Beating the International Prognostic Index for high-risk DLBCL patients

Master thesis final report

Lukas Geßl

July 9th, 2024

Chair of Statistical Bioinformatics, Regensburg University

Recap: The goal of this thesis

- MMML-Predict: develop a cost-efficient classifier that filters DLBCL patients with progression-free survival ≤ 2 years more reliably than the International Prognostic Index for non-Hodgkin's lymphoma (IPI).
- The IPI [7] is a simple risk score (0–5) based on five clinical features. The cohorts IPI $\geq i, i = 0, 1, \ldots, 5$, lack precision ($< 50\%$) or are too small to be clinically relevant (prevalence $< 10\%$).
- Our classifier should label at least 15% of patients as high-risk with a precision of at least $\max(50\%, \text{precision of IPI} \geq 4)$.
- Unlike the IPI, the new classifier can incorporate the whole range of modern features (like transcriptomic, genetic, clinical data, already-existent signatures) measured at diagnosis and even dynamic features measured during the treatment.

MMML-Predict will enroll 300 DLBCL patients in a prospective trial.

- Data for the first 200 patients *will* arrive here and will be our sole foundation to train classifiers and finally submit a single one.
- A group in Leipzig will test the submitted classifier on the remaining 100 patients.

For this thesis, we also play by these rules, but on already existing data.

How to find and sell the best model

A two-step approach

Validation Of those models we have trained, we want to find and choose the model that performs best on new data to the best of our knowledge.

Testing We need to demonstrate the performance of the chosen model to outside people on new, independent data.

To this end, we split the data (X, y) into a train cohort $(X_{\text{train}}, y_{\text{train}})$ (also for validation) and test cohort $(X_{\text{test}}, y_{\text{test}})$ (no more repeated splitting).

Validation

We start with a set of tuples of hyperparameters H, where every $h \in H$ defines a model up to its parameters.

For every hyperparameter tuple $h \in H$, we

1. fit the model to the train cohort in a cross-validation, yielding a vector of cross-validated predictions $\hat{y}_{\text{train}} = \text{cv}(h)$.
2. We use the cross-validated predictions to calculate the cross-validated error $\text{err}(y_{\text{train}}, \hat{y}_{\text{train}})$.

We select the model m^* with hyperparameter tuple

$$h^* = \underset{h \in H}{\arg\min}\ \text{err}(y_{\text{train}}, \text{cv}(h)).$$

Testing

We calculate m^*'s predictions $m^*(X_{\text{test}}) = \hat{y}_{\text{test}}$ on the test cohort and estimate its performance on independent data via

$$\text{err}(y_{\text{test}}, \hat{y}_{\text{test}}).$$

For our problem, we choose $\text{err}(y, \hat{y})$ as the minimum of the negative precisions with a prevalence of at least 17% (model output usually needs thresholding).

Strictly speaking, the threshold for the model output is another hyperparameter, but it is a platform-dependent one [1]. On a new data set, one might take the 17% quantile of the model output as the threshold.

Let's talk about H: candidate models

Model-agnostic hyperparameters ...

...apply for every model. In our case, they concern the predictor matrix $X \in \mathbb{R}^{n \times p}$ and the response vector $y \in \{0,1\}^n \cup (\mathbb{R} \times \{0,1\})^n$.

- We add all combinations of at most n_{combi} discrete features that are positive in a share of at least $s_{\min}$ patients to X; e.g. we add a column "female and ABC-type tumor" if at least 5% of patients have this property.
- For $T > 0$, we provide the fitting algorithm a modified reponse y, namely
  - for the binary response, we set $y_i = 1$ if the patient's progression-free survival is $< T$, $y_i = 0$ otherwise,
  - for the Cox response, we censor all samples with time to event exceeding T at T.
- A-priori feature selection: which features do we include in X in the first place?

The most model-specific hyperparameter: model class

At the core, our models consist of

- Cox proportional-hazards,
- logistic regression and
- ordinary linear (or Gauss) regression

models [4],

- ℓ_1 or ℓ_2 regularization,
- the zero-sum constraint on a subset of features [1],
- standardization of the predictor.

