\boldsymbol{\Lambda}^T = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T.$$

Hence we can use loading matrix $\boldsymbol{\Lambda}^1 = \boldsymbol{\Lambda}\mathbf{R}$ and achieve the same likelihood.[4]

This means that there is no unique $\boldsymbol{\Lambda}$ that best explains the data.
We do not have to worry about effects of orthonormality since we do not impose that on $\boldsymbol{\Lambda}$, but at the same time we lose uniqueness.

[4] In general a model for which the true parameters can be recovered in the limit of the data is called identifiable. Factor analysis is only partially identifiable.

Factor analysis applications

Most recently it has been used as means to aggregate expression measurements across multiple gene expression platforms in The Cancer Genome Atlas.

- Wang XV et al., Unifying Gene Expression Measures from Multiple Platforms Using Factor Analysis. PLoS One, 2011.
- Verhaak RG et al., Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. Cancer Cell 2010.

Sparse FA gaining more prominence; we need to cover MCMC to be able to work with this

- Engelhardt BE and Stephens M, Analysis of Population Structure: A Unifying Framework and Novel Methods Based on Sparse Factor Analysis, PLoS Genetics, 2010.
- Pruteanu-Malinici I., Automatic Annotation of Spatial Expression Patterns via Sparse Bayesian Factor Models.PLoS Comp Bio 2011.

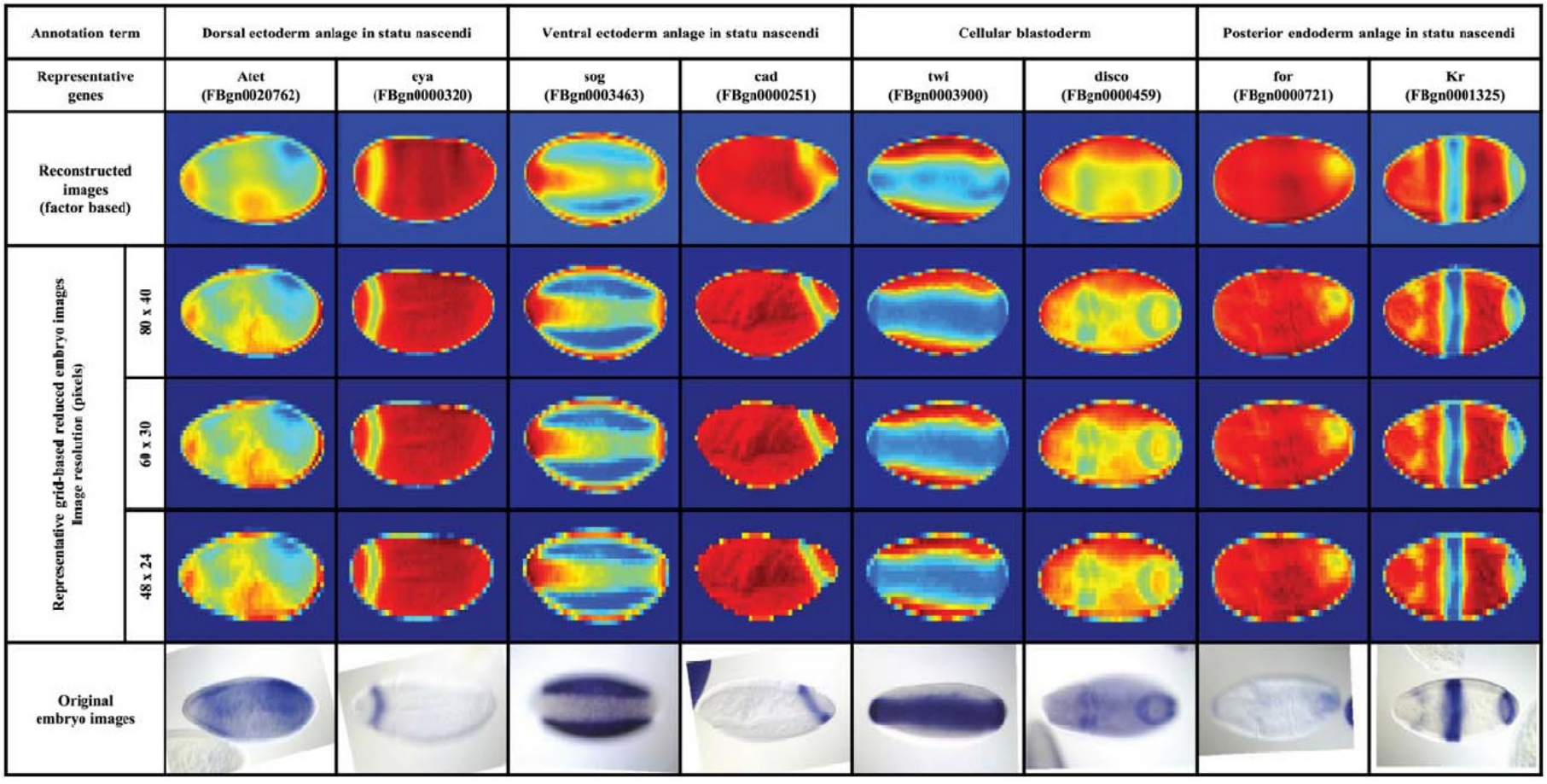


Figure 1. Original, grid-based and reconstructed factor-based images, using the estimated factors and factor loading matrix. Selected annotation terms with the highest number of associated genes; each annotation term is represented by two of its corresponding genes (with the original, the grid-based factor-based embryo images), from the time window of developmental stages 4—6. These examples reveal that images with the same annotation term can show different orientations and quite different patterns, for instance because they are taken during a relatively large temporal window during which expression can change. In the false color display, blue color indicates strong *in situ* staining while red indicates no staining.

