Beating the International Prognostic Index for high-risk DLBCL patients

Master thesis progress report

Lukas Geßl

March 12, 2024

Chair of Statistical Bioinformatics, Regensburg University

Set the arena

DLBCL: a heterogeneous cancer with a homogeneous therapy

In practice, there is only one treatment regimen: immunochemotherapy with R-CHOP. It cures two thirds of the patients.

Cure rates among relapsed and refractory patients are low. Hence, we should not send them to R-CHOP therapy in the first place, but into alternative treatments including clinical trials.

We define high-risk patients as those with a progression-free survival (PFS) < 2 years.

The goal of this thesis (and the MMML-Predict project) is ...

... to develop a cost-efficent ($<$ 1500 euros) classifier filtering out high-risk DLBCL patients before an R-CHOP treatment begins or at least at an early stage of it.

Candidate input features for the new classifier are

- clinical data (like the IPI, see next slide),
- transcriptomic (RNA-seq, signatures like LAMIS, ABC vs. GCB),
- proteomic signatures,
- somatic genetic factors (translocations like MYC),

all of which are measured at diagnosis, as well dynamic features like

- the tumor burden according to a liquid biopsy after 2 and 4 cycles of R-CHOP.

To beat: the International Prognostic Index (IPI) for non-Hodgkin's lymphoma

The IPI [1] is a simple risk score ranging from 0 to 5 depending on how many of the following clinical questions for a patient one can answer with "yes":

- Age > 60?
- Ann Arbor stage III or IV: is the cancer advanced?
- Serum LDH (lactacte dehydrogenase) level: higher than normal?
- Performance status: is the patient no longer ambulatory?
- Number of extranodal sites (like bone marrow, liver, lung) involved: more than one?

The lower the IPI, the better the patient's outlook: higher progression-free survival (PFS) and overall survival (OS).

The IPI is a 30-year old dinosaur

Yet, it's still state of the art in clinical practice when it comes to assessing a DLBCL patient's risk because it's simple, cheap and robust (after all, it's based on a rigorous statistical analysis and Cox regression).

Still, just six values the IPI can attain mean it's very rough. In particular, it fails to identify a clinically relevant high-risk group:

- The cohort with IPI $= 5$ is too small to get attention from clincians.
- The cohort with IPI ≥ 4 lacks precision[1] in identifying high-risk patients: 16% of patients have an IPI ≥ 4, but only 40% of them are high-risk. This is too low to persuade a clinician to change the treatment plan.

[1] Proportion of true positives among all positives.

What does ”beating the IPI” mean?

We need data to demonstrate the new classifier’s superiority. MMML-Predict aims to enroll 300 DLBCL patients (200 training, 100 test cohort). For my thesis, I need to use already existent data.

Beating the IPI means, on the test cohort the new classifier needs to

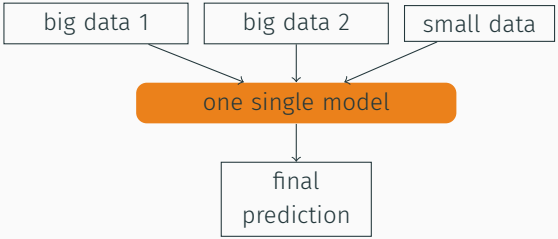
- be more precise in identifying high-risk patients than the IPI: the 95% confidence interval (CI) of the precision according to Clopper-Pearson must not include 35%[2],
- yield two cohorts with significantly differing survival (PFS): logrank-test p-value < 0.05.

Calculations with the size of the test cohort ($n = 100$) suggest that a precision $\geq 50\%$ with a prevalence[3] $\geq 15\%$ is enough.

[2] This is the precision of IPI ≥ 4 on pooled data from DSNHNL trials (2721 samples).
[3] The rate of positive predictions.

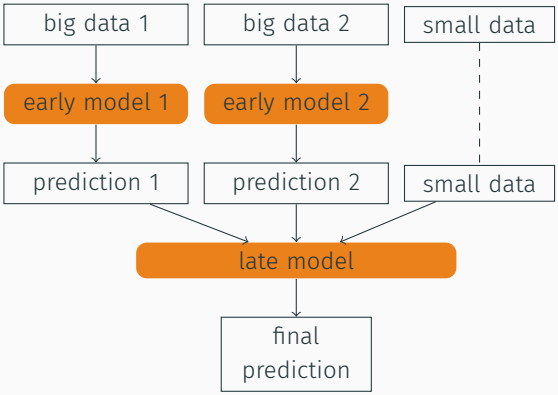
Meet the players

Early versus late integration



- Provide all data as input features to a single well-known model.
- Upside: easy to implement, one algorithm fits and picks the model including a cross validation.
- Downside: data on vastly different scales may confuse the model and its minimizer.

