Gaussians

Kevin P. Murphy

Last updated November 24, 2006

* Denotes advanced sections that may be omitted on a first reading.

1 Univariate Gaussians

1.1 Basic properties

Recall the one dimensional Gaussian (normal) distribution

$$\mathcal{N}(x|\mu,\sigma) \stackrel{\text{def}}{=} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \tag{1}$$

where μ is the mean and σ is the standard deviation.

$$\mu \stackrel{\text{def}}{=} EX = \int_{-\infty}^{\infty} xp(x)dx \tag{2}$$

$$\sigma^2 \stackrel{\text{def}}{=} \text{Var}X = E(X-\mu)^2 \tag{3}$$

$$= \int (x-\mu)^2 p(x)dx \tag{4}$$

$$= \int x^2 p(x)dx + \mu^2 \int p(x)dx - 2\mu \int xp(x)dx \tag{5}$$

$$= E[X^2] - \mu^2 \tag{6}$$

(from which we infer the useful result $E[X^2] = \mu^2 + \sigma^2$). See Figure 1 for an example. The term $\frac{1}{\sqrt{2\pi\sigma^2}}$ is a normalization constant. Note that it is possible that $\mathcal{N}(x|\mu,\sigma) > 1$ for some x, as long as $\int \mathcal{N}(x|\mu,\sigma)dx = 1$, since this is a **probability density function (pdf)**.

If $Z \sim \mathcal{N}(0,1)$, we say Z follows a **standard normal** distribution. Its **cumulative distribution function (cdf)** is defined as

$$\Phi(x) = \int_{-\infty}^{x} p(z)dz \tag{7}$$

which is called the **probit distribution**. This has no closed form expression, but is built in to most software packages (eg. **normcdf** in the matlab statistics toolbox). In particular, we can compute it in terms of the **error (erf) function**

$$\Phi(x;\mu,\sigma) = \tfrac{1}{2}[1 + erf(z/\sqrt{2})] \tag{8}$$

where $z = (x-\mu)/\sigma$ and

$$erf(x) \stackrel{\text{def}}{=} \frac{2}{\sqrt{p}i} \int_0^x e^{-t^2} dt \tag{9}$$

Let us see how we can use the cdf to compute how much probability mass is contained in the interval $\mu \pm 2\sigma$. If $X \sim \mathcal{N}(\mu,\sigma^2)$, then $Z = (X-\mu)/\sigma \sim \mathcal{N}(0,1)$. The amount of mass contained inside the 2σ interval is given by

$$p(a < X < b) = p(\frac{a-\mu}{\sigma} < Z < \frac{b-\mu}{\sigma}) \tag{10}$$

$$= \Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma}) \tag{11}$$

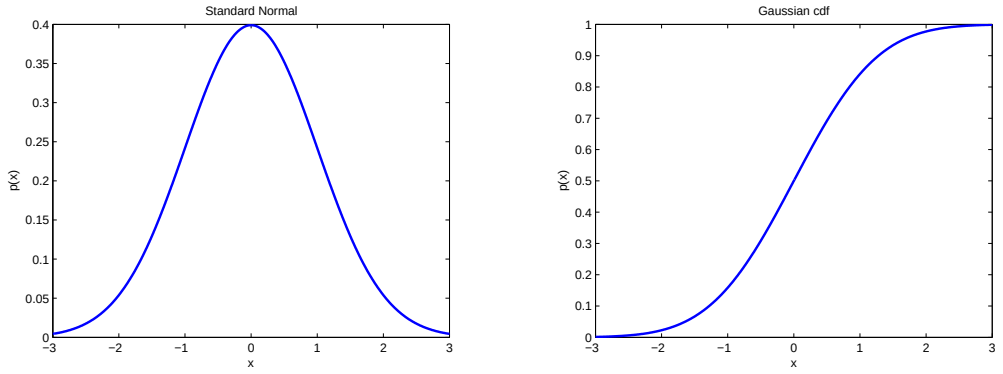

Figure 1: A standard normal pdf and cdf. The matlab code used to produce these plots is `xs=-3:0.01:3; plot(xs,normpdf(xs,mu,sigma)); plot(xs,normcdf(xs,mu,sigma));` , where $xs = [-3, -2.99, -2.98, \ldots, 2.99, 3.0]$ is a vector of points at which the density is evaluated.

Function	Matlab	R
density	`normpdf`	`dnorm`
cdf	`normcdf`	`pnorm`
inverse cdf (quantiles)	`norminv`	`qnorm`
sampling	`randn`	`rnorm`

Table 1: Translation between Matlab and R for common functions related to univariate gaussians.

Since

$$p(Z \leq -1.96) = \text{normcdf}(-1.96) = 0.025 \tag{12}$$

we have

$$p(-1.96\sigma < X - \mu < 1.96\sigma) = 1 - 2 \times 0.025 = 0.95 \tag{13}$$

Often we approximate this by replacing 1.96 with 2, and saying that the interval $\mu \pm 2\sigma$ contains 0.95 mass. See Figure **??** for an illustration.

It is also useful to compute quantiles of a distribution. A α**-quantile** is the value $f_\alpha = x$ s.t., $f(X \leq x) = \alpha$, where f is the pdf. For example, the median is the 50%-quantile. Also, if $Z \sim \mathcal{N}(0, 1)$, then the 2.5% quantile is $N_{0.025} = \Phi^{-1}(0.025) = -1.96$, where Φ^{-1} is the inverse of the Gaussian cdf:

$$z = \text{norminv}(0.025) = -1.96 \tag{14}$$

$$p(Z \leq z) = \text{normcdf}(z) = 0.025 \tag{15}$$

By symmetry of the Gaussian, $\Phi^{-1}(0.025) = -\Phi^{-1}(1 - 0.025) = \Phi^{-1}(0.975)$.

