Cross Validation for Correlated Data in Classification Models

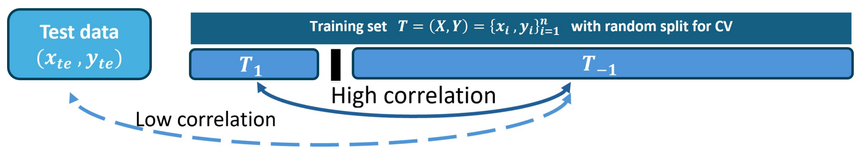
Oren Yuval Saharon Rosset

Department of Statistics and OR, Tel Aviv University

SCAN FOR PAPER

Motivation

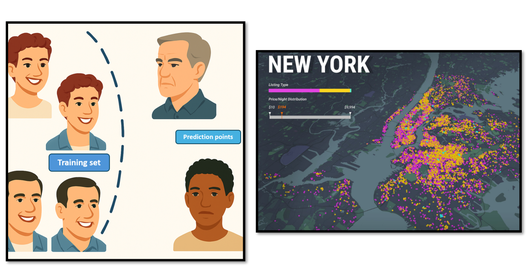
Standard cross-validation (CV) may result in a deceptive evaluation of the generalization error(GenErr) when the correlation structure within the training data does not match that between test data

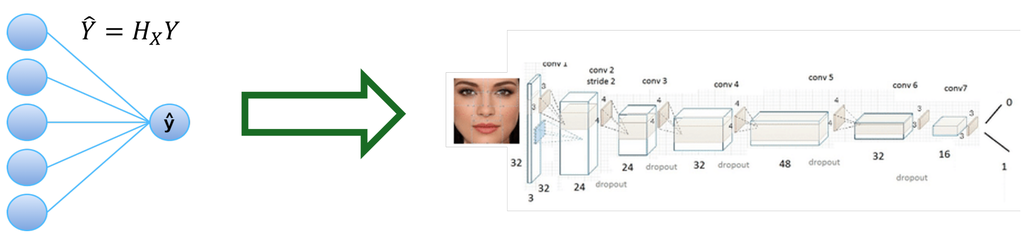


$$\text{GenErr} = \mathbb{E}\left[L\left(y_{te}, \hat{y}(x_{te}; T_{-1})\right)\right] \quad ; \quad \text{CV} = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, \hat{y}(x_i; T_{-i})\right) = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{cv}\right)$$

Such changes in correlation may occur in real-world situations:

- Data sets with clustered structures.
- Spatially-informed data sampling.
- Temporal-informed data sampling.



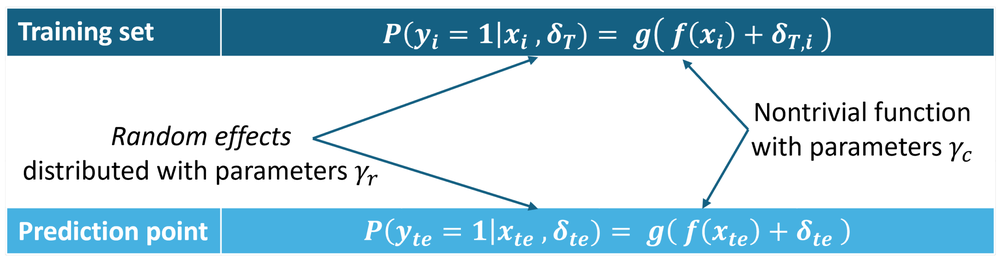


A recently introduced method addresses these issues by measuring and correcting bias in standard CV, focusing on linear models with squared loss.

We suggest a bias-corrected CV estimator that can be applied to any learning model, including deep neural networks, and to standard criteria for prediction performance for classification tasks.

Modeling the correlation with random effects

We consider a comprehensive framework that includes the GLMM and more advanced models like deep neural networks and decision trees.



In the case of data with spatial structure, the covariance between two observations is defined by a kernel function $\mathcal{K}(z_{i_1}, z_{i_2})$, on pairs of coordinates. In particular:

$$\begin{bmatrix}\delta_T \\ \delta_{te}\end{bmatrix} \sim \mathcal{MN}\left(0_{n+1}, K_{tot}\right) \;\; ; \;\; K_{tot,i_1,i_2} = \mathcal{K}(z_{i_1}, z_{i_2}) \;\; ; \;\; i_1, i_2 \in \{1, \cdots, n+1\}$$

If the data is gathered from randomly chosen geographical areas, this typically leads to a stronger correlation between observations within the training set compared to those from different sets.

