AI6102 Assignment 1

Sourabh Vyas

16/09/2025

Question 1 (10 marks)

Consider a multi-class classification problem of C classes. Based on the parametric forms of the conditional probabilities of each class introduced on the 39th Page ("Extension to Multiple Classes") of the lecture notes of L4, derive the learning procedure of regularized logistic regression for multi-class classification problems.

Hint: define a loss function by borrowing an idea from binary classification, and derive the gradient descent rules to update $\{\mathbf{w}^{(c)}\}$ for $c = 1, \ldots, C-1$.

Answer

1. Conditional Probabilities We treat class 0 as the baseline and model the others via a "softmax" over negative scores. Then for $c = 0, \ldots, C-1$:

$$P(y = 0 \mid \mathbf{x}) = \frac{1}{1 + \sum_{k=1}^{C-1} \exp\big(-\mathbf{w}^{(k)\top}\mathbf{x}\big)}, \quad P(y = c \mid \mathbf{x}) = \frac{\exp\big(-\mathbf{w}^{(c)\top}\mathbf{x}\big)}{1 + \sum_{k=1}^{C-1} \exp\big(-\mathbf{w}^{(k)\top}\mathbf{x}\big)}.$$

This ensures all C probabilities sum to 1.

2. Multiclass Negative Log Likelihood We assume each example (x_i, y_i) is drawn independently, so the likelihood of all labels given all inputs is

$$L(\{\mathbf{w}^{(c)}\}) = \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i)$$

Taking the logarithm turns the product into a sum:

$$\ell(\{\mathbf{w}^{(c)}\}) = \sum_{i=1}^{N} \log P(y_i \mid \mathbf{x}_i)$$

Two cases for each i:
If $y = c \mid c = 1, \ldots, C-1$:

$$P\big(y_i = c \mid \mathbf{x}_i\big) = \frac{\exp\big(-\mathbf{w}^{(c)\top}\mathbf{x}_i\big)}{D_i}, \quad D_i = 1 + \sum_{k=1}^{C-1} \exp\big(-\mathbf{w}^{(k)\top}\mathbf{x}_i\big)$$

Taking its log

$$\log P(y_i = c \mid \mathbf{x}_i) = -\,\mathbf{w}^{(c)\top}\mathbf{x}_i - \log D_i$$

If $y_i = 0$:

$$P(y_i = 0 \mid \mathbf{x}_i) = \frac{1}{D_i}, \quad \log P(y_i = 0 \mid \mathbf{x}_i) = -\log D_i$$

Introduce $\mathbb{I}(y_i = c)$ which is 1 when $y_i = c$. Then for each i

$$-\log P(y_i \mid \mathbf{x}_i) = \log D_i + \sum_{c=1}^{C-1} \mathbb{I}(y_i = c)\, \mathbf{w}^{(c)\top} \mathbf{x}_i$$

Summing the negative log-probabilities over $i = 1, \ldots, N$ yields the total unregularized loss:

$$\mathcal{L} = \sum_{i=1}^{N} \Big[\log\big(1 + \sum_{k=1}^{C-1} e^{-\mathbf{w}^{(k)\top} \mathbf{x}_i}\big) + \sum_{c=1}^{C-1} \mathbb{I}(y_i = c)\, \mathbf{w}^{(c)\top} \mathbf{x}_i \Big]$$

3. Regularized Negative Log-Likelihood The unregularized negative log-likelihood is

$$\mathcal{L} = \sum_{i=1}^{N} \Big[\log\big(1 + \sum_{k=1}^{C-1} e^{-\mathbf{w}^{(k)\top} \mathbf{x}_i}\big) + \sum_{c=1}^{C-1} \mathbb{I}(y_i = c)\, \mathbf{w}^{(c)\top} \mathbf{x}_i \Big]$$

Add an ℓ_2 penalty:

$$\mathcal{L}_{\text{reg}} = \mathcal{L} \; + \; \frac{\lambda}{2} \sum_{c=1}^{C-1} \big\| \mathbf{w}^{(c)} \big\|^2.$$

4. Gradient Computation Regularized negative log-likelihood for class c:

$$\mathcal{L}_{\text{reg}} = \sum_{i=1}^{N} \Big[\log\big(1 + \sum_{k=1}^{C-1} e^{-\mathbf{w}^{(k)\top} \mathbf{x}_i}\big) + \sum_{k=1}^{C-1} \mathbb{I}(y_i = k)\, \mathbf{w}^{(k)\top} \mathbf{x}_i \Big] + \frac{\lambda}{2} \sum_{k=1}^{C-1} \|\mathbf{w}^{(k)}\|^2$$