Moreover, we deploy random forests [8].

Nested models

Given some "early" models $f_i : \mathbb{R}^p \to \mathbb{R}, i = 1, \ldots, m$, we can nest them into another, "late" model $f : \mathbb{R}^m \to \mathbb{R}$ and get a new model $f \circ (f_1, \ldots, f_m)$.

- Often, the early models have been trained on another data set, so we observe their output as features in our data set (like the Lamis signature): such f_i are merely projections onto a feature.
- If we need to fit some of the early models to our data, how can we get reliable cross-validated predictions for f? See next slide.

Typically, we train the early model on the high-dimensional part of the data (like gene expression) and use its output together with the remaining features as input for the late model.

Algorithm 1 Nested pseudo cross validation

1: **Input:** Predictor matrix X, response y, hyperparameter tuple $h = (h_1, h_2)$
2: Fit f_1 to $(X; y)$ subject to h_1 in a k-fold cross-validation, yielding cross-validated predictions $\hat{y}^{(1)}$.
3: Fit f to $(\hat{y}^{(1)}, f_2(X), f_3(X), \ldots, f_m(X); y)$ subject to h_2 in a k-fold cross-validation, yielding cross-validated predictions $\hat{y}$.
4: $g \leftarrow f \circ (f_1, \ldots, f_n)$
5: **Output:** $(\hat{y}, g)$

The pseudo cross-validated prediction for every sample in $\hat{y}$ slightly depends on the sample itself. Benefit: save a factor k in time complexity.

Procede greedily (first tune h_1, then h_2) to avoid overfitting of cross-validated predictions to the training cohort.

The R package patroklos

patroklos [**?**] solves this and analogous problems with the presented methods.

How this plays out on real data

Meet the data

	Schmitz [5]	Reddy [3]	Lamis test [6]
# samples	229	604	466
# genes	25 066	13 302	145
technology	RNA-seq	RNA-seq	NanoString
high risk [%]	36.6	31.5[1]	24.3
IPI-45 prev. [%]	12.9	21.6	17.0
IPI-45 prec. [%]	65.2	54.1	38.2
other features[2]	signature “genetic subtype”, continuous IPI features	genetic events: high expression, translocation, mutation	

[1] High risk is defined as overall survival < 2.5 years.

[2] All datasets include the IPI features in thresholded form, gender, cell of origin, and the LAMIS signature.

Intra-trial: Validate and test on the same data set

	Schmitz	Reddy	Lamis test
# samples	58	151	117
high risk [%]	37.0	31.6	24.3
(prev./prec.) IPI ≥ 4	(0.170/0.500)	(0.192/0.421)	(0.139/0.364)
(prev./prec.) m^*	(0.351/0.684)	(0.230/0.556)	(0.346/0.459)
ROC-AUC m^*	0.80	0.65	0.66
logrank m^*	3.69×10^{-4}	1.82×10^{-3}	9.38×10^{-4}

Table 1: Randomly split a single data set into a train and test cohort; train and validate on the train cohort, test on the test cohort. All numbers refer to the test set.

m^*'s architecture in a nutshell

Schmitz

Nested model as in Alg. 1 with

- the early model (Gauss) trained on the RNA-seq features,
- the late model (Cox) trained on the early model's output plus the remaining features (IPI in all verions), $n_{\text{combi}} = 2$.