1.2 Maximum likelihood estimation

Let $X_i \sim \mathcal{N}(\mu, \sigma^2)$. Then

$$p(\mathcal{D}|\mu, \sigma^2) = \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \sigma^2) \tag{16}$$

$$\ell(\mu, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (x_i - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi) \tag{17}$$

To find the maximum, we set the partial derivatives to 0 and solve. Starting with the mean, we have

$$\frac{\partial L}{\partial \mu} = -\frac{2}{2\sigma^2}\sum_i (x_i - \mu) = 0 \tag{18}$$

$$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{19}$$

which is just the empirical mean. Similarly, letting $v = \sigma^2$, and $\mu = \hat{\mu}_{ML}$,

$$\frac{\partial L}{\partial v} = \frac{1}{2}v^{-2}\sum_i (x_i - \mu) - \frac{N}{2v} = 0 \tag{20}$$

$$\hat{\sigma^2}_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{ML})^2 \tag{21}$$

$$= \frac{1}{N}\left[\sum_i x_i^2 + \sum_i \hat{\mu}^2 - 2\sum_i x_i\hat{\mu}\right] \tag{22}$$

$$= \frac{1}{N}\sum_i (x_i^2) - (\frac{1}{N}\sum_i x_i)^2 \tag{23}$$

since $\sum_i x_i = N\hat{\mu}_{ML}$. This is just the empirical variance. Note that we divide by N and not by $N-1$. We shall discuss this issue in Section **??**.

Since we can express the MLEs in terms of $\sum_i x_i$ and $\sum_i (x_i^2)$, we say these are the **sufficient statistics** for the data. In other words, if we know the sufficient statistics, we can "throw the data away" without losing any information.

2 Multivariate Gaussians

Let $Z = (Z_1, \ldots, Z_r)$ consist of r iid $\mathcal{N}(0,1)$ random variables, L be an $p \times r$ matrix and $\vec{\mu}$ a $p \times 1$ vector. Then

$$X = (X_1, \ldots, X_p) = LZ + \vec{\mu} \tag{24}$$

has a **multivariate Gaussian** or **multivariate normal (MVN)** distribution given by

$$\mathcal{N}(\vec{x}|\vec{\mu}, \Sigma) \stackrel{\text{def}}{=} \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp[-\tfrac{1}{2}(\vec{x} - \vec{\mu})^T\Sigma^{-1}(\vec{x} - \vec{\mu})] \tag{25}$$

where $\vec{\mu}$ is a $p \times 1$ vector, Σ is a $p \times p$ **symmetric positive definite (pd)** matrix, and p is the dimensionality of $\vec{x}$. It can be shown that $E[X] = \vec{\mu}$ and $\text{Cov}[X] = \Sigma$ (see e.g., [Bis06, p82]). (Note that in the 1D case, σ is the standard deviation, whereas in the multivariate case, Σ is the covariance matrix.) In fact we have

$$\text{Cov}X = \Sigma = L \text{ Cov } ZL = LL^T \tag{26}$$

So a MVN is a linear combination of scalar iid Gaussians. We can also show that linear combinations of MVN are MVN:

$$A \sim N(\mu, \Sigma) \quad \Rightarrow \quad AX \sim N(A\mu, A\Sigma A') \tag{27}$$

This implies that marginals of a MVN are also Gaussian. To see this, suppose that $X \in \mathbb{R}^4$ and we want to compute $p(X_1, X_2)$: we can just use the projection matrix

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{28}$$

The **quadratic form** $\Delta = (\vec{x} - \vec{\mu})^T\Sigma^{-1}(\vec{x} - \vec{\mu})$ in the exponent is called the **Mahalanobis distance** between $\vec{x}$ and $\vec{\mu}$. The equation $\Delta = \text{const}$ defines an ellipsoid, which are the level sets of constant density: see Figure 2, where the region inside the ellipses contains 95% of the mass of each Gaussian. We will explain this in more detail below.

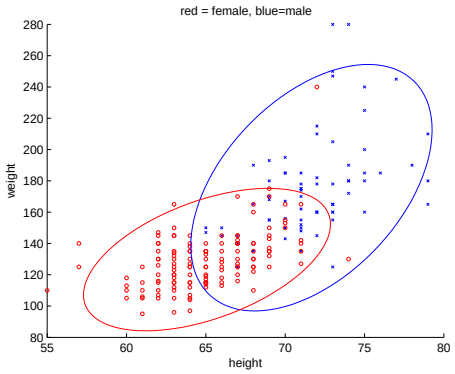

Figure 2: A scatterplot of height and weight of various men (blue crosses) and women (red circles) with fitted 2D Gaussians superimposed. Figure produced by `biometric_plot`.

μ and Σ are called **moment parameters**. An alternative is to use the **canonical parameters** or **information form**, defined as follows

$$p(x|\eta,\Lambda) = \exp[a + \eta^T x - \frac{1}{2}x^T \Lambda x] \tag{29}$$

$$\Lambda = \Sigma^{-1} \tag{30}$$

$$\eta = \Sigma^{-1}\mu \tag{31}$$

$$a = -\tfrac{1}{2}(n\log(2\pi) - \log|\Lambda| + \eta^T \Lambda \eta \tag{32}$$

where Λ is called the **precision matrix** or **concentration matrix**, and a is the normalization constant.

2.1 Bivariate Gaussians

In the 2D case, define the **correlation coefficient** between X and Y as

$$\rho = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} \tag{33}$$

Hence the covariance matrix is

$$\Sigma = \begin{pmatrix} \sigma_X & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y \end{pmatrix} \tag{34}$$

and the pdf (for the zero mean case) becomes

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} - \frac{2\rho xy}{(\sigma_x\sigma_y)}\right)\right) \tag{35}$$

It should be clear from this example that when doing multivariate analysis, using matrices and vectors is easier than working with scalar variables.

2.2 Eigenvectors and eigenvalues

To understand MVNs, it is necessary to know some **linear algebra**. Here we provide a quick review.

