1 Multivariate Gaussians: A review

Multivariate Gaussian distributions crop up everywhere in machine learning, from priors on model parameters to assumptions on noise distributions. Being able to manipulate multivariate Gaussians also becomes important for analyzing correlations in data and preprocessing it for better regression and classification. We want to make sure to first cover the MVG fundamentals here.

Note that the probability density function of a non-degenerate (i.e. the covariance matrix is positive definite and, thus, invertible) multivariate Gaussian RV with mean vector, $\boldsymbol{\mu} \in \mathbb{R}^2$, and covariance matrix, $\Sigma \in \mathbb{R}^{2\times 2}$, is:

$$f(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^2\,|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{z}-\boldsymbol{\mu})\right)$$

(a) Consider a two dimensional, zero mean random variable $Z = \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}^\top \in \mathbb{R}^2$. In order for the random variable to be jointly Gaussian, a necessary and sufficient condition which we call the *first characterization* is that

- Z_1 and Z_2 are each marginally Gaussian, and
- $Z_1|Z_2 = z$ is Gaussian, and $Z_2|Z_1 = z$ is Gaussian.

A *second characterization* of a jointly Gaussian zero mean RV $Z \in \mathbb{R}^2$ is that it can be written as $Z = AX$, where $X \in \mathbb{R}^2$ is a collection of i.i.d. standard normal RVs and $A \in \mathbb{R}^{2\times 2}$ is a matrix.

Let X_1 and X_2 be i.i.d. standard normal RVs. Let U denote a binary random variable uniformly that is equal to 1 with probability $\frac{1}{2}$ and -1 with probability $\frac{1}{2}$, independent of everything else.

For each of the below subproblems, complete the following *two* steps: (1) Using one of the characterizations given above, determine whether the RVs are jointly Gaussian. If using the second characterization, clearly specify the A matrix. (2) Calculate the covariance matrix of Z (regardless of whether the RVs are jointly Gaussian or not).

(i.) $Z_1 = X_1$ and $Z_2 = X_2$.

(ii.) $Z_1 = X_1$ and $Z_2 = X_1 + 2X_2$. If using the first characterization, assume that you already know $(Z_1|Z_2 = z)$ is Gaussian.

(iii.) $Z_1 = X_1$ and $Z_2 = -X_1$.

(iv.) $Z_1 = X_1$ and $Z_2 = UX_1$.

Solution:

(i.) $Z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} X$, $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

(ii.) $Z = \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} X$, $\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}$.

(iii.) $Z = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} X$, $\Sigma = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$.

(iv.) Z_1, Z_2 are both marginally Gaussian. But $(Z_2|Z_1 = z) = \begin{cases} z & \frac{1}{2} \\ -z & \frac{1}{2} \end{cases}$ is not a Gaussian distribution.

$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

(b) Show that two Gaussian random variables can be uncorrelated, but not independent (*Hint: use one of the examples in part (a)*). On the other hand, show that two uncorrelated, jointly Gaussian RVs are independent.

Solution:

Uncorrelated but not independent : $Z_1 = X_1$ and $Z_2 = UX_1$.

Uncorrelated and independent : $Z_1 = X_1$ and $Z_2 = X_2$.

(c) With the setup in (a), let $Z = VX$, where $V \in \mathbb{R}^{2\times 2}$, and $Z, X \in \mathbb{R}^2$. What is the covariance matrix Σ_Z? If X is not a multivariate Gaussian but has the identity matrix $I \in \mathbb{R}^{2\times 2}$ as its covariance matrix, is your computed Σ_Z still the covariance of Z?

Solution:

(1) $\Sigma_Z = \mathbb{E}[(Z-\mu)(Z-\mu)^\top] = \mathbb{E}[ZZ^\top] - \mathbb{E}[\mu\mu^\top] = \mathbb{E}[VXX^\top V] = V\,\mathbb{E}[XX^\top]V^\top$.

(2) $\mathbb{E}[XX^\top] = \Sigma_X = I \implies \Sigma_Z = VV^\top$ is still the covariance of Z.

(d) Given a jointly Gaussian zero mean RV $Z = [Z_1\ Z_2]^\top \in \mathbb{R}^2$ with covariance matrix $\Sigma_Z = \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{12} & \Sigma_{22}\end{bmatrix}$, derive the conditional distribution of $(Z_1|Z_2 = z)$.

