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' solutions or code. 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 files on **Gradescope** by the deadline shown at the top right corner.

1. Theoretical problems: upload your solutions to 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.

2. Programming assignment: upload all code you have written for the project (do not include the training and test datasets) and a README file with clear descriptions of each file. The Gradescope autograder will test your "barrel_detector.py" on the test set. You are allowed infinite number of attempts before the deadline but each attempt should terminate in 20 minutes which is the autograder timeout limit. The autograder will show you the test cases that you have passed or failed.

3. Report: upload your report. You are encouraged but not required to use an IEEE conference template[1] for your report.

Problems

In square brackets are the points assigned to each part.

1. [26 pts] Let $U_1, \ldots, U_n$ be independent, identically distributed *Uniform* random variables with (continuous) support on $(0, b)$, where $b > 0$ is a parameter.

   (a) Define the random variable $Y := -\sum_{i=1}^{n} \log(U_i)$, where log is the natural logarithm function. Determine the probability density function (pdf) $p(y; b)$ of Y by explicitly computing it.

   (b) Based on the pdf you found in part (a) above, determine the third moment of Y, i.e., $\mathbb{E}[Y^3]$.

   (c) Suppose now that you are given two independent observations y_1 and y_2 of Y. Determine the maximum likelihood estimate of b based on the observations y_1 and y_2.

2. [26 pts] Consider the Logistic Regression (LR) and Gaussian Naive Bayes (GNB) models in the K-class setting with D features. As a reminder, the generative and discriminative models used by LR and GNB, respectively, for a given labeled example $(\mathbf{x}, y)$ with $\mathbf{x} \in \mathbb{R}^D$ and $y = k \in \{1, \ldots, K\}$ are:

$$\text{LR}: \qquad p(y \mid \mathbf{x}) = \frac{\exp\left(\sum_{d=1}^{D} \omega_{d,k} x_d\right)}{\sum_{j=1}^{K} \exp\left(\sum_{d=1}^{D} \omega_{d,j} x_j\right)}$$

$$\text{GNB}: \qquad p(y, \mathbf{x}) = \theta_k \prod_{d=1}^{D} \phi(x_d; \mu_{d,k}, \sigma_{d,k}^2)$$

   (a) How many parameters must be estimated by:

   - Logistic Regression
   - Logistic Regression with additional quadratic features of the form $x_i x_j$ for all $i, j \in \{1, \ldots, D\}$?
   - Gaussian Naive Bayes with arbitrary variances
   - Gaussian Naive Bayes with variance shared among the classes, i.e., the variance of feature d for class k is $\sigma_{d,k}^2 = \sigma_d^2$ (independent of k)?

[1] https://www.ieee.org/conferences_events/conferences/publishing/templates.html

(b) In this part, we will derive the gradient ascent algorithm for optimizing the weights $\boldsymbol{\omega}_k := [\omega_{1,k}, \cdots, \omega_{D,k}]^T \in \mathbb{R}^D$ for $k = 1, \ldots, K$ of K-ary Logistic Regression.

- Explicitly write down the log-likelihood as a function of the parameters, $J(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_K)$.
- Note that there is no closed-form solution to $\max_{\boldsymbol{\omega}_k} J(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_K)$ but we can still find a solution using gradient ascent. Derive an expression for the k-th component of the gradient of $J(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_K)$ with respect to $\boldsymbol{\omega}_k$.
- Beginning with initial weights $\boldsymbol{\omega}_k^{(t)}$, write down the update rule for $\boldsymbol{\omega}_k^{(t)}$ using gradient ascent with step size α.

3. [26 pts] Consider a data set $\mathcal{D} = \left\{ \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ * \end{pmatrix}, \begin{pmatrix} * \\ 6 \end{pmatrix} \right\}$, where the $*$ symbol indicates missing values. Suppose that the data was generated using independent draws from a two-dimensional *Uniform* distribution:

$$p(\mathbf{x}; \boldsymbol{\theta}) = \begin{cases} \frac{1}{|u_1 - l_1||u_2 - l_2|}, & \text{if } l_1 \leq x_1 \leq u_1 \text{ and } l_2 \leq x_2 \leq u_2 \\ 0, & \text{otherwise,} \end{cases}$$

where $\boldsymbol{\theta} := \begin{bmatrix} l_1 & u_1 & l_2 & u_2 \end{bmatrix}^T$ are the distribution parameters. Our goal is to determine the maximum likelihood estimate of $\boldsymbol{\theta}$, based on the known data values $\mathcal{D}_{known}$, by solving:

$$\max_{\theta} \; \log p(\mathcal{D}_{known}; \boldsymbol{\theta})$$

Since there is missing data, we will use the EM algorithm to estimate $\boldsymbol{\theta}$ by thinking of the missing values as latent variables.

(a) (E step) Starting with an initial estimate $\boldsymbol{\theta}^{(0)} := \begin{bmatrix} 0 & 10 & 0 & 10 \end{bmatrix}^T$, explicitly compute an upper bound $\mathcal{T}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(0)})$ to the data log-likelihood.

(b) (M step) Find $\boldsymbol{\theta}^{(1)}$ that maximizes $\mathcal{T}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(0)})$.

(c) Make a 2-D plot of the data and the bounding box determined by your parameter estimate $\boldsymbol{\theta}^{(1)}$.

(d) Determine the final MLE estimate $\boldsymbol{\theta}_{MLE}$ that the EM algorithm will eventually converge to.

4. [90 pts] Train a probabilistic color model from image data and use it to segment unseen images, detect a blue barrel, and draw a bounding box around it. 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., blue, yellow, brown, etc.) and finally a blue barrel detector. You should then use your algorithm to obtain the bounding box of a detected barrel in the image frame on new test images. Instructions and tips follow.

- **Training data**: available in this package.
- **Test data**: will be released on 01/28/19.
- (Optional) We encourage you to set up a Python virtual environment so that the packages you use are compatible with the Gradescope autograder. You may follow the bash commands below

```
# install python3 and pip3
$ apt-get install python3 python3-pip
# install virtualenv, virtualenvwrapper
$ apt-get install python-virtualenv
$ pip3 install virtualenvwrapper
# make a virtual environment
$ export WORKON_HOME=~/Envs
$ mkdir -p $WORKON_HOME
$ source /usr/local/bin/virtualenvwrapper.sh
$ mkvirtualenv -p `which python3` ece276a_hw1
# install required packages
$ cd hw1_starter_code
$ pip3 install -r requirements.txt
# exit virtual environment
$ deactivate
```

```
# open virtual environment
$ source /usr/local/bin/virtualenvwrapper.sh
$ workon ece276a_hw1
```

- Hand-label appropriate regions in the training images with discrete color labels. For this project, we will be especially interested in regions containing the blue 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 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 blue regions and compute a "barrelness" score for each one. Identify the coordinates of a bounding box for the regions with high "barrelness" score. Your algorithm should be able to quickly classify and display results on a new set of test images.
- You should use the provided starter code "barrel_detector.py" and implement the two functions "segment_image()" and "get_bounding_box()". For this file, please do not change the file name, class name, function names, function arguments or use other packages not listed in "requirements.txt". 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:

  ```
  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. [32 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 and (b) the bounding box coordinates of the barrel for each test image.