KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Machine Learning on Belgian Health Expenditure Data

Data-driven screening for type 2 diabetes

Marc Claesen

Supervisor:
Prof. dr. ir. Bart De Moor

Co-supervisor:
Prof. dr. ir. dr. Frank De Smet

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

November 2015

Machine Learning on Belgian Health Expenditure Data

Data-driven screening for type 2 diabetes

Marc CLAESEN

Examination committee:
Prof. dr. ir. The Chairman, chair
Prof. dr. ir. Bart De Moor, supervisor
Prof. dr. ir. dr. Frank De Smet, co-supervisor
Prof. dr. ir. Johan Suykens
Prof. dr. Chantal Mathieu
Prof. dr. ir. Hendrik Blockeel
Prof. dr. ir. Jesse Davis
Prof. dr. Marco Loog
(TU Delft)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

November 2015

Uitgegeven in eigen beheer, Marc Claesen, Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven (Belgium)

Preface

. . .

Abstract

. . .

Beknopte samenvatting

...

Abbreviations

A1C	glycated hemoglobin
ADA	American Diabetes Association
ATC	anatomical therapeutic chemical
AUC	area under the curve
AUPR	area under the PR curve
AUROC	area under the ROC curve
BAG	bagging SVM
BMI	body mass index
CDF	cumulative distribution function
CI	confidence interval
CMA-ES	covariance matrix adaptation evolutionary strategy
CVD	cardiovascular disease
CWSVM	class-weighted support vector machine
DDD	defined daily dose
DPP-4	dipeptidyl peptidase-4
ECDF	empirical cumulative distribution function
FN	false negative
FP	false positive
FPR	false positive rate
GA	genetic algorithm
gcc	GNU compiler collection
GDM	gestational diabetes mellitus
GLA	glucose lowering agent

GLP-1	glucagon-like peptide-1
GP	general practitioner
HR	hazard ratio
IDDM	insulin-dependent diabetes mellitus
IFG	impaired fasting glucose
IGT	impaired glucose tolerance
JSON	Javascript object notation
NACM	National Alliance of Christian Mutualities
NIDDM	noninsulin-dependent diabetes mellitus
NIHDI	National Institute for Health and Disability Insurance
NNT	number needed to treat
OAD	oral antidiabetic drug
OGTT	oral glucose tolerance test
PR	precision-recall
PSO	particle swarm optimization
PU learning	learning from positive and unlabeled data
RBF	radial basis function
RBM	restricted Boltzmann machine
RESVM	robust ensemble of support vector machines
ROC	receiver operating characteristic
SU	sulfonylurea (anti-diabetic medication)
SV	support vector
SVC	support vector classifier
SVM	support vector machine
T2D	type 2 diabetes mellitus
TN	true negative
TP	true positive
TPE	tree-structured Parzen estimators
TPR	true positive rate
TZD	thiazolidinediones (anti-diabetic medication)
WHO	World Health Organization

List of Symbols

Ω	Asymptotic lower bound on (time) complexity
τ	Bound on rank cumulative distribution function
$\mathcal{C}$	Classifier, e.g. trained SVM model
$\mathbf{X}^{(te)}$	Data set for testing
$\mathbf{X}^{(tr)}$	Data set for training
$\mathbf{X}$	Data set
$\mathcal{U}$	Data without label (unlabeled)
$\mathcal{R}$	Data, ranked based on model output
$\hat{\beta}$	Fraction of positives in the unlabeled set (estimated)
β	Fraction of positives in the unlabeled set (truth)
λ	Hyperparameters of a learning/modeling method
φ	Kernel embedding function (input space → kernel feature space)
κ	Kernel function: $\kappa(\mathbf{u}, \mathbf{v}) = \langle \varphi(\mathbf{u}), \varphi(\mathbf{v}) \rangle$
$\mathcal{N}$	Negative class
$\mathcal{N}_L$	Negatives with known label (known negatives)
$\mathcal{P}$	Positive class
$\mathcal{P}_U^{\star}$	Positive surrogates, unlabeled instances treated as positives
$\mathcal{P}_L$	Positives with known label (known positives)
$\mathcal{P}_U$	Positives without known label (latent positives)

$\mathcal{P}_\Omega$	Positives (all): set union of known and latent positives
$\mathcal{P}_\Omega^\star$	Positives: union of known and surrogate positives ($\approx \mathcal{P}_\Omega$)
$C_\mathcal{P}$	SVM hyperparameter: misclassification penalty on positives
$C_\mathcal{U}$	SVM hyperparameter: misclassification penalty on unlabeled data
C	SVM hyperparameter: misclassification penalty
ρ	SVM model bias
$\mathbf{w}$	SVM model separating hyperplane