Reddy

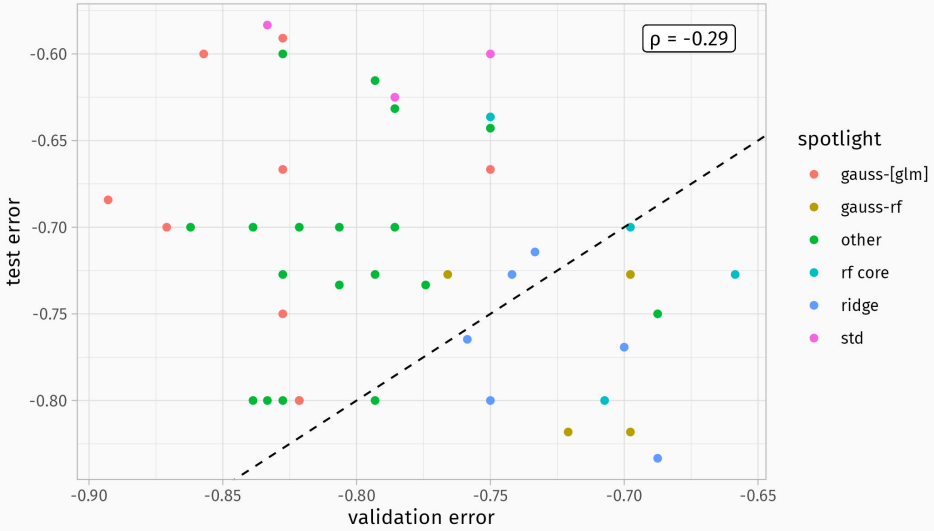
Nested model as in Alg. 1 with

- the early model (Gauss) trained on the RNA-seq features,
- the late model (Cox) trained on the early model's output plus the remaining features (five IPI features discretized), $n_{\text{combi}} = 3$.

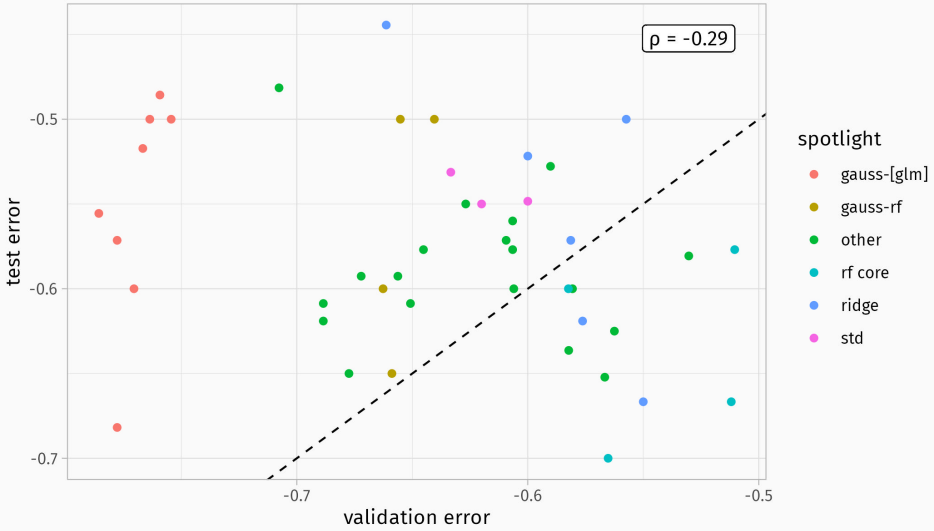
Lamis test

A logistic model trained on all features except for the NanoString gene counts, $n_{\text{combi}} = 1$.

m^* seems to the winner of a lottery: Schmitz



m^* seems to the winner of a lottery: Reddy



m^* seems to the winner of a more predictable lottery: Lamis test

Inter-trial: Train and validate on one data set, test on another

	Schmitz	Reddy	Lamis test
Schmitz	(12.9/65.2)	(17.7/59.6)	(17.1/50.7)
Reddy	(17.8, 71.1)	(21.6/54.1)	(18.0/53.2)
Lamis test	(17.4/75.7)	(22.5/50.4)	(17.0/38.2)

Table 2: Rows i hold training cohorts, columns j hold test cohorts. Diagonal entries (i, i) hold (prevalence/precision) of IPI ≥ 4 on cohort i. Off-diagonal entries (i, j) hold (prevalence/precision) on cohort j of the best validated model m_i^* trained on cohort i.

