# Project 1: Color Segmentation

*Collaboration in the sense of discussion is allowed, however, the work you turn in should be your own - you should not split parts of the assignments with other students and you should certainly not copy other students' code or papers. See the collaboration and academic integrity statement here:* https://natanaso. github.io/ece276a. *Books may be consulted but not copied from.*

## Submission

You should submit the following two files by the deadline shown on the top right corner.

1. **FirstnameLastname_P1.pdf** on **Gradescope**: upload your solutions to the theoretical problems (Problem 1-3). You may use latex, scanned handwritten notes (write legibly!), or any other method to prepare a pdf file. Do not just write the final result. Present your work in detail explaining your approach at every step. Also, attach to the **same** pdf the report for Problem 5. You are encouraged but not required to use an IEEE conference template[1] for your report.

2. **FirstnameLastname_P1.zip** on **TritonEd**: upload all code you have written for Problem 4 (do not include the training and test datasets) and a README file with clear instructions for running it. Please try to generate an executable file from your code following the instructions online[2]. While generating an executable is **not required**, it will simplify the task of running your code with different library versions across different platforms.

## Problems

In square brackets are the points assigned to each problem.

1. [2 pts] Suppose $\mathcal{U}$ and $\mathcal{V}$ are independent random variables, each distributed uniformly in the unit interval.

   (a) Determine the distributions of the random variables $R = \sqrt{-2\log(1 - \mathcal{U})}$ and $\Theta = 2\pi\mathcal{V}$.

   (b) Determine the joint distribution of $X = R\cos\Theta$ and $Y = R\sin\Theta$.

2. [2 pts] Consider two color classes $\{1, 2\}$ and associated feature vectors $x \in \mathbb{R}^m$. Let the feature distribution conditioned on the class label $Y$ be Gaussian, i.e., $X|Y = 1 \sim \mathcal{N}(\mu_1, \sigma^2 I_m)$ and $X|Y = 2 \sim \mathcal{N}(\mu_2, \sigma^2 I_m)$.

   (a) What is the expected square intra-class distance $\mathbb{E}\big[(X - X')^T(X - X') \mid Y = 1, Y' = 1\big]$ (both $X$ and $X'$ belong to class 1)?

   (b) What is the expected square inter-class distance $\mathbb{E}\big[(X - X')^T(X - X') \mid Y = 1, Y' = 2\big]$ ($X$ belongs to class 1, $X'$ belongs to class 2)?

   (c) Suppose that only one dimension is informative about class values, i.e. $\mu_{11} \neq \mu_{21}$ but $\mu_{1j} = \mu_{2j}$ for $j = 2, \ldots, m$. What is the ratio of the expected intra-class to inter-class distances under this assumption? As $m$ increases, what does this ration approach and what does that imply about the classification accuracy in this case?

3. [4 pts] The Poisson distribution is a discrete probability distribution useful for modeling occurrences per unit time. For example, in networking, packet arrival density is often modeled using a Poisson distribution. The probability mass function (pmf) of a Poisson-distributed random variable $X$ is $p(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$, where $\lambda > 0$ is a parameter specifying the average number of events per interval. It can be shown that $\lambda$ is the mean of the Poisson distribution. In this problem, we will estimate the parameter $\lambda$ from a number of observed packets per unit time $X_1, \ldots, X_n$. Assume that $X_i \sim Poisson(\lambda)$.

   (a) Show that $\hat{\lambda} := \frac{1}{n}\sum_{i=1}^n X_i$ is the maximum likelihood estimate of $\lambda$ and that it is an *unbiased* estimator of $\lambda$, i.e., $\mathbb{E}[\hat{\lambda}] = \lambda$

---

[1]https://www.ieee.org/conferences_events/conferences/publishing/templates.html
[2]https://natanaso.github.io/ece276a/executable.html

(b) Your friend in networking hands you a typical plot showing the counts of computers at the university cluster with different average packet arrival densities:
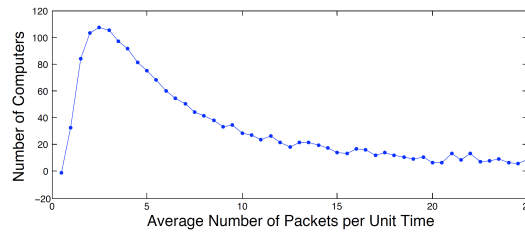


Your extensive experience in statistics tells you that the plot resembles the pdf of a Gamma distribution. Hence, you believe that a good *prior distribution* for $\lambda$ may be a Gamma distribution $\Gamma(\alpha, \beta)$ with pdf[3]:

$$p(\lambda \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}, \quad \lambda > 0,$$

Also, if $\lambda \sim \Gamma(\alpha, \beta)$, then $\lambda$ has mean $\alpha/\beta$ and mode $(\alpha - 1)/\beta$ for $\alpha > 1$. Compute the posterior distribution of $\lambda$ after observing the $n$ packet arrival times. *Hint*: $\lambda^{\sum_i X_i + \alpha - 1} e^{-\lambda(n+\beta)}$ looks like a Gamma distribution.

(c) Derive an analytic expression for the maximum a posteriori (MAP) estimate of $\lambda$ under a $\Gamma(\alpha, \beta)$ prior.