Contents

Abstract iii

Contents xi

List of Figures xvii

List of Tables xxi

1 Introduction 1

1.1 Diabetes mellitus 1

1.1.1 Regulation of blood glucose levels 2

1.1.2 Complications and comorbidities 2

1.1.3 Classification of diabetes mellitus 3

1.1.4 Treatment of diabetes mellitus 4

1.1.5 Glucose level regulation as a state-space system 5

1.2 Screening for diabetes 7

1.2.1 Using Belgian health expenditure data 8

1.3 Belgian mutual health insurance 8

1.3.1 Health expenditure records 9

1.3.2 A note on privacy 9

1.4 Machine learning challenges and contributions 9

1.4.1 Learning from positive and unlabeled data 10

1.4.2 Automated hyperparameter optimization 10

1.4.3 Open-source software 10

1.5 Structure of the thesis 10

2 Mortality in individuals treated with glucose lowering agents: a large, controlled cohort study 13

2.1 Introduction 15

2.2 Research design and methods 16

2.2.1 Study cohort selection 16

2.2.2 Control cohort selection 19

2.2.3 Therapy changes within cohorts 19

2.2.4 Censoring 19

2.2.5 Statistical analysis 21

2.3 Results 21

2.3.1 Baseline cohort characteristics 21

2.3.2 Five-year survival in individuals on different glucose lowering agents 22

2.3.3 Age-dependent 5-year survival of individuals on different glucose lowering agents 26

2.3.4 Statins and survival in individuals on different glucose lowering therapy 26

2.4 Conclusions 30

3 EnsembleSVM: A Library for Ensemble Learning Using Support Vector Machines 33

3.1 Introduction 34

3.2 Software Description 34

3.2.1 Implementation 35
3.2.2 Tools 36
3.3 Benchmark Results 36
3.4 Conclusions 37

4 SVM Ensemble Learning from Positive and Unlabeled Data 39

4.1 Introduction 40
4.2 Related work 41
4.2.1 Class-weighted SVM 41
4.2.2 Bagging SVM 42
4.3 Robust Ensemble of SVMs 43
4.3.1 Bootstrap resampling contaminated sets 43
4.3.2 Bagging predictors 44
4.3.3 Justification of the RESVM algorithm 45
4.3.4 RESVM training 46
4.3.5 RESVM prediction 47
4.4 Experimental setup 48
4.4.1 Simulation setup 48
4.4.2 Data sets 51
4.5 Results and discussion 53
4.5.1 Results for supervised classification 53
4.5.2 Results for PU learning 55
4.5.3 Results of semi-supervised classification 55
4.5.4 A note on the number of repetitions per experiment . . 57
4.5.5 Trend across data sets 58
4.5.6 Effect of contamination 60
4.5.7 RESVM optimal parameters 62

4.6 Conclusion 63

5 Hyperparameter Search in Machine Learning 65

5.1 Introduction 66

5.1.1 Example: controlling model complexity 66

5.1.2 Formalizing hyperparameter search 67

5.2 Challenges in hyperparameter search 67

5.2.1 Costly objective function evaluations 67

5.2.2 Randomness 68

5.2.3 Complex search spaces 68

5.3 Current approaches 69

5.4 Conclusion 69

6 Easy Hyperparameter Search Using Optunity 71

6.1 Introduction 72

6.2 Optunity 72

6.2.1 Functional Overview 72

6.2.2 Available Solvers 73

6.2.3 Software Design and Implementation 74

6.2.4 Development and Documentation 75

6.3 Related Work 75

6.4 Solver Performance Benchmark 75

7 Assessing Binary Classifiers Using Only Positive and Unlabeled Data 77

7.1 Introduction 78

7.2 Background and definitions 79

7.2.1 Rank distributions and contingency tables 79

7.2.2 ROC and PR curves 80

7.2.3 Evaluation with partially labeled data 81

7.3 Relationship between the rank CDF of positives and contingency tables 81

7.3.1 Rank distributions and contingency tables based on subsets of positives within a ranking 82

7.3.2 Contingency tables based on partially labeled data . . . 82

7.4 Efficiently computing the bounds 83

7.4.1 Computing the contingency table with greatest-lower bound on FPR at given rank r 83

7.4.2 Bounds on the rank distribution of $\mathcal{P}_U$ 85

7.5 Constructing ROC and PR curve estimates 85

7.6 Discussion and Recommendations 87

7.6.1 Determining $\hat{\beta}$ and its effect 87

7.6.2 Model selection 88

7.6.3 Empirical quality of the estimates 89

7.6.4 Relative importance of known negatives compared to known positives 90

7.7 Conclusion 91

Appendices 92

7.A Proofs 92

7.B Effect of $\hat{\beta}$ on contingency table entries and common performance metrics 94