A closer look at m^* for Reddy $\rightarrow$ Lamis test

We train a logistic model with ℓ_1 penalty and standardization of the predictor, for $T = 2.3$ and $n_{\text{combi}} = 2$, providing as features

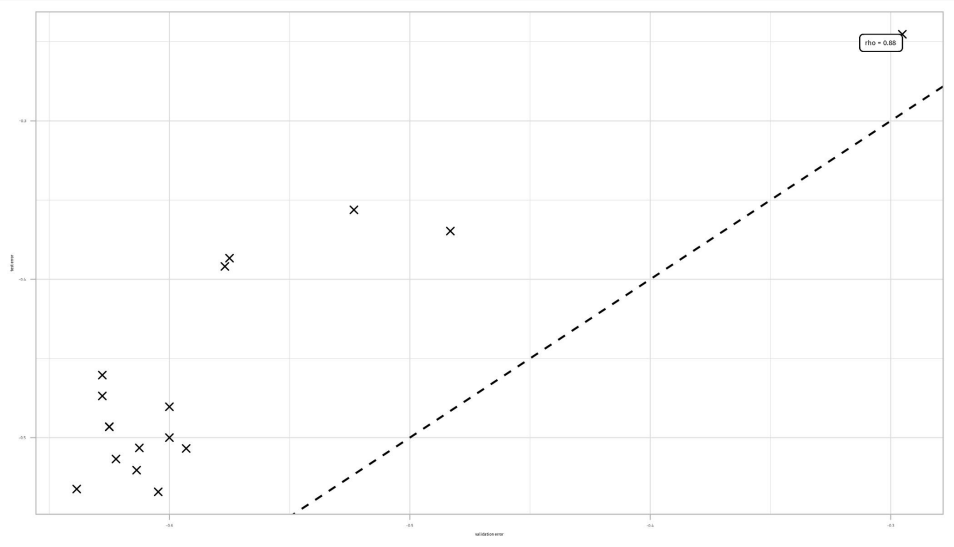
- LAMIS score,
- cell of origin,
- IPI group: low (0–1), intermediate (2–3), high (4–5),
- the five thresholded IPI features,
- gender.

A closer look at m^* for Reddy $\rightarrow$ Lamis test

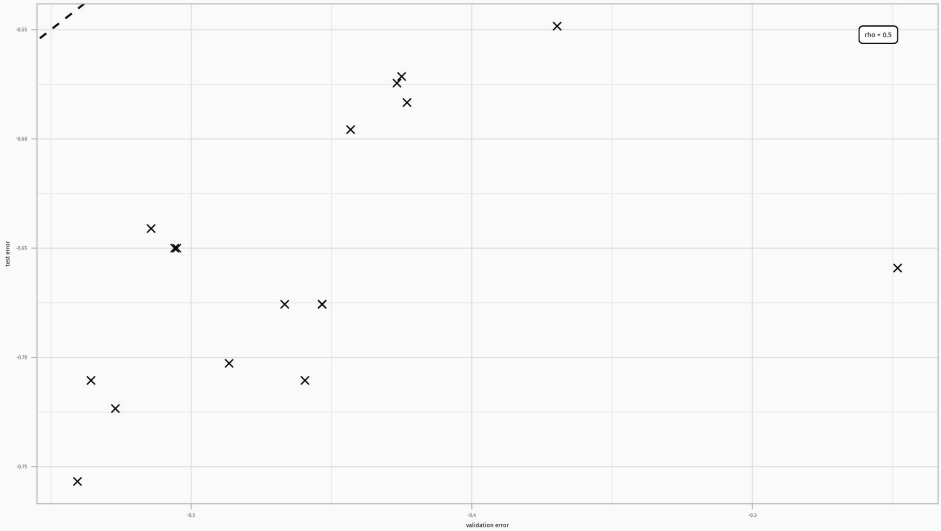
Feature	Coefficient
Lamis score	0.531
IPI group = intermediate & age > 60	0.144
IPI group = low	−0.266
IPI group = low & gender = male	−0.695
IPI group = low & cell of origin = unclassified	−0.033
IPI group = low & ann arbor stage > 2	1.596
gender = male & # extranodal sites > 1	0.456
ABC/GCB unclassified & performance status > 1	0.236
age > 60	0.379
age > 60 & LDH ratio > 1	0.252
age > 60 & performance status > 1	0.042
LDH ratio > 1	0.130
performance status > 1	0.904

Table 3: Features with non-zero coefficients of the logistic model m^*_{Reddy}.