Instead of estimating the parameter $\lambda$, next we consider estimating a *nonlinear function* of $\lambda$, namely $\eta = e^{-2\lambda}$ from a *single* sample $X \sim Poisson(\lambda)$.

(d) Let $\hat{\eta} = e^{-2X}$. Show that $\hat{\eta}$ is the maximum likelihood estimate of $\eta$.

(e) Show that the bias of $\hat{\eta}$ is $e^{\lambda(1/e^2 - 1)} - e^{-2\lambda}$

(f) It turns out that $(-1)^X$ is the *only* unbiased estimate of $\eta$. Prove that it is indeed unbiased and briefly explain why this is a bad estimator to use. It may be instructive to plot the values of the MLE and unbiased estimate for $X = 1, \ldots, 10$ (You do not need to hand in the plot).

4. [10 pts] Train a probabilistic color model from image data and use it to segment unseen images, detect a red barrel, and find the relative distance to the barrel. Given a set of training images, you should hand-label examples of different colors. From these examples, you should build color classifiers for several colors (e.g., red, yellow, brown, etc.) and finally a red barrel detector. You should then use your algorithm to obtain the bounding box of a detected barrel in the image frame and use linear regression to estimate the distance to the barrel on new test images. Instructions and tips follow.

- **Training data**: now available at:
  https://drive.google.com/open?id=0B241vEW29598NnFyUTZRbWNQYTA
- **Test data**: released on 10/21/17 at:
  https://drive.google.com/open?id=0B241vEW29598TUxGUmVtNVI4VVk
- Hand-label appropriate regions in the training images with discrete color labels. For this project, we will be especially interested in regions containing the red barrel (positive examples) and images containing similar colored-areas that are not a barrel (negative examples). If you are ambitious, you can try to implement automated ways of labeling images, e.g., by unsupervised image segmentation, or an adaptive region flooding algorithm. Lighting invariance will be an issue, so you should think carefully about the best color space to use, and perhaps some low-level adaptation on the images.
- Use a learning algorithm to partition the color space into appropriate color class regions. You *must* implement and present results from an approach discussed in class (Single Gaussian, Gaussian Mixture, or Logistic Regression) but you are also free to try other machine learning approaches if you have time, e.g., decision trees, support vector machines, etc. You need to make your algorithm

---

[3]$\Gamma(\alpha)$ refers to the Gamma function but its particular form is not important for this problem

so that it is able to generalize to new images. To prevent overfitting the training images, split them into training and validation sets. Train your algorithms using the training set and evaluate their performance on the validation set. This will allow you to compare different parameters for the probabilistic models and different color space representations.

- Once the color regions are identified, you can use shape statistics and other higher-level features to decide where the barrel is located in the images. Try all possible combinations of red regions and compute a "barrelness" score for each one. Identify the coordinates of abounding box for the regions with high "barrelness" score. You should also compute an estimate of the distance to the barrel, e.g., using linear regression. Your algorithm should be able to quickly classify and display results on a new set of test images. Write a script to display your test result as follows:

```python
import cv2, os
folder = "testset"
for filename in os.listdir(folder):
  # read one test image
  img = cv2.imread(os.path.join(folder,filename))
  # Your computations here!
  blX, blY, trX, trY, d = myAlgorithm(img)
  # Display results:
  # (1) Segmented image
  # (2) Barrel bounding box
  # (3) Distance of barrel
  # You may also want to plot and display other diagnostic information
  cv2.waitKey(0)
  cv2.destroyAllWindows()
```

- Your report should include your test results in the following manner:

```
ImageNo = [01], BottomLeftX = 507.0605, BottomLeftY = 506.7571,
              TopRightX = 662.0208, TopRightY=826.7004, Distance = 5.4418
ImageNo = [02], BottomLeftX = 507.0605, BottomLeftY = 826.7004,
              TopRightX = 662.0208, TopRightY=826.7004, Distance = 2.3840
....
```

- You may rely on some useful python functions for this project:
  - hand-labeling: roipoly: https://github.com/jdoepfert/roipoly.py
  - conversion: cvtColor: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html
  - region analysis: regionprops: http://scikit-image.org/docs/dev/api/skimage.measure.html

  However, **do not use any built-in functions that implement a core part of this project** (Gaussian Mixtures, EM, Logistic Regression). If you are not sure, then ask the TAs. Examples of allowed code:

```python
import cv2
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2YCR_CB)
from skimage import data, util
from skimage.measure import label, regionprops
img = util.img_as_ubyte(data.coins()) > 110
label_img = label(img, connectivity=img.ndim)
props = skimage.measure.regionprops(label_img)
```

5. [6 pts] Write a project report describing your approach to the color segmentation and barrel detection problem. Your report should include the following sections:

- **Introduction**: discuss why the problem is important and present a brief overview of your approach
- **Problem Formulation**: state the problem you are trying to solve in mathematical terms. This section should be short and clear and should define the quantities you are interested in.
- **Technical Approach**: describe your approach to color segmentation and barrel detection
- **Results**: present your training results, test results, and discuss them – what worked, what did not, and why. Make sure your results include (a) a segmented color image, (b) the bounding box coordinates of the barrel, (c) the distance to the barrel estimate **in meters** for each test image.