

Multi-Class Classification

Sanjay Lall and Stephen Boyd

EE104

Stanford University

Multi-class classification

- ▶ multi-class classification with $\mathcal{V} = \{1, \dots, K\}$
- ▶ embed the K classes as $\psi_1, \dots, \psi_K \in \mathbf{R}^m$
- ▶ use nearest-neighbor un-embedding, $\hat{v} = \operatorname{argmin}_i \|\hat{y} - \psi_i\|_2$
- ▶ use RERM to fit predictor
- ▶ validate using Neyman-Pearson metric on test data
 - ▶ Neyman-Pearson metric is $\sum_j \kappa_j \mathbf{E}_j$
 - ▶ \mathbf{E}_j is rate of mistaking $v = j$
 - ▶ κ_j is our relative distaste for mistaking $v = j$
 - ▶ with $\kappa_1 = \dots = \kappa_K = 1$, reduces to error rate

Signed distances

When is a vector closer to one given vector than another?

► when is $\hat{y} \in \mathbf{R}^m$ closer to a than b , where $a \neq b$?

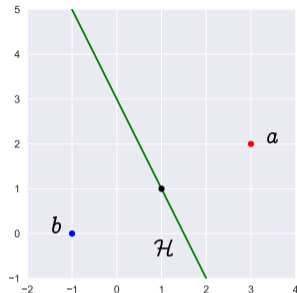
► square both sides of $\|\hat{y} - a\|_2 < \|\hat{y} - b\|_2$ to get

$$2(b - a)^\top \hat{y} - \|b\|_2^2 + \|a\|_2^2 < 0$$

► the *decision boundary* is given by $\|\hat{y} - a\|_2 = \|\hat{y} - b\|_2$, i.e.,

$$2(b - a)^\top \hat{y} - \|b\|_2^2 + \|a\|_2^2 = 0$$

► this defines a *hyperplane* \mathcal{H} in \mathbf{R}^m , with *normal vector* $b - a$, passing through the midpoint $(a + b)/2$

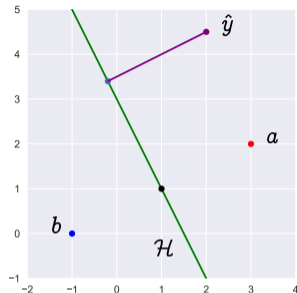


Signed distance to the decision boundary

- ▶ the *signed distance* of \hat{y} to \mathcal{H} is

$$D(\hat{y}, a, b) = \frac{2(b-a)^\top \hat{y} - \|a\|_2^2 + \|b\|_2^2}{2\|b-a\|_2}$$

- ▶ $D(\hat{y}, a, b) < 0$ when \hat{y} is closer to a than b
- ▶ $|D(\hat{y}, a, b)|$ is the distance of \hat{y} to \mathcal{H}
- ▶ $D(\hat{y}, a, b) = 0$ gives the decision boundary
- ▶ $D(\hat{y}, a, b)$ is an affine function of \hat{y}



Signed distances

- ▶ now consider un-embedding the prediction $\hat{y} \in \mathbf{R}^m$, i.e., finding which ψ_i is closest
- ▶ define signed distance functions D_{ij} , for $i \neq j$, as

$$D_{ij}(\hat{y}) = D(\hat{y}, \psi_i, \psi_j) = \frac{2(\psi_j - \psi_i)^\top \hat{y} - \|\psi_i\|_2^2 + \|\psi_j\|_2^2}{2\|\psi_j - \psi_i\|_2}$$

- ▶ $D_{ij}(\hat{y}) < 0$ means \hat{y} is closer to ψ_i than ψ_j
- ▶ \hat{y} is closest to ψ_i when $D_{ij} < 0$ for $j \neq i$, or

$$\max_{j \neq i} D_{ij}(\hat{y}) < 0$$

- ▶ loss function should encourage this, when $y = \psi_i$

Examples

- ▶ Boolean, with $\psi_1 = -1$ and $\psi_2 = 1$

$$D_{12}(\hat{y}) = \hat{y}, \quad D_{21}(\hat{y}) = -\hat{y}$$

so, when $y = -1$, we'd like $\hat{y} < 0$; when $y = +1$, we'd like $\hat{y} > 0$