Hint: The following identity may be useful

$$\begin{bmatrix}a & b\\ b & c\end{bmatrix}^{-1} = \begin{bmatrix}1 & 0\\ -\frac{b}{c} & 1\end{bmatrix}\begin{bmatrix}\left(a-\frac{b^2}{c}\right)^{-1} & 0\\ 0 & \frac{1}{c}\end{bmatrix}\begin{bmatrix}1 & -\frac{b}{c}\\ 0 & 1\end{bmatrix}$$

Solution: Let $\Sigma_Z = \begin{bmatrix}a & b\\ b & c\end{bmatrix}$.

$$\begin{aligned}\begin{bmatrix}a & b\\ b & c\end{bmatrix} &= \begin{bmatrix}1 & \frac{b}{c}\\ 0 & 1\end{bmatrix}\begin{bmatrix}a-\frac{b^2}{c} & 0\\ 0 & c\end{bmatrix}\begin{bmatrix}1 & 0\\ \frac{b}{c} & 1\end{bmatrix}\\ &= \begin{bmatrix}1 & \frac{b}{c}\\ 0 & 1\end{bmatrix}\begin{bmatrix}\sqrt{a-\frac{b^2}{c}} & 0\\ 0 & \sqrt{c}\end{bmatrix}^2\begin{bmatrix}1 & 0\\ \frac{b}{c} & 1\end{bmatrix}.\end{aligned}$$

Let $Z = VX$, $X = [X_1, X_2]^\top$, $\Sigma_Z = VV^\top$. So,

$$V = \begin{bmatrix}1 & \frac{b}{c}\\ 0 & 1\end{bmatrix}\begin{bmatrix}\sqrt{a-\frac{b^2}{c}} & 0\\ 0 & \sqrt{c}\end{bmatrix} = \begin{bmatrix}\sqrt{a-\frac{b^2}{c}} & \sqrt{\frac{b^2}{c}}\\ 0 & \sqrt{c}\end{bmatrix}.$$

$$\begin{cases}Z_1 &= \sqrt{a-\frac{b^2}{c}}X_1 + \frac{b}{c}\sqrt{c}X_2\\ Z_2 &= \sqrt{c}X_2\end{cases}$$

$(Z_1|Z_2 = z) = \sqrt{a-\frac{b^2}{c}}X_1 + \frac{b}{c}z$. So $(Z_1|Z_2 = z) \sim N(\frac{\Sigma_{12}}{\Sigma_{22}}z, \Sigma_{11} - \frac{\Sigma_{12}^2}{\Sigma 22})$.

2 Projections and Linear Regression

We are given $X \in \mathbb{R}^{n \times d}$ where $n > d$ and $\text{rank}(X) = d$. We are also given a vector $y \in \mathbb{R}^n$. Define the orthogonal projection of y onto $\text{range}(X)$ as $P_{\text{range}(X)}(y)$.

Background on orthogonal projections For any finite-dimensional subspace W (here, $\text{range}(X)$) of a vector space V (here, $\mathbb{R}^n$), any vector $v \in V$ can be decomposed as

$$v = w + u, \qquad w \in W, \quad u \in W^\perp,$$

where $W^\perp$ is the orthogonal complement of W. Furthermore, this decomposition is unique: if $v = w' + u'$ where $w' \in W$, $u' \in W^\perp$, then $w' = w$ and $u' = u$. These two facts allow us to define P_W, the orthogonal projection operator onto W. Given a vector v with decomposition $v = w + u$, we define

$$P_W(v) = w.$$

It can also be shown using these two facts that P_W is linear. For more information on orthogonal projections, see `https://gwthomas.github.io/docs/math4ml.pdf`.

(a) Prove that $P_{\text{range}(X)}(y) = \underset{w \in \text{range}(X)}{\arg\min} \|y - w\|_2^2$.

Solution:

Let $y = u + v, \qquad u \in W, \quad v \in W^\perp$.

$$\begin{aligned}
\underset{w \in \text{range}(X)}{\arg\min} \|y - w\|_2^2 &= \underset{w \in \text{range}(X)}{\arg\min} \|u + v - w\|_2^2 \\
&= \underset{w \in \text{range}(X)}{\arg\min} \|u - w + v\|_2^2 \\
&= \underset{w \in \text{range}(X)}{\arg\min} \|u - w\|_2^2 + \|v\|_2^2 + 2(u - w)^\top v \qquad = \underset{w \in \text{range}(X)}{\arg\min} \|u - w\|_2^2.
\end{aligned}$$

So $P_{\text{range}(X)}(y) = \underset{w \in \text{range}(X)}{\arg\min} \|y - w\|_2^2$.