Consider the loss for a single example i:

$$\ell_i = \log D_i + \sum_{k=1}^{C-1} \mathbb{I}(y_i = k)\, \mathbf{w}^{(k)\top} \mathbf{x}_i, \quad D_i = 1 + \sum_{k=1}^{C-1} e^{-\mathbf{w}^{(k)\top} \mathbf{x}_i}$$

Taking thr gradient w.r.t $\mathbf{w}^{(c)}$ splits into two: Derivative of $\log D_i$:

$$\frac{\partial}{\partial \mathbf{w}^{(c)}} \log D_i = \frac{1}{D_i} \frac{\partial}{\partial \mathbf{w}^{(c)}} \big(1 + \sum_{k} e^{-\mathbf{w}^{(k)\top} \mathbf{x}_i}\big) = -\frac{e^{-\mathbf{w}^{(c)\top} \mathbf{x}_i}}{D_i}\, \mathbf{x}_i = -P(y = c \mid \mathbf{x}_i)\, \mathbf{x}_i$$

Derivative of indicator term:

$$\frac{\partial}{\partial \mathbf{w}^{(c)}} \big[\mathbb{I}(y_i = c)\, \mathbf{w}^{(c)\top} \mathbf{x}_i \big] = \mathbb{I}(y_i = c)\, \mathbf{x}_i$$

Combining these two for an i:

$$\frac{\partial \ell_i}{\partial \mathbf{w}^{(c)}} = \big(\mathbb{I}(y_i = c) - P(y = c \mid \mathbf{x}_i) \big)\, \mathbf{x}_i$$

Summing the per-example gradients over $i = 1, \ldots, N$ yields

$$\nabla_{\mathbf{w}^{(c)}} \mathcal{L} = \sum_{i=1}^{N} \big(\mathbb{I}(y_i = c) - P(y = c \mid \mathbf{x}_i) \big)\, \mathbf{x}_i$$

Gradient for regularizer

$$\frac{\partial (\frac{\lambda}{2} \big\| \mathbf{w}^{(c)} \big\|^2)}{\partial \mathbf{w}^{(c)}} = \lambda \mathbf{w}^{(c)}$$

Adding the ℓ_2 Regularizer

$$\nabla_{\mathbf{w}^{(c)}} \mathcal{L}_{\text{reg}} = \sum_{i=1}^{N} \big(\mathbb{I}(y_i = c) - P(y = c \mid \mathbf{x}_i) \big)\, \mathbf{x}_i + \lambda\, \mathbf{w}^{(c)}$$

5. Gradient Descent Update With learning rate $\eta > 0$, update each class-weight:

$$\mathbf{w}^{(c)} \leftarrow \mathbf{w}^{(c)} - \eta\Big[\nabla_{\mathbf{w}^{(c)}}\mathcal{L}_{\text{reg}}\Big]$$

$$\mathbf{w}^{(c)} \leftarrow \mathbf{w}^{(c)} - \eta\Big[\sum_{i=1}^{N}\big(\mathbb{I}(y_i = c) - P(y = c \mid \mathbf{x}_i)\big)\,\mathbf{x}_i + \lambda\,\mathbf{w}^{(c)}\Big]$$

5. Prediction Rule For a new input $\mathbf{x}^*$,

$$y^* = \arg\max_{c\in\{0,\ldots,C-1\}} P(y = c \mid \mathbf{x}^*).$$

1 Question 2 (5 marks)

This is a hands-on exercise to use the SVC API of scikit-learn1 to train a SVM with the linear kernel and the rbf kernel, respectively, on a binary classification dataset. The details of instructions are described as follows.

1.1 Download the a9a dataset ... Detailed information is available here.

1.2 Regarding the linear kernel, show 3-fold cross-validation ... training set (in accuracy).

Table 1: The 3-fold cross-validation results of varying values of C in SVC with linear kernel on the a9a training set (in accuracy).

$C = 0.01$	$C = 0.05$	$C = 0.1$	$C = 0.5$	$C = 1$
0.8435	84.70%	84.72%	84.80%	84.83%

Parameters:

- `StratifiedKFold`:
  - `n_splits` = 3
  - `shuffle` = `True`
  - `random_state` = 64
- `SVC`:
  - `kernel` = `"linear"`
  - `C` takes values in {0.01, 0.05, 0.1, 0.5, 1}
- `cross_val_score`:
  - `cv` = `skf` (the StratifiedKFold)
  - `scoring` = `"accuracy"`

1.3 Regarding the rbf kernel, show 3-fold ... Some examples can be found here.

Table 2: The 3-fold cross-validation results of varying values of γ and C in SVC with RBF kernel on the a9a training set (in accuracy).