7.C The effect of the fraction of known positives, known negatives and $\hat{\beta}$ 97

8 Building Classifiers to Predict the Start of Glucose-Lowering Pharmacotherapy Using Belgian Health Expenditure Data 99

8.1 Introduction 100

8.2 Existing Type 2 Diabetes Risk Profiling Approaches 101

8.3 Health Expenditure Data 102

8.3.1 Records Related to Drug Purchases 103

8.3.2 Records Related to Medical Provisions 103

8.3.3 Advantages of Health Expenditure Data 104

8.3.4 Limitations of Health Expenditure Data 104

8.4 Methods 105

8.4.1 Experimental Setup 106

8.4.2 Data Set Construction 109

8.4.3 Learning Methods 113

8.5 Results and Discussion 115

8.5.1 Benchmark of learning methods 115

8.5.2 Performance Curves for RESVM 118

8.5.3 Feature Importance Analysis for the RESVM Model . . 119

8.6 Conclusion 119

9 This is conclusion 121

A This is myappendix 123

Bibliography 125

This is curriculum 147

List of Figures

1.1 Dependencies of the key contributions made to machine learning research during this project. To enable diabetes screening based on Belgian health expenditure data (Chapter 8), we made contributions to semi-supervised learning (Chapters 4 and 7), automated hyperparameter search (Chapters 5 and 6) and open-source machine learning software (Chapters 3 and 6). 11

2.1 Flowchart describing the selection protocol for study and control patients. Patients can move from the bottom right (monotherapy) to the bottom left group (combination therapy), but not vice versa. All listed counts are for unique patients. 17

2.2 5-year survival for increasing age per cohort. 23

2.3 5-year survival for increasing age per cohort. 28

2.4 5-year survival for increasing age per cohort with matched controls. 29

4.1 Contamination of bootstrap resamples for increasing size of resamples when the original sample has 10% contamination. Errorbars indicate the 95% confidence interval (CI) of contamination in resamples. The contamination varies greatly between small resamples as shown by the CIs. 44

4.2 Overview of a single benchmark iteration. 49

4.3 Empirical densities of the `synthetic` data used for training per problem setting (visualized in input space). The supervised densities (top row) are based on samples of the underlying positive and negative classes. The use of high contamination (30%) induces similar empirical densities for $\mathcal{P}$ and $\mathcal{U}$ in the semi-supervised setting (bottom row). 52

4.4 Performance in semi-supervised setting on `mnist`, digit 7 as positive. 58

4.5 Critical difference diagrams for each setting. Groups of algorithms that are not significantly different at the 5% significance level are connected. 59

4.6 Effect of different levels of contamination in $\mathcal{U}$ and $\mathcal{P}$ on generalization performance. The plots show point estimates of the mean area under the PR curve across experiments and the associated 95% confidence intervals. 61

6.1 Integrating OPTUNITY in non-Python environments. 74

6.2 Evolution of average error per solver while minimizing the RASTRIGIN function (contours depicted on the right) based on 1,000 repetitions. The winner line shows the statistically significant best solver per evaluation count, if any ($\alpha = 5\%$). . . 76

7.1 Rank CDF of two sets of positives $\mathcal{P}_1 = \{B, D, A, C\}$ and $\mathcal{P}_2 = \{E, G, F\}$ within an overall ranking $\mathcal{R} = \{B, E, D, G, H, A, F, C, I\}$, with $|\mathcal{P}_1| = 4$ and $|\mathcal{P}_2| = 3$. In practice $\mathcal{R}$ is obtained by sorting the data according to classifier score. The rank CDF of a set $\mathcal{S} \subseteq \mathcal{R}$ is based on the positions of elements of $\mathcal{S}$ in $\mathcal{R}$. 80

7.2 The effect of $|\mathcal{P}_L|$ on estimated AUC. Based on $|\mathcal{U}| = 100,000$, $\mathcal{N}_L = \emptyset$ and $\hat{\beta} = \beta = 0.2$. Bounds on rank CDF were obtained via bootstrap. The depicted confidence intervals are based on 200 repeated experiments. 88