We can compute the eigenvectors u_i and eigenvalues λ_i of any square matrix A:

$$Au_i = \lambda_i u_i \tag{36}$$

We can write this in matrix form as

$$A = U\Lambda U^T = \sum_{i=1}^{p} \lambda_i \vec{u}_i \vec{u}_i^T \tag{37}$$

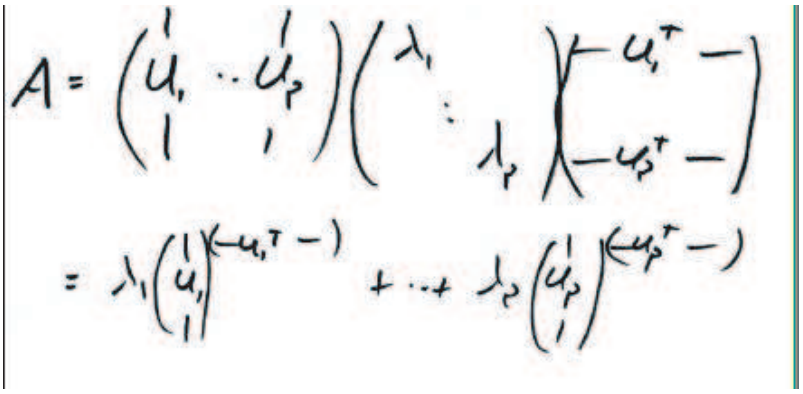

Figure 3: Diagonalizing a square matrix $A = U\Lambda U^T$.

where the columns of U are the u_i and $\Lambda = \text{diag}(\lambda_i)$. This is called **diagonalizing** A. (In matlab, just type `[U,Lam]=eig(A)`.) See Figure 3.

If A is **real and symmetric**, then the eigenvalues are real and the eigenvectors are orthonormal, so that

$$u_i^T u_j = I_{ij} \tag{38}$$

or

$$U^T U = I \tag{39}$$

where I is the **identity matrix**. We say that U is an **orthogonal matrix**.

To see why it is possible to write $A = U\Lambda U^{-1}$, suppose A is a linear transformation. It can always be decomposed into a rotation U, a scaling Λ and a reverse rotation $U^T = U^{-1}$.

The **rank** of A is the number of non-zero eigenvalues. If all $\lambda_i \geq 0$, then A is **positive semi definite (psd)**, i.e., $x^T Ax \geq 0$ for all x. (If we consider the Mahalanobis distance for a 0-mean Gaussian, $\Delta = x^T \Sigma^{-1} x$, it seems reasonable to require $\Delta \geq 0$.) Note that if all elements of A are positive, it does not mean A is psd. For example,

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \tag{40}$$

has all positive elements, but is not psd, since $\lambda_1 = 5.37$ and $\lambda_2 = -0.37$. Intuitively you can think of psd matrices as defining concave "bowls", since the corresponding **quadratic form** will satisfy $x^T Ax \geq 0$, and therefore always curve up.

2.3 Degenerate MVNs

A **degenerate** multivariate Gaussian is one for which the covariance matrix is singular, $\det \Sigma = 0$. Consider for example

$$X = \begin{pmatrix} 2 & 0 \\ 5 & 0 \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{41}$$

Here the rank of L is $1 < p = 2$, since the rows are linearly dependent. Hence $\Sigma = LL^T$ is also rank 1, and is therefore not invertible. This corresponds to a 1D density embedded in a 2D space.

2.4 Visualizing the covariance matrix

Letting $\Sigma = U\Lambda U^T$ we find

$$\Sigma^{-1} = U^{-T}\Lambda^{-1}U^{-1} = U\Lambda^{-1}U = \sum_{i=1}^{p} \frac{1}{\lambda_i} \vec{u}_i \vec{u}_i^T \tag{42}$$

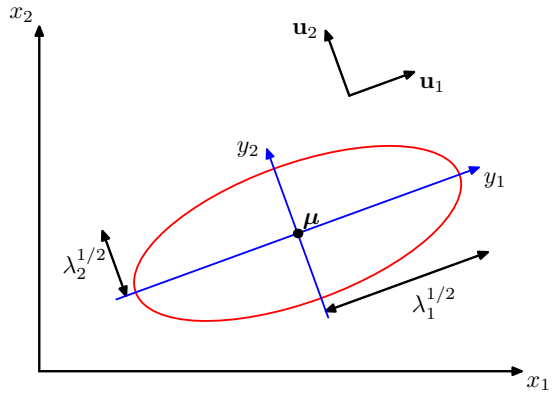

Figure 4: Visualization of a 2 dimensional Gaussian density. The major and minor axes of the ellipse are defined by the first two eigenvectors of the covariance matrix, namely $\vec{u}_1$ and $\vec{u}_2$. Source: [Bis06] Figure 2.7.

Hence

$$(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu}) = (\vec{x}-\vec{\mu})^T\left(\sum_{i=1}^{p}\frac{1}{\lambda_i}\vec{u}_i\vec{u}_i^T\right)(\vec{x}-\vec{\mu}) \quad (43)$$

$$= \sum_{i=1}^{p}\frac{1}{\lambda_i}(\vec{x}-\vec{\mu})^T\vec{u}_i\vec{u}_i^T(\vec{x}-\vec{\mu}) \quad (44)$$

$$= \sum_{i=1}^{p}\frac{y_i^2}{\lambda_i} \quad (45)$$

where $y_i \stackrel{\text{def}}{=} \vec{u}_i^T(\vec{x}-\vec{\mu})$. The y variables define a new coordinate system that is shifted (by $\vec{\mu}$) and rotated (by U) with respect to the original x coordinates: $\vec{y} = U(\vec{x}-\vec{\mu})$.

Recall that the equation for an ellipse in 2D is

$$\frac{y_1^2}{\lambda_1}+\frac{y_2^2}{\lambda_2}=1 \quad (46)$$

Hence we see that the contours of equal probability density of a Gaussian lie along ellipses: see Figure 4 and Figure 5.