(b) An orthogonal projection is a linear transformation. That is, $P_{\text{range}(X)}(y) = Py$ for some projection matrix P. Specifically, given $1 \leq d \leq n$, a matrix $P \in \mathbb{R}^{n \times n}$ is said to be a rank-d orthogonal projection matrix if

- $\text{rank}(P) = d$
- $P = P^T$
- $P^2 = P$.

Prove that P is a rank-d projection matrix if and only if there exists a $U \in \mathbb{R}^{n \times d}$ such that $P = UU^\top$ and $U^\top U = I$.

Hint Use the eigendecomposition of P to prove the forward direction.

Solution:

(i) Prove a matrix $P \in \mathbb{R}^{n \times n}$ satisfied :

- $\text{rank}(P) = d$
- $P = P^T$
- $P^2 = P$

Then there exists a $U \in \mathbb{R}^{n \times d}$ such that $P = UU^\top$ and $U^\top U = I$.

$P = Q\Lambda Q^{-1}$, $\Lambda^2 = \Lambda$, so $\Lambda = \begin{bmatrix} I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{n-d} \end{bmatrix}$.

Let $Q = [U|V]$, $U \in \mathbb{R}^{n \times d}$, $V \in \mathbb{R}^{n \times (n-d)}$, $P = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{n-d} \end{bmatrix} \begin{bmatrix} U^T \\ V^T \end{bmatrix} = UU^\top$.

$\begin{bmatrix} U^T \\ V^T \end{bmatrix} \begin{bmatrix} U & V \end{bmatrix} = I_n$, so $U^\top U = I$.

(ii) Prove a matrix $P \in \mathbb{R}^{n \times n}$ and there exists a $U \in \mathbb{R}^{n \times d}$ such that $P = UU^\top$ and $U^\top U = I$, satisfied :

- $\text{rank}(P) = d$
- $P = P^T$
- $P^2 = P$

(1) $P^\top = (UU^\top)^\top = P$.

(2) $\text{rank}(U) = d$, $\text{rank}(P) = d$.

(3) $P^2 = (UU^\top)(UU^\top) = UIU^\top = UU^\top = P$.

(c) The Singular Value Decomposition theorem states that we can write any matrix X as

$$X = \sum_{i=1}^{\min\{n,d\}} \sigma_i u_i v_i^\top = \sum_{i:\sigma_i>0} \sigma_i u_i v_i^\top$$

where $\sigma_i \geq 0$, and $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^d$ are orthonormal bases for $\mathbb{R}^n$ and $\mathbb{R}^d$ respectively. Some of the singular values σ_i may equal 0, indicating that the associated left and right singular vectors u_i and v_i do not contribute to the sum, but sometimes it is still convenient to include them in the SVD so we have complete orthonormal bases for $\mathbb{R}^n$ and $\mathbb{R}^d$ to work with. Show that

(i) $\{u_i : \sigma_i > 0\}$ is an orthonormal basis for the columnspace of X

(ii) Similarly, $\{v_i : \sigma_i > 0\}$ is an orthonormal basis for the row space of X
Hint: consider $X^\top$.

Solution:

$X = U\Sigma V^\top$. Let $k = \text{rank}(X)$.

For $y \in \mathbb{R}^d$, $Xy = U\Sigma V^\top y$, let $a = V^\top y$. So $\text{Col}(X) = \{U\Sigma a \mid a \in \mathbb{R}^d\}$.

$U = \begin{bmatrix} A & B \end{bmatrix}$, $A \in \mathbb{R}^{n\times k}$, $B \in \mathbb{R}^{n\times(n-k)}$. $\Sigma a = \begin{bmatrix} \sigma_1 a_1 \\ \sigma_2 a_2 \\ \cdots \\ \sigma_k a_k \\ 0 \\ \cdots \\ 0 \end{bmatrix}$.