7.3 The effect of $\hat{\beta}$ on estimated ROC curves, based on 2,000 known positives, 100,000 unlabeled instances and $\beta = 0.3$. 88

7.A.1Illustration of Lemma 1: higher TPR at a given rank r implies lower FPR at r for two positive sets of the same size. 92

7.A.2Illustration of Lemma 2: $\mathcal{F}(\cdot)$ denotes feasible region. The rank distribution of the union $\mathcal{P}_\Omega$ of two sets of positives $\mathcal{P}_1$ and $\mathcal{P}_2$ lies between their respective rank distributions. 93

8.1 Overview of the full learning approach: data set vectorization, normalization and the nested cross-validation setup. Per iteration, hyperparameter optimization and model training is done based exclusively on $\mathbf{X}_{train}^{(outer)}$ 107

8.2 Visualization and vectorization of trees. In the tree representation, the value of internal nodes is the sum of the values of its children. The unnormalized vector representations $\mathcal{V}_A$ and $\mathcal{V}_B$ contain the values per node in the tree representation in some fixed order. Inner products between unnormalized representations $\mathcal{V}_A$ and $\mathcal{V}_B$ are mainly influenced by the top level nodes, since those have the largest value by construction. This undesirable effect can be fixed through feature-wise scaling. The scaling vector $\mathcal{S}$ was constructed using node-wise maxima. The normalized vector representations $\mathcal{V}_A^\star$ and $\mathcal{V}_B^\star$ are obtained by dividing the vector representations ($\mathcal{V}_A$, $\mathcal{V}_B$) element-wise by entries in the scaling vector $\mathcal{S}$. $\mathcal{V}_A^\star$ and $\mathcal{V}_B^\star$ are used as input to classifiers in the remainder of this work. As desired, the inner product of normalized vector representations is increasingly influenced by similarities at higher depths in the tree representations. 111

8.3 Tensor formulation of medical provisions with three components: patients, physicians and provisions. Each entry in the tensor is the frequency of the given tuple. This provision tensor is very sparse. The patient matrix is obtained by summing counts over all physicians (transposed). The physician matrix is obtained by summing counts over all patients. These matrices capture complementary information. 111

8.4 Structure of the provision similarity matrix $\mathbf{S}_{prov}$ based on providing physicians. 113

8.1 Performance curves for the best model: RESVM classifier based on ATC | PROVS vectorization. The lower and upper bounds are estimated using $\hat{\beta}_{lo} = 5\%$ and $\hat{\beta}_{up} = 10\%$, respectively. 118

List of Tables

1.1 ADA criteria for testing for diabetes in asymptomatic adult individuals, taken from [6]. 8

2.1 Definitions of drug categories and cardiovascular events. 18

2.2 Partitioning of final treatment regimen of patients starting a specific therapy (rows). The final treatment regimen (columns) is based on the last 9 months of individual followup. Patients are censored in all survival analyses after 9 consecutive months of renunciation from metformin, sulfonylurea and insulin. . . . 20

2.3 Overview of survival for the study cohorts compared to a fully matched control cohort, stratified by cv history. The control group is sampled from the general population and matched for age, gender and use of statins, antihypertensives and antiplatelet drugs. For every patient in the study cohorts, 5 patients with completely matching profiles were used in control. • indicates that the proportional hazards assumption was rejected for the associated hazards ratio ($p < 0.01$). 24

2.4 Baseline characteristics of the study cohorts. Individuals starting starting mono therapy are selected such that they have no prior history of diabetes-related drugs. Individuals starting combination therapy may have a prior history of diabetes-related drugs. All differences in use of associated therapy are statistically significant between study cohorts, except for use of antihypertensives in insulin and sulfonylurea mono cohorts (no significant difference). All associated therapy use is statistically significantly elevated in subgroups with prior cardiovascular events within each study cohort. All comparisons of study group characteristics use significance level $\alpha = 0.05$ and are computed using Tukey's test in conjunction with ANOVA to adjusts for multiple comparison. 25

2.5 Analysis of the effect of statins within each study cohort. Presented hazard ratios are associated to the fraction of follow-up on statins. Patients are classified as statin users if they were on statins for at least half the follow-up. The proportional hazards models used here control for age, gender, use of antihypertensive and antiplatelet drugs and an age-gender interaction. • indicates that the proportional hazards assumption was rejected for the associated hazards ratio ($p < 0.01$). 27