The Matlab code to plot the 2D Gaussian in Figure 5 is shown below. It uses some useful commands: `meshgrid` to generate a 2D array of points, `surfc` to plot the surface, and `contour` to plot the contours.

```
function plotGauss2dDemo()

mu = [1 0]';        % mean (must be row vector for mvnpdf)
S  = [4 3; 3 4];    % covariance
figure(1); clf
plotSurf(mu, S, 1)

% Compute whitening transform:
[U,D] = eig(S);      % U = eigenvectors, D= diagonal matrix of eigenvalues.
A = sqrt(inv(D))*U';
mu2 = A*mu;
S2  = A*S*A';

plotSurf(mu2, S2, 3)

%%%%%%%%%%

function plotSurf(mu, S, figndx)

[U,D] = eig(S);      % U = eigenvectors, D= diagonal matrix of eigenvalues.
```

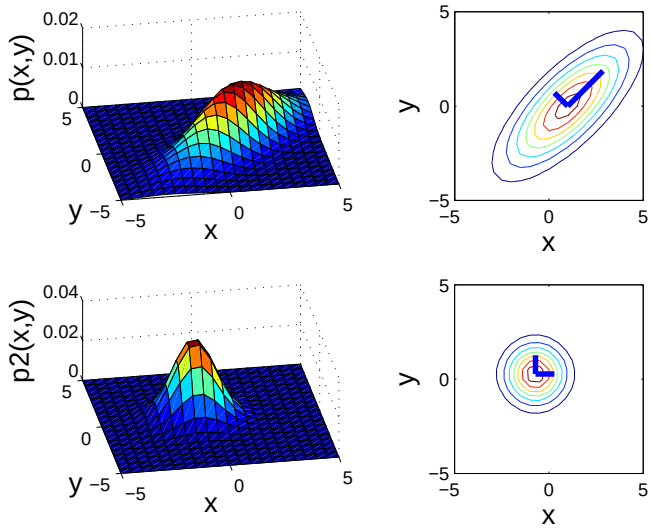

Figure 5: Visualization of a 2 dimensional Gaussian density. This figure was produced by `plotGauss2dDemo`.

```
% Evaluate p(x) on a grid.
stepSize = 0.5;
[x,y] = meshgrid(-5:stepSize:5,-5:stepSize:5); % Create grid.
[r,c]=size(x);

% data(k,:) = [x(k) y(k)] for pixel k
data = [x(:) y(:)];
p = mvnpdf(data, mu', S);
p = reshape(p, r, c);

% scale density so it sums to 1
p=p*stepSize^2;  %  p2(x,y)  defeq  p(x: x+dx, y: y+ dy) approx p(x,y) dx dy
assert(approxeq(sum(p(:)), 1, 1e-1))

subplot(2,2,fignd x)
surfc(x,y,p);                  % 3D plot
view(-10,50);
xlabel('x','fontsize',15);
ylabel('y','fontsize',15);
zlabel('p(x,y)','fontsize',15);

subplot(2,2,fignd x+1)
contour(x,y,p);           % Plot contours
axis('square');
xlabel('x','fontsize',15);
ylabel('y','fontsize',15);
% Plot first eigenvector
line([mu(1) mu(1)+sqrt(D(1,1))*U(1,1)],[mu(2) mu(2)+sqrt(D(1,1))*U(2,1)],'linewidth',3)
% Plot second eigenvector
line([mu(1) mu(1)+sqrt(D(2,2))*U(1,2)],[mu(2) mu(2)+sqrt(D(2,2))*U(2,2)],'linewidth',3)
```

A faster alternative to using the `contour` command is to just to realise that if X is a matrix of points on the circle, then $Y = U\Lambda^{\frac{1}{2}}X$ is a matrix of points on the ellipse represented by $\Sigma = U\Lambda U^T$. Hence we can use the code below to plot ellipsoids of constant density.

Using the fact that the Mahalanobis distance $\Delta = (x-\mu)^T\Sigma^{-1}(x-\mu)$ is a sum of squares of p Gaussian random variables, we have $\Delta \sim \chi^2_p$ (see Section **??**). Hence we can find the value of Δ that corresponds to a 95% confidence interval by using `delta = chi2inv(0.95, 2)` where the 2 is because $p = 2$. If we plot the locus of points for which $\Delta = \delta$, we will enclose 95% of the probability mass.

```
function h=plot2dgauss(mu, Sigma, color)
```

```
% plot2dgauss, based on code by Mark Paskin
% function h=plot2dgauss(mu, Sigma, color)
% Plot an ellipse representing the covariance matrix of a Gaussian

if size(Sigma) ~= [2 2], error('Sigma must be a 2 by 2 matrix'); end
%if length(mu) ~= 2, error('mu must be a 2 by 1 vector'); end
if nargin < 3, color = 'r'; end

mu = mu(:);
[U, D] = eig(Sigma);
n = 100;
t = linspace(0, 2*pi, n);
xy = [cos(t); sin(t)];
%k = 1;
k = sqrt(conf2mahal(0.95, 2));
w = (k * U * sqrt(D)) * xy;
z = repmat(mu, [1 n]) + w;
h = plot(z(1, :), z(2, :), color);

%%%%%%%%%

function m = conf2mahal(c, d)
m = chi2inv(c, d);
```

2.5 Whitening

Any Gaussian random variable can be transformed so its covariance matrix is spherical. This is called **whitening**. Let $X \sim \mathcal{N}(\mu, \Sigma)$ and $\Sigma = U \Lambda U^T$. Let

$$Y = \Lambda^{-\frac{1}{2}} U^T X \tag{47}$$

where $\Lambda^{-\frac{1}{2}} = \text{diag}(1/\sqrt{\Lambda_{ii}})$, Then

$$\begin{aligned} \text{Cov}Y &= \Lambda^{-\frac{1}{2}} U^T \Sigma U \Lambda^{-\frac{1}{2}} && (48)\\ &= \Lambda^{-\frac{1}{2}} U^T U \Lambda U^T U \Lambda^{-\frac{1}{2}} && (49)\\ &= \Lambda^{-\frac{1}{2}} \Lambda \Lambda^{-\frac{1}{2}} && (50)\\ &= I && (51) \end{aligned}$$

using $\text{Cov}[AY] = A\text{Cov}[Y]A^T$ and $UU^T = U^T U = I$. and

$$EY = \Lambda^{-\frac{1}{2}} U^T E[X] \tag{52}$$

So

$$Y \sim \mathcal{N}(\Lambda^{-\frac{1}{2}} U^T \mu, I) \tag{53}$$

has a spherical covariance. See Figure 5 for an example.

