CS 5135/6035 Learning Probabilistic Models

Lecture 8: MLE, Gradient Descent, Multivariate Gaussian

Gowtham Atluri

September 17, 2018

Reading Material

- Properties of MLE

https://ocw.mit.edu/courses/mathematics/18-443-statistics-for-applications-fall-2006/lecture-notes/lecture3.pdf

- Singer, Advanced Optimization, Lecture 9

https://people.seas.harvard.edu/~yaron/AM221-S16/lecture_notes/AM221_lecture9.pdf

- Jordan, Chapter 13. The Multivariate Gaussian

https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter13.pdf

Maximum Likelihood Estimation

- Review
- Properties of Estimators
  - Consistency
  - Bias
  - Variance

Maximum Likelihood Estimation - Recap

- I.I.D assumption
  - $x_1, x_2, \dots, x_n$ are i.i.d.
    - x_i's are *independently* sampled
    - Ever x_i is drawn from the *same* probability distribution
- Likelihood

$$p(x_1, \dots, x_n|\theta) = p(x_1|\theta)p(x_2|\theta)\dots p(x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta) = L(\theta|x)$$

- Log-likelihood

$$\ell(\theta) = logL(\theta)$$

- Maximization

$$\hat{\theta}_{MLE} = argmax_{\theta} p(D|\theta) = argmax_{\theta} \ell(\theta)$$

- Examples
  - Bernoulli (Discrete)
  - Gaussian (Continuous)

Properties of Estimators - Recap

Consistency

An estimator is consistent if the estimate $\hat{\theta}$ it constructs is guaranteed to converge to the true parameter value θ as the quantitiy of data to which it is applied increases.

Bias

The bias of an estimator η is defined as the deviation of the expectation of the estimate from the true value: $E[\hat{\theta}_\eta]$
When the sampling of data is viewed as a stochastic process, then the estimated parameter $\hat{\theta}_\eta$ cab be viewed as a continuous random variable.
When $E[\hat{\theta}_\eta] = \theta$ we say the estimator is unbiased.}

Variance (and efficiency)

$Var[\hat{\theta}_\eta]$
All else being equal, an estimator with smaller variance is preferable to one with greater variance.

MLE for univariate Gaussian variable

The temperatures, in Celsius, in Minneapolis during the first week of March 2018 are observed as $(-2.5, -9.9, -12.1, -8.9, -6.0, -4.8, 2.4)$
What is the distribution from which this data was generated (assuming it was Gaussian)?

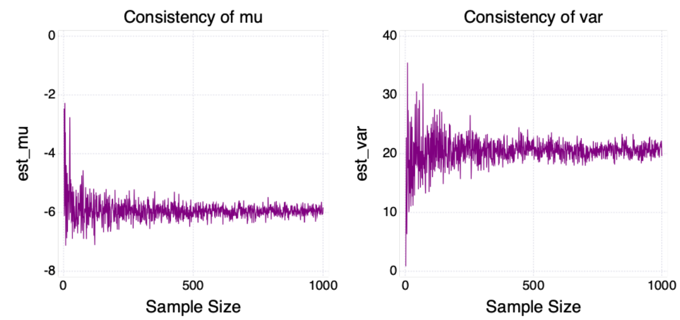
$$\hat{\mu} = \frac{\sum_{i=1}^{7} x_i}{7} = -5.97$$

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{7}(x_i - \mu)^2}{7} = 20.72$$