So $\text{Col}(X) = Ab$, $b \in \mathbb{R}^k$. Therefore, $\{u_i : \sigma_i > 0\}$ is an orthonormal basis for the columnspace of X.

Similarly, $\text{Row}(X) = \text{Col}(X^\top)$, so $\{v_i : \sigma_i > 0\}$ is an orthonormal basis for the row space of X.

(d) Let $X \in \mathbb{R}^{n\times d}$ such that $\text{rank}(X) = d$. Prove that $X(X^TX)^{-1}X^T$ is a rank-d orthogonal projection matrix.
Hint: Consider the SVD decomposition of X.

Solution:

$X(X^\top X)^{-1}X^\top = U\Sigma V^\top((U\Sigma V^\top)^\top U\Sigma V^\top)^{-1}(U\Sigma V^\top)^\top = U\Sigma V^\top(V\Sigma^\top\Sigma V^\top)^{-1}V\Sigma^\top U^\top$.

$\Sigma^\top\Sigma$ is a $d \times d$ full rank diagonal matrix.

$A = X(X^\top X)^{-1}X^\top = U\Sigma(\Sigma^\top\Sigma)^{-1}\Sigma^\top U^\top$

Let $\Sigma = \begin{bmatrix} \hat{\Sigma} \\ 0 \end{bmatrix}$, $U = \begin{bmatrix} \hat{U_1} & \hat{U_2} \end{bmatrix}$ $A = U\hat{\Sigma}(\hat{\Sigma})^{-2}\hat{\Sigma}^\top U^\top = \hat{U_1}\hat{U_1}^\top$.

(e) Prove that $X(X^TX)^{-1}X^T$ is a projection onto range(X).

Solution: $A = X(X^TX)^{-1}X^T$, according to (e) and (b), $A^2 = A$. so A is a projection onto range(X).

(f) Show that $w^* = (X^T X)^{-1} X^T y$ is the solution to the optimization problem

$$\arg\min_w \|y - Xw\|_2^2$$

using only facts proved in this problem.

Solution: $Xw^* = P_{\text{range}(X)}(y) = X(X^T X)^{-1} X^T y$, so $w^* = (X^T X)^{-1} X^T y$.

3 Some MLEs

For this question, assume you observe n (data point, label) pairs $(x_i, y_i)_{i=1}^n$, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for all $i = 1, \ldots, n$. We denote X as the data matrix containing all the data points and y as the label vector containing all the labels:

$$X = \begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix} \in \mathbb{R}^{n \times d}, \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n.$$

(a) Ignoring y for now, suppose we model the data points as coming from a d-dimensional Gaussian with diagonal covariance:

$$\forall i = 1, \ldots, n, \quad x_i \overset{i.i.d.}{\sim} N(\mu, \Sigma); \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_d^2 \end{bmatrix}.$$

If we consider $\mu \in \mathbb{R}^d$ and $(\sigma_1^2, \ldots, \sigma_d^2)$, where each $\sigma_i^2 > 0$, to be unknown, the parameter space here is $2d$-dimensional. When we refer to Σ as a parameter, we are referring to the d-tuple $(\sigma_1^2, \ldots, \sigma_d^2)$, but inside a linear algebraic expression, Σ denotes the diagonal matrix $\text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$.

Solve the following problems:

(i) Prove that log-likelihood $\ell(\mu, \Sigma) = \log p(X \mid \mu, \Sigma)$ is equal to

$$-\frac{n}{2}\left(d\log(2\pi) - \sum_{j=1}^{d} \log\left(\frac{1}{\sigma_j^2}\right)\right) - \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^T\Sigma^{-1}(x_i - \mu).$$

(ii) Find the MLE of μ assuming Σ is known.

(iii) Find the MLE of Σ assuming μ is known.
Hint: you can re-parameterize σ_j^2 by defining $v_j = \frac{1}{\sigma_j^2}$

(iv) Find the joint MLE of (μ, Σ) in terms of the maximum likelihood estimates computed above.

