MACHINE LEARNING FUNDAMENTALS - LS2026
SEMINAR: LINEAR MODELS

CZECH TECHNICAL UNIVERSITY IN PRAGUE
V. FRANC

Assignment 1. Given i.i.d. training samples $T_m = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \in (\mathbb{R}^d \times \mathbb{R})^m$, consider the class of linear regressors $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$. Formulate the learning task as an empirical risk minimization problem with the quadratic loss $\ell(y, \hat{y}) = (y - \hat{y})^2$. Then derive the optimality conditions and show that the resulting problem has a closed-form solution, provided the required matrix is invertible.

Solution 1. We consider the class of linear regressors

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b, \qquad \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}.$$

Given the training sample T_m, the empirical risk under the quadratic loss $\ell(y, \hat{y}) = (y - \hat{y})^2$ is

$$\widehat{R}(T_m, \mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \big(h(\mathbf{x}_i) - y_i\big)^2 = \frac{1}{m} \sum_{i=1}^{m} \big(\mathbf{w}^\top \mathbf{x}_i + b - y_i\big)^2.$$

Hence, learning the linear regressor can be formulated as the empirical risk minimization problem

$$(\mathbf{w}_m, b_m) \in \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^d,\, b \in \mathbb{R}} \widehat{R}(T_m, \mathbf{w}, b) = \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^d,\, b \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^{m} \big(\mathbf{w}^\top \mathbf{x}_i + b - y_i\big)^2 = \operatorname*{arg\,min}_{\mathbf{w},\, b} \sum_{i=1}^{m} \big(\mathbf{w}^\top \mathbf{x}_i + b - y_i\big)^2.$$

Matrix form. Introduce the augmented parameter vector

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \in \mathbb{R}^{d+1},$$

the design matrix

$$X = \begin{bmatrix} \mathbf{x}_1^\top & 1 \\ \vdots & \vdots \\ \mathbf{x}_m^\top & 1 \end{bmatrix} \in \mathbb{R}^{m \times (d+1)},$$

and the target vector

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m.$$

Then the vector of predictions is $X\boldsymbol{\theta}$, and the optimization problem becomes

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{d+1}} \frac{1}{m} \|X\boldsymbol{\theta} - \mathbf{y}\|_2^2.$$

Optimality conditions. Consider the objective

$$f(\boldsymbol{\theta}) = \frac{1}{m}\|X\boldsymbol{\theta} - \mathbf{y}\|_2^2 = \frac{1}{m}(X\boldsymbol{\theta} - \mathbf{y})^\top(X\boldsymbol{\theta} - \mathbf{y}).$$

Its gradient is

$$\nabla f(\boldsymbol{\theta}) = \frac{2}{m}X^\top(X\boldsymbol{\theta} - \mathbf{y}).$$

A necessary condition for optimality is therefore

$$X^\top(X\boldsymbol{\theta} - \mathbf{y}) = 0,$$

which yields the *normal equations*

$$X^\top X\,\boldsymbol{\theta} = X^\top\mathbf{y}.$$

Closed-form solution. If the matrix $X^\top X$ is invertible, then the normal equations have the unique solution

$$\boldsymbol{\theta}_m = (X^\top X)^{-1}X^\top\mathbf{y}.$$

Therefore,

$$\begin{bmatrix}\mathbf{w}_m \\ b_m\end{bmatrix} = (X^\top X)^{-1}X^\top\mathbf{y}.$$

Thus, provided $X^\top X$ is invertible, the empirical risk minimization problem for linear regression with quadratic loss admits a closed-form solution.

Assignment 2. Given a linearly separable training sample $T_m = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$, finding parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ such that the generic linear classifier

$$h(x; \boldsymbol{\theta}) \in \arg\max_{y \in \mathcal{Y}} \boldsymbol{\theta}^\top \boldsymbol{\phi}(x, y)$$

achieves zero training error on T_m, i.e. $h(x_i; \boldsymbol{\theta}) = y_i, \qquad \forall i \in \{1, \ldots, m\}$, can be accomplished by the Perceptron Learning Algorithm (PLA).

Perceptron Learning Algorithm

---

(1) Initialize

$$\boldsymbol{\theta} \leftarrow \mathbf{0}.$$

(2) Find a misclassified example $(x_u, y_u) \in T_m$ such that

$$y_u \neq \hat{y}_u, \qquad \hat{y}_u = \arg\max_{y \in \mathcal{Y}} \boldsymbol{\theta}^\top \boldsymbol{\phi}(x_u, y).$$

(3) If no misclassified example exists, return $\boldsymbol{\theta}$. Otherwise update

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\phi}(x_u, y_u) - \boldsymbol{\phi}(x_u, \hat{y}_u),$$

and go back to step 2.

