

CSC6022 Homework 3

Homework Due Date: **Dec 12, 2025**

Instructions (Please read carefully)

- Submit your answers as an electronic copy only in **pdf** format on Blackboard.
- No late submissions will be accepted. Zero credit will be assigned for late submissions. Email requests for late submission will not be replied.
- Please type in Latex or provide handwritten submissions scanned to **pdf**.
- Explicitly mention your collaborators if any. Collaborations should be limited to discussing and learning from each other but please do your own work and write your own codes. We will actively monitor any attempt to copy solutions from each other or from the internet.
- The full score of this homework is 150 pts.

1 Learning Distance Metric [30 pts]

1. Connecting Metric Learning and Cross-Entropy Classification

You are given a setting where an embedding network and a classifier are used for learning similarity measures. In the metric learning setup, one commonly used loss is the **N-pair loss**, and in classification, we use the **cross-entropy loss**. The goal is to establish a connection between these losses by showing that the ridge-regularized cross-entropy loss upper bounds a pairwise metric-learning loss.

The overall task is divided into the following subproblems.

- (a) Given an $(N + 1)$ -tuple of training examples

$$\{x, x^+, x_1^-, x_2^-, \dots, x_{N-1}^-\}$$

where x^+ is a positive example (same class as x) and $\{x_i^-\}$ are negative examples (different classes), let $z = \phi(x)$ be the embedding of x by a deep neural network. The N-pair loss is defined by

$$\mathcal{L}(\{x, x^+, \{x_i^-\}\}; z) = \log \left(1 + \sum_{i=1}^{N-1} \exp(z^\top z_i^- - z^\top z^+) \right)$$

Show that the above loss can be rewritten as

$$\mathcal{L} = -\log \frac{\exp(z^\top z^+)}{\exp(z^\top z^+) + \sum_{i=1}^{N-1} \exp(z^\top z_i^-)} = -z^\top z^+ + \log \left(\exp(z^\top z^+) + \sum_{i=1}^{N-1} \exp(z^\top z_i^-) \right).$$

- (b) Consider an encoder $\phi_W(x) = z$ and a classifier $f_\theta(z)$ that maps the embedding to a vector of class probabilities for K classes via the softmax function:

$$p_{ik} = \frac{\exp(\theta_k^\top z_i)}{\sum_{j=1}^K \exp(\theta_j^\top z_i)}$$

Define the ridge-regularized cross-entropy loss (for n samples) as

$$\mathcal{L}_{CE} = -\frac{1}{n} \sum_{i=1}^n \log p_{i,y_i} + \lambda \sum_{k=1}^K \|\theta_k\|^2$$

(1) Rewrite the loss by splitting it into two parts:

- A linear part (denoted $f_1(\theta)$)
- A log-sum-exp part (denoted $f_2(\theta)$)

$$\mathcal{L}_{CE} = \underbrace{-\frac{1}{n} \sum_{i=1}^n \theta_{y_i}^\top z_i + \frac{\lambda}{2} \sum_{k=1}^K \|\theta_k\|^2}_{f_1(\theta)} + \underbrace{\frac{1}{n} \sum_{i=1}^n \log \left(\sum_{j=1}^K \exp(\theta_j^\top z_i) \right) + \frac{\lambda}{2} \sum_{k=1}^K \|\theta_k\|^2}_{f_2(\theta)}$$

(2) Also, derive the gradients with respect to θ_k for both parts.

- (c) Demonstrate that both $f_1(\theta)$ and $f_2(\theta)$ are convex functions in θ . In particular, for $f_2(\theta)$, you are encouraged to show that its Hessian matrix is positive semidefinite using the Diagonal Dominance Theorem.

Hint: A symmetric matrix M is positive semidefinite if for every index i ,

$$M_{ii} \geq \sum_{j \neq i} |M_{ij}|$$

- (d) Using the convexity properties obtained in Subproblem (c) and the gradient expressions from Subproblem (2), show that the following inequality holds:

$$\mathcal{L}_{CE} \geq -\frac{1}{2\lambda n^2} \sum_{i=1}^n \sum_{j:y_i=y_j} z_i^\top z_j + \frac{1}{n} \sum_{i=1}^n \log \sum_{k=1}^K \exp \left(-\frac{1}{\lambda n} \sum_{j=1}^n p_{jk} z_i^\top z_j \right) + \frac{1}{2\lambda} \sum_{k=1}^K \|c_k^s\|^2$$

where

$$c_k^s = \frac{1}{n} \sum_{i=1}^n p_{ik} z_i$$

Then, explain briefly why letting $\lambda \rightarrow 0$ shows that minimizing the cross-entropy loss approximately minimizes a pairwise metric-learning loss.

