TNM Staging of Hepatocellular Carcinoma Using Clinical Image Data (CT and MR) and Clinical Tabular Data

Ujjwal Yadav Waheb Hasan Zaidi

Friday 10^{th} January, 2025

Motivation

Gaining insights from the Clinical Image and Tabular data

- We have 100 patients and their data in two formats.
- To try several classic and advanced methods to predict the TNM stage of the patient based on their CT and MR scans.
- Doing similar predictions of various metrics such as TNM based on other numerical data

Table of Contents

TNM Staging Using Clinical Image Data

- TNM staging importance for cancer treatment.
- Use of CT and MR scans (˜50 GB, 100 patients).
- Using several methods for classifying the scan with a stage.

Challenges in Image-Based Staging

- Variability in image acquisition protocols.
- High noise and artifacts in medical images.
- Computational cost for 3D segmentation.
- Class imbalance for TNM stages.

Proposed Approach for Image Data

Pipeline for Image Data

1. Preprocessing: Check consistency using metrics. Convert DICOM to NIfTI (while preserving metadata), resize, normalize, pad.
2. Segmentation: Use SAM (Segment Anything Model).
3. Classification: XGBoost trained on extracted (selected) features.

State of the Art for Image Data and Results

Segment Anything Model (SAM)

- Pretrained model for robust segmentation.
- Reduces need for supervised segmentation.

Advanced Classifiers

Table: Performance of Different Classifiers on the Dataset

Classifier	**Accuracy (%)**
Gradient Boosting	63.8 ± 0.03
Random Forest	62.0 ± 0.35
3D CNN	50.8 ± 0.04
SAM (After merging class 0 with class 1)	49.7 ± 0.01
SAM	43.6 ± 0.02

Model Overview

1. Data Ingestion
(a) Read patient scans (NIfTI format).
(b) Read metadata (patient ID, scan date, TNM labels).
(c) Match scans to TNM stage based on closest scan date.
(d) Exclude incomplete or invalid records.

↓

2. Preprocessing
(a) Basic normalization (zero-mean, unit variance).
(b) Resize volumes (e.g., 128×128 2D or $64 \times 64 \times 64$ 3D).
(c) Apply Otsu thresholding masks.

↓

3. Feature Extraction
(a) SAM-based embedding for each slice (through the SAM encoder).
(b) Collect slice-level features and compute aggregate (e.g., mean).

↓

4. Feature Selection
(a) Mutual information-based ranking.
(b) XGBoost-based feature importance.
(c) Combine and retain top features (e.g., 50).

↓

5. Classification
(a) Patient-level split into training/validation.
(b) Train XGBoost on selected features.
(c) Evaluate: accuracy, balanced accuracy, AUC.

Challenges and Future work

Challenges

- Limited data and class imbalance and noise in scans.
- Unavailability of stronger GPU and thus only using slices .

Possible Future Work

- Synthetic data generation to avert the effects of imbalance.
- To use radiomics features which is taking 6 hours for one patient on current system
- Make use of contrast-enhanced MRI data and Tumor masks

Introduction to Tabular Data

- Use of clinical records for TNM staging.
- Dataset: Demographics, biochemical markers, pathological details.
- Challenges: Missing values, class imbalance, non-linear feature interactions.

Challenges in Tabular Data

- Missing values require careful handling.
- Feature heterogeneity complicates modeling.
- Imbalanced data for rare TNM stages (e.g., Stage 0).

Proposed Approach for Tabular Data

Pipeline for Tabular Data

1. Preprocessing: Handle missing values, scale numerical data, encode categorical variables.
2. Feature Engineering: TabZilla for polynomial and interaction features.
3. Classification: Random Forest, SVM, and XGBoost.

Theoretical Background for Tabular Data

- Tabular data consists of heterogeneous numerical and categorical features.
- Machine learning models (Random Forest, XGBoost, SVM) require robust preprocessing.
- TabZilla automates feature engineering and optimizes hyperparameters.

State of the Art for Tabular Data

TabZilla Framework

- Automates feature engineering.
- Multi-output classification for TNM stages.
- Optimizes hyperparameters using Bayesian optimization.

Results from Tabular Data

Table: Performance Metrics with Mean and Standard Deviation (Std)

Metric	Random Forest (Mean $\pm$ Std)	SVM (Mean $\pm$ Std)	XGBoost (Mean $\pm$ Std)
Accuracy	56.08% $\pm$ 1.5	54.22% $\pm$ 2.1	62.35% $\pm$ 1.8
AUC	66.58% $\pm$ 2.3	63.91% $\pm$ 2.0	71.42% $\pm$ 1.9
Log Loss	2.235 $\pm$ 0.12	2.450 $\pm$ 0.15	1.870 $\pm$ 0.10
F1 Score	0.4644 $\pm$ 0.03	0.4517 $\pm$ 0.02	0.5235 $\pm$ 0.02

Discussion of Tabular Results

Key Insights

- TabZilla enhanced XGBoost's performance through feature interactions.
- Random Forest was reliable but less robust.
- SVM struggled with multi-output classification.

Challenges

- Class imbalance for rare TNM stages.

Conclusion of Tabular Data

Takeaways

- TabZilla simplifies feature engineering.
- XGBoost is the best-performing classifier.

Future Work

- Incorporate domain-specific features for interpretability.

QA

Thank you! Questions?