---

a) Instantiate the PLA for learning a binary linear classifier

$$\hat{y} = h(\mathbf{x}; \mathbf{w}, b) = \operatorname{sign}(\mathbf{w}^\top\mathbf{x} + b),$$

with parameters $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

b) Instantiate the PLA for learning a multiclass linear classifier

$$\hat{y} = h(\mathbf{x}; \mathbf{W}, \mathbf{b}) = \arg\max_{y \in \mathcal{Y}} \big(\mathbf{w}_y^\top \mathbf{x} + b_y\big),$$

with parameters

$$\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_Y) \in \mathbb{R}^{d \times Y}, \qquad \mathbf{b} = (b_1, \ldots, b_Y) \in \mathbb{R}^Y.$$

Solution 2. a) Binary linear classifier For the binary linear classifier, we use the generic representation

$$\mathcal{Y} = \{-1, +1\}, \qquad \boldsymbol{\theta} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \in \mathbb{R}^{d+1}, \qquad \boldsymbol{\phi}(\mathbf{x}, y) = \begin{bmatrix} y\mathbf{x} \\ y \end{bmatrix} \in \mathbb{R}^{d+1}.$$

Then

$$\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}, y) = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}^\top \begin{bmatrix} y\mathbf{x} \\ y \end{bmatrix} = y(\mathbf{w}^\top \mathbf{x} + b),$$

and therefore

$$h(\mathbf{x}; \boldsymbol{\theta}) = \arg\max_{y \in \{-1,+1\}} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}, y) = \arg\max_{y \in \{-1,+1\}} y(\mathbf{w}^\top \mathbf{x} + b) = \operatorname{sign}(\mathbf{w}^\top \mathbf{x} + b).$$

Hence, the Perceptron Learning Algorithm specializes as follows.

(1) Initialize

$$\mathbf{w} \leftarrow \mathbf{0}, \qquad b \leftarrow 0.$$

(2) Find a misclassified example $(\mathbf{x}_u, y_u) \in T_m$ such that

$$y_u \neq \hat{y}_u, \qquad \hat{y}_u = \operatorname{sign}(\mathbf{w}^\top \mathbf{x}_u + b).$$

Equivalently, find $(\mathbf{x}_u, y_u)$ such that

$$y_u(\mathbf{w}^\top \mathbf{x}_u + b) \leq 0.$$

(3) If no such example exists, return $(\mathbf{w}, b)$. Otherwise, perform the update

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\phi}(\mathbf{x}_u, y_u) - \boldsymbol{\phi}(\mathbf{x}_u, \hat{y}_u).$$

Since

$$\boldsymbol{\phi}(\mathbf{x}_u, y_u) = \begin{bmatrix} y_u \mathbf{x}_u \\ y_u \end{bmatrix}, \qquad \boldsymbol{\phi}(\mathbf{x}_u, \hat{y}_u) = \begin{bmatrix} \hat{y}_u \mathbf{x}_u \\ \hat{y}_u \end{bmatrix},$$

we obtain

$$\begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} + \begin{bmatrix} (y_u - \hat{y}_u)\mathbf{x}_u \\ y_u - \hat{y}_u \end{bmatrix}.$$

Therefore,

$$\mathbf{w} \leftarrow \mathbf{w} + (y_u - \hat{y}_u)\mathbf{x}_u, \qquad b \leftarrow b + (y_u - \hat{y}_u).$$

Since for a misclassified example we have $\hat{y}_u = -y_u$, it follows that

$$y_u - \hat{y}_u = y_u - (-y_u) = 2y_u.$$

Hence the update can be written as

$$\mathbf{w} \leftarrow \mathbf{w} + 2y_u \mathbf{x}_u, \qquad b \leftarrow b + 2y_u.$$

Up to a constant scaling factor, this is equivalent to the standard perceptron update

$$\mathbf{w} \leftarrow \mathbf{w} + y_u \mathbf{x}_u, \qquad b \leftarrow b + y_u.$$

Therefore, the PLA instantiated for the binary linear classifier is:

(1) Initialize
$$\mathbf{w} \leftarrow \mathbf{0}, \qquad b \leftarrow 0.$$
(2) Find $(\mathbf{x}_u, y_u) \in T_m$ such that
$$y_u(\mathbf{w}^\top \mathbf{x}_u + b) \leq 0.$$
(3) If no such example exists, return $(\mathbf{w}, b)$. Otherwise update
$$\mathbf{w} \leftarrow \mathbf{w} + y_u \mathbf{x}_u, \qquad b \leftarrow b + y_u,$$
and go back to step 2.

b) Multiclass linear classifier For the multiclass linear classifier, we use the generic representation $\mathcal{Y} = \{1, 2, \ldots, Y\}$, with parameter vector

$$\boldsymbol{\theta} = [\mathbf{w}_1, b_1, \mathbf{w}_2, b_2, \ldots, \mathbf{w}_Y, b_Y] \in \mathbb{R}^{(d+1)Y}.$$

The corresponding joint feature map is

$$\boldsymbol{\phi}(\mathbf{x}, y) = \big[[\mathbf{0}, 0], \ldots, \underbrace{[\mathbf{x}, 1]}_{y\text{-th block}}, \ldots, [\mathbf{0}, 0]\big] \in \mathbb{R}^{(d+1)Y}.$$

Then

$$\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}, y) = \mathbf{w}_y^\top \mathbf{x} + b_y,$$

and therefore

$$h(\mathbf{x}; \boldsymbol{\theta}) = \arg\max_{y \in \mathcal{Y}} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}, y) = \arg\max_{y \in \mathcal{Y}} (\mathbf{w}_y^\top \mathbf{x} + b_y).$$

Hence, the Perceptron Learning Algorithm specializes as follows.

(1) Initialize
$$\mathbf{w}_y \leftarrow \mathbf{0}, \qquad b_y \leftarrow 0, \qquad \forall y \in \mathcal{Y}.$$
(2) Find a misclassified example $(\mathbf{x}_u, y_u) \in T_m$ such that
$$y_u \neq \hat{y}_u, \qquad \hat{y}_u = \arg\max_{y \in \mathcal{Y}} (\mathbf{w}_y^\top \mathbf{x}_u + b_y).$$
(3) If no such example exists, return $(\mathbf{W}, \mathbf{b})$. Otherwise, perform the update
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\phi}(\mathbf{x}_u, y_u) - \boldsymbol{\phi}(\mathbf{x}_u, \hat{y}_u).$$
Since $\boldsymbol{\phi}(\mathbf{x}_u, y_u)$ inserts $[\mathbf{x}_u, 1]$ into the y_u-th block and $\boldsymbol{\phi}(\mathbf{x}_u, \hat{y}_u)$ inserts $[\mathbf{x}_u, 1]$ into the $\hat{y}_u$-th block, only the parameters of classes y_u and $\hat{y}_u$ are updated. Therefore,
$$\mathbf{w}_{y_u} \leftarrow \mathbf{w}_{y_u} + \mathbf{x}_u, \qquad b_{y_u} \leftarrow b_{y_u} + 1,$$
$$\mathbf{w}_{\hat{y}_u} \leftarrow \mathbf{w}_{\hat{y}_u} - \mathbf{x}_u, \qquad b_{\hat{y}_u} \leftarrow b_{\hat{y}_u} - 1,$$
while for all $y \notin \{y_u, \hat{y}_u\}$,
$$\mathbf{w}_y \leftarrow \mathbf{w}_y, \qquad b_y \leftarrow b_y.$$

Therefore, the PLA instantiated for the multiclass linear classifier is:

(1) Initialize

$$\mathbf{w}_y \leftarrow \mathbf{0}, \qquad b_y \leftarrow 0, \qquad \forall y \in \mathcal{Y}.$$

(2) Find $(\mathbf{x}_u, y_u) \in T_m$ such that

$$\hat{y}_u = \arg\max_{y \in \mathcal{Y}} (\mathbf{w}_y^\top \mathbf{x}_u + b_y), \qquad \hat{y}_u \neq y_u.$$

(3) If no such example exists, return $(\mathbf{W}, \mathbf{b})$. Otherwise update

$$\mathbf{w}_{y_u} \leftarrow \mathbf{w}_{y_u} + \mathbf{x}_u, \qquad b_{y_u} \leftarrow b_{y_u} + 1,$$
$$\mathbf{w}_{\hat{y}_u} \leftarrow \mathbf{w}_{\hat{y}_u} - \mathbf{x}_u, \qquad b_{\hat{y}_u} \leftarrow b_{\hat{y}_u} - 1,$$

and go back to step 2.