Notes on computational details mentioned in the manuscript

Predictive Power of Dynamic (vs. Static) Risk Factors in the Finnish Risk and Needs Assessment Form

Elastic net logistic regression

Elastic net logistic regression is a mix of two penalized regression methods: *ridge regression* (Hoerl & Kennard, 1970) and *lasso regression* (Tibshirani, 2011). They seek to increase predictive power in a regression model (in this case logistic regression) by increasing parsimony. Parsimony is increased by imposing a penalty parameter, λ, and thus shrinking the estimated regression coefficients towards 0. Ridge regression imposes this penalty in relation to coefficient size and thus shrinks large coefficients more aggressively and no coefficient all the way to 0. Lasso regression on the other hand imposes an equally strong penalty to all coefficients, all the way down to 0 if λ is large enough. In both ridge and lasso regression, when parameter $\lambda = 0$ the model is an ordinary logistic regression. As λ is increased, the model approaches a null model where all coefficients are 0. In both cases the appropriate value for λ is treated as a *tuning parameter*, decided through testing different values and choosing the value that produces the best results in cross-validation.

Ridge and lasso regression perform well under different conditions and what penalty will give the best result is unknown before testing the models. Often the best solution is a mix of the two types of penalties in what is called an *elastic net* (Zou & Hastie, 2005). The elastic net introduces a second tuning parameter, α, a mixing parameter, which describes the nature of the penalty imposed. Parameter α, takes a value between 0 and 1, from pure ridge to pure lasso regression.

Random forest

Classification trees (Breiman, Friedman, Olshen, & Stone, 1984) splits the sample repeatedly into subsets based on the available predictors. In this case, whatever predictor that can best be used to separate reoffending individuals from those individuals that do not reoffend, is chosen first. This is evaluated based on how *pure* (i.e. homogenous in respect to recidivism status) the resulting groups are. A new split is added based on what predictor can achieve the biggest improvement in purity in either of the current groups. Successively a *tree* is *grown* by again splitting the current subgroups.

Random forest (Breiman, 2001) summarizes the prediction of many classification trees and introduces variability in trees via two processes. First the sample on which the tree is grown is a new bootstrapped sample for each new tree. Second, the number of predictors that the algorithm considers for each split in the tree is constrained to a limited number of randomly selected predictors. Randomly excluding some predictors has the result of allowing weaker predictors, that otherwise would be out-powered, to contribute to the prediction. Introducing variability in trees, in combination of averaging predictions over many trees, has proved to result in very good predictive performance.

The number of random predictors to consider at each split is defined by the modeler as a tuning parameter, m_{try}. The optimal value for m_{try} is not known before testing and is best chosen via cross-validation. Though the value of m_{try} generally does not affect predictive performance very much, a lower value has the effect of giving a bigger role to relatively weak predictors and models with lower m_{try} performs well when these weaker predictors have unique contributions.

The strength of the random forest algorithm lies in its flexibility. By growing trees in multiple layers it can accommodate for complex interactions between predictors. However, while it splits predictors at optimal points, it does not capture linear relationships with the outcome as well as logistic regression. From an interpretability standpoint it is also inferior to logistic regression (Hastie, Tibshirani, & Friedman, 2009; Kuhn & Johnson, 2013).

Choosing Tuning Parameters

We chose form a set of candidate values. For the mixing parameter lambda in elastic net these were increments of 0.1 between 0 and 1. For the penalty parameter λ we tested all values in increments of 0.01 between 0 and 1. For m_{try} in random forest we tested 11 candidates based on the number of predictors, including the often recommended square root of the number of predictors. We test five smaller and five bigger values in relation to this. The sequence of tested values are the number of predictors raised to the power of 1/12, 1/6, 1/4, 1/3, 5/12, 1/2, (i.e. the square root), 7/12, 2/3, 3/4, 5/6, and 11/12.

Choosing the best tuning parameters

Tuning parameters were tested via cross-validation in the training set. We first trained and tested the models with the candidate parameters in two repeats of a fourfold cross-validation. The best parameters in these 8 resamples was identified and tuning parameters that resulted in worse log-likelihood in all 8 resamples were excluded from further analyses. (This reduces the number of models tested and speeds up the analyses). All other tuning parameters were retested in 25 repeats of four folds. The best tuning parameters, per model, based on these 100 resamples were chosen and rerun in 250 repeats of four folds to create the 1000 resamples used to compare models.

Statistical Software

We used the software environment *R* (Version 3.4.3; R Core Team, 2017) for all analyses. For training the model and cross-validating them in the training set we used the package *caret* (Version 6.0-78; Kuhn, 2008). Together with caret we used the packages *glmnet* (Version 2.0-13; Friedman, Hastie, & Tibshirani, 2010) for elastic net regression and *randomForest* (Version 4.6-12; Liaw & Wiener, 2002) for random forest. AUC values and their bootstrapped confidence intervals were computed using the package *pROC* (Version 1.10-0; Robin et al., 2011). Plots were built using the package *ggplot2* (Version 2.2.1; Wickham, 2009).

References

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. New York: Wadsworth International Group.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. New York, NY: Springer New York. https://doi.org/10.1007/b94608

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, *12*(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, *28*(5), 1–26. https://doi.org/10.18637/jss.v028.i05

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-6849-3

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. Retrieved from http://cran.r-project.org/doc/Rnews/

R Core Team. (2017). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.r-project.org

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., & Müller, M. (2011).

pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, *12*(77), 1–8. https://doi.org/10.1186/1471-2105-12-77

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273–282. https://doi.org/10.1111/j.1467-9868.2011.00771.x

Wickham, H. (2009). *ggplot2: Elegant graphics for data analysis*. New York, NY: Springer. https://doi.org/10.1007/978-0-387-98141-3

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x