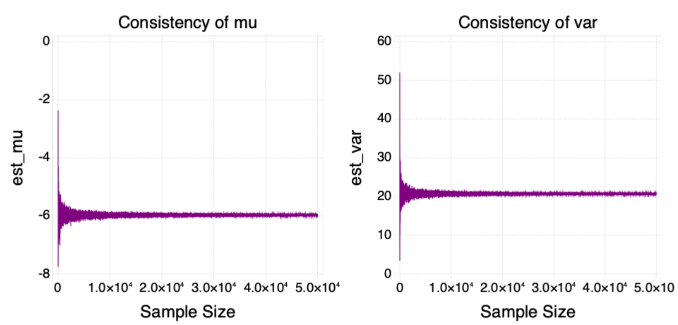
Estimating Consistency (Julia)

```
d = Normal(-5.97,sqrt(20.72));
sample_size = collect(2:1000);
est_mean = zeros(length(sample_size));
est_var = zeros(length(sample_size));
for i=1:length(sample_size)
  sample = rand(d,sample_size[i]);
  est_mean[i] = sum(sample)/length(sample);
  est_var[i] = sum((sample.-est_mean[i]).^2)/length(sample);
end
myplot1 = Gadfly.plot(x=sample_size,y=est_mean,Geom.line,
              Guide.xlabel("Sample Size"), Guide.ylabel("est_mu"),
              Guide.title("Consistency of mu"),white_panel);
myplot2 = Gadfly.plot(x=sample_size,y=est_var,Geom.line,
              Guide.xlabel("Sample Size"), Guide.ylabel("est_var"),
              Guide.title("Consistency of var"),white_panel);
final_plot = hstack(myplot1,myplot2);
draw(PNG("./figs/univ_normal_consistency.png", 10inch, 5inch), fina
```


Estimating Consistency (Julia)



Estimating Consistency (Julia)



Estimating Bias ($\hat{\mu}$)

Estimator of mean is unbiased if $E[\hat{\theta}_{MLE}] = \theta$

$$
\begin{aligned}
\hat{\mu} &= \frac{\sum_{i=1}^{n} x_i}{n} \\
E(\hat{\mu}) &= E[\frac{1}{n}\sum_{i=1}^{n} x_i] \\
&= \frac{1}{n}E[\sum_{i=1}^{n} x_i] \\
&= \frac{1}{n}.n.E[x] \\
&= \mu \qquad (\text{as } E(x) = \mu)
\end{aligned}
$$

The mean estimator is unbiased.

Estimating Bias ($\hat{\sigma}^2$)

Estimator of variance is unbiased if $E[\hat{\sigma}^2_{MLE}] = \sigma^2$

$$
\begin{aligned}
\hat{\sigma}^2 &= \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n} \\
E(\hat{\sigma}^2) &= E(\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}) \\
&= \ldots \\
&= \frac{n-1}{n}\sigma^2
\end{aligned}
$$

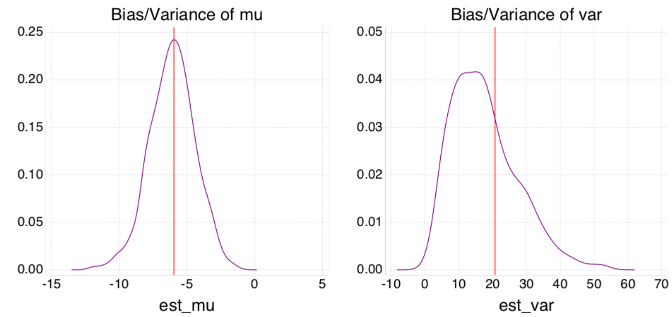
The variance estimator is biased.

Estimating Bias and Variance (Julia)

```
d = Normal(-5.97,sqrt(20.72));
est_mean = zeros(1000);
est_var = zeros(1000);
for i=1:1000
  sample = rand(d,7);
  est_mean[i] = sum(sample)/length(sample);
  est_var[i] = sum((sample.-est_mean[i]).^2)/length(sample);
end
myplot1 = Gadfly.plot(x=est_mean,Geom.density, Guide.xlabel("est_mu"
              xintercept=[-5.97], Geom.vline(color=colorant"red"),
              Guide.title("Bias/Variance of mu"),white_panel);
myplot2 = Gadfly.plot(x=est_var,Geom.density, Guide.xlabel("est_var"
              xintercept=[20.72], Geom.vline(color=colorant"red"),
              Guide.title("Bias/Variance of var"),white_panel);
final_plot = hstack(myplot1,myplot2);
#draw(PNG("./figs/univ_normal_bias_var_7.png", 10inch, 5inch), fina
```