Early versus late integration



- Early models deal with high-throughput data and its curses: curse of high dimensionality, measurement errors.
- Upside: modularizes the model selection process, allows for very sophisticated late models.
- Downside: implementing the model selection process becomes more complicated, how to deal with cross validation in the early models?

The key player for early-stage models: zeroSum

We feed high-troughput data into

- Cox proportional-hazards models and
- logistic models,

with LASSO regularization and the zero-sum constraint. Both aim to estimate the response y_i of sample i by a predictor $x_i \in \mathbb{R}^p$ via

$$y_i = f(\beta_0 + x_i^T \beta) + \varepsilon_i \tag{1}$$

for a link function $f : \mathbb{R} \to \mathbb{R}$, a vector of coefficients (β_0, β), and a residual ε_i.

The zero-sum constraint, $\sum_{j=1}^{p} \beta_j = 0$, enforces scale-invariance, i.e., the model output for $\alpha \cdot x_i$ $(\alpha > 0)$ *after taking the log* is the same as for x_i.

Wrap it all into a loss function

Training such a model comes down to minimizing a loss function of the form

$$\mathcal{L}_{X,y,\lambda,u,v,w}(\beta_0, \beta) = -\sum_{i=1}^{n} w_i \ell_{X,y,\beta}(\tilde{y}_i, \beta_0 + x_i^T \beta) + \lambda \sum_{j=1}^{p} v_j |\beta_j| \quad \text{subject to } \sum_{j=1}^{p} u_j \beta_j = 0 \qquad (2)$$

for hyperparameters

- $\lambda > 0$, the LASSO penalty factor (tuned in a cross-validation),
- $u \in \mathbb{R}^p_{\geq 0}$, the zero-sum weights (often $u = \mathbf{1}$),
- $v \in \mathbb{R}^p_{\geq 0}$, the LASSO penalty weights (often $v = \mathbf{1}$), and
- $w \in \mathbb{R}^n_{\geq 0}$, the sample weights (often $w = \frac{1}{n}\mathbf{1}$).

$\ell_{X,y,\beta} : \mathbb{R}^2 \to \mathbb{R}$ is some kind of model-dependent log likelihood. $\tilde{y}_i$ is closely related to y_i (if not the same), its nature again depends on the model.

More on ℓ and y_i

In the Cox model,

- y_i is the relative hazard of sample i: the higher it is, the earlier we expect sample i to face the event compared to the other samples.
- $\tilde{y}_i$ is the time to event.
- $\ell_{X,y,\beta}$ tries to enforce the correct ordering: $\beta_0 + x_i^T\beta$ should be monotonic in $\tilde{y}_i$.

In the logistic model,

- we need to threshold the time to event to get a binary response: $\tilde{y}_i = y_i = 1$ if the event happens before a certain time T, 0 otherwise. One can view T as yet another hyperparameter.
- $\ell_{X,y,\beta}$ forces $\beta_0 + \beta x_i^T$ to be high if $\tilde{y}_i = 1$ and low else.

What this means for right-censoring

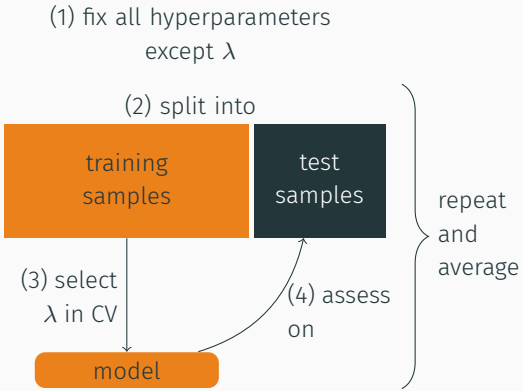
In reality, the data contains patients that dropped out of the study before the event could occur (right-censoring).

The Cox model, more precisely $\ell_{X,y,\beta}$, can take this information into account.

For the logistic model, however, we cannot use patients censored before T.

Train the players and select the best

Train-test paradigm



Hyperparameters excluding λ may be

- the model type,
- zero-sum weights, regularization weights, sample weights,
- the threshold T for the logistic model[4].

We assess the models

- with a scalar metric (like the ROC-AUC) to get a pre-selection,
- in scatter plots (like prevalence versus precision) to threshold the scores output by the pre-selected models.

[4] Similarly for the Cox model, we can right-censor samples with time to event $> T$ at T.