2.6 Sampling from a multivariate Gaussian

It is often necessary to sample from a multivariate Gaussian, $Y \sim \mathcal{N}(\mu, \Sigma)$,. One way to do this is to use a **Cholesky decomposition**, $\Sigma = L^T L$. Specifically, we first sample $X \sim \mathcal{N}(0, I)$ and then set $Y = L^T X + \mu$. This is valid since

$$\text{Cov}[Y] = L^T \text{Cov}[X] L = L^T I L = \Sigma \tag{54}$$

(If you have the Matlab statistics toolbox, you can just call `mnvrnd`, but it is useful to know this other method.) For example, the code to generate the plot in Figure 6 is shown below.

```
% gaussSampleDemo.m
% sample data from a spherical, diagonal and full cov Gaussian in 2D

figure(1); clf
N = 500;
z = 10;
```

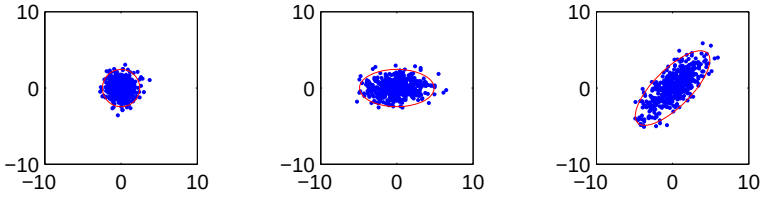

Figure 6: Samples from a spherical, diagonal and full covariance Gaussian, with 95% confidence ellipsoid superimposed. This figure was generated using `gaussSampleDemo`.

```
Sigma{1} = [1 0; 0 1];
Sigma{2} = [4 0; 0 1];
Sigma{3} = [4 3; 3 4];
mu = [0 0];


for i=1:3
  %x = mvnrnd(mu, Sigma{i}, N);
  L  = chol(Sigma{i});
  x = (L' * randn(2, N))' + repmat(mu, N, 1);
  subplot(1,3,i)
  plot(x(:,1), x(:,2), '.');
  hold on
  plot2dgauss(mu(:), Sigma{i}, 'r');
  axis([-z z -z z])
  axis square
end
```

2.7 Parsimonious covariance matrices

A full covariance matrix has $p(p+1)/2$ parameters. Hence it may be hard to estimate from data. We can restrict Σ to be diagonal; this has p parameters. Or we can use a **spherical (isotropic)** covariance, $\Sigma = \sigma^2 I$. See Figure 6 for a visualization of these different assumptions. We will consider other **parsimonious representations** for high dimensional Gaussian distributions later in the book. The problem of estimating a structured covariance matrix is called **covariance selection**.

2.8 Independence and correlation

RVs X, Y have a joint Gaussian distribution if $p(X, Y)$ is multivariate Gaussian, $(X, Y) \sim \mathcal{N}(\mu, \Sigma)$. *In this case,* if X and Y are uncorrelated, then they are independent, i.e.,

$$\Sigma_{X,Y} = 0 \iff X \perp Y \tag{55}$$

To prove this, note that

$$p(X, Y) = \mathcal{N}\left(\begin{pmatrix} X \\ Y \end{pmatrix} | \mu, \begin{pmatrix} \sigma_X^2 & 0 \\ 0 & \sigma_Y^2 \end{pmatrix}\right) = \mathcal{N}(X, \mu_X, \sigma_X^2)\mathcal{N}(Y, \mu_Y, \sigma_Y^2) = p(X)p(Y) \tag{56}$$

since the cross-product terms vanish when we multiply out the terms in the exponent. This is an exception to the general rule established in Section **??**, which showed that uncorrelated does not mean independernt.

However, now suppose X and Y are marginally Gaussian, but *not jointly Gaussian*. In particular, let $X \sim \mathcal{N}(0, 1)$ and $Y = WX$, where $p(W = -1) = p(W = 1) = 0.5$. It is clear that X and Y are not independent, since Y is a function of X. However, we can show (Exercise **??**) that $Y \sim \mathcal{N}(0, 1)$ and $\text{Cov}(X, Y) = 0$. So uncorrelated does not mean independent even if the variables are Gaussian (only if they are *jointly* Gaussian).

2.9 Sparse precision matrices encode conditional independence

Zeros in the precision matrix correspond to conditional independences. More precisely, if $\Lambda_{ij} = 0$, where $\Lambda = \Sigma^{-1}$, then $X_i \perp X_j | X_{rest}$. To prove this, let $i = 1, j = 2, rest = 3 : p$ and let us compute $p(x_{1,2}|x_r)$. where block 1

denote variables i, j and block 2 denote the rest. Then using the Equations in Section 2.10, we have

$$\begin{aligned} p(x_{12}|x_r) &= \mathcal{N}(\mu_{12|r}, \Sigma_{12,12|r}) && (57)\\ \Sigma_{12,12|r} &= \Lambda^{-1}_{12,12} && (58)\\ &= \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix}^{-1} && (59)\\ &= \frac{1}{\lambda_{11}\lambda_{22} - (\lambda_{12}^2)} \begin{pmatrix} \lambda_{22} & -\lambda_{21} \\ -\lambda_{12} & \lambda_{11} \end{pmatrix} && (60) \end{aligned}$$

The correlation coefficient between X_1 and X_2 in this conditional distribution is called the **partial correlation coefficient**, and is given by

$$\begin{aligned} \rho_{12|r} &= \frac{\Sigma_{12,12|r}(1,2)}{\sqrt{\Sigma_{12,12|r}(1,1)\Sigma_{12,12|r}(2,2)}} && (61)\\ &= \frac{-\lambda_{12}}{\sqrt{\lambda_{11}\lambda_{22}}} && (62) \end{aligned}$$

So we see that $\rho_{12|r} = 0 \iff \lambda_{12} = 0$. And since uncorrelated implies independent (for jointly Gaussian variables), we have shown that zeros in the precision matrix represent conditional independence.

Another way to see this is to expand out the canonical form

$$p(x) = \exp[a + \eta^T x - \tfrac{1}{2} x^T \Lambda x] = \exp[a + \sum_j \eta_j x_j - \tfrac{1}{2} \sum_i \sum_j \lambda_{ij} x_i x_j] \quad (63)$$

and to notice that if $\lambda_{ij} = 0$, this factorizes into independent terms involving i and j. By the **factorization theorem**, $X \perp Y|Z$ iff the joint factorizes as follows

$$p(x, y, z) = f(x, z)g(y, z) \quad (64)$$

for some functions f, g.

