DS-GA 3001, Introduction to Data Science (PhD)
Prof. Arthur Spirling, Prof. Julia Kempe
Assignment date: November 25, 2020, 2020

Homework 3: Solutions

This homework must be returned to your TA by **5pm Eastern Time, December 5th, 2020**. Late work will incur penalties of the equivalent of one third of a letter grade per day late.

It must be your own work, and your own work only—you must not copy anyone's work, or allow anyone to copy yours. This extends to writing code. You may consult with others, but when you write up, you must do so alone.

Your homework submission must be in one of the following formats: (1) A set of answers and a clearly commented `Python` code appendix (use comments to identify code relevant to each answer you produced), (2) A report consisting of clearly marked answers, each accompanied by the relevant code (e.g., a report generated using Python markdown or similar). **In either case, your code must be included in full, such that your understanding of the problems can be assessed.**

---

1. In this exercise, you will investigate entropy and derive some of its fundamental properties. We will only focus on the discrete case (Shannon entropy), i.e. $H(p) = -\sum_{i=1}^{n} p_i \log p_i$ where $p = (p_1, p_2, \ldots, p_n)$ is some discrete probability distribution so that $p_i \geq 0$ and $\sum_{i=1}^{n} p_i = 1$. Throughout this exercise, you may assume $0 \log 0 = 0$. You may also use concavity of $-x \log x$ where $x \geq 0$ without proof.

   (a) Show that $H(p) \geq 0$. Give an example of a distribution p_0 such that $H(p_0) = 0$.
   Given a distribution $p = (p_1, p_2, \ldots, p_n)$, $p_i \leq 1$ and, hence, $\log p_i \leq 0$ for all $i \in [n]$. Therefore, $H(p) = -\sum_{i=1}^{n} p_i \log p_i \geq 0$. If $p = (1, 0, 0, \ldots, 0)$, $H(p) = 0$.

   (b) An n-variate scalar function f is *concave* if, for all $x, y \in \text{dom}(f)$ and $\lambda \in [0, 1]$, we have $f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda) f(y)$. Prove that $H(p)$ is concave when viewed as an n-variate function of $p_1, p_2, \ldots, p_n$.
   Given discrete distributions $p_1, p_2 \in \mathbb{R}^n$ and $\lambda \in [0, 1]$, we have

$$\begin{aligned} H(\lambda p_1 + (1-\lambda)p_2) &= \sum_{i=1}^{n} -(\lambda p_{1i} + (1-\lambda)p_{2i}) \log(\lambda p_{1i} + (1-\lambda)p_{2i}) \\ &\geq \sum_{i=1}^{n} -\lambda p_{1i} \log p_{1i} + \sum_{i=1}^{n} -(1-\lambda) p_{2i} \log p_{2i} = \lambda H(p_1) + (1-\lambda) H(p_2), \end{aligned}$$

   where we used the fact that $-x \log x$ is concave.

   (c) Prove that if f is concave, then, for any set of n values $\lambda_i \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$ and any $x_1, x_2, \ldots, x_n$ from the domain of f, we have $f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \geq \sum_{i=1}^{n} \lambda_i f(x_i)$. (Hint: use induction on n). This result is called Jensen's inequality.
   First, note that inequality is trivially satisfied for $n = 1$ having $f(x)$ on both sides.

Moreover, the desired inequality degenerates to the definition of concavity in the case of $n = 2$. Now, assume it holds for $n = k$; i..e., $f\left(\sum_{i=1}^{k} \lambda_i x_i\right) \geq \sum_{i=1}^{k} \lambda_i f(x_i)$ for any λ and x satisfying the assumptions. We will show that inequality holds for $n = k + 1$. First, fix some $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{k+1})$ and $x = (x_1, x_2, \ldots, x_{k+1})$. Note that if $\lambda_{k+1} = 1$, then $\lambda_1 = \lambda_2 = \ldots, \lambda_k = 0$ and the inequality is again trivially satisfied by having $f(x_{k+1})$ on both sides. Otherwise, define $y = (\lambda_1 x_1 + \lambda_2 x_2 + \ldots, +\lambda_k x_k)(1 - \lambda_{k+1})^{-1}$, which gives

$$\begin{aligned} f\left(\sum_{i=1}^{k+1} \lambda_i x_i\right) &= f(\lambda_{k+1} x_{k+1} + (1 - \lambda_{k+1}) y) \geq \lambda_{k+1} f(x_{k+1}) + (1 - \lambda_{k+1}) f(y) \\ &= \lambda_{k+1} f(x_{k+1}) + (1 - \lambda_{k+1}) f\left(\sum_{i=1}^{k} \lambda_i \cdot f(x_i)/(1 - \lambda_{k+1})\right), \\ &\geq \lambda_{k+1} f(x_{k+1}) + (1 - \lambda_{k+1})(1 - \lambda_{k+1})^{-1} \sum_{i=1}^{k} \lambda_i x_i = \sum_{i=1}^{k+1} \lambda_i f(x_i), \end{aligned}$$

where we used the definition of concavity as well as the inductive hypothesis. This concludes proof by induction.

(d) Use Jensen's inequality to prove that $H(p) \leq \log n$. For which distribution p will $H(p) = \log n$?

Given some discrete distribution $p \in \mathbb{R}^n$, define $\lambda_i = 1/n$ and $x_i = p_i$ for all $i \in [n]$. Then, applying Jensen's inequality with a concave function $f(x) = -x \log x$, we discover

$$\begin{aligned} f\left(\sum_{i=1}^{n} \lambda_i x_i\right) &= f\left((1/n) \sum_{i=1}^{n} p_i\right) = f(1/n) = (1/n) \log n \\ \sum_{i=1}^{n} \lambda_i f(x_i) &= (1/n) \sum_{i=1}^{n} -p_i \log p_i = (1/n) H(p), \end{aligned}$$

which immediately implies $H(p) \leq \log n$ by Jensen's inequality. Note that $H(p_u) = \log n$ where $p_u = (1/n, 1/n, \ldots, 1/n)$ is the uniform distribution.

2. Noisy coding theorem: In this exercise you will complete the proof of the noisy coding theorem seen in the Lecture. Namely, assume we have a source of iid bits b with probability $\Pr[b = 1] = p$, and without loss of generality $p < \frac{1}{2}$. We want to show:

Theorem: For all $p < \frac{1}{2}$ and $\epsilon, \delta > 0$ and large enough n, there exists an encoder $E : \{0,1\}^n \to \{0,1\}^m$ and decoder $D : \{0,1\}^m \to \{0,1\}^n$ such that $m \leq n \cdot (H(x) + \epsilon)$ and $Pr_x[D(E(x)) \neq x] < \delta$. Recall

- Stirling's formula: $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$
- Chernoff bound: Let $X_1, \ldots, X_n$ be iid random variables in $[0,1]$ with expectation p. Then
$$\Pr\left[\left|\frac{1}{n}\sum_i X_i - p\right| \geq \lambda\right] \leq 2e^{-n\lambda^2}$$
- Define $wt(x) = \#1$ in x. Define a set $A = \{x \in \{0,1\}^n | (p - \lambda)n \leq wt(x) \leq (p + \lambda)n\}$ for some λ such that $\lambda \to 0$ for large n and that you will determine in (b).

(a) Show that A is "small": $\log |A| \leq n(H(p) + \epsilon')$, where ϵ' is a function of λ and $\epsilon' \to 0$ as n grows large.

Let n be large enough so that $p + \lambda < 1/2$. In this case,

$$|A| = \binom{n}{n(p-\lambda)} + \binom{n}{n(p-\lambda)+1} + \ldots + \binom{n}{n(p+\lambda)} \leq (2\lambda n + 1)\binom{n}{n(p+\lambda)}.$$

since the last binomial coefficient is the largest. Now, we apply Stirling's formula to the binomial coefficient on the right hand side:

$$\begin{aligned}\log \binom{n}{n(p+\lambda)} &= \log\sqrt{2\pi n} + \log n^n - \log\sqrt{2\pi(n(p+\lambda))} - n(p+\lambda)\log n(p+\lambda) \\ &\quad - \log\sqrt{2\pi(n(1-(p+\lambda)))} - n(1-(p+\lambda))\log(n(1-(p+\lambda))) \\ &\le n\log n - n(p+\lambda)\log(p+\lambda) - n(p+\lambda)\log n - n(1-(p+\lambda))\log n \\ &\quad - n(1-(p+\lambda))\log(1-(p+\lambda))) \\ &= -n(p+\lambda)\log(p+\lambda) - n(1-(p+\lambda))\log(1-(p+\lambda))) = nH(p+\lambda).\end{aligned}$$

where we used the fact that square root and logarithm are concave functions. Further, note that entropy is a continuous function, so $H(p)$ is arbitrarily close to $H(p+\lambda)$ for large enough n. Therefore,

$$\log|A| \le \log(2\lambda n+1) + nH(p+\lambda) = n(n^{-1}\log(2\lambda n+1) + H(p+\lambda)) = n(H(p)+\epsilon'),$$

where $\epsilon' = n^{-1}\log(2\lambda n+1) + H(p+\lambda) - H(p)$, which vanishes to 0 with $n\to\infty$.

(b) A randomly drawn x (according to the source distribution) is $\in A$ with high probability: Use the Chernoff bound to show that $\Pr[x\notin A] < \delta$ for a choice of λ.
Let $\lambda = \sqrt{-n^{-1}\log\delta/2}$; note that $\sum_{i=1}^n X_i$ is the number of ones in a vector of independent Bernoulli(p) variables X_i. Therefore, $2e^{-n\lambda^2} = \delta$ and so

$$1-\delta \le \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n X_i - p\right| \le \lambda\right) = \mathbb{P}\left(n(p-\lambda) \le \sum_{i=1}^n X_i \le n(p+\lambda)\right) = \mathbb{P}(X\in A),$$

hence, $\mathbb{P}(X\notin A) = 1-\mathbb{P}(X\in A) \ge \delta$, as desired. Not that $\lambda\to 0$ as $n\to\infty$ with this definition, so the result of part (a) applies to thus selected λ.

(c) Combine (a) and (b) to show that for large enough n, $\log|A| \le n(H(p)+\epsilon)$. Finish the proof of the theorem by using an encoding that gives an index in A.
In part (a), we concluded that $\log|A| \le n(H(p)+\epsilon')$ where $\epsilon'\to 0$ as $n\to\infty$. Therefore, with large enough n, we will achieve $\log|A| \le n(H(p)+\epsilon)$ for any given ϵ. Consider an encoding function $x\mapsto i(A,x)$, where $i(A,x)$ is the index of element x in A in case $x\in A$ and $i(A,x)=0$ otherwise. Moreover, define the decoder function by $i(A,x)\mapsto x$. Then, all messages $x\in A$ will be decoded correctly. Therefore, $\mathbb{P}(D(E(x))\neq x) \le \mathbb{P}(x\notin A) \le \delta$ under selecting $\lambda = \sqrt{-n^{-1}\log\delta/2}$ for a sufficiently large n. If $m = \log|A| \le n(H(p)+\epsilon)$, we find that all "typical" sequences (sequences from A) can be coded as strings of m bits, concluding the proof of the theorem.

3. SOM and clustering: Show (as a brief outline only, no formulas or calculations needed) how a SOM becomes a type of k-means clustering when the width of the neighborhood is set to zero. Note that the difference to conventional k-means clustering will be that training data points are added one at a time for each update, as opposed to being present in its entirety at the beginning.
Imagine that one iteration of SOM updates has access to all training samples. Then, one iteration of a zero-width SOM is marked by (1) partitioning all training points among neurons from the grid, and (2) bringing those neurons (best matching units) closer to their assigned points. Recall that the K-means algorithm has exactly the same structure.

4. VC-dimension:

   (a) *Threshold* and *Interval* classifiers: Compute the VC dimension of (1) the class of threshold classifiers $T = \{t_a(x) = I_{x<a} | a \in \mathbb{R}\}$ and (2) the class of interval classifiers $H = \{h_{ab}(x) = I_{x\in(a,b)} | a, b \in \mathbb{R}, a < b\}$
   The class of threshold classifiers has a VC-dimension of 1. One point is easily shattered: given an input $x \in \mathbb{R}$, t_{x-1} (t_{x+1}) will have zero error when x has a positive (negative) label. However, having two points $a < b$, there is no threshold classifier with zero error if a is positive but b is negative. The class of interval classifiers has a VC-dimension of 2. It is easy to see that all four labelings of a two-point arrangement can be shattered; however, there is no interval classifier that will achieve zero error on $a < b < c$ with a, c positive and b negative.

   (b) Show that the set of functions $\{I(\sin(\alpha x) > 0)\}$ can shatter the following points on the line:
   $$z_1 = 10^{-1}, \ldots, z^l = 10^{-l}$$
   for any l. Hence the VC dimension of the class $\{I(\sin(\alpha x) > 0)\}$ is infinite.
   Define $N = \{i \in [l] \colon y_i = 0\}$ to be the set of indices corresponding to negative observations; further,
   $$\alpha = \pi \left(1 + \sum_{i=1}^{l} (1 - y_i) 10^i\right) = \pi \left(1 + \sum_{i \in N} 10^i\right).$$
   Now, for any $j \in N$ define $N_j = \{i \in N \colon i < j\}$ and $N^j = \{i \in N \colon i > j\}$; we then have
   $$\alpha 10^{-j} = \pi \left(10^{-j} + 10^{-j} \sum_{i \in N} 10^i\right) = \pi \left(10^{-j} + 1 + \sum_{i \in N_j} 10^{i-j} + \sum_{i \in N^j} 10^{i-j}\right)$$
   where $\sum_{i \in N^j} 10^{i-j} = 2k$ for some $k \in \mathbb{Z}$ and $\sum_{i \in N_j} 10^{i-j} < \sum_{i=1}^{\infty} 10^{-i} = 10/9 - 1 = 1/9$. In addition, $10^{-j} \leq 0.1$; thus, $\pi + 2k\pi < \alpha 10^{-j} < \pi(1 + 1/9 + 0.1) + 2k\pi$, implying that $\sin(\alpha 10^{-j}) < 0$. Hence, all negative points are correctly classified by $I(\sin(\alpha x))$. Similarly, for any $j \in [l] \setminus N$, we have
   $$\alpha 10^{-j} = \pi \left(10^{-j} + 10^{-j} \sum_{i \in N} 10^i\right) = \pi \left(10^{-j} + \sum_{i \in N_j} 10^{i-j} + \sum_{i \in N^j} 10^{i-j}\right),$$
   and, for the same reasons as above, we conclude $2k\pi < \alpha 10^{-j} < \pi(1/9 + 0.1) + 2k\pi$, implying $\sin(\alpha 10^{-j}) > 0$. Hence, all positive points are correctly classified by $I(\sin(\alpha x))$ as well.

5. Gaussian Naive Bayes: in Lab10, we derived the maximum likelihood estimators for Bernoulli and Multinomial conditional models; in this exercise you will do the same for the Gaussian Naive Bayes model. In particular, let X be a d-dimensional continuous random variable and Y be a discrete random variable with range $[K] = \{1, 2, \ldots, K\}$. Moreover, assume $X_j | Y = y \sim \mathcal{N}(\mu_{jy}, \sigma_{jy}^2)$ for any $j \in [d]$ and any $y \in [K]$, so that the conditional model $X|Y$

has $2Kd$ parameters. Given data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ generated by the joint distribution of X and Y, find the maximum likelihood estimators for these parameters (hint: you will need to use *the naive assumption*).
The likelihood and log-likelihood functions $\mathcal{L}$ and l for μ and σ^2 given $\mathcal{D}$ are

$$\mathcal{L}(\mu, \sigma^2|\mathcal{D}) = p_{\mu,\sigma^2}(\mathcal{D}) = \prod_{i=1}^{n} p_{\mu,\sigma^2}(x_i, y_i) = \prod_{i=1}^{n} p_{\mu,\sigma^2}(x_i|y_i)p(y_i) = \prod_{i=1}^{n} p(y_i) \prod_{j=1}^{d} p_{\mu,\sigma^2}(x_{ij}|y_i)$$
$$= \prod_{k=1}^{K} \prod_{i \in C_k} p(k) \prod_{j=1}^{d} p_{\mu,\sigma^2}(x_{ij}|y_i) = \prod_{k=1}^{K} p(k) \prod_{i \in C_k} \prod_{j=1}^{d} \frac{\exp\left(-\frac{(x_{ij}-\mu_{jk})^2}{2\sigma_{jk}^2}\right)}{\sigma_{jk}\sqrt{2\pi}}$$

$$l(\mu, \sigma^2|\mathcal{D}) = \log \mathcal{L}(\mu, \sigma^2|\mathcal{D}) = \sum_{k=1}^{K} \log p(k) \sum_{i \in C_k} \sum_{j=1}^{d} -\frac{(x_{ij} - \mu_{jk})^2}{2\sigma_{jk}^2} - \log \sigma_{jk} - \log \sqrt{2\pi}$$

where we defined $C_k = \{i \in [n] \colon y_i = k\}$. Now, we take the derivative of the log-likelihood with respect to μ_{jk} and obtain

$$\frac{\partial l(\mu, \sigma^2)}{\partial \mu_{jk}} = \log p(k) \sum_{i \in C_k} -\frac{2(x_{ij} - \mu_{jk})}{\sigma_{jk}^2} = 0 \quad \Longrightarrow \quad \hat{\mu}_{jk} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$$

Note that the second derivative is positive, so that the above estimator is indeed the maximum likelihood estimator. Now, take the derivative of the log-likelihood with respect to σ_{jk} and obtain

$$\frac{\partial l(\mu, \sigma^2)}{\partial \sigma_{jk}} = \log p(k) \sum_{i \in C_k} \frac{(x_{ij} - \mu_{jk})^2}{\sigma_{jk}^3} - \frac{1}{\sigma_{jk}} = 0 \quad \Longrightarrow \quad \hat{\sigma}_{jk}^2 = \frac{1}{|C_k|} \sum_{i \in C_k} (x_{ij} - \hat{\mu}_{jk})^2$$

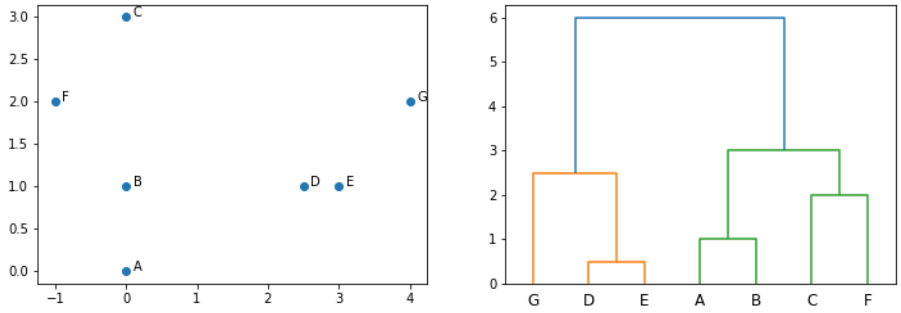
Note that $\hat{\sigma}_{jk}^2$ is a biased estimator of the true parameter σ_{jk}^2 .

6. In this exercise, you will design a text classification pipeline to categorize movies into four genres: comedy, drama, horror, and action, based on their official textual plot descriptions (`movie-plots-student.csv`). You are free to choose your own strategy, i.e. train/validation splits, tokenization, normalization, vectorization, modeling, validation and other choices are yours (see Lab11 for reference). Describe your methodology in detail, comment on your selected approach, report all intermediate results, and provide your code in a jupyter notebook, ready to be executed (we will run your model on a private hold-out test set). In addition, please provide a function `test_model(test_data)` that will label our hold-out movie descriptions (given as a list of raw unprocessed strings) according to your processing and modeling pipeline. Top-3 performing submissions (measured by macro-F1 score) will receive 5 extra points towards this homework assignment.
See `question6_solution.ipynb`

7. Hierarchical clustering:

(a) Recall the agglomerative clustering algorithm from Lab9: it starts with all samples as their own clusters and proceeds by merging pairs of current clusters based on a specified linkage function; most common linkage functions allow for non-euclidean metrics. Let $\mathcal{D} \subseteq \mathbb{R}^2$ be a dataset of points $A = (0,0), B = (0,1), C = (0,3), D = (2.5,1), E = (3,1), F = (-1,2), G = (4,2)$. Use Manhattan distance together with a complete (maximum) linkage function to produce a full dendrogram describing the agglomerative clustering procedure on $\mathcal{D}$.

See below:



(b) Perform agglomerative clustering with at least 5 different linkage-metric combinations to cluster 5000 MNIST samples (`mnist-sample-X.npy` and `mnist-sample-y.npy` are normalized and flattened, ready-to-use) into 10 groups. For each linkage-metric selection, label the resulting clusters by majority voting and assess the training error. Repeat the procedure for 10-means and 10-means++ methods. Which one worked best?

See `question7_solution.ipynb`