F-005

混同行列に基づく不均衡データ分類ニューラルネットワーク

Imbalanced Data Classification Neural Networks Based on Confusion Matrix

田儀 樹 * 大崎 美穂 * 片桐 滋 * 大西 圭 †
Itsuki Tagi Miho Ohsaki Shigeru Katagiri Kei Ohnishi

1. はじめに

不均衡データとはクラス間で事例の数・比率が大きく異なるデータを指す (図 1 参照)．多数の通常的事例 (陰性) と少数の危機的事例 (陽性) の 2 クラス分類は，特に重要である．例えば通信ネットワークの不正侵入検知，ロボットの故障・誤動作検知，がんの診断などがある．事例を多数の陰性に割り当て少数の陽性を見落としてしまう問題に対して，サンプリング，コストを含む損失関数，アンサンブルといった解決アプローチがある [1]．損失関数の改善は全アプローチに併用可能で適用範囲が広いため，我々はこれに着目する．コストアプローチは真陽性 (TP)，偽陽性 (FP)，真陰性 (TN)，偽陰性 (FN) に与えるコストを介して分類性能を間接的に高めるが，コストと分類性能のバランス・向上との関係は明らかではない．

以上を踏まえ，ドメインやユーザが重視する分類性能を直接的にバランス良く高める CM-KLOGR が提案された [2]．混同行列から得られる分類性能は一般に，陽性的中率・適合率 (PPV)，感度・再現率 (Sens)，陰性的中率 (NPV)，特異度 (Spec)，精度 (Acc) である．これらを融合した損失関数 (混同行列損失) をカーネルロジスティック回帰 (KLOGR) に組み込んだ分類器が，CM-KLOGR である．本研究では混同行列損失を汎用的な形で再定式化し，この損失と自動微分を活用した NNs(CM-NNs) を提案する．まずは影響要因を極力排除して，原理的有効性を明らかにしたい．そこで，最も基本的な多層パーセプトロン分類器 (MLPC) に混同行列損失を組み込み，CM-NNs を具現化した CM-MLPC を開発する．そして，ベンチマークデータセットに適用する実験を行い，標準的な交差エントロピーを持つ MLPC と比較評価する．

2. 提案手法

以下では CM-KLOGR の旧混同行列損失を解説後に，我々の新しい混同行列損失を定式化する．そして，自動微分を伴って NNs に組み込むことで提案手法 CM-NNs を実現する．CM-KLOGR では，MCE [3] の平滑化 0-1 損失 $l(d_{k_n}(\mathbf{x}_n))$ を誤分類の個数と見なし，これを分類パターン TP，FP，TN，FN ごとに数える．1 バッチあたりの TP の個数 N_{TP} は式 (1) と定式化できる．本式の $d_{k_n}(\mathbf{x}_n)$ は事例 $\mathbf{x}_n$ の分類結果，$l(d_{k_n}(\mathbf{x}_n))$ は結果に対する損失である．$\delta_{k_n,\mathrm{pos}}$ は，$\mathbf{x}_n$ の正解クラス C_{k_n} が陽性と一致すれば 1，しなければ 0 となる指示関数である．他のパターンの個数 N_{FP}, N_{TN}, N_{FN} も同様に定義される．これらを式 (2) に代入すれば，微分可能な分類性能が得られる．

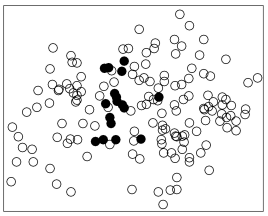

図 1: 文献 [2] より引用した不均衡データの一例.

$$N_{\mathrm{TP}} \approx \sum_{n=1}^{N}(1 - l(d_{k_n}(\mathbf{x}_n))\delta_{k_n,\mathrm{pos}} \tag{1}$$

$$\begin{aligned}\mathrm{PPV} = N_{\mathrm{TP}}/(N_{\mathrm{TP}} + N_{\mathrm{FP}}), \mathrm{Sens} = N_{\mathrm{TP}}/(N_{\mathrm{TP}} + N_{\mathrm{FN}}) \\ \mathrm{NPV} = N_{\mathrm{TN}}/(N_{\mathrm{TN}} + N_{\mathrm{FN}}), \mathrm{Spec} = N_{\mathrm{TN}}/(N_{\mathrm{TN}} + N_{\mathrm{FP}}) \\ \mathrm{Acc} = (N_{\mathrm{TP}} + N_{\mathrm{TN}})/(N_{\mathrm{TP}} + N_{\mathrm{FP}} + N_{\mathrm{TN}} + N_{\mathrm{FN}})\end{aligned} \tag{2}$$