Decomposing the loss function L in classification

Our observation: Any loss function $L(y, \hat{y}) : \{0,1\} \times \mathbb{R} \to \mathbb{R}$ that assesses the discrepancy between the actual response variable y and the predicted value $\hat{y}$, can be decomposed as follows:

$$L(y, \hat{y}) = y \cdot L(1, \hat{y}) + (1-y)L(0, \hat{y}) = \underbrace{L(0, \hat{y})}_{L_1(\hat{y})} - \underbrace{[L(0, \hat{y}) - L(1, \hat{y})]}_{L_2(\hat{y})} \cdot y.$$

This can be illustrated on standard classification settings:

1. The output $\hat{y} \in (0,1)$ aims for the probability that $y = 1$, and the performance is measured by the **cross entropy loss**. In this case, we can write:
$$L(y, \hat{y}) = -y \cdot log(\hat{y}) - (1-y) \cdot log(1-\hat{y}) = \underbrace{-log(1-\hat{y})}_{L_1(\hat{y})} - \underbrace{log\left(\frac{\hat{y}}{1-\hat{y}}\right)}_{L_2(\hat{y})} \cdot y.$$
2. The output $\hat{y} \in \{0,1\}$ is the best guess for the actual value of y, and the performance is measured by the **zero-one loss**. We can express this as follows:
$$L(y, \hat{y}) = y \cdot (1-\hat{y}) + (1-y) \cdot \hat{y} = \underbrace{\hat{y}}_{L_1(\hat{y})} - \underbrace{(2\hat{y}-1)}_{L_2(\hat{y})} \cdot y.$$
3. The output $\hat{y} \in \{-\infty, \infty\}$ is some score in favor of $y = 1$, and the performance is measured by the **Hinge loss**. In this case, the penalty is zero if $y = 1$ and $\hat{y} \geq 1$, or $y = 0$ and $\hat{y} \leq -1$, and increases linearly in $\hat{y}$ otherwise. This can be expressed as follows:
$$L(y, \hat{y}) = y \cdot \text{ReLU}(1-\hat{y}) + (1-y) \cdot \text{ReLU}(1+\hat{y}) = \underbrace{\text{ReLU}(1+\hat{y})}_{L_1(\hat{y})} - \underbrace{[\text{ReLU}(1+\hat{y}) - \text{ReLU}(1-\hat{y})]}_{L_2(\hat{y})} \cdot y.$$

Key result: Formulation of CV bias in classification

The term w_{cv} is defined as the bias of the CV estimator relative to the generalization error, and we provide an explicit expression:

$$w_{cv} := \text{GenErr} - \mathbb{E}[\text{CV}] = \mathbb{E}\, L\left(y_{te}, \hat{y}(x_{te}; T_{-1})\right) - \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\, L\left(y_i, \hat{y}_i^{cv}\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_X \mathbb{C}ov\left(L_2\left(\hat{y}_i^{cv}\right), y_i | X\right) - \mathbb{E}_{X_{-1}, x_{te}} \mathbb{C}ov\left(L_2\left(\hat{y}(x_{te}; T_{-1})\right), y_{te} | X_{-1}, x_{te}\right)$$

This key result applies to any decomposable loss function and makes no assumptions about the learning model.

Given an estimator $\hat{w}_{cv}$ of w_{cv}, the corrected CV estimator is:

$$\text{CV}_c = \text{CV} + \hat{w}_{cv}.$$

Methods for estimating w_{cv}

We suggest a parametric bootstrap in which we utilize estimates $\hat{\gamma}_c$ and $\hat{\gamma}_r$ of the true parameters that can be obtained from the training sample T.

Initially, it is essential to simplify the formula for w_{cv} according to the given scenario to eliminate the expectation over the unobserved x_{te}.

In the scenario of spatial correlation, we suggest the following simplified formula:

$$\hat{w}_{cv} = \frac{1}{n}\sum_{i=1}^{n} \widehat{\mathbb{C}ov}\left(L_2\left(\hat{y}_i^{cv}\right), y_i | \, X \, ; \, \widetilde{K}\right) \;\; ; \;\; \widetilde{K}_{i_1 i_2} = \mathcal{K}(z_{i_1}, z_{i_2}; \hat{\gamma}_r) - \overline{K},$$

where $\overline{K}$ is the estimated average kernel value between two observations from distinct regions.