Solution:

(i) $|\Sigma| = \prod_{i=1}^d \sigma_i^2$,

$$\begin{aligned}\ell(\mu, \Sigma) = \log p(X \mid \mu, \Sigma) &= \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}(x_i - \mu)^\top\Sigma^{-1}(x_i - \mu)\right)\right) \\ &= -\frac{n}{2}\left(d\log(2\pi) - \sum_{j=1}^{d} \log\left(\frac{1}{\sigma_j^2}\right)\right) - \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^T\Sigma^{-1}(x_i - \mu).\end{aligned}$$

(ii)

$$\begin{aligned}\mu^* &= \arg\max_{\mu} \ell(\mu, \Sigma) \\ &= \arg\min_{\mu} \sum_{i=1}^{n}(x_i - \mu)^T\Sigma^{-1}(x_i - \mu)\end{aligned}$$

Let $f_i(\mu) = (x_i - \mu)^T \Sigma^{-1}(x_i - \mu)$, $\nabla_\mu f_i(\mu) = 2\Sigma^{-1}(\mu - x_i)$. So

$$\begin{aligned}\nabla_\mu \sum_{i=1}^{n} f_i(\mu) &= \sum_{i=1}^{n} 2\Sigma^{-1}(\mu - x_i) \\ &= 2\Sigma^{-1}(n\mu - \sum_{i=1}^{n} x_i)\end{aligned}$$

.

Therefore, $\mu^* = \frac{1}{n}\sum_{i=1}^{n} x_i$.

(iii)

$$\begin{aligned}\Sigma^* &= \arg\max_{\Sigma} \ell(\mu, \Sigma) \\ &= \arg\min_{\mu} \sum_{i=1}^{n} (x_i - \mu)^T \Sigma^{-1}(x_i - \mu) - n\sum_{j=1}^{d} \log\left(\frac{1}{\sigma_j^2}\right) \\ &= \arg\min_{\mu} \sum_{i=1}^{n} (x_i - \mu)^T V(x_i - \mu) - n\sum_{j=1}^{d} \log\left(v_j\right).\end{aligned}$$

$$\frac{\partial\left(\sum_{i=1}^{n}(x_i-\mu)^T V(x_i-\mu) - n\sum_{j=1}^{d}\log\left(v_j\right)\right)}{\partial v_k} = (x_{ik} - \mu_k)^2 - \frac{n}{v_k}$$

$(\sigma_k^*)^2 = \frac{(x_{ik} - \mu_k)^2}{n}$.

$\Sigma^* = \text{diag}\left(\frac{(x_{i1} - \mu_1)^2}{n}, \frac{(x_{i2} - \mu_2)^2}{n}, \dots\right)$.

(iv) $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\hat{\Sigma}^* = \text{diag}\left(\frac{(x_{i1} - \hat{\mu}_1)^2}{n}, \frac{(x_{i2} - \hat{\mu}_2)^2}{n}, \dots\right)$.

(b) Suppose that we have a training set $\{(x_i, y_i) \mid i = 1 \dots n\}$ of n independent examples but in which the residual terms had different variances. That is, we assume

$$y_i \sim N(w^T x_i, \sigma_i^2).$$

Show that the MLE estimate of w can be found by solving the following optimization problem

$$w_{\text{MLE}} = \arg\min_w \|A(Xw - y)\|_2^2.$$

Clearly state what the matrix A equals.

Solution:

$$L(w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y_i - w^\top x_i)^2}{2\sigma^2})$$

$$\begin{aligned} w_{\text{MLE}} = \arg\max_w L(w) &= \arg\max_w \log L(w) \\ &= \arg\max_w \sum_{i=1}^{n} \frac{-(y_i - w^\top x_i)^2}{\sigma_i^2} \\ &= \arg\min_w \sum_{i=1}^{n} \frac{(y_i - w^\top x_i)^2}{\sigma_i^2} \\ &= \arg\min_w \sum_{i=1}^{n} \frac{(y - Xw)_i^2}{\sigma_i^2} \\ &= \arg\min_w \sum_{i=1}^{n} \left(\frac{(Xw - y)_i}{\sigma_i}\right)^2 \end{aligned}$$

$$A = \text{diag}(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_n})$$

(c) Consider the Categorical$(\theta_1, \theta_2, \ldots, \theta_k)$ distribution. Recall, for categorical distributions, there are two constraints on θ_k:

- $\theta_k \geq 0$ for all k
- $\sum_{k=1}^{K} \theta_k = 1$

The distribution describes a random process that selects one of the K possible categories, with category k being chosen with probability θ_k.