2.6 Hazard ratios between statin users in the study group a control group matched for age and gender. The remaining characteristics of the control groups used here follow the distribution of the total NACM member population (after matching for age and gender). The proportional hazard models used to compute these hazard ratios control for age, gender, the use of antihypertensive and antiplatelet drugs and an age-gender interaction. • indicates that the proportional hazards assumption was rejected for the associated hazards ratio ($p < 0.01$). 30

3.1 Summary of benchmark results per data set: test set accuracy, number of support vectors and training time. Accuracies are listed for a single `LIBSVM` model, `LIBLINEAR` model and an ensemble model. 37

4.1 Overview of the data sets used in simulations: number of features, contamination (when applicable), training set size as used in the experiments and test set size. The `mnist` data set consists of 10 classes and the test set is almost uniformly distributed. The `sensit` data set has 3 classes with uneven class distribution in the test set, so we treat it separately here. 53

4.2 95% CIs for mean test set performance in a fully supervised setup, the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $h_1 : AUC^{RESVM} > AUC^{BAG}$ and the number of times each approach had best test set performance. Test result encoding: $\bullet\ p < 0.05$, $\bullet\bullet\ p < 0.01$ and $\bullet\ \bullet\ \bullet\ p < 0.001$. . . . 54

4.3 95% CIs for mean test set performance in a PU learning setup, the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $h_1 : AUC^{RESVM} > AUC^{BAG}$ and the number of times each approach had best test set performance. Test result encoding: $\bullet\ p < 0.05$, $\bullet\bullet\ p < 0.01$ and $\bullet\ \bullet\ \bullet\ p < 0.001$. 56

4.4 95% CIs for mean test set performance in a semi-supervised setup, the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $h_1 : AUC^{RESVM} > AUC^{BAG}$ and the number of times each approach had best test set performance. Test result encoding: $\bullet\ p < 0.05$, $\bullet\bullet\ p < 0.01$ and $\bullet\ \bullet\ \bullet\ p < 0.001$. . . . 57

4.5 Number of wins in simulations for each method per setting. The bottom half shows normalized number of wins, where wins in multiclass data sets (mnist and sensit) are divided by the number of classes. 60

4.6 Medians of optimal hyperparameters per digit obtained via cross-validation and mean of all medians per setting. The normalized relative weight on positives versus unlabeled instances (w_{pos}) is associated with the relative size and contamination of the positive and unlabeled training sets. 62

8.1 Example of the ATC classification system: classification of metformin per level. 103

8.1 Summary of vectorization schemes used for records of drug purchases. 110

8.2 Summary of vectorization schemes used for records of medical provisions. 113

8.1 Average bounds on area under the ROC curve and p-value of the Mann-Whitney U test over all folds for different feature sets per learning approach in a long-term prediction setup. The lower and upper bounds on AUC were computed with $\hat{\beta}_{lo} = 0.05$ and $\hat{\beta}_{up} = 0.10$, respectively. The ATC | PROVS feature set is the concatenation of the best performing sets per aspect, namely ATC 1–5 and PROVS BOTH. Stars ($*$) denote p-values below 0.005. 116

Chapter 1

Introduction

In this work we explored the ability of screening for (type 2) diabetes mellitus based on Belgian health insurance data from a machine learning perspective. The thesis is organized as a collection of papers concerning various aspects related to this application. This chapter provides some context of the project, both in terms of the medical situation and the challenges in terms of data analysis.

We will first briefly introduce diabetes mellitus and describe the main characteristics of the disease in Section 1.1. Then, we will summarize existing methods of screening for diabetes that are actively used in practice, and how our approach relates to existing methods in Section 1.2. The Belgian health insurance landscape is described in Section 1.3. Subsequently, Section 1.4 contains a discussion of the main challenges of this project from a machine learning perspective to explain the necessity of each aspect of our research. Finally, Section 1.5 summarizes the structure of the text and indicates how all chapters (papers) are related to each other.

1.1 Diabetes mellitus

Diabetes mellitus is a metabolic disorder characterized by chronic hyperglycemia, which is primarily caused by insufficient insulin secretion and/or insulin resistance [4].

In the remainder of this Section we provide a brief overview of various aspects of the disease: the underlying biological problem, common complications and comorbidities, classification of diabetes based on etiology, treatment approaches and finally a simple mathematical model to describe blood glucose regulation.

Type 2 diabetes is considered an epidemic [142].

todo: epidemiology

1.1.1 Regulation of blood glucose levels

To allow a better understanding of the underlying problems inherent to diabetes mellitus, we will briefly describe the critical role of insulin in the regulation of blood glucose levels.