doi:10.1371/journal.pcbi.1002098.g001

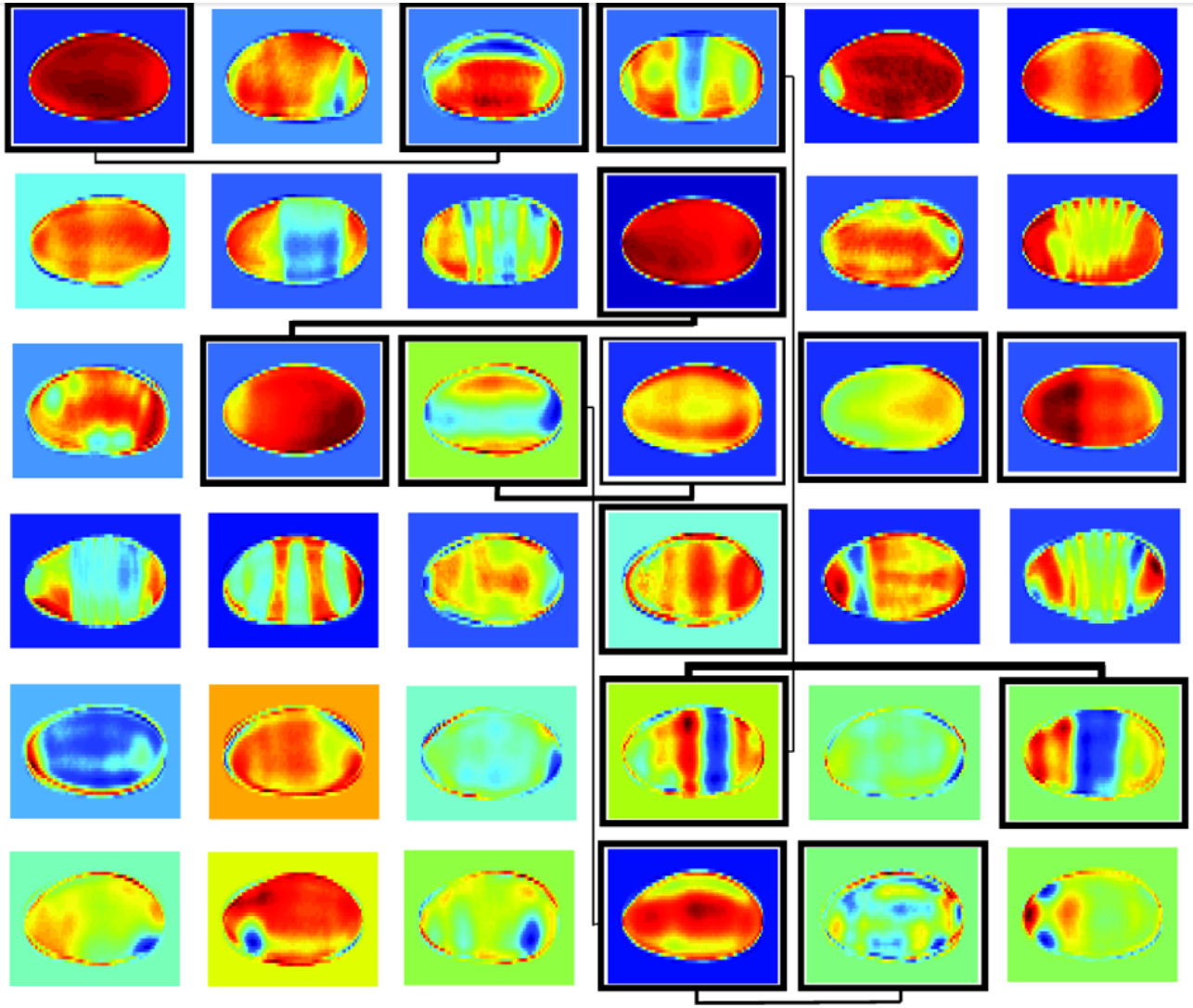


Figure 2. Selected factors estimated from a total of $k=60$ factors, for a grid size of 80×40 (data set S_{4-6}. As factors can have negative loadings, patterns may be inverse to the *in situ* staining pattern. The different background colors are an artifact and not part of the model. The bordered factors are the centroids of the largest clusters, while representative occurrences of genes shared among clusters are indicated by the weighted lines.

doi:10.1371/journal.pcbi.1002098.g002

Factor analysis and other subspace methods

Just like PCA, a basic tool for preprocessing, but also a starting point for more sophisticated models.

We have not covered various means of rotating the factors (like VARIMAX).

There is another popular dim reduction method – independent component analysis (ICA). We will cover this along with Latent Dirichlet Allocation.

There are other matrix factorization methods, such as Non-Negative Matrix Factorization, that might be of interest to you.

We will stick with methods that can be cast as learning algorithms in probabilistic models.

Today we covered

- Multivariate Gaussians
- Covariance and Precision matrix
- Factor analysis