2. Construction of mini-batches in training

Consider a multi-class problem where you have a total of L classes. Suppose L is very large. It is desirable for the triplet loss to involve negative examples across all classes but it is impractical in the case when the number of output classes L is large; even if we restrict the number of negative examples per class to one, it is still computation expensive to perform the optimization, such as stochastic gradient descent (SGD).

Now, you are asked to design an effective batch construction to avoid excessive computational burden. Let

$$\{(x_1, x_1^+), \dots, (x_N, x_N^+)\} \quad (1)$$

be N pairs of examples from N different classes, i.e., labels $y_i \neq y_j, \forall i \neq j$. Please use these N pairs of examples to build N distinct $(N + 1)$ -tuplets, denoted as $\{S_i\}_{i=1}^N$, to form batches in training. **Write out your design of $\{S_i\}_{i=1}^N$.**

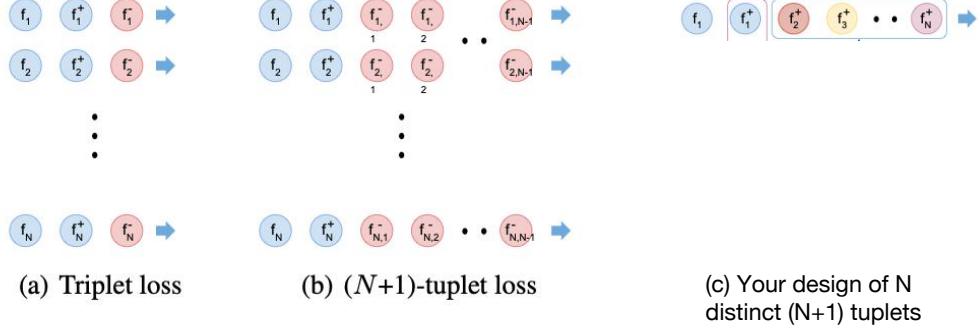


Figure 1: Let $\{f_i\}_{i=1}^N$ be queries. For a batch consisting of N distinct queries, triplet loss requires $3N$ passes to evaluate the necessary embedding vectors, $(N + 1)$ -tuple loss requires $(N + 1)N$ passes and your design only requires $2N$.

2 Gaussian Process Regression [30 pts]

1. Show ridge regression can be considered as a special case of Gaussian process regression
2. Show for the noise-free case, i.e., we observe the function value $f(x)$ directly, the functions sampled from the posterior GP actually interpolate the observed data points. *Hint.* you show the for the observed data point x , the mean is $f(x)$ and the variance is zero.
3. Reproduce Figure 17.7 by specifying a different GP, i.e., you need to specify the mean function and the kernel function, and add random observation points, plot the case for two points, 4 points and 8 points.
4. Derive Equations (17.44-46) using the Sherman-Morrison-Woodbury formula for matrix inversion.

$$p(f_* | \mathcal{D}, \mathbf{x}_*) = \mathcal{N}(f_* | \boldsymbol{\mu}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \quad (17.44)$$

$$\boldsymbol{\mu}_{*|X} = \phi_*^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}^T (\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \mathbf{y} = \mathbf{k}_*^T \mathbf{K}_\sigma^{-1} \mathbf{y} \quad (17.45)$$

$$\boldsymbol{\Sigma}_{*|X} = \phi_*^T \boldsymbol{\Sigma}_w \phi_* - \phi_*^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}^T (\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \boldsymbol{\Phi} \boldsymbol{\Sigma}_w \phi_* = k_{**} - \mathbf{k}_*^T \mathbf{K}_\sigma^{-1} \mathbf{k}_* \quad (17.46)$$

5. Reproduce Figure 17.9, first using 7 data points, and then use 50 data points, and compare the marginal likelihood as the number of data points increases.