Insulin is a peptide hormone which regulates the metabolism of fats and carbohydrates. In normal circumstances, insulin is released in response to changes in blood glucose concentration to prevent glucose levels from reaching toxic concentrations. Specifically, insulin promotes the absorption of glucose from the blood to skeletal muscles and fat tissue, thereby lowering the glucose level in the bloodstream, and additionally inhibits hepatic glucose output [181]. Insulin is exclusively produced by pancreatic β cells, which are located in clusters known as the islets of Langerhans.

Hyperglycemia occurs when the regulation of the blood glucose level fails and hence toxic concentrations of glucose remain in the bloodstream. Such failures are typically caused by excessive glucose intake, insufficient insulin secretion and/or insulin resistance.

1.1.2 Complications and comorbidities

Exposure to chronic hyperglycemia can cause serious damage to many of the body's systems, including dysfunction and failure of various organs, and in severe cases may induce ketoacidosis[1] or the development of a non-ketotic hyperosmolar state[2] which can lead to stupor, coma or even death [4]. On the long-term, diabetes mellitus may lead to the progressive development of retinopathy (which potentially results in blindless), neuropathy (which can induce problems like foot ulcers) and nephropathy (which can lead to renal failure). Finally, patients with diabetes mellitus are at increased risk for cardiovascular, peripheral vascular and cerebrovascular disease.

todo: comorbidities

[1] This means the body's cells use ketones (toxic acids) as energy source, rather than glucose.

[2] When blood sugar levels rise, the body attempts to excrete excess glucose via urine.

1.1.3 Classification of diabetes mellitus

Diabetes mellitus is a disorder of multiple etiologies and is subdivided into several more specific classes. The primary distinction between subclasses is essentially whether or not external insulin is necessary for survival. The first widely accepted classification was made by the World Health Organization (WHO) in 1980 [146], proposing two main classes of diabetes mellitus:

- **Insulin-dependent diabetes mellitus (IDDM)**, also known as **type 1** diabetes (T1D), refers to a condition characterized by insufficient insulin secretion caused by the autoimmune destruction of pancreatic β-cells. T1D patients require insulin for survival. T1D can start at an early age and has a strong genetic component.
- **Noninsulin-dependent diabetes mellitus (NIDDM)**, also known as **type 2** diabetes (T2D), results from defect(s) in insulin secretion, almost always paired with insulin resistance. T2D includes patients that require insulin for metabolic control (but not survival) and those that don't require insulin at all. The onset of T2D typically occurs during adulthood and is often a result of chronic obesity.

The WHO deprecates the terms IDDM and NIDDM, as their use has frequently led to patients being classified based on treatment, rather than pathogenesis [146]. Hence, contemporary terminology uses type 1 and type 2 diabetes to denote the main classes of diabetes mellitus. Though our work focuses on type 1 and particularly type 2 diabetes mellitus, it must be noted that more forms exist, such as gestational diabetes mellitus which may occur during pregnancy and then disappear.

Type 2 diabetes is typically preceded by the following metabolic abnormalities:

- *Impaired fasting glucose (IFG)*, also known as prediabetes, is a condition in which fasting blood glucose levels are consistently higher than normal, but not high enough to warrant a diabetes diagnosis. Some patients with IFG can also be diagnosed with impaired glucose tolerance.
- *Impaired glucose tolerance (IGT)* is a prediabetic state of hyperglycemia which may precede T2D by many years. Patients with IGT exhibit raised glucose levels after 2 hours compared to healthy people, but not high enough to qualify for T2D. Patients with IGT present a higher risk for diabetes than patients with IFG.

Both IFG and IGT are associated with insulin resistance and increased risk of diabetes and cardiovascular pathologies, with IGT being more strongly associated with cardiovascular outcomes [191]. Although the transition of IFG and/or IGT to diabetes may take many years, the majority of individuals with these pre-diabetic states eventually develop diabetes [188, 60, 142].

Finally, the *metabolic syndrome* encompasses the most dangerous heart attack risk factors, which include diabetes, IFG/IGT, abdominal obesity, high cholesterol and high blood pressure [2, 73].

1.1.4 Treatment of diabetes mellitus

The fact chronic hyperglycemia may manifest in various ways is reflected in a wide variety of treatments for diabetes mellitus. Treatment of type 1 diabetes patients revolves around timely administration of external insulin, which they need for survival. In contrast, a wide range of treatment options exist for patients with type 2 diabetes.

