Maestría en Ciencia y Análisis de Datos- Universidad Mayor de San Andrés

Modelos lineales y modelos lineales generalizados

Rolando Gonzales Martinez, PhD

Fellow postdoctoral Marie Skłodowska-Curie

Universidad de Groningen (Países Bajos)

Investigador (researcher)

Iniciativa de Pobreza y Desarrollo Humano de la Universidad de Oxford (UK)

Integración de modelos lineales en machine learning

$$M_1, M_2, \ldots, M_k$$

$$\hat{\beta}_{\text{train},i} = (X_{\text{train},i}^T X_{\text{train},i})^{-1} X_{\text{train},i}^T y_{\text{train}}$$

$$\hat{y}_{\text{test},i} = X_{\text{test},i} \hat{\beta}_{\text{train},i}$$

$$\text{MSE}_i = \frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} (y_{\text{test},j} - \hat{y}_{\text{test},ij})^2$$

$$R_i^2 = 1 - \frac{\sum_{j=1}^{n_{\text{test}}} (y_{\text{test},j} - \hat{y}_{\text{test},ij})^2}{\sum_{j=1}^{n_{\text{test}}} (y_{\text{test},j} - \bar{y}_{\text{test}})^2}$$

$$M_{\text{best}} = \arg\min_i \text{MSE}_i$$

$$M_{\text{best}} = \arg\max_i R_i^2$$

Contenido del curso

(5) Aplicaciones de Modelos Lineales en Machine Learning

- Integración de modelos lineales en machine learning.
- Regularización en modelos lineales: Ridge, LASSO, Elastic Net.
- Laboratorio: Implementación de algoritmos de regularización en modelos lineales.

(6) Modelos Lineales Generalizados en Machine Learning

- Uso de GLM en problemas de clasificación y regresión.
- Comparación con otros algoritmos de machine learning.
- Laboratorio: Aplicación de GLM en machine learning.

Integración de modelos lineales en machine learning

$$y_{\text{train}} = X_{\text{train}}\beta_{\text{train}} + \epsilon_{\text{train}}$$

$$\hat{\beta}_{\text{train}} = (X_{\text{train}}^T X_{\text{train}})^{-1} X_{\text{train}}^T y_{\text{train}}$$

$$\hat{y}_{\text{test}} = X_{\text{test}}\hat{\beta}_{\text{train}}$$

$$\text{MSE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (y_{\text{test},i} - \hat{y}_{\text{test},i})^2$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{\text{test}}} (y_{\text{test},i} - \hat{y}_{\text{test},i})^2}{\sum_{i=1}^{n_{\text{test}}} (y_{\text{test},i} - \bar{y}_{\text{test}})^2}$$

Regularización

Lasso (L1), Ridge (L2), y Elastic Nets son técnicas de regularización que permiten reducir el sobreajuste y lidiar con multicolinealidad y variables redundantes:

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \left(\sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{n} \beta_j^2 \right)$$

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \left(\sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{n} |\beta_j| \right)$$

$$\hat{\beta}_{\text{elastic}} = \arg\min_{\beta} \left(\sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{n} |\beta_j| + \lambda_2 \sum_{j=1}^{n} \beta_j^2 \right)$$

Modelos Lineales Generalizados en Machine Learning

En MLG, es necesario considerar medidas de performance para variables continuas y variables discretas:

$$M_1, M_2, \ldots, M_k$$

$$\hat{\beta}_{\text{train},i} = \arg\max_{\beta} \log \mathcal{L}(\beta; X_{\text{train},i}, y_{\text{train}})$$

$$\hat{y}_{\text{test},i} = g^{-1}(X_{\text{test},i}\hat{\beta}_{\text{train},i})$$

$$\text{MSE}_i = \frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} (y_{\text{test},j} - \hat{y}_{\text{test},ij})^2$$

$$R_i^2 = 1 - \frac{\sum_{j=1}^{n_{\text{test}}} (y_{\text{test},j} - \hat{y}_{\text{test},ij})^2}{\sum_{j=1}^{n_{\text{test}}} (y_{\text{test},j} - \bar{y}_{\text{test}})^2}$$

$$\text{AUC-ROC}_i = \text{AUC}(\text{Curva ROC}_i)$$

$$M_{\text{best}} = \arg\min_i \text{MSE}_i$$

$$M_{\text{best}} = \arg\max_i R_i^2$$

$$M_{\text{best}} = \arg\max_i \text{AUC-ROC}_i$$

Regularización en MLG

Estas técnicas se aplican también para modelos lineales generalizados en base a una penalización a la función de verosimilitud:

- **Ridge:**

$$\hat{\beta}_{\text{train},i} = \arg\min_{\beta} \left(-\log \mathcal{L}_{\text{train},i}(\beta) + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

- **Lasso:**

$$\hat{\beta}_{\text{train},i} = \arg\min_{\beta} \left(-\log \mathcal{L}_{\text{train},i}(\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

- **Elastic Net:**

$$\hat{\beta}_{\text{train},i} = \arg\min_{\beta} \left(-\log \mathcal{L}_{\text{train},i}(\beta) + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right)$$

Comparación de ML y MLG con algoritmos de machine learning

Modelos lineales y MLGs	Modelos de machine learning
white-box	algunos modelos son cajas negras (black box)
Menor poder predictivo (en algunas ocasiones)	Mayor poder predictivo (en algunas ocasiones)
Mayor explicabilidad/comunicabilidad	Menor explicabilidad/comunicabilidad
Mayor interpretabilidad	Menor interpretabilidad

Comparación de ML y MLG con algoritmos de machine learning: matriz de confusión

En problemas de clasificación, métricas basadas en la **matriz de confusión** pueden utilizarse para comparar MLGs de clasificación con modelos de machine learning:

	Predicción Positiva	Predicción Negativa
Clase Positiva	Verdadero Positivo (TP)	Falso Negativo (FN)
Clase Negativa	Falso Positivo (FP)	Verdadero Negativo (TN)

Comparación de ML y MLG con algoritmos de machine learning: exactitud (accuracy)

- La exactitud se define como la proporción de predicciones correctas (tanto positivas como negativas) sobre el total de predicciones realizadas.

$$\text{Exactitud (Accuracy)} = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}}$$

$$\text{Exactitud (Accuracy)} = \frac{TP+TN}{TP+TN+FP+FN}$$

- En caso de datos desbalanceados otras medidas como el score F1 y la sensibilidad (recall) son más apropiadas

Comparación de ML y MLG con algoritmos de machine learning: laboratorio

- MLMLG_0500: Implementación de algoritmos de regularización en modelos lineale.
- MLMLG_0600: Comparación de un MLG logit con algoritmos de machine learning (SVM y bosques aleatorios), empleando matrices de confusión, en el contexto de credit scoring.