Ignoring the data points X, suppose that for all i from 1 to n, we sample y_i from a categorical distribution:

$$y_i \overset{i.i.d.}{\sim} \text{Categorical}(\theta_1, \ldots, \theta_K).$$

Compute the MLE of $\theta = (\theta_1, \ldots, \theta_K)$. Use the fact that the KL divergence is nonnegative:

$$\text{KL}(\pi \,\|\, \theta) = \sum_{\omega \in \Omega} \pi(\omega) \log \left(\frac{\pi(\omega)}{\theta(\omega)} \right) \geq 0.$$

Solution:

$$\begin{aligned}
\theta_{\text{MLE}} &= \arg\max_{\theta \in P} \prod_{i=1}^{n} \theta_{y_i} = \arg\max_{\theta \in P} \sum_{i=1}^{n} \log \theta_{y_i} \\
&= \arg\max_{\theta \in P} \sum_{\omega \in \Omega} \sum_{i=1}^{n} [\omega = y_i] \log \theta_\omega \\
&= \arg\max_{\theta \in P} \sum_{\omega \in \Omega} \text{count}_\omega \log \theta_\omega \\
&= \arg\max_{\theta \in P} \sum_{\omega \in \Omega} \frac{\text{count}_\omega}{n} \log \theta_\omega \\
&\sum_{\omega \in \Omega} \frac{\text{count}_\omega}{n} \log \frac{\frac{\text{count}_\omega}{n}}{\theta_\omega} \geq 0 \\
&\sum_{\omega \in \Omega} \frac{\text{count}_\omega}{n} \log \theta_\omega \leq \sum_{\omega \in \Omega} \frac{\text{count}_\omega}{n} \log \frac{\text{count}_\omega}{n} \\
\theta_{\text{MLE}} &= \left(\frac{\text{count}_1}{n}, \frac{\text{count}_2}{n}, \ldots, \frac{\text{count}_K}{n} \right)
\end{aligned}$$

(d) Again consider X fixed. This time, we suppose that each y_i is binary-valued (0 or 1). We choose to model y as

$$y_i \overset{ind.}{\sim} \text{Ber}(s(x_i^\top w)) \quad \forall i = 1, \ldots, n,$$

where $s(z) = \frac{1}{1+e^{-z}}$ is the *sigmoid* function and $\text{Ber}(p)$ denotes the Bernoulli distribution which takes value 1 with probability p and 0 with probability $1 - p$.

(i) Write down the log-likelihood $\ell(w) = \log p(y \mid w)$ and show that finding the MLE of w is equivalent to minimizing the cross entropy between $\text{Ber}(y_i)$ and $\text{Ber}(s(x_i^\top w))$ for each i:

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} H(\text{Ber}(y_i), \text{Ber}(s(x_i^\top w))). \tag{1}$$

Definition of cross entropy: given two discrete probability distributions $\pi : \Omega \to [0, 1]$ and $\theta : \Omega \to [0, 1]$ on some outcome space Ω, we define the cross entropy between π and θ as

$$H(\pi, \theta) = \sum_{\omega \in \Omega} -\pi(\omega) \log \theta(\omega).$$

Solution: TODO

(ii) Show that (1) (and therefore finding the MLE) is equivalent to the following problem:

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \log(1 + \exp(-z_i x_i^\top w)) \tag{2}$$

where $z_i = 1$ if $y_i = 1$ and $z_i = -1$ if $y_i = 0$.
Note: both (1) and (2) are referred to as logistic regression.

Solution: TODO

(iii) Let $J(w) = \log(1+\exp(-zx^\top w))$ where, again, $z = 1$ if $y = 1$ and $z = -1$ if $y = 0$ (we are only considering a single (x, y) pair in this subpart). Prove the following:

i. J is not strictly convex.
Hint: A necessary condition for a twice-differentiable function to be strictly convex is that its Hessian is positive definite.

ii. The gradient descent update rule for minimizing $J(w)$ with learning rate ϵ is

$$w' = w - \epsilon \left(\frac{1}{1 + e^{-x^T w}} - y \right) x$$

Solution: TODO

4 Geometry of Ridge Regression

You recently learned ridge regression and how it differs from ordinary least squares. In this question we will explore the properties of ridge regression in more depth. Recall that the ridge regression problem is given by the following optimization problem:

$$\min_{w} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \nu\|\mathbf{w}\|_2^2. \tag{3}$$