CM-KLOGR の混同行列損失は，式 (3) のように，分類性能の加重調和平均に過学習を抑制する L2 正則化を加えたものである．なお，J は損失関数，f_i は任意の分類性能，γ_i はその重みでありユーザが指定できる．$\boldsymbol{\alpha}_k$ は KLOGR のパラメータベクトル，$\boldsymbol{\mathcal{K}}$ はグラム行列である．式 (3) は確かに，分類性能の重要度とバランスを γ_i で反映し，かつ，分類性能を微分可能な形で直接的に表現している．

$$J = -\left(\frac{1}{\sum_i \gamma_i}\sum_i \gamma_i \frac{1}{f_i}\right)^{-1} + \frac{\lambda}{2}\sum_k \boldsymbol{\alpha}_k^T \boldsymbol{\mathcal{K}} \boldsymbol{\alpha}_k \tag{3}$$

上記の旧混同行列損失はクラスに属する確率 $\Pr(C_k|\mathbf{x}_n)$ を出力する分類器に制限し，$l(d_{k_n}(\mathbf{x}_n))$ を導出していた．我々はこの制限を外して，クラスに属する度合いを表す量 $y_k(\mathbf{x}_n)$ であれば何でも可とする．すなわち，NNs の出力値そのもの，出力値を Softmax 関数で変換して求めた確率のどちらでも混同行列損失の第 1 項を求められる．旧混同行列損失の第 2 項は KLOGR の全パラメータの L2 正則化に限定していた．新混同行列損失ではどのような正則化も許容する．以上を反映して再定式化した混同行列損失を，式 (4) に示す．第 1 項は加重調和平均で融合した分類性能，第 2 項は Lq ノルム正則化項である．第 2 項では正則化に含まれる度合い α_j を分類器のパラメータ w_j にかけ，全パラメータに対する総和を取っている．α_j, w_j, q の設定により，様々な正則化を包括的に表現できる．旧混同行列損失の人手による記号微分とは異なり，新混同行列損失に

*同志社大学 Doshisha University
†九州工業大学 Kyushu Institute of Technology

表 2: 実験の試験性能 [%]．符号と矢印は BCE-MLPC に対する CM-MLPC の性能向上を，太字は総合性能の最大値を表す．

Dataset	Classifier	PPV	Sens	NPV	Spec	Acc	HM (All)	HM (Pos)
Breast	CM-MLPC (All)	93.23 −0.14 ↓	97.08 −0.84 ↓	98.45 −0.45 ↓	96.29 0.00	96.57 −0.28 ↓	96.29 −0.34 ↓	95.12 −0.47 ↓
Imbalance Ratio 1.90 (Low)	CM-MLPC (Pos)	94.12 +0.75 ↑	97.40 −0.52 ↓	98.62 −0.28 ↓	96.67 +0.38 ↑	96.92 +0.07 ↑	96.72 +0.09 ↑	**95.73** +0.14 ↑
	BCE-MLPC	93.37	97.92	98.90	96.29	96.85	96.63	95.59
Dataset	Classifier	PPV	Sens	NPV	Spec	Acc	HM (All)	HM (Pos)
Yeast-1_vs_7	CM-MLPC (All)	48.57 +17.98 ↑	33.33 +6.67 ↑	95.45 +0.58 ↑	97.44 +2.57 ↑	93.25 +2.84 ↑	**60.94** +12.07 ↑	**39.53** +11.04 ↑
Imbalance Ratio 14.30 (Relatively High)	CM-MLPC (Pos)	56.67 +26.07 ↑	33.33 +6.67 ↑	95.46 +0.59 ↑	97.67 +2.80 ↑	93.47 +3.05 ↑	**63.24** +14.38 ↑	**41.98** +13.48 ↑
	BCE-MLPC	30.60	26.67	94.87	94.87	90.41	48.86	28.50

は自動微分を用いる．これにより，様々な NNs に組み込めるようにした．

$$J = -\left(\frac{1}{\sum_i \gamma_i}\sum_i \gamma_i \frac{1}{f_i}\right)^{-1} + \lambda \sum_j |\alpha_j w_j|^q \quad (4)$$