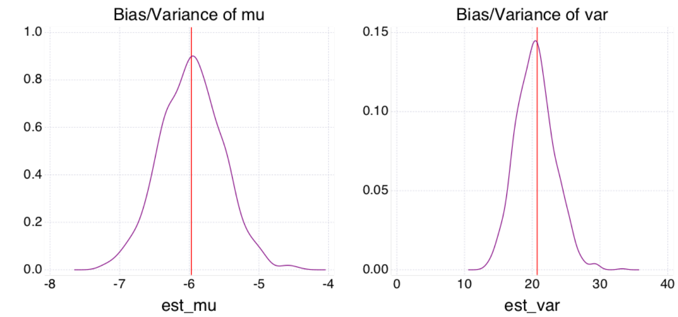
Estimating Bias and Variance (Julia)

$n = 7$



Estimating Bias and Variance (Julia)

$n = 100$

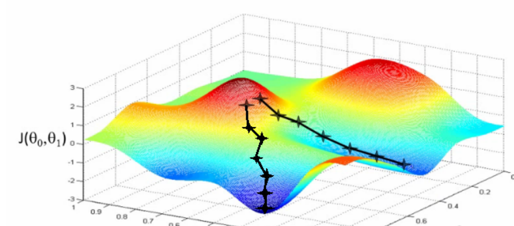


Gradient Descent Approach for MLE

- Overview
- General algorithm
- MLE for Gamma Distr.
- Julia code
- Limitations of Gradient Descent

Gradient Descent for MLE: Approach - II

- When approach I is not possible (particularly when the model involves many parameters and its PDF is highly non-linear), use gradient descent approach.
  - Use negative log-likelihood (also referred to as a cost function)
  - Randomly initialize and then incrementally update our weights by calculating the slope of our objective function
  - When applying the cost function, we want to continue updating our weights until the slope of the gradient gets as close to zero as possible.



Gradient Descent

- A numerical optimization technique used to find the parameter vector $\boldsymbol{w}$ that minimizes an objective function $E(\boldsymbol{w})$.

$$\boldsymbol{w}^* = argmin_{\boldsymbol{w}} E(\boldsymbol{w})$$

- An iterative approach to estimate $\boldsymbol{w}$
  - Starts with an initial estimate $\boldsymbol{w}_1$ (often a random vector)
  - Steepest descent from this point is to follow the negative gradient $-\nabla E$

$$\nabla E = [\frac{dE}{dw_1} \; \frac{dE}{dw_2} \; \cdots \; \frac{dE}{dw_n}]$$

  - Next estimate $\boldsymbol{w}_2$ is estimated as $\boldsymbol{w}_2 \leftarrow \boldsymbol{w}_1 - \lambda \nabla E|_{\boldsymbol{w}_1}$
  - Generally $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i-1} - \lambda \nabla E|_{\boldsymbol{w}_{i-1}}$
    - λ is the learning rate
  - Stops after a given maxIter or when estimate $\boldsymbol{w}$ converges

Gradient Descent: a general algorithm

Step 1: Pick initial value $\boldsymbol{w}_1$

Step 2: $maxIter = 10000$

Step 3: **for** $i = 2 : maxIter$

Step 4: $\quad \boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i-1} - \lambda \nabla E|_{\boldsymbol{w}_{i-1}}$

Step 5: $\quad$ **if** $|\boldsymbol{w}_i - \boldsymbol{w}_{i-1}| < \epsilon$ terminate; **end**

Step 6: **end for**

MLE for Gamma distribution

Probability density function of Gamma distribution is

$$\frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{\alpha-1} e^{-x/\beta}$$

where $\Gamma(\alpha)$ is the gamma function and (α, β) are parameters that take positive values.