The solution to ridge regression is given by

$$\hat{\mathbf{w}}_\mathbf{r} = (\mathbf{X}^\top\mathbf{X} + \nu\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}. \tag{4}$$

1. One reason why we might want to have small weights $\mathbf{w}$ has to do with the sensitivity of the predictor to its input. Let $\mathbf{x}$ be a d-dimensional list of features corresponding to a new test point. Our predictor is $\mathbf{w}^\top\mathbf{x}$. What is an upper bound on how much our prediction could change if we added noise $\boldsymbol{\epsilon} \in \mathbb{R}^d$ to a test point's features $\mathbf{x}$, in terms of $\|\mathbf{w}\|_2$ and $\|\boldsymbol{\epsilon}\|_2$?
   Hint: Use the Cauchy-Schwarz inequality.

   Solution: TODO

2. Note that in computing $\hat{\mathbf{w}}_\mathbf{r}$, we are trying to invert the matrix $\mathbf{X}^\top\mathbf{X} + \nu\mathbf{I}$ instead of the matrix $\mathbf{X}^\top\mathbf{X}$. If $\mathbf{X}^\top\mathbf{X}$ has eigenvalues $\sigma_1^2, \ldots, \sigma_d^2$, what are the eigenvalues of $(\mathbf{X}^\top\mathbf{X} + \nu\mathbf{I})^{-1}$? Comment on why adding the regularizer term $\nu\mathbf{I}$ can improve the inversion operation numerically.

   Solution: TODO

3. Let the number of parameters $d = 4$ and the number of datapoints $n = 6$, and let the eigenvalues of $\mathbf{X}^\top \mathbf{X}$ be given by 500, 10, 1, and 0.001. We must now choose between two regularization parameters $\nu_1 = 50$ and $\nu_2 = 0.1$. Which do you think is a better choice for this problem and why?

   Solution: TODO

4. Another advantage of ridge regression can be seen for under-determined systems. Say we have the data drawn from a $d = 5$ parameter model, but only have $n = 4$ training samples of it, i.e. $\mathbf{X} \in \mathbb{R}^{4\times 5}$. Now this is clearly an underdetermined system, since $n < d$. Show that ridge regression with $\nu > 0$ results in a unique solution, whereas ordinary least squares has an infinite number of solutions.

   Hint: To make this point, it may be helpful to consider $\mathbf{w} = \mathbf{w}_0 + \mathbf{w}^*$ where $\mathbf{w}_0$ is in the null space of $\mathbf{X}$ and $\mathbf{w}^*$ is a solution.

   Solution: TODO

5. What will the solution to ridge regression (4) converge to if you take the limit $\nu \to 0$? Your answer should be a simple expression in terms of $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}, \mathbf{y}$, and ν where $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD of $\mathbf{X}$.

   Solution: TODO

6. Tikhonov regularization is a general term for ridge regression, where the implicit constraint set takes the form of an ellipsoid instead of a ball. In other words, we solve the optimization problem

$$\mathbf{w} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \nu\|\mathbf{\Gamma}\mathbf{w}\|_2^2$$

for some full rank matrix $\mathbf{\Gamma} \in \mathbb{R}^{d\times d}$. Derive a closed form solution for $\mathbf{w}$.

Solution: TODO

5 Robotic Learning of Controls from Demonstrations and Images

Huey, a home robot, is learning to retrieve objects from a cupboard. The goal is to push obstacle objects out of the way to expose a goal object. Huey's robot trainer, Anne, provides demonstrations via tele-operation. When tele-operating the robot, Anne can look at the images captured by the robot and provide controls to Huey remotely.

During a demonstration, Huey records the RGB images of the scene for each of the n timesteps, $x_1, ..., x_n$, where $x_i \in \mathbb{R}^{30\times30\times3}$ and the controls for his body for each of the n timesteps, $u_1, \ldots, u_n$, where $u_i \in \mathbb{R}^3$. The controls correspond to making small changes in the 3D pose (i.e. translation and rotation) of his body. Examples of the data are shown in the figure.

Under an assumption (sometimes called the Markovian assumption) that all that matters for the current control is the current image, Huey can try to learn a linear *policy* π (where $\pi \in \mathbb{R}^{2700\times3}$) which linearly maps image states to controls (i.e. $\pi^\top x = u$). We will now explore how Huey can recover this policy using linear regression.