2.10 Marginals and conditionals of a MVN

In Section **??**, we discussed how to compute marginals and conditionals from joint distributions, i.e., how to compute $p(x_1) = \sum_{x_2} p(x_1, x_2)$ and $p(x_1|x_2) = p(x_1, x_2)/p(x_2)$. For discrete random variables, this is straightforward. But for parametric distributions, one has to replace brute force summation with integration, which can require some messy algebra. Below we derive the relevant results for Gaussians.

Suppose $x = (x_1, x_2)$ is jointly Gaussian with parameters

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \quad \Lambda = \Sigma^{-1} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}, \quad (65)$$

Below we will show that we can factorize the joint as

$$\begin{aligned} p(x_1, x_2) &= p(x_2)p(x_1|x_2) && (66)\\ &= \mathcal{N}(x_2|\mu_2, \Sigma_{22})\mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2} && (67) \end{aligned}$$

where the marginal parameters for $p(x_2)$ are just gotten by extracting rows and columns for x_2, and the conditional parameters for $p(x_1|x_2)$ are given by

$$\begin{aligned} \mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) && (68)\\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} && (69) \end{aligned}$$

Note that the new mean is a linear function of x_2, and the new covariance is independent of x_2. Note that both the marginal and conditional distributions are themselves Gaussian: see Figure 7.

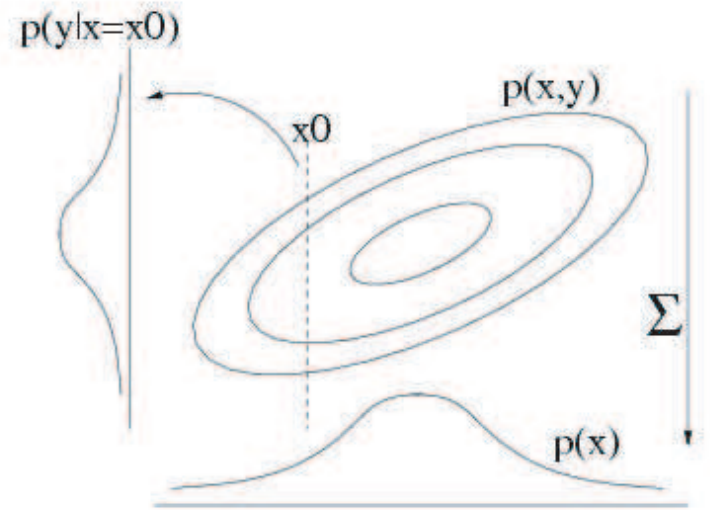

Figure 7: Marginalizing and conditionalizing a 2D Gaussian results in a 1D Gaussian. Source: Sam Roweis.

The equivalent results in canonical parameters are given below. Let

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Lambda = \Sigma^{-1} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}, \quad \eta = \Lambda\mu \tag{70}$$

Then the marginal is given by

$$\begin{aligned} p(x_2) &= \mathcal{N}(x_2|\eta_2^m, \Lambda_2^m) && (71)\\ \eta_2^m &= \eta_2 - \Lambda_{21}\Lambda_{11}^{-1}\eta_1 && (72)\\ \Lambda_2^m &= \Lambda_{22} - \Lambda_{21}\Lambda_{11}^{-1}\Lambda_{12} && (73) \end{aligned}$$

and the conditional is given by

$$\begin{aligned} p(x_1|x_2) &= \mathcal{N}(x_1; \eta_{1|2}, \Lambda_{1|2}) && (74)\\ \eta_{1|2} &= \eta_1 - \Lambda_{12}x_2 && (75)\\ \Lambda_{1|2} &= \Lambda_{11} && (76) \end{aligned}$$

Note that marginalization is easier in moment form, but conditioning is easier in canonical form.

2.11 Partitioned matrices

To derive the above results, it is helpful to know the following results for manipulating block structured matrices. (This section is based on [Jor06, ch13].)

Consider a general partioned matrix

$$M = \begin{pmatrix} E & F \\ G & H \end{pmatrix} \tag{77}$$

where we assume E and H are invertible. The goal is to derive an expression for M^{-1}. If we could block diagonalize M, it would be easier, since then the inverse would be a diagonal matrix of the inverse blocks. To zero out the top right we can pre-multiply as follows

$$\begin{pmatrix} I & -FH^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} E - FH^{-1}G & 0 \\ G & H \end{pmatrix} \tag{78}$$

Similarly, to zero out the bottom right we can post-multiply as follows

$$\begin{pmatrix} I & -FH^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} E & F \\ G & H \end{pmatrix} \begin{pmatrix} I & 0 \\ -H^{-1}G & I \end{pmatrix} = \begin{pmatrix} E - FH^{-1}G & 0 \\ 0 & H \end{pmatrix} \tag{79}$$

The top left corner is called the **Schur complement** of M wrt H, and is denoted M/H:

$$M/H = E - FH^{-1}G \tag{80}$$

If we rewrite the above as

$$XYZ = W \tag{81}$$

where $Y = M$, we get the following expression for the determinant of a partitioned matrix:

$$|X||Y||Z| = |W| \tag{82}$$

$$|M| = |M/H||H| \tag{83}$$

since $|X| = |Z| = 1$ and $|W| = |M/H||H|$. Also, we can derive the inverse as follows

$$Z^{-1}Y^{-1}X^{-1} = W^{-1} \tag{84}$$

$$Y^{-1} = ZW^{-1}X \tag{85}$$

hence

$$\begin{pmatrix} E & F \\ G & H \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ -H^{-1}G & I \end{pmatrix} \begin{pmatrix} (M/H)^{-1} & 0 \\ 0 & H^{-1} \end{pmatrix} \begin{pmatrix} I & -FH^{-1} \\ 0 & I \end{pmatrix} \tag{86}$$

$$= \begin{pmatrix} (M/H)^{-1} & -(M/H)^{-1}FH^{-1} \\ -H^{-1}G(M/H)^{-1} & H^{-1} + G(M/H)^{-1}FH^{-1} \end{pmatrix} \tag{87}$$