3. 評価実験

3.1. 目的と条件

本実験では「提案手法 CM-NNs は，ユーザが重視する分類性能を直接的にバランス良く高めるのか」を調べる．損失自体の有効性を明確にすべく，NNs に最も単純なモデル MLPC を採用した CM-MLPC と，標準的な二値交差エントロピー (BCE) を持つ BCE-MLPC を比較する．また，混同行列損失の本質を担う第 1 項の効果を調べるため，正則化を行う第 2 項の λ はあえて 0 に設定する．CM-MLPC には次の設定を用いる．CM-MLPC (All): 全分類性能を等しく重視．情報が重複する Acc のみ重み 0，他の全ては重み 1．CM-MLPC (Pos): 陽性の分類性能を重視．Acc, NPV, Spec は重み 0, 陽性に関する PPV と Sens は重み 1．CM-MLPC は不均衡度が低いと BCE-MLPC と同等に働き，高いと BCE-MLPC を凌ぐと期待される．そこで異なる不均衡度のデータセットを使う [4]．1 つは特徴量 10，データサイズ 699，不均衡度 (陰性/陽性)1.90 の Haberman，もう 1 つは特徴量 7，データサイズ 459，不均衡度 14.30 の Yeast-1_vs_7 である．表 1 の設定のもと，層別入れ子交差検証により訓練・検証・試験を行う．

表 1: 訓練と検証の条件．

隠れ層の数	1, 2, 3
隠れ層のニューロン数	30, 60, 90, 120
0-1 損失の傾き	5, 10, 15
学習率	0.001, 0.010
最大エポック数	8000

3.2. 結果と考察

実験で得られた試験性能を表 2 に示す．列ごとに，PPV から Acc までの各分類性能と総合分類性能を掲載した．総合分類性能は，情報重複がある Acc 以外の全分類性能の調和平均 HM (All) と，陽性に関する Sens と Spec の調和平均 HM (Pos) である．Breast の結果を見ると，事前に想定した通りに CM-MLPC は BCE-MLPC と同等の性能である．不均衡度が低い場合，混同行列損失は BCE から極端に逸脱せず適切に動作すると確認できた．次に，不均衡度が比較的高い Yeast-1_vs_7 の結果を検討する．CM-MLPC は，PPV から HM (Pos) までの全てで BCE-MLPC よりも性能が高く，その向上量は最大で 26.07 [%] を達成した．分類性能間の差が大きい BCE-MLPC に比べ，CM-MLPC は性能差の抑制にも成功した．よって，CM-MLPC が分類性能を直接的にバランス良く高めると言える．

4. おわりに

不均衡データ分類では，多数の陰性事例の強い影響によって少数の陽性事例の分類が困難になる．この解決策として過去に，混同行列から求まる分類性能を融合した混同行列損失により，全分類性能の向上とバランスを両立する CM-KLOGR が提案された．本研究では様々な深層学習への展開を見据え，CM-KLOGR の混同行列損失を再定式化して NNs に組み込んだ CM-NNs を提案した．原理的有効性を明らかにすべく，最も基本的な MLPC で CM-NNs を実装し (CM-MLPC)，ベンチマークデータセットを用いた評価実験を行った．その結果，CM-MLPC は二値交差エントロピーを持つ MLPC よりも高性能，かつ，ユーザ指定の分類性能をバランス良く高めることができた．

謝辞

本研究は JSPS 科研費 21K12018 の助成を受けた．

参考文献

[1] G. Haixiang et al., “Learning from Class-Imbalanced Data: Review of Methods and Applications”, Expert Systems with Applications, vol.73, pp.220–239 (2017).

[2] M. Ohsaki et al., “Confusion-Matrix-Based Kernel Logistic Regression for Imbalanced Data Classification”, IEEE Trans. Knowl. Data Eng., vol.29, no.9, pp.1806–1819 (2020).

[3] S. Katagiri et al., “Pattern Recognition using a Family of Design Algorithms Based upon the Generalized Probabilistic Descent Method”, Proc. of the IEEE, vol.86, no.11, pp. 2345–2373 (1998).

[4] M. Lichman et al., “UCI machine learning repository”, http://archive.ics.uci.edu/ml