Likelihood function

$$L(\theta|x) = \frac{1}{\Gamma(\alpha)^n \beta^{n\alpha}} (\prod_i x_i^{\alpha-1}) e^{-\sum_i x_i/\beta}$$

Log-Likelihood function

$$\ell(\theta) = -n \log \Gamma(\alpha) - n\alpha \log \beta + (\alpha - 1) \sum_i \log x_i - \frac{\sum_i x_i}{\beta}$$

Negative Log-Likelihood function

$$-\ell(\theta) = n \log \Gamma(\alpha) + n\alpha \log \beta - (\alpha - 1) \sum_i \log x_i + \frac{\sum_i x_i}{\beta}$$

MLE for Gamma distribution

Negative Log-Likelihood function

$$-\ell(\theta) = n \log \Gamma(\alpha) + n\alpha \log \beta - (\alpha - 1) \sum_i \log x_i + \frac{\sum_i x_i}{\beta}$$

Computing partial derivatives:

$$\frac{\partial \ell}{\partial \alpha} = n \frac{\partial}{\partial \alpha} \log \Gamma(\alpha) + n \log \beta - \sum_i \log x_i$$

$$\frac{\partial \ell}{\partial \beta} = n \frac{\alpha}{\beta} - \frac{\sum_i x_i}{\beta^2}$$

Gradient Descent update rules:

$$\alpha \leftarrow \alpha - \gamma \frac{\partial \ell}{\partial \alpha} \qquad \beta \leftarrow \beta - \gamma \frac{\partial \ell}{\partial \beta}$$

where γ is the learning rate.

MLE using Gradient Descent (Julia)

```
# generating 10 samples from a Gamma distribution
d = Gamma(5,5);
sample = rand(d,1000)
```

```
## 1000-element Array{Float64,1}:
##  22.5362
##  25.1435
##  45.6914
##  26.7435
##  37.8724
##  26.1113
##  21.9101
##  18.8474
##  16.8973
##  23.8151
##
##  18.313
##  16.7582
##   6.79626
```


MLE using Gradient Descent (Julia)

```
function dl_by_da(sample,a,b)
    n = length(sample);
    result = n*digamma(a) + n*log(b) - sum(log.(sample));
    return result;
end
```

```
## dl_by_da (generic function with 1 method)
```

```
function dl_by_db(sample,a,b)
    n = length(sample);
    result = (n*a/b) - (sum(sample)/(b^2));
    return result;
end
```

```
## dl_by_db (generic function with 1 method)
```


MLE using Gradient Descent (Julia)

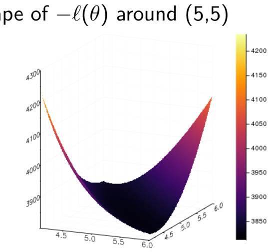
```
function gradient_descent_gamma(sample)
    n  = length(sample);
    max_itr = 1000; # maximum num. iterations
    gm = 0.01; # rate of learning
    a = rand()*10; # random initialization
    b = rand()*10;

    for i=1:max_itr
      a_new = a - gm*dl_by_da(sample,a,b);
      b_new = b - gm*dl_by_db(sample,a,b);
      if(a_new<0) a_new = rand()*10; end;
      if(b_new<0) b_new = rand()*10; end;
      if(abs(a_new-a)<0.0001 && abs(b_new-b)<0.0001) break; end;
      a = a_new; b = b_new;
    end
    return a,b;
end
```

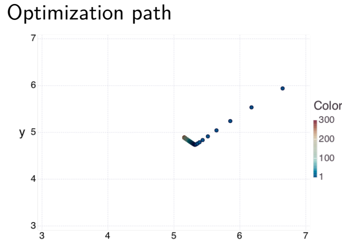
```
## gradient_descent_gamma (generic function with 1 method)
```


MLE using Gradient Descent (Julia)

Shape of $-\ell(\theta)$ around (5,5)