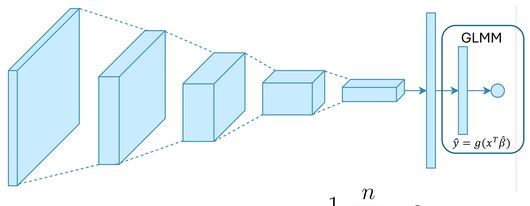
This brings us to the subsequent bootstrap algorithm:

1. Evaluate estimates of the unknown parameters, denoted by $\hat{\gamma}_c$, and $\hat{\gamma}_r$.
2. For $b = 1, \cdots, B$:
   - 2.a. Draw $\delta_b \sim MN(0, \widetilde{K})$, and draw Y_b from Bernoulli distribution with mean $g\left(f_{\hat{\gamma}_c}(X) + \delta_b\right)$
   - 2.b. Apply the learning procedure to $T_{b,-i} = (X_{-i}, Y_{b,-i})$, and calculate $l_{b,i} = L_2(\hat{y}(x_i; T_{b,-i}))$, for any i.
3. Calculate the sample covariance: $\widehat{C}_i = \frac{1}{B}\sum_{b=1}^{B}(l_{b,i} - \overline{l_i})(y_{b,i} - \overline{y_i})$.
4. Calculate the mean: $\hat{w}_{cv} = \frac{1}{n}\sum_{i=1}^{n} \widehat{C}_i$.

To address the demand for considerable computing powe in Step 2.b., we suggest an approximate estimate of $l_{b,i}$, instead of the full fitting process!

We view the last layer as a GLMM, while the rest of the network is fixed.

Then, we approximate the GLMM estimator $\hat{\beta}$ by embracing quadratic approximation of the Quasi-Likelihood equations:

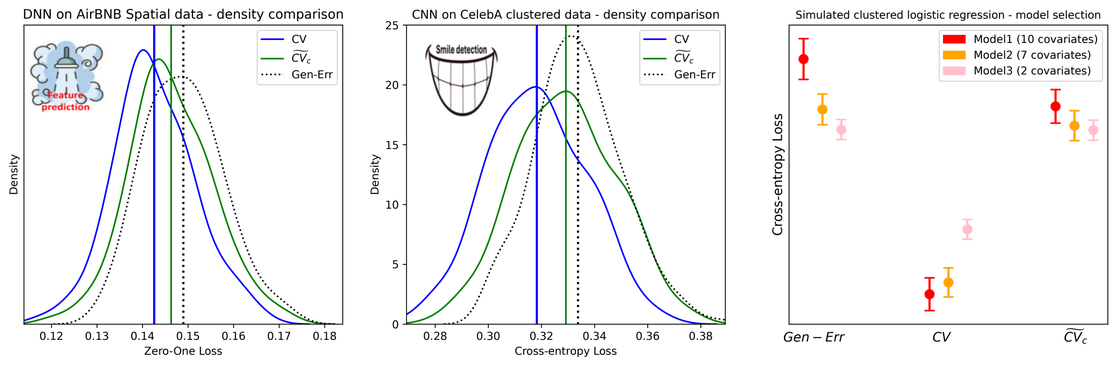


$$\tilde{\beta}_b = \beta + (X^t D V^{-1} D X)^{-1} X^t D V^{-1}(Y_b - \mu) \;\Rightarrow\; \tilde{l}_{b,i} = L_2(g(x_i^T \tilde{\beta}_{b,-i})) \;\Rightarrow\; \tilde{w}_{cv} = \frac{1}{n}\sum_{i=1}^{n} \widehat{\mathbb{C}ov}(\tilde{l}_{b,i}, y_i)$$

$$\mu_i = \mathbb{E}_\delta\left[g(x_i^t\beta + \delta_i)|X; \beta, \hat{\gamma}_r\right] \;\; ; \;\; D_{ii} = \mathbb{E}_\delta\left[g'(x_i^t\beta + \delta_i)|X; \beta, \hat{\gamma}_r\right] \;\; ; \;\; V = \mathbb{C}ov(Y|X; \beta, \hat{\gamma}_r).$$

Experiments

We demonstrate the application of the proposed methodology in various scenarios by comparing the standard CV estimator to our proposed approximated estimator $\widetilde{\text{CV}}_c = CV + \tilde{w}_{cv}$. The comparison focuses on deviation from the true GenErr and the model selection strategy.



Takeaway

We have shown that CV can be adjusted to consider correlation patterns in the data. Our approach has been validated through simulations and real-world data applications.

Our approach relies on a thorough examination of how covariance structures impact model evaluation, leading to an explicit formula for the bias between standard CV and generalization error, and a practical estimation methodology.

The outlined methodology is applicable not only to classification using standard evaluation metrics but also to the evaluation of the area under the ROC curve (AUC) and to non-linear regression models, as detailed in arXiv:2502.14808.