A strong link between validation and test error: Reddy $\rightarrow$ Lamis test



A strong link between validation and test error: Lamis test $\rightarrow$ Schmitz



Conclusions and discussion

Take aways

We wanted to deliver a classifier that defines a high-risk group of DLBCL patients which is larger and more precise than that defined by the IPI.

- In intra-trial experiments, we could deliver on this promise for three data sets. Inter-trial experiments worked even better.
- While simple, ℓ_1-penalized models predicting from high-dimensional gene expression levels only sometimes already beat the IPI, one usually needs to integrate more features.
- Integrating *already-existent* transcriptomic and genetic signatures and the IPI features into another model reliably beats the IPI.
- Transferring these models from one data set (and platform) to another works very well (especially Reddy $\rightarrow$ Lamis test). Apparently, the size of the data set matters most.

Discussion

- **Validation:** Ensure a reliable link between validated and tested performance. How?
  - Validating a smaller H (proceding more greedily, relying on prior, general knowledge).
  - A refined cross validation following [2] to estimate the generalization error more reliably.
  - Is our choice of err too unstable? ROC-AUC isn't any more stable.
  - More samples.
- **Training:** Deploy other, more complex models in the integration step like boosted trees or neural networks. Balance classification problem via sample weights in loss function.
- For MMML-Predict: rather more samples, less features.

Thank you!
Questions?

References

[1] M. Altenbuchinger, P. Schwarzfischer, T. Rehberg, J. Reinders, et al. Molecular signatures that can be transferred across different omics platforms. *Bioinformatics*, 33(14):i333–i340, 07 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx241.

[2] S. Bates, T. Hastie, and R. Tibshirani. Cross-validation: What does it estimate and how well does it do it? *Journal of the American Statistical Association*, 119(546):1434–1445, May 2023. ISSN 1537-274X. doi: 10.1080/01621459.2023.2197686.

[3] A. Reddy, J. Zhang, N. S. Davis, A. B. Moffitt, et al. Genetic and functional drivers of diffuse large b cell lymphoma. *Cell*, 171(2):481–494.e15, 2017. ISSN 0092-8674. doi: 10.1016/j.cell.2017.09.027.

[4] T. Rehberg. zerosum: R package for elastic net regularized regression with zero sum constraint. URL `https://github.com/rehbergT/zeroSum`. Version 2.0.7.

References ii

[5] R. Schmitz, G. W. Wright, D. W. Huang, C. A. Johnson, et al. Genetics and pathogenesis of diffuse large b-cell lymphoma. *New England Journal of Medicine*, 378(15):1396–1407, 2018. doi: 10.1056/NEJMoa1801445.

[6] A. M. Staiger, M. Altenbuchinger, M. Ziepert, C. Kohler, et al. A novel lymphoma-associated macrophage interaction signature (lamis) provides robust risk prognostication in diffuse large b-cell lymphoma clinical trial cohorts of the dshnhl. *Leukemia*, 34(2):543–552, 2020. doi: 10.1038/s41375-019-0573-y.

[7] The International Non-Hodgkin's Lymphoma Prognostic Factors Project. A predictive model for aggressive non-hodgkin's lymphoma. *New England Journal of Medicine*, 329(14):987–994, 1993. doi: 10.1056/NEJM199309303291402. PMID: 8141877.

[8] M. N. Wright. ranger: A fast implementation of random forests. URL `https://github.com/imbs-hl/ranger`. Version 0.16.2.

The ultra cute Pumuckl is taken from `https://irp-cdn.multiscreensite.com/08191d67/dms3rep/multi/Pumuckl_Rennend.png`.