Note the dimensions in this problem! Previously, you saw linear regression in problems in which the learned weight w^* was a vector and the predicted value y was a scalar. Here, we are predicting 3D controls. This means that the learned policy is a matrix. In essence, we are performing 3 regressions at the same time, one for each element of the predicted control u.

Please stick to **numpy** (and **numpy.linalg**) only for performing any computations in this assignment. We will ask that you edit the file `robotic_ridge_code.py` directly, instead of working in a Python notebook, and submit it to the Gradescope autograder after you are finished. Please don't rename the file, or change any of the function signatures!

(a) To get familiar with the structure of the data, **please visualize the 0th, 10th and 20th images in the training dataset. Also find their corresponding control vectors.**
Note: the training and testing images are currently stored as float32 numpy arrays, with pixel values in the range [0.0, 255.0]. You may have to convert to these images to the np.uint8 format to visualize them.

Solution: TODO

(b) Load the n training examples from `x_train.p` and compose the matrix X, where $X \in \mathbb{R}^{n \times 2700}$. Note that you will need to flatten the images and reduce them to a single vector. The flattened image vector will be denoted by $\bar{x}$ (where $\bar{x} \in \mathbb{R}^{2700 \times 1}$). Next, load the n examples from `y_train.p` and compose the matrix U, where $U \in \mathbb{R}^{n \times 3}$. Try to perform ordinary least squares by forming the matrix $(X^\top X)^{-1} X^\top$ for solving

$$\min_{\pi} \|X\pi - U\|_F$$

in order to learn the optimal *policy* $\pi^* \in \mathbb{R}^{2700 \times 3}$. **Report what happens as you attempt to do this and explain why**.

Solution: TODO

(c) Now try to perform ridge regression:

$$\min_{\pi} \|X\pi - U\|_F^2 + \lambda\|\pi\|_F^2$$

on the dataset for regularization values $\lambda = \{0.1, 1.0, 10, 100, 1000\}$. Measure the average squared Euclidean distance for the accuracy of the policy on the training data:

$$\frac{1}{n}\sum_{i=0}^{n-1} \|\bar{x}_i^T\pi - u_i^\top\|_2^2$$

In the expression above, we are taking the ℓ_2 norm of a row vector, which here we take to mean the ℓ_2 norm of the column vector we get by transposing it. **Report the training error results for each value of λ.**

Solution: TODO

(d) Next, we are going to try standardizing the states. For each pixel value in each data point, $\bar{x}$, perform the following operation:

$$\bar{x} \mapsto \frac{\bar{x}}{255} \times 2 - 1$$

We know that the maximum pixel value is 255, so this operation rescales the data to be in the range $[-1, 1]$. **Repeat the previous part and report the average squared training error for each value of λ.**

Solution: TODO

(e) Evaluate both *policies* (i.e. with and without standardization) on the new validation data `x_test.p` and `y_test.p` for the different values of λ. **Report the average squared Euclidean loss and qualitatively explain how changing the values of λ affects the performance in terms of bias and variance**.

Solution: TODO

(f) To better understand how standardizing improved the loss function, we are going to evaluate the *condition number* κ of the optimization problem above, which is defined as

$$\kappa = \frac{\sigma_{\max}(X^T X + \lambda I)}{\sigma_{\min}(X^T X + \lambda I)}$$

or the ratio of the maximum singular value to the minimum singular value of the relevant matrix. Roughly speaking, the condition number of the optimization process measures how stable the solution will be when some error exists in the observations. More precisely, given a linear system $Ax = b$, the condition number of the matrix A is the maximum ratio of the relative error in the solution x to the relative error of b.

For the regularization value of $\lambda = 100$, **report the condition number with the standardization technique applied and without**.

Solution: TODO

6 Honor Code

1. **List all collaborators. If you worked alone, then you must explicitly state so.**

   Solution: TODO

2. **Declare and sign the following statement**:

   "I certify that all solutions in this document are entirely my own and that I have not looked at anyone else's solution. I have given credit to all external sources I consulted."

   Signature : _____

   While discussions are encouraged, *everything* in your solution must be your (and only your) creation. Furthermore, all external material (i.e., *anything* outside lectures and assigned readings, including figures and pictures) should be cited properly. We wish to remind you that the consequences of academic misconduct are *particularly severe*!

   Solution: TODO