$C \backslash \gamma$	0.01	0.05	0.1	0.5	1
0.01	75.92%	81.97%	81.96%	75.92%	75.92%
0.05	83.04%	83.60%	83.41%	78.97%	75.92%
0.1	83.74%	83.96%	83.90%	80.63%	76.10%
0.5	84.33%	84.51%	84.71%	83.33%	78.95%
1	84.48%	84.72%	84.76%	83.74%	79.94%

1.4 Based on the results shown in Tables ... the following table:

Table 3: Test results of SVC on the a9a test set (in accuracy).

Kernel	Parameter setting	Test accuracy
RBF	$C = 1,\ \gamma = 0.1$	85.03%

Question 3 (5 marks)

The optimization problem of linear soft-margin SVMs can be re-formulated as an instance of empirical structural risk minimization (refer to Page 37 on L5 notes). Show how to reformulate it. Hint: search reference about the hinge loss.

Answer

1. The standard soft-margin SVM solves

$$\min_{w,b,\{\xi_i\}} \ \frac{1}{2}\|w\|_2^2 \ + \ C\sum_{i=1}^{N}\xi_i \quad \text{subject to} \quad y_i\big(w\cdot x_i + b\big) \geq 1-\xi_i,\ \xi_i \geq 0$$

2. Introduce the hinge loss

$$\ell_{\text{hinge}}\big(f(x_i), y_i\big) = \max\big(0,\ 1 - y_i\, f(x_i)\big),$$

where $f(x) = wx + b$
At optimum $\xi_i = \max\big(0,\ 1 - y_i\, f(x_i)\big)$ Substituting back gives the unconstrained form

$$\min_{w,b} \ \frac{1}{2}\|w\|_2^2 \ + \ C\sum_{i=1}^{N}\max\big(0,\ 1 - y_i(w\cdot x_i + b)\big)$$

3. Empirical Risk Minimization The empirical risk minimization template is

$$\hat{\theta} = \arg\min_{\theta} \ \sum_{i=1}^{N}\ell\big(f(x_i;\theta), y_i\big) \ + \ \lambda\,\Omega(\theta).$$

$$\text{Set } \theta = (w,b), \quad \ell(f(x), y) = \max\big(0,\, 1 - y\, f(x)\big), \quad \Omega(\theta) = \tfrac{1}{2}\|w\|_2^2, \quad \lambda = \tfrac{1}{C}.$$

Question 4 (5 marks)

Using the kernel trick introduced in L5 to extend the regularized linear regression model (L3) to solve nonlinear regression problems. Derive a closed-form solution (i.e., to derive a kernelized version of the closed-form solution on Page 50 of L3).

Answer

1. The primal objective is

$$\min_{w \in \mathbb{R}^d} \; J(w) \;=\; \|Xw - y\|_2^2 \;+\; \lambda \, \|w\|_2^2.$$

By the representer theorem, the minimizer $w*$ can be expressed in the span of the rows of X. Introduce a coefficient vector $\alpha \in \mathbb{R}^N$:

$$w = X^T \alpha, \qquad \alpha \in \mathbb{R}^N.$$

2. Objective in Terms of *alpha* and Differentiation Define the kernel (Gram) matrix $K = XX^T \in \mathbb{R}^{N \times N}$. Substitute $w = X^T \alpha$ into $J(w)$:

$$\begin{aligned} J(\alpha) = \|X(X^T\alpha) - y\|_2^2 + \lambda \, \|X^T\alpha\|_2^2 = \|K\alpha - y\|_2^2 + \lambda \, \alpha^T (XX^T)\alpha \\ = \alpha^T K^2 \alpha - 2 \, y^T K \alpha + y^T y + \lambda \, \alpha^T K \alpha. \end{aligned}$$

Differentiate $J(\alpha)$ wrt α:

$$\begin{aligned} \nabla_\alpha J(\alpha) &= 2 \, K \, (K\alpha - y) \;+\; 2 \, \lambda \, K \, \alpha \\ &= 2 \, K \big(K\alpha - y + \lambda \, \alpha \big). \end{aligned}$$

Set $\nabla_\alpha J(\alpha) = 0$

$$K \big(K\alpha - y + \lambda \, \alpha \big) = 0 \implies (K + \lambda I) \, \alpha = y.$$

3. Solving for α yields the dual(kernel) closed form

$$\alpha = (K + \lambda I)^{-1} \, y.$$

Substitute back to get the primal weight vector:

$$w = X^T \alpha = X^T (K + \lambda I)^{-1} \, y = X^T (XX^T + \lambda I)^{-1} \, y.$$