- ▶ one-hot, with $\psi_j = e_i$, $j = 1, \dots, K$

$$D_{ij} = \frac{y_j - y_i}{\sqrt{2}}, \quad i \neq j$$

so, when $y = e_i$, we want $\max_{j \neq i} D_{ij}(\hat{y}) < 0$, i.e., $\operatorname{argmax}_j \hat{y}_j = i$

Multi-class loss functions

Loss function for multi-class classification

- ▶ we need to give the K functions of \hat{y}

$$\ell(\hat{y}, \psi_i), \quad i = 1, \dots, K$$

- ▶ $\ell(\hat{y}, \psi_i)$ is how much we dislike predicting \hat{y} when $y = \psi_i$
- ▶ loss function $\ell(\hat{y}, \psi_i)$ should be
 - ▶ small when $\max_{j \neq i} D_{ij}(\hat{y}) < 0$
 - ▶ larger when $\max_{j \neq i} D_{ij}(\hat{y}) \nearrow 0$

Neyman-Pearson loss

- ▶ Neyman-Pearson loss is

$$\ell(\hat{y}, \psi_i) = \begin{cases} 0 & \max_{j \neq i} D_{ij} < 0 \\ \kappa_i & \text{otherwise} \end{cases}$$

i.e., zero when ψ_i is decoded correctly, κ_i otherwise

- ▶ it's hard to minimize $\mathcal{L}(\theta)$
- ▶ we do better with a *proxy loss* that
 - ▶ approximates, or at least captures the flavor of, the Neyman-Pearson loss
 - ▶ is more easily optimized (e.g., is convex, differentiable)

Multi-class hinge loss

- ▶ *hinge loss* is

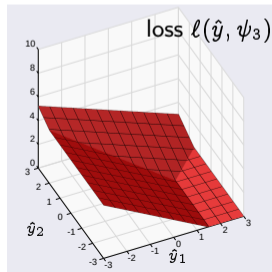
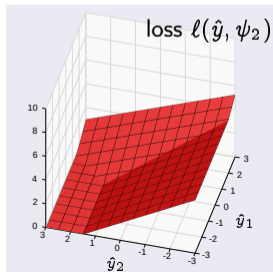
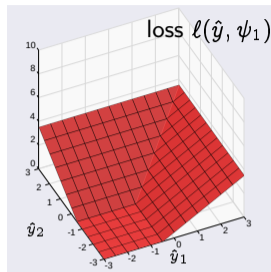
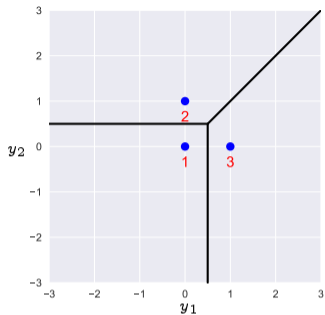
$$\ell(\hat{y}, \psi_i) = \kappa_i \max_{j \neq i} (1 + D_{ij}(\hat{y}))_+$$

- ▶ $\ell(\hat{y}, \psi_i)$ is zero when \hat{y} is correctly un-embedded, with a margin at least one
- ▶ convex but not differentiable
- ▶ with quadratic regularization, called *multi-class SVM*
- ▶ for Boolean embedding with $\psi_1 = -1$, $\psi_2 = 1$, reduces to

$$\ell(\hat{y}, -1) = \kappa_1 (1 + \hat{y})_+, \quad \ell(\hat{y}, 1) = \kappa_2 (1 - \hat{y})_+$$

usual hinge loss when $\kappa_1 = 1$

Multi-class hinge loss



Multi-class logistic loss

- ▶ *logistic loss* is

$$\ell(\hat{y}, \psi_i) = \kappa_i \log \left(\sum_{j=1}^K \exp(D_{ij}(\hat{y})) \right)$$