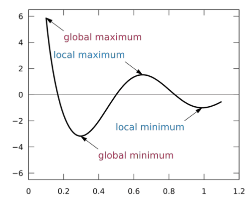
Optimization path



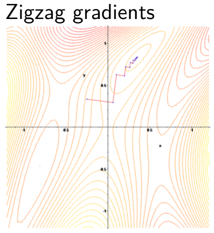
Gradient Descent: limitation

- Can converge to a local minimum
  - can result in a different value in different runs
- Tends to be slow when it is close to the minimum
- In poorly conditioned convex problems, 'zigzags' when gradients point nearly orthogonally to the shortest direction

Convergence to local minimum



Zigzag gradients



Multivariate Gaussian

- Functional form
- Covariance matrix
- Isocontours
- Multivariate Gaussian as a product of univariate distributions
- Properties of Multivariate Gaussian

Multivariate Gaussian

Univariate Gaussian

$$p(x|\mu,\sigma^2) = \mathcal{N}(x|\mu,\sigma^2) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\mu)^2/2\sigma^2}$$

Multivariate Gaussian

$$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \equiv \frac{1}{\sqrt{det(2\pi\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

Univariate

- exponent $-\frac{1}{2\sigma^2}(x-\mu)^2$
  - quadratic in x
  - negative sign
- coefficient in front $\frac{1}{\sqrt{2\pi\sigma^2}}$
  - normalization

Multivariate

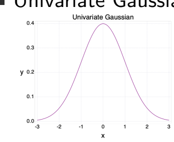
- $-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$
  - quadratic in x
  - negative sign
- $\frac{1}{\sqrt{det(2\pi\boldsymbol{\Sigma})}}$
  - normalization

Multivariate Gaussian

```
rand(Normal(0,1),4)
```

```
## 4-element Array{Float64,1}:
##   0.654719
##  -1.36141
##   0.830842
##   0.192311
```

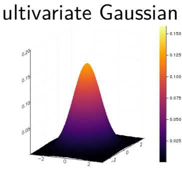
Univariate Gaussian




```
rand(MvNormal([0,0],eye(2)), 4)'
```

```
## 4×2 Array{Float64,2}:
##  -0.792111  -0.465018
##  -1.02169    0.512884
##  -1.10957    0.821939
##   0.271616  -1.05763
```

Multivariate Gaussian



Multivariate Gaussian

$$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \equiv \frac{1}{\sqrt{det(2\pi\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

where $\boldsymbol{\mu}$ is the mean vector of the distribution

and $\boldsymbol{\Sigma}$ is the covariance matrix.

Covariance Matrix:

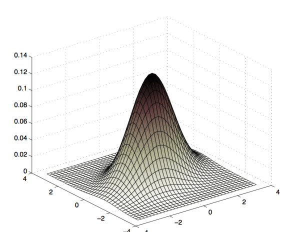
- For two random variables x, y,
  $Cov[x,y] = E[(x-E(x))(y-E(y))] = E[xy] - E[x]E[y]$
  - $\boldsymbol{\Sigma}_{ij} = Cov(x_i, x_j)$
- $\boldsymbol{\Sigma} \in \boldsymbol{S}^n{}_{++}$ (Positive Definite)

  $\boldsymbol{S}^n{}_{++} = \{A \in \boldsymbol{R}^{n\times n} : A = A^T \text{ and } x^TAx > 0 \ \forall x \in \boldsymbol{R}^n, \text{ such that } x \neq 0\}$
  - If all eigenvalues are positive, then the matrix is positive definite

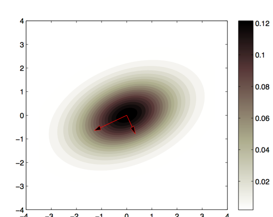
Isocontours

For a function $f: \boldsymbol{R}^2 \to R$, an isocontour is a set of the form

$$x \in \boldsymbol{R}^2 : f(x) = c, \quad \text{for some } c \in \boldsymbol{R}$$