Alternatively, we could have decomposed the matrix M in terms of E and M/E, yielding

$$\begin{pmatrix} E & F \\ G & H \end{pmatrix}^{-1} = \begin{pmatrix} E^{-1} + E^{-1}F(M/E)^{-1}GE^{-1} & E^{-1}F(M/E)^{-1} \\ -(M/E)^{-1}GE^{-1} & (M/E)^{-1} \end{pmatrix} \tag{88}$$

Equating these two expression yields the following two formulae, the first of which is known as the **matrix inversion lemma** (aka **Sherman-Morrison-Woodbury formula**)

$$(E - FH^{-1}G)^{-1} = E^{-1} + E^{-1}F(H - GE^{-1}F)^{-1}GE^{-1} \tag{89}$$

$$(E - FH^{-1}G)^{-1}FH^{-1} = E^{-1}F(H - GE^{-1}F)^{-1} \tag{90}$$

In the special case that $H = -1$, $F = u$ a column vector, $G = v'$ a row vector, we get the following formula for a **rank one update of an inverse**

$$(E + uv')^{-1} = E^{-1} + E^{-1}u(-I - v'E^{-1}u)^{-1}v'E^{-1} \tag{91}$$

$$= E^{-1} - \frac{E^{-1}uv'E^{-1}}{1 + v'E^{-1}u} \tag{92}$$

2.12 Marginals and conditionals of MVNs: derivation

Armed with knowledge of Schur complements etc, we can derive the results in Section 2.10.

Let us factor the joint $p(x_1, x_2)$ as $p(x_2)p(x_1|x_2)$ by applying Equation 86 to the matrix inverse in the exponent term.

$$\exp\left\{-\frac{1}{2}\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}^{-1} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}\right\} \tag{93}$$

$$= \exp\left\{-\tfrac{1}{2}\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T \begin{pmatrix} I & 0 \\ -\Sigma_{22}^{-1}\Sigma_{21} & I \end{pmatrix} \begin{pmatrix} (\Sigma/\Sigma_{22})^{-1} & 0 \\ 0 & \Sigma_{22}^{-1} \end{pmatrix} \begin{pmatrix} I & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}\right\} \tag{94}$$

$$= \exp\left\{-\tfrac{1}{2}(x_1 - \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2))^T(\Sigma/\Sigma_{22})^{-1}(x_1 - \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2))\right\} \tag{95}$$

$$\times \exp\left\{-\tfrac{1}{2}(x_2 - \mu_2)^T\Sigma_{22}^{-1}(x_2 - \mu_2)\right\} \tag{96}$$

This is of the form

$$\exp(\text{quadratic form in } x_1, x_2) \times \exp(\text{quadratic form in } x_2) \tag{97}$$

Using Equation 83 we can also split up the normalization constants

$$(2\pi)^{(p+q)/2}|\Sigma|^{\frac{1}{2}} = (2\pi)^{(p+q)/2}(|\Sigma/\Sigma_{22}||\Sigma_{22}|)^{\frac{1}{2}} \tag{98}$$

$$= (2\pi)^{p/2}|\Sigma/\Sigma_{22}|^{\frac{1}{2}} \ (2\pi)^{q/2}|\Sigma_{22}|^{\frac{1}{2}} \tag{99}$$

Hence we have succesfully factorized the joint as

$$p(x_1, x_2) = p(x_2)p(x_1|x_2) \tag{100}$$

$$= \mathcal{N}(x_2|\mu_2, \Sigma_{22})\mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2}) \tag{101}$$

where the parameters of the marginal and conditional distribution can be read off from the above equations, using

$$(\Sigma/\Sigma_{22})^{-1} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{102}$$

2.13 Bayes rule for MVNs

Consider representing the joint distribution on X and Y in **linear Gaussian** form:

$$p(x) = \mathcal{N}(x|\mu, \Lambda^{-1}) \tag{103}$$

$$p(y|x) = \mathcal{N}(y|Ax + b, S^{-1}) \tag{104}$$

where Λ and S are precision matrices. It can be shown (see e.g., [Bis06, p93]) that we can invert this model as follows

$$p(y) = \mathcal{N}(y|A\mu + b, S^{-1} + A\Lambda^{-1}A^T) \tag{105}$$

$$p(x|y) = \mathcal{N}(x|\Sigma[A^T S(y - b)] + \Lambda\mu, \Sigma) \tag{106}$$

$$\Sigma = (\Lambda + A^T SA)^{-1} \tag{107}$$

Consider the following 1D example, where we try to estimate x from a noisy observation y, where the noise level is τ^2. If the prior on x is Gaussian, and the likelihood $p(y|x)$ is Gaussian, then the posterior $p(x|y)$ is also Gaussian:

$$p(x) = \mathcal{N}(x|\mu, \sigma^2) \tag{108}$$

$$p(y|x) = \mathcal{N}(y|x, \tau^2) \tag{109}$$

$$p(y) = \mathcal{N}(y|\mu, \sigma^2 + \tau^2) \tag{110}$$

2.14 Maximum likelihood parameter estimation

Given N iid datapoints $\vec{x}_i$ stored in rows of X, the log-likelihood is

$$\log p(X|\mu, \Sigma) = -\frac{Np}{2}\log(2\pi) - \frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(\vec{x}_i - \mu)^T\Sigma^{-1}(\vec{x}_i - \mu) \tag{111}$$

Below we drop the first term since it is a constant. Also, using the fact that

$$-\log|\Sigma| = \log|\Sigma^{-1}| \tag{112}$$

we can rewrite this as

$$\log p(X|\mu, \Sigma) = -\frac{Np}{2}\log(2\pi)\frac{N}{2}\log|\Lambda| - \frac{1}{2}\sum_{i=1}^{N}(\vec{x}_i - \mu)^T\Lambda(\vec{x}_i - \mu) \tag{113}$$

where $\Lambda = \Sigma^{-1}$ is called the **precision matrix**.