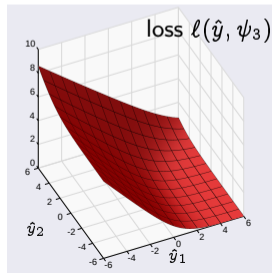
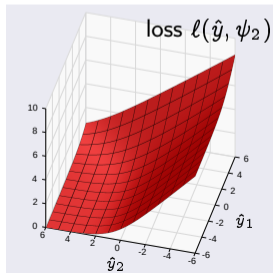
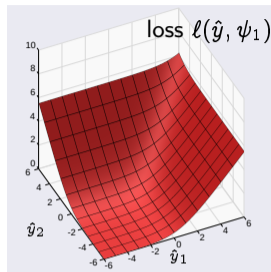
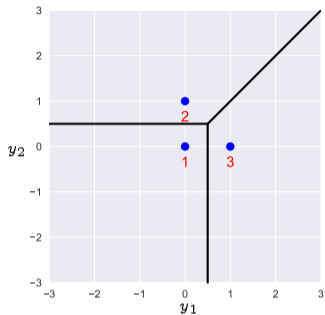
(where we take $D_{jj} = 0$)

- ▶ convex and differentiable
- ▶ called *multi-class logistic regression*
- ▶ for Boolean embedding with $\psi_1 = -1$, $\psi_2 = 1$, reduces to

$$\ell(\hat{y}, -1) = \kappa_1 \log(1 + e^{\hat{y}}), \quad \ell(\hat{y}, 1) = \kappa_2 \log(1 + e^{-\hat{y}})$$

usual logistic loss when $\kappa_1 = 1$

Multi-class logistic loss



Log-sum-exp function

- ▶ the function $f : \mathbf{R}^n \rightarrow \mathbf{R}$

$$f(\mathbf{x}) = \log \sum_{i=1}^n \exp(x_i)$$

is called the *log-sum-exp* function

- ▶ it is a convex differentiable approximation to the max function
- ▶ sometimes called the *softmax* function; but that term is also used for other functions
- ▶ we have

$$\max\{x_1, \dots, x_n\} \leq f(\mathbf{x}) \leq \max\{x_1, \dots, x_n\} + \log(n)$$

Example: Iris

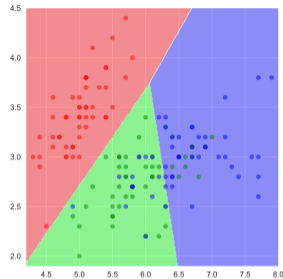
Example: Iris

- ▶ famous example dataset by Fisher, 1936
- ▶ measurements of 150 plants, 50 from each of 3 species
- ▶ iris setosa, iris versicolor, iris virginica
- ▶ four measurements: sepal length, sepal width, petal length, petal width

Example: Iris



Classification with two features



- ▶ using only `sepal_length` and `sepal_width`
- ▶ one-hot embedding, multi-class logistic loss with $\kappa_i = 1$ for all i , trained on all data

- ▶ confusion matrix $C = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 38 & 13 \\ 0 & 12 & 37 \end{bmatrix}$

Classification with all four features

- ▶ use all four features
- ▶ one-hot embedding, multi-class logistic loss with $\kappa_i =$ for all i , trained on all data

- ▶ confusion matrix $C = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 49 & 1 \\ 0 & 1 & 49 \end{bmatrix}$

Summary

Summary

- ▶ loss functions for multi-class classification should encourage correct un-embedding, *i.e.*,
 - ▶ $\ell(\hat{y}, \psi_i)$ is small when \hat{y} is closest to ψ_i
 - ▶ $\ell(\hat{y}, \psi_i)$ is not small when \hat{y} is not closest to ψ_i
- ▶ most common losses are multi-class hinge loss and multi-class logistic
 - ▶ associated classifiers are called multi-class SVM and multi-class logistic
 - ▶ both losses are convex, so easy to solve ERM or RERM problems