3 SVM [15 pts]

Exercise 17.1 [Fitting an SVM classifier by hand]

Consider a dataset with 2 points in 1d: $x_1 = 0$ with label $y_1 = -1$ and $x_2 = \sqrt{2}$ with label $y_2 = 1$. Consider mapping each point to 3d using the feature vector $\phi(x) = [1, \sqrt{2}x, x^2]^T$. (This is equivalent to using a second order polynomial kernel.) The max margin classifier has the form

$$\min \|\mathbf{w}\|^2 \quad \text{s.t.} \quad (17.119)$$

$$y_1(\mathbf{w}^T \phi(x_1) + w_0) \geq 1 \quad (17.120)$$

$$y_2(\mathbf{w}^T \phi(x_2) + w_0) \geq 1 \quad (17.121)$$

- a. Write down a vector that is parallel to the optimal vector \mathbf{w} . Hint: recall from Figure 17.12(a) that \mathbf{w} is perpendicular to the decision boundary between the two points in the 3d feature space.
- b. What is the value of the margin that is achieved by this \mathbf{w} ? Hint: recall that the margin is the distance from each support vector to the decision boundary. Hint 2: think about the geometry of 2 points in space, with a line separating one from the other.
- c. Solve for \mathbf{w} , using the fact that the margin is equal to $1/\|\mathbf{w}\|$.
- d. Solve for w_0 using your value for \mathbf{w} and Equations 17.119 to 17.121. Hint: the points will be on the decision boundary, so the inequalities will be tight.
- e. Write down the form of the discriminant function $f(x) = w_0 + \mathbf{w}^T \phi(x)$ as an explicit function of x .

4 Trees and Boosting [30 pts]

1. What are the similarities and key differences between AdaBoost and gradient boosting? Can you briefly outline how to fit a tree for training data with weights?
2. Using XGBoost in Python: Please use XGBoost to solve a regression problem on the Boston Housing data.

The dataset is taken from the UCI Machine Learning Repository and is also present in sklearn's datasets module. It has 14 explanatory variables describing various aspects of residential homes in Boston, the challenge is to predict the median value of owner-occupied homes per \$1000s.

Please build a XGBoost model based on a 3-fold cross validation on the original dataset. Report your prediction results using "cv_results" and visualize the boosting tree using "plot_tree()".

Hint: See Tutorial for more guidance.

5 VAE (25 pts)

1. Fisher identity. Let

$$p_\theta(x) = \int p_\theta(x, z) dz, \quad p_\theta(z|x) = p_\theta(x, z)/p_\theta(x)$$

Show

$$\frac{\partial}{\partial \theta} \log p_\theta(x) = \mathbf{E}_{p_\theta(z|x)} \left[\frac{\partial}{\partial \theta} \log p_\theta(x, z) \right]$$

2. In VAE, let $p_{\text{data}}(x)$ be the underlying data distribution, We can approach the joint distribution for (x, z) in two different ways

$$p_\theta(x, z) \equiv p_\theta(x|z)p_\theta(z), \quad q_\phi(x, z) \equiv p_{\text{data}}(x)q_\phi(z | x)$$

- Show that

$$\begin{aligned} \text{KL}(q_\phi(x, z) \| p_\theta(x, z)) &= -\mathbf{E}_{p_{\text{data}}(x)} [\mathbf{E}_{q_\phi(z|x)} [\log p_\theta(x, z) - \log q_\phi(z | x)] - \log p_{\text{data}}(x)] = \\ &= -\mathbf{E}_{p_{\text{data}}(x)} \text{ELBO}(\theta, \phi, x) + \text{const} \end{aligned}$$

- Also show that

$$\text{KL}(q_\phi(x, z) \| p_\theta(x, z)) = \text{KL}(p_{\text{data}}(x) \| p_\theta(x)) + \mathbf{E}_{p_{\text{data}}(x)} \text{KL}(q_\phi(z | x) \| p_\theta(z | x))$$

6 Mixture Models (20 pts)

- Consider a D -dimensional variable \mathbf{x} each of whose components i is itself a multinomial variable of degree M so that \mathbf{x} is a binary vector with components x_{ij} where $i = 1, \dots, D$ and $j = 1, \dots, M$, subject to the constraint that $\sum_j x_{ij} = 1$ for all i . Suppose that the distribution of these variables is described by a mixture of the discrete multinomial distributions so that

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k p(\mathbf{x} | \boldsymbol{\mu}_k)$$

where

$$p(\mathbf{x} | \boldsymbol{\mu}_k) = \prod_{i=1}^D \prod_{j=1}^M \mu_{kij}^{x_{ij}}.$$

The parameters μ_{kij} represent the probabilities $p(x_{ij} = 1 | \boldsymbol{\mu}_k)$ and must satisfy $0 \leq \mu_{kij} \leq 1$ together with the constraint $\sum_j \mu_{kij} = 1$ for all values of k and i . Given an observed data set $\{\mathbf{x}_n\}$, where $n = 1, \dots, N$, derive the **E** and **M** step equations of the EM algorithm for optimizing the mixing coefficients π_k and the component parameters μ_{kij} of this distribution by maximum likelihood.