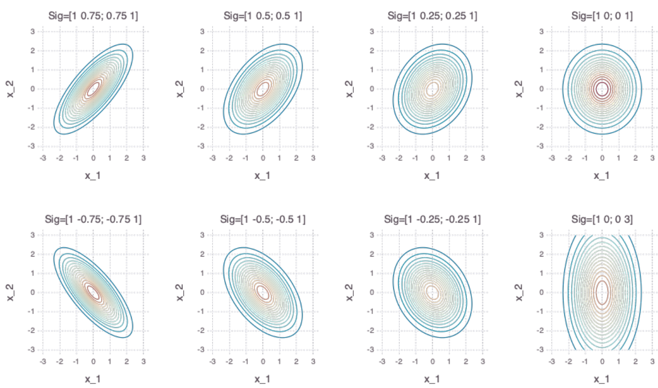
(a)



(b)

Mean $(0,0)$, Covariance $[1, 0.5; 0.5, 1.75]$

Multivariate Gaussian: Covariance matrix - Geometric view



Multivariate Gaussian - Geometric view

- Every real symmetric matrix $D \times D$ has an eigen-decomposition

$$\boldsymbol{\Sigma} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^{\boldsymbol{T}}$$

  where $\boldsymbol{E}^T\boldsymbol{E} = \boldsymbol{I}$ and $\boldsymbol{\Lambda} = diag(\lambda_1, \ldots, \lambda_D)$
- In the case of covariance matrix, all eigenvalues λ_i are positive.
- One can then use

$$\boldsymbol{y} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{E}^T(\boldsymbol{x} - \boldsymbol{\mu})$$

  so that

$$(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) = (\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{E}\boldsymbol{\Lambda}^{-1}\boldsymbol{E}^T(\boldsymbol{x} - \boldsymbol{\mu}) = \boldsymbol{y}^T\boldsymbol{y}$$

- The multivariate Gaussian reduces to a product of D univariate zero-mean unit variance Gaussians.

$$X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \iff X \sim \mu + \boldsymbol{E}\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Lambda}^{1/2}) \iff X \sim \boldsymbol{\mu} + \boldsymbol{E}\boldsymbol{\Lambda}^{1/2}\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

- We can view multivariate Gaissian as a shifted, scaled, and rotated version of 'standard' Gaussian in which the center is given by the mean, rotation by the eigen vectors and scaling by sqroot of

Product of Gaussians

Product of two Gaussians is another Gaussian with a multiplicative factor

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\frac{exp(-\frac{1}{2}(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})\boldsymbol{S}^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})}{\sqrt{det(2\pi\boldsymbol{S})}}$$

where $\boldsymbol{S} \equiv \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$ and the mean and covariance are given by

$$\mu = \boldsymbol{\Sigma}_1\boldsymbol{S}^{-1}\mu_2 + \boldsymbol{\Sigma}_2\boldsymbol{S}^{-1}\mu_1 \qquad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1\boldsymbol{S}^{-1}\boldsymbol{\Sigma}_2$$

Linear Transform of a Gaussian

Let $\boldsymbol{y}$ be linearly related to $\boldsymbol{x}$ through

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{\eta}$$

where $\boldsymbol{x} \perp\!\!\!\perp \boldsymbol{\eta}$, $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu_x}, \boldsymbol{\Sigma_x})$

Then marginal $p(\boldsymbol{y}) = \int_x p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$ is a Gaussian

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{M}\boldsymbol{\mu_x} + \boldsymbol{\mu}, \boldsymbol{M}\boldsymbol{\Sigma_x}\boldsymbol{M}^T + \boldsymbol{\Sigma})$$

Partitioned Gaussian

Consider a distribution $\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ defined jointly over two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of potentially different dimensions,

$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}$$

with corresponding mean and partitioned covariance

$$\mu = \begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{pmatrix}$$

where $\boldsymbol{\Sigma}_{yx} \equiv \boldsymbol{\Sigma}_{xy}^T$.

The marginal distribution is given by

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$$

and conditional

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y), \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx})$$