2.15 Mean

Using the following results for taking derivatives wrt vectors (where $\vec{a}$ is a vector and A is a matrix)

$$\frac{\partial(\vec{a}^T\vec{x})}{\partial\vec{x}} = \vec{a} \tag{114}$$

$$\frac{\partial(\vec{x}^T A\vec{x})}{\partial x} = (A + A^T)\vec{x} \tag{115}$$

and using the substitution $\vec{y}_i = \vec{x}_i - \mu$, we have

$$\frac{\partial}{\partial\mu}(\vec{x}_i - \mu)^T\Sigma^{-1}(\vec{x}_i - \mu) = \frac{\partial}{\partial y}\frac{\partial y}{\partial\mu}\vec{y}_i^T\Sigma^{-1}\vec{y}_i \tag{116}$$

$$= -1(\Sigma^{-1} + \Sigma^{-T})\vec{y}_i \tag{117}$$

Hence

$$\frac{\partial}{\partial\mu}\log p(X|\mu,\Sigma) = -\frac{1}{2}\sum_{i=1}^{N} -2\Sigma^{-1}(\vec{x}_i - \mu) \tag{118}$$

$$= \Sigma^{-1}\sum_{i=1}^{N}(\vec{x}_i - \mu) \tag{119}$$

$$= 0 \tag{120}$$

so

$$\mu_{ML} = \frac{1}{N}\sum_i \vec{x}_i \tag{121}$$

which is just the empirical mean.

2.16 Covariance

To compute Σ_{ML} is a little harder We will need to take derivatives wrt a matrix of a quadratic form and a determinant. We introduce the required algebra, since we will be using multivariate Gaussians a lot.

First, recall $\text{tr}(A) = \sum_i A_{ii}$ is the **trace** of a matrix (sum of the diagonal elements). This satisfies the **cyclic permutation property**

$$\text{tr}(ABC) = \text{tr}(CAB) = \text{tr}(BCA) \tag{122}$$

We can therefore derive the **trace trick**, which reorders the scalar inner product $x^T Ax$ as follows

$$x^T Ax = \text{tr}(x^T Ax) = \text{tr}(xx^T A) \tag{123}$$

Hence the log-likelihood becomes

$$\ell(\mathcal{D}|\Sigma,\hat{\mu}) = \frac{N}{2}\log|\Sigma^{-1}| - \tfrac{1}{2}\sum_i(x_i - \mu)^T\Sigma^{-1}(x_i - \mu) \tag{124}$$

$$= \frac{N}{2}\log|\Sigma^{-1}| - \tfrac{1}{2}\sum_i \text{tr}[(x_i - \mu)(x_i - \mu)^T\Sigma^{-1}] \tag{125}$$

$$= \frac{N}{2}\log|\Sigma^{-1}| - \tfrac{1}{2}\sum_i \text{tr}[S\Sigma^{-1}] \tag{126}$$

where S is the **scatter matrix**

$$S \stackrel{\text{def}}{=} \sum_i(\vec{x}_i - \overline{x})(\vec{x}_i - \overline{x})^T = (\sum_i \vec{x}_i\vec{x}_i^T) - N\overline{x}\overline{x}^T \tag{127}$$

We need to take derivatives of this expression wrt Σ^{-1}. We use the following results

$$\frac{\partial}{\partial A}\text{tr}(BA) = B^T \tag{128}$$

$$\frac{\partial}{\partial A}\log|A| = A^{-T} \tag{129}$$

Hence

$$\frac{\partial \ell(\mathcal{D}|\Sigma)}{\partial \Sigma^{-1}} = \frac{N}{2}\Sigma - \frac{1}{2}S \tag{130}$$

$$= 0 \tag{131}$$

yields

$$\Sigma_{ML} = \frac{1}{N}\sum_{i=1}^{N}(\vec{x}_i - \mu_{ML})(\vec{x}_i - \mu_{ML})^T = \frac{1}{N}S \tag{132}$$

$\sum_i \vec{x}_i$ and $\sum_i \vec{x}_i\vec{x}_i^T$ are called **sufficient statistics**, because if we know these, we do not need the original raw data X in order to estimate the parameters. In matlab, just type `Sigma = cov(X,1)`.

We can write the MLEs in matrix form as follows (where $\vec{1}$ is a vector of $N \times 1$ ones)

$$\mu_{ML} = \frac{1}{N}X^T\vec{1} \tag{133}$$

$$\Sigma_{ML} = \frac{1}{N}\ X^T(I - \vec{1}\vec{1}^T/N)X \tag{134}$$

where $(I - \vec{1}\vec{1}^T/N)$ is the centering matrix, since it removes the mean from any matrix of data it multiplies. If the data is already centered, then $\Sigma_{ML} = \frac{1}{N}X^TX$.

To find a MLE for a diagonal covariance matrix, we just compute $\hat{\Sigma}_{ML}$ as above and set the off-diagonal elements to 0. (We leave the proof of this fact as an exercise.) To find the MLE for a spherical covariance matrix, we set $\Sigma = \sigma^2 I$. Then the log-likelihood becomes

$$\log p(X|\Sigma) = -\frac{N}{2}p\log(\sigma^2) - \tfrac{1}{2}\sum_{i=1}^{N}(\sigma^2)^{-1}(\vec{x}_i - \vec{\mu})^T(\vec{x}_i - \vec{\mu})^T \tag{135}$$

Hence

$$0 = \frac{\partial}{\partial \sigma^2}\log p(X|\Sigma) \tag{136}$$

$$= -\frac{Np}{2\sigma^2} + \tfrac{1}{2}\sum_{i=1}^{N}(\sigma^2)^{-2}(\vec{x}_i - \vec{\mu})^T(\vec{x}_i - \vec{\mu})^T \tag{137}$$

$$\sigma^2_{ML} = \frac{1}{Np}\sum_{i=1}^{N}(\vec{x}_i - \vec{\mu})^T(\vec{x}_i - \vec{\mu}) \tag{138}$$

In the case $p = 1$, this reduces to the standard result

$$\sigma^2_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{139}$$

as required.

References

[Bis06] C. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[Jor06] M. I. Jordan. *An Introduction to Probabilistic Graphical Models*. 2006. In preparation.