Software

zeroSum R package [4] for fitting and cross-validating the logistic and Cox models. It extends the glmnet package by the zero-sum constraint.

When integrating a model selected in a cross validation into another model I want to continue the cross validation of the early model. zeroSum does not report enough details, so I added this functionality in a fork zeroSumLI [5].

Training and assessing a bunch of models on several data sets means a lot of administrative, repetitive work. I automized and outsourced this part into an R package patroklos [2].

Watch the game: the results

The data

I trained models predicting progression-free survival < 2 years on data including bulk RNA-seq taken from Schmitz et al. [6].

- It has $n = 229$ patients with survival information, $p = 25\,066$ genes.
- 78 (34%) of these are high risk.
- 135 (59%) are low risk.
- 16 (7%) we cannot assign.
- All IPI features are available in pheno data.
- The IPI does a pretty good job on it, see Table 1.

IPI $\geq$	prevalence	precision
5	0.01	1.00
4	0.13	0.65
3	0.35	0.55
2	0.59	0.48
1	0.85	0.41
0	1.00	0.37

Table 1: Classifying PFS < 2 years on [6].

The line-up: the models tried out

hyperparamter	choices
model family	Cox, logistic regression.
input data	Gene expression only; gene expression with early integrated IPI: as one continuous variable (”with ipi cont”) or five binary variables (”with disc ipi feat”).
zero-sum weights	$u = 0$; $u_i = 1$ for gene-expression features, else $u_i = 0$ (”zerosum”).
LASSO weights	Standardization, i.e., v_i is the standard deviation of feature i (”std”); $v_i = 1$ for gene-expression features, else 0.
sample weights	$w = \mathbf{1}/n$.
T	$1, 1.25, 1.5, \ldots, 2.5$; additionally ∞ for Cox.

Table 2: Terms in quotation marks are used in model names on the following slides.

Pre-selection via ROC-AUC

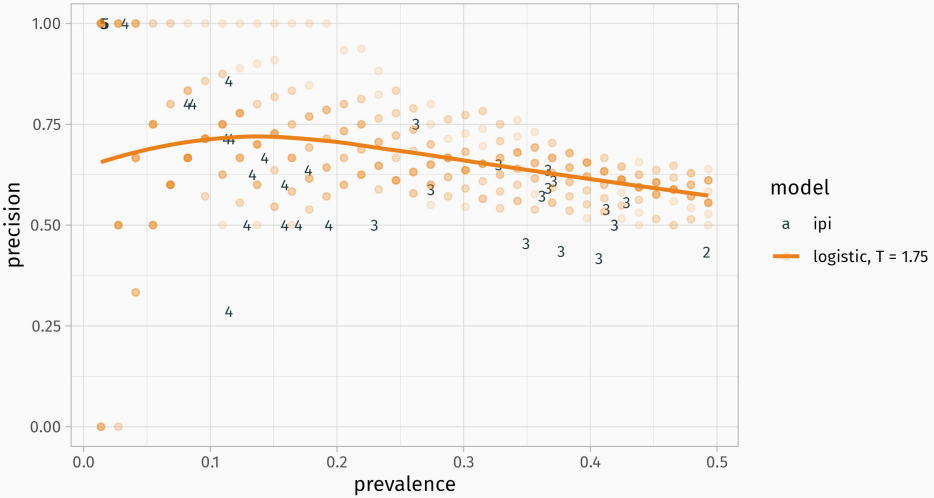
rank	model	T	AUC
1	logistic	1.75	0.770
2	logistic zerosum	1.75	0.769
3	cox	1.75	0.762
4	cox zerosum	1.75	0.761
5	cox	2	0.755
6	logistic zerosum	1.5	0.755
7	cox zerosum	1.5	0.752
8	cox	1.5	0.750
9	cox	Inf	0.749
10	cox zerosum	Inf	0.748

Table 3: Split into train and test cohort, train a model on train cohort, calculate ROC-AUC on test cohort. Repeat this 15 times and average.

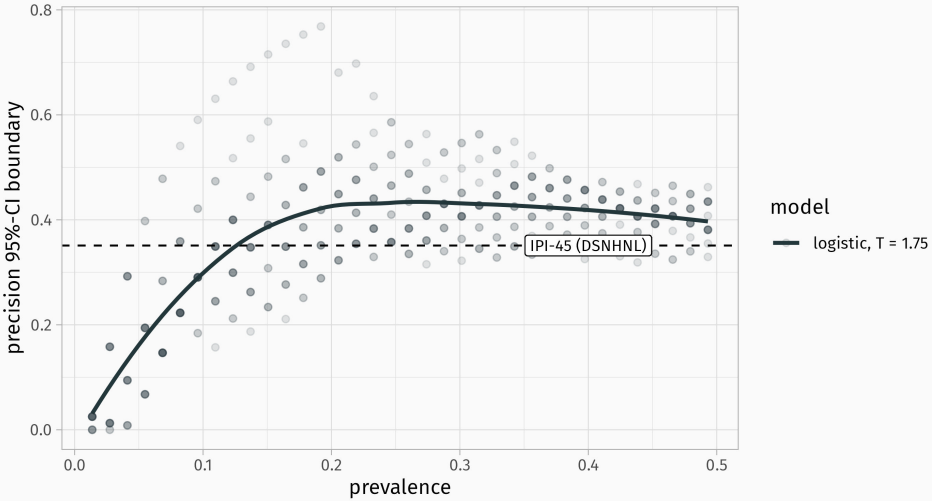
What does this mean for choosing the hyperparameters?

- Logistic versus Cox regression not that important.
- With zero-sum constraint usually a bit worse than without.
- Choose $T = 1.75$.
- Early integration and standardization disappoint.

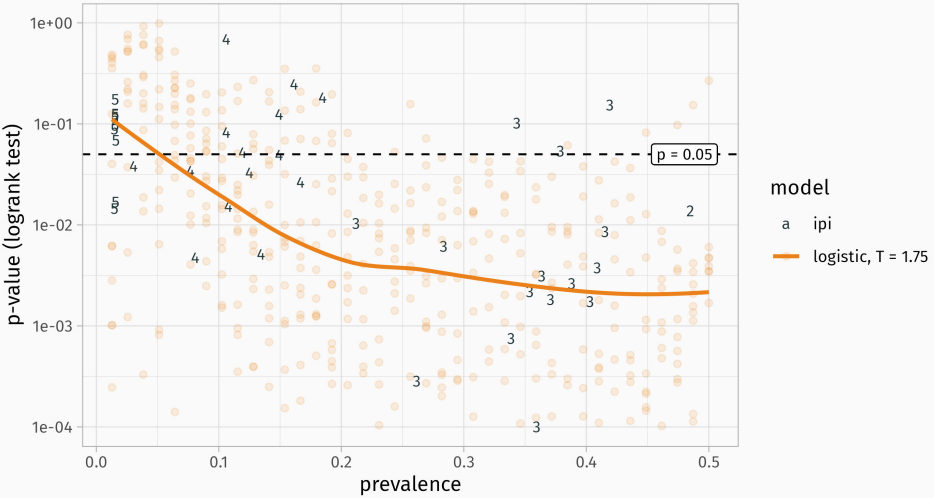
Is this enough to meet our initially stated goals?



Is this enough to meet our initially stated goals?



Is this enough to meet our initially stated goals?



What comes next?

On the data front, I want to

- do the afore-mentioned on a bigger ($n = 624 \gg 229$) data set by Reddy et al. [3].
- Downside: only overall, no progression-free survival included (fine for this thesis, less helpful for MMML-Predict).

On the methods front, I want to

- try to get early integration with zeroSum working,
- use sample weights $w \neq \frac{1}{n}\mathbf{1}$ to give less weight to patients with a PFS close to 2 years in the loss,
- implement late integration: with, e.g., linear regression, random forest as second-stage models,
- integrate more features.

Thank you for your attention!

Questions?

References

[1] **“A Predictive Model for Aggressive Non-Hodgkin’s Lymphoma”.** In: *New England Journal of Medicine* 329.14 (1993). PMID: 8141877, pp. 987–994. DOI: 10.1056/NEJM199309303291402.

[2] Lukas Gessl. ***patroklos: An R package pipelining omics-based cancer survival analysis.*** R package version 0.4.0. 2024. URL: https://lgessl.github.io/patroklos/.

[3] Anupama Reddy et al. **“Genetic and Functional Drivers of Diffuse Large B Cell Lymphoma”.** In: *Cell* 171.2 (2017), 481–494.e15. DOI: 10.1016/j.cell.2017.09.027.

[4] Thorsten Rehberg. ***zeroSum.*** URL: https://github.com/rehbergT/zeroSum.

[5] Thorsten Rehberg and Lukas Geßl. ***zeroSumLI.*** URL: https://github.com/lgessl/zeroSumLI.

[6] Roland Schmitz et al. **“Genetics and Pathogenesis of Diffuse Large B-Cell Lymphoma”.** In: *New England Journal of Medicine* 378.15 (2018), pp. 1396–1407. DOI: 10.1056/NEJMoa1801445.