Management of type 2 diabetes includes several lifestyle interventions: healthy eating (specifically high-fiber, low-fat foods like fruits and vegetables), regular exercise, smoking cessation and blood sugar monitoring and management. Pharmacotherapy via glucose lowering agents (GLAs) is used when lifestyle changes alone are insufficient for adequate glycemic control or when diabetes is already in a progressed stage at the time of clinical diagnosis.

We will elaborate on pharmacological treatment of type 2 diabetes, as this has played a crucial role in our work. Pharmacological therapies for (type 2) diabetes may be based on various biological mechanisms:

- **External insulin** is administered when insufficient insulin is secreted by the pancreas. External insulin is always necessary to treat type 1 diabetes but may also be required to treat insulin deficiency in type 2 diabetes. Insulin cannot be taken as a pill because it would be broken down during digestion just like protein in food. Instead, it must be injected or inhaled. Insulins are classified via ATC code `A10A`.
- **Sensitizers** reduce the insulin resistance that is central in type 2 diabetes.
  - Biguanides suppress hepatic glucose output and increase uptake of glucose by the periphery. The most common agent in this class is *metformin* (brand names include *Glucophage*, *Glucovance*) [114]. Biguanides are classified via ATC code `A10BA`.
  - Thiazolidinediones (TZDs) enhance the effects of insulin by increasing insulin-dependent glucose disposal and reducing hepatic glucose output (as a result of increased hepatic insulin sensitivity) [167, 207].

An example of TZDs is *pioglitazone* (brand name *Actos*).
Thiazolidinediones are classified via ATC code `A10BG`.

- **Secretagogues** increase insulin output from the pancreas. The main type of secretagogues – sulfonylurea (SU) – stimulate endogenous insulin secretion from pancreatic β-cells [157]. Hypoglycemia is a major concern when using sulfonylureas [24]. The most common SU are *glimepiride* (brand name *Amaryl*), *glibenclamide* (*Euglucon* and *Daonil*), *gliclazide* (*Diamicron*), *glipizide* (*Glucotrol*) and *gliquidone* (*Glurenorm*).
  These agents are classified via ATC code `A10BB`.

- **Alpha-glucosidase inhibitors** slow down digestion of starch in the small intestine, thereby reducing the rate at which the resulting glucose enters the bloodstream. These agents do not have a direct effect on insulin secretion or sensitivity, but can (i) be sufficiently efficient in early stages of impaired glucose tolerance or (ii) be used in combination with other antidiabetic agents. A common alpha-glucosidase inhibitor is *acarbose* (brand name *glucobay*).
  These agents are classified via ATC code `A10BF`.

- **Incretin-based therapies** use the antidiabetic properties of the incretin hormone glucagon-like peptide 1 (GLP-1), namely that GLP-1 augments glucose-induced insulin secretion in a highly glucose-dependent manner [143, 131]. As this form of insulin secretion occurs in a glucose-dependent manner, incretin-based therapies are less prone to cause hypoglycemia.

  – GLP-1 receptor agonists activate GLP-1 receptors, resulting in increased insulin synthesis and release [62]. Common GLP-1 receptor agonists include *exenatide* (brand name *Byetta*) and *lixisenatide* (brand name *Lyxumia*).
  These agents are classified via ATC code `A10BX`.

  – Dipeptidyl peptidase-4 (DPP-4) inhibitors increase the blood concentration of GLP-1 by inhibiting its degradation caused by the enzyme DPP-4. Common DPP-4 inhibitors include *sitagliptin* (brand name *Januvia*), *vildagliptin* (brand name *Galvus*).
  These agents are classified via ATC code `A10BH`.

1.1.5 Glucose level regulation as a state-space system

With significant simplifications, we can summarize the glucose level regulatory mechanism as a dynamical system that is controlled via a basic feedback loop. This coarse approximation allows us to understand the main mechanisms of diabetes mellitus treatments on a mathematical level.

The system's states are $[glucose](t)$ and $[insulin](t)$ (concentrations) and its inputs are $\Delta glucose(t) \geq 0$ and $\Delta insulin(t) \geq 0$ which enter the bloodstream.

