Introduction to Machine Learning

ECE 580
Spring 2025

HW #3, Due 02/20/25 11:59pm

Submission Instructions

Submit your work to the corresponding assignment in Gradescope. Although Gradescope accepts multiple file formats, they strongly recommend submitting your assignment as a single PDF file.

It is your responsibility to ensure the uploaded file is: 1) the correct file, 2) complete (includes all pages), 3) legible, and 4) submitted on-time as determined by the Gradescope server system clock.

It is your responsibility to submit a multi-page PDF file and tag the pages that correspond to each question. Pages may be tagged after submission, even if the submission deadline has passed. If you are submitting close to the submission deadline, submit your assignment first then immediately return to tag pages.

When code is requested, submit a PDF print-out of your code. Submitting a URL for a cloud-based repository is insufficient.

Late Submissions

Late submissions will be accepted up to 5 days after the submission deadline, with the following point penalty applied if its late submission is not excused: [1]

1 day (0^+ to 24 hours) late: 2 point deduction ($\frac{1}{5}$ letter grade)
2 days (24^+ to 48 hours) late: 5 point deduction ($\frac{1}{2}$ letter grade)
3 days late: 10 point deduction (1 letter grade)
4 days late: 20 point deduction (2 letter grades)
5 days late: 30 point deduction (3 letter grades)
6 or more days late: score = 0 (not accepted for credit)

The late policy is designed to be minimally punitive for submissions up to 3 days late, yet encourages staying current with the coursework for our course by not allowing one assignment's late submission to overlap with the next assignment's submission.

A homework score will not drop below 0 as a result of applying the late penalty point deduction.

[1] One day = one 24-hour period or fraction thereof.

Maximum Likelihood Estimation

1. Let $\mathscr{D}(x_1,y_1)\ldots(x_n,y_n)$ denote a training dataset of feature, label pairs, with $x_i \in \mathbb{R}^d$ and $y_i \in \{0,1\}$.
   - Let $\mu_0 \in \mathbb{R}^d$ and let $\mu_1 \in \mathbb{R}^d$ be arbitrary unobserved random variables.
   - For $i \in \{1\ldots n\}$, let $y_i = 0$ with probability $1/2$ and let $y_i = 1$ with probability $1/2$ (independently).
   - For $i \in \{1\ldots n\}$, let $x_i \sim \mathscr{N}(\mu_{y_i}, I)$.

   The multivariate normal distribution $\mathscr{N}(\mu, I)$, has density

   $$q(x;\mu) = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|x-\mu\|_2^2\right)$$

   .

(5) (a) For a single sample (x_i,y_i), write down the likelihood $\mathbb{P}((x_i,y_i);\mu_0,\mu_1)$.

(5) (b) Write down the log-likelihood of the training dataset: $\log(\mathbb{P}(\mathscr{D}|\mu_0,\mu_1))$

(5) (c) Write down the closed-form expressions for the MLE estimates μ_0^{MLE} and μ_1^{MLE}. i.e.

$$\arg\min_{\mu_0,\mu_1} Pr(\mathscr{D}|\mu_0,\mu_1).$$

You may find the following definitions useful:

- Let $S_0 \subset \{1\ldots n\}$ denote the set of indices where $y_i = 0$, i.e. $i \in S_0 \Leftrightarrow y_i = 0$. Let $n_0 := |S_0|$ denote the size of S_0.
- Let $S_1 := \{1\ldots n\} \backslash S_0$. Let $n_1 := n - n_0$ denote the size of S_1

(10) (d) Given μ_0 and μ_1, and a new query point x, show that

$$\mathbb{P}(y=1|x;\mu_0,\mu_1) = \frac{\exp\left((\mu_1-\mu_0)^\top x\right)}{C + \exp\left((\mu_1-\mu_0)^\top x\right)}, \tag{1}$$

where C is a constant that depends on μ_1 and μ_0. State the expression for C.

(5) (e) Recall from lecture that for binary logistic regression setup, where $y \in \{0,1\}$, parameterized by $\mu_0, \mu_1 \in \mathbb{R}^d$,

$$\mathbb{P}_{logistic}(y=1|x;\mu_0,\mu_1) = \frac{\exp\left(\mu_1^\top x\right)}{\exp\left(\mu_0^\top x\right) + \exp\left(\mu_1^\top x\right)}. \tag{2}$$

Under what condition on μ_0, μ_1 is $\mathbb{P}_{logistic}$ from (2) equal to $\mathbb{P}$ from (1)?

2. **(MLE for Gaussian Mean and Covariance)** Let $\Sigma \in \mathbb{R}^{d\times d}$ denote some symmetric, positive-definite matrix, and let $w \in \mathbb{R}^d$. For $i = 1\ldots n$, assume that

   $$x_i \sim \mathscr{N}(w, \Sigma),$$

   The training dataset is $\mathscr{D} = (x_1 \ldots x_n)$.

(5) (a) Give the log-likelihood $\mathbb{P}(\mathscr{D}; w, \Sigma)$.

(10) (b) Assume Σ is fixed and known, what is the maximum likelihood estimator of w?

(10) (c) Assume $w = 0$, and assume that Σ is a diagonal matrix, i.e. for all $i, j \in \{1 \ldots d\}$, $\Sigma_{ij} = 0$ for $i \neq j$ and $\Sigma_{ii} = 1/v_i$ for some unknown parameters v_i. Rewrite the log-likelihood of part (a) as a function of $v_1 \ldots v_d$.

(15) (d) Continuing from part (c), what is the maximum likelihood estimator for $v_1 \ldots v_d$?

(20) (e) Now let $\Sigma = U \Lambda U^\top$ denote the eigenvalue decomposition of Σ. Assume for simplicity that

- $w = 0$.
- $U \in \mathbb{R}^{d \times d}$ is **fixed and known**.

Let $\lambda_1 \ldots \lambda_d$ denote the diagonal entries of Λ (i.e. eigenvalues of Σ). Write the log-likelihood as a function of $\lambda_1 \ldots \lambda_d$, i.e. $\mathbb{P}(\mathscr{D}; \lambda_1 \ldots \lambda_d)$.

What is the maximum likelihood estimator $\hat{\lambda}_i^{MLE}$, for $i = 1 \ldots d$?

Hint: $\det(\Sigma) = \det(\Lambda) = \prod_{i=1}^d \lambda_i$. This is useful for simplifying the multivariate Gaussian density.

Cross Validation on Image Block

Choose λ by Cross-Validation

In this project, we apply LASSO regression to each K×K image block, and our goal is to determine the optimal regularization parameter λ for each block to achieve the best reconstruction results. To select the optimal λ, we will employ cross-validation.

As outlined in HW #2, for each K×K block, we sample S pixels, which we refer to as the ***sensed*** pixels (i.e., known pixels). For the purposes of this reconstruction task, we assume that the remaining $K^2 - S$ pixels are unobserved and unknown.

For each candidate λ (from the list of λ you are considering), we will:

- Partition the block into m validation pixels and $(S - m)$ training pixels
- Determine the DCT coefficients for the $(S - m)$ pixels in the training set
- Estimate approximation "error" using the m pixels in the validation set
- Use Mean Square Error (MSE) as a measurement of the "error"
- Choose the λ with the lowest error (may have different λ for each block!)

Cross-Validation with Random Subsets

In this section, we will compare **K-fold cross-validation** with **cross-validation with random subsets**, with the latter being the approach applied in our image recovery task.

K-fold cross validation

In this section, we will review the process of K-fold cross-validation. The steps involved in K-fold cross-validation are outlined as follows:

- Distribute all data into K folds such that the number of samples in every fold is equal (or as equal as possible)
- Take one fold as the validation set at each iteration
- Training-and-validation process is repeated K times

Cross validation with random subsets

The cross-validation with random subsets of the S sampled pixels follows these steps:

- At each iteration, randomly draw m samples to form the validation set and use the remaining $(S - m)$ samples as the training set
- Repeat the training-and-validation process M times

(Note: With this approach to cross-validation there is a "risk" that a data point (sample) may not be included in any of the validation sets, or any of the training sets)

Cross validation in our task

We will apply **cross-validation with random subsets** in our task for the following reasons:

• When the data set is small, using this method with large M is more accurate than K-fold cross validation.

• Easy to implement if the data set cannot be divided equally into K folds

Question

3. Demonstrate you are able to apply **random subsets cross-validation** to select the regularization parameter for LASSO to estimate model weights for **a single corrupted image block**.

   Apply random subsets cross-validation to choose the regularization parameter for LASSO to reconstruct your corrupted image block with $S = 30$ sensed pixels (same setting as Question 2 in *LASSO on Image Block*, from HW #2). For this proof-of-concept, you may choose the regularization parameters so there are 6 logarithmically-spaced values that span 6 decades. For example $\lambda = \{10^{-3}, 10^{-2}, \ldots, 10^{2}\}$.[2]

   - We will use m = floor(S/6) in this project, where S is the total number of samples (sensed pixels in a block).
   - We will use M = 20 in this project.

(10) (a) Pick the first random subset for cross-validation. For this random subset, for each of the 6 values of λ, display:

1. the reconstructed ***sensed*** image block
2. the original (corrupted) image block
3. the model weights via a stem plot

(Neither chip will contain the missing pixels (i.e., $K^2 - S$ unknown pixels) for this cross-validation subset.)

(Okay to use `matplotlib.pyplot.stem` function for the stem plot)

(5) (b) **Plot the MSE** between the sensed pixels held out for validation (i.e., m held out pixels) and the estimates of those held out pixels as a function of λ (use a logarithmic scale on the λ axis of the plot).

(10) (c) Repeat the reconstruction of the held out ***sensed*** pixels for the other 19 random cross-validation subsets. (It is not necessary to display the reconstructions or weights for these subsets, as you did in part (a) for the first subset.)

Create a single plot that **shows the average MSE vs. λ curve** on a single set of axes.

(5) (d) Reconstruct the ***missing*** pixels (i.e., $K^2 - S$ pixels) in the image block by applying LASSO with the value of λ that minimizes the cross-validated MSE (the value of λ that minimizes the average MSE vs. λ curve).

Display following:

1. the reconstructed image block
2. the original (corrupted) image block

[2] A recommended starting range for λ is provided in the class notes for Image Recovery Project.

3. the model weights via a stem plot

(Okay to use `matplotlib.pyplot.stem` function for the stem plot)

(5) (e) Submit a PDF print-out of your code.
(Submitting a URL for a cloud-based repository is insufficient.)

Recover a Whole Image

Quality of Recovered Image

The quality of a recovered image is measured using the mean square error between the recovered image and the original image:

$$\frac{1}{WH} \sum_{1 \leq x \leq W; 1 \leq y \leq H;} [\hat{g}(x,y) - g(x,y)]^2$$

In this equation, W = image width (in pixels); H = image height (in pixels); $g(x,y)$ are pixels of the recovered image; $g(x,y)$ are pixels of the original image.

Question

4. Demonstrate you are able to extend the single-block reconstruction to **whole-image reconstruction** (looping over all blocks in the image).

(10) (a) Divide the "fishing_boat" image using 8×8 block size. Use **S = 30** sample size to corrupt each image block. Combine (concatenate) all corrupted image blocks into a full image.

Display your **whole corrupted image**.

(30) (b) Run cross-validation with random subsets using m and M values (m = floor(S/6), M=20) in the previous question **for all image blocks** to find the best λ (which minimizes average MSE) for every image block. ($\lambda = \{10^{-4}, 10^{-3}, \ldots, 10^{3}\}$)

Reconstruct every image block using the best λ found by cross-validation. Combine (concatenate) all recovered image blocks into a full image.

Display the following:

1. the original (uncorrupted) image
2. the corrupted image
3. the reconstructed image

(5) (c) **Calculate the MSE** to measure the quality of the reconstructed whole image

(5) (d) Submit a PDF print-out of your code.
(Submitting a URL for a cloud-based repository is insufficient.)