- The glucose concentration is influenced negatively by the concentration of insulin ($A_{i\to g} < 0$) and positively by external input $\Delta glucose(t)$ ($B_g > 0$):

$$\frac{d[glucose](t)}{dt} = A_{i\to g}[insulin](t) + B_g \Delta glucose(t)$$

- The insulin concentration is influenced by insulin injection $\Delta insulin(t)$ and insulin secretion $\mathcal{S}_{insulin}(t)$, which in turn is influenced by the glucose level, i.e. $\mathcal{S}_{insulin}(t) = K[glucose](t)$ (with $K > 0$, $B_s > 0$ and $B_i > 0$):

$$\begin{aligned}\frac{d[insulin](t)}{dt} &= B_s \mathcal{S}_{insulin}(t) + B_i \Delta insulin(t), \\ &= B_s K[glucose](t) + B_i \Delta insulin(t),\end{aligned}$$

  where B_i is a normalization based on volume and K is a positive constant defining the amount of insulin secreted at a given glucose level.

This can be written as the following state-space system with two inputs:

$$\begin{bmatrix} \frac{d[glucose]}{dt} \\ \frac{d[insulin]}{dt} \end{bmatrix} = \begin{bmatrix} 0 & A_{i\to g} \\ B_s K & 0 \end{bmatrix} \begin{bmatrix} [glucose](t) \\ [insulin](t) \end{bmatrix} + \begin{bmatrix} B_g & 0 \\ 0 & B_i \end{bmatrix} \begin{bmatrix} \Delta glucose(t) \\ \Delta insulin(t) \end{bmatrix}.$$

Nature has somehow determined an optimal feedback gain $K^\star$ that usually works (or none of us would be here). The essential problems underlying diabetes can now be formalized as follows:

- Insufficient insulin secretion can be formalized as $K < K^\star$,
- Insulin resistance can be formalized as $A_{i\to g} > A^\star_{i\to g}$.

Both of these effects imply a real control action that is smaller than optimal, directly leading to hyperglycemia. Treatment of diabetes mellitus generally attempts to do one of the following:

- Decrease the amount of glucose entering the bloodstream.
  Equivalent to $\Delta glucose(t) \searrow$.
- Increase insulin levels manually by injecting insulin, or increase secretion.
  Equivalent to $\Delta insulin(t) \nearrow$ and $K \nearrow$, respectively.

- Decrease insulin resistance in cells, to make them absorb glucose faster. Equivalent to $A_{i \to g} \searrow$.

The approximated state-space system also indicates that for type 1 diabetes – in which $\mathcal{S}_{insulin}(t)$ has a low saturation level – treatments that attempt to increase insulin secretion or decrease insulin resistance are ineffective. Hence, treatment of type 1 diabetes requires insulin injections which explains its alias of insulin-dependent diabetes mellitus (IDDM).

1.2 Screening for diabetes

Diabetes mellitus may remain asymptomatic for extended periods of time.

Type 2 diabetes is frequently not diagnosed until complications appear [6] . The ADA estimates that one-fourth of the US population may have undiagnosed diabetes [6].

uit [6]: Mass screening of asymptomatic individuals has not effectively identified those with prediabetes or diabetes, and rigorous clinical trials to provide such proof are unlikely to occur. In a large randomized controlled trial (RCT) in Europe, general practice patients between the ages of 40–69 years were screened for diabetes, then randomized by practice to routine diabetes care or intensive treatment of multiple risk factors. After 5.3 years of follow-up, CVD risk factors were modestly but significantly improved with intensive treatment. Incidence of first CVD event and mortality rates were not significantly different between groups [83]. This study would seem to add support for early treatment of screen-detected diabetes, as risk factor control was excellent even in the routine treatment arm and both groups had lower event rates than predicted. The absence of a control unscreened arm limits the ability to definitely prove that screening impacts outcomes. Mathematical modeling studies suggest that screening, independent of risk factors, beginning at age 30 or 45 years is highly cost-effective $11,000 per quality-adjusted life-year gained) [108].

1. Testing should be considered in all adults who are overweight (BMI ≥ 25 kg/m^{2*}) and have additional risk factors:
   - physical inactivity,
   - first-degree relative with diabetes,
   - high-risk race/ethnicity (e.g., African American, Latino, Native American, Asian American, Pacific Islander),
   - women who delivered a baby weighing > 9 lb or were diagnosed with GDM,
   - hypertension ($\geq 140/90$ mmHg or on therapy for hypertension),
   - HDL cholesterol level < 35 mg/dL (0.90 mmol/L) and/or a triglyceride level > 250 mg/dL (2.82 mmol/L),
   - women with polycystic ovarian syndrome,
   - A1C $\geq 5.7\%$, IGT, or IFG on previous testing,
   - other clinical conditions associated with insulin resistance (e.g., severe obesity, acanthosis nigricans)
   - history of CVD.
2. In the absence of the above criteria, testing for diabetes should begin at age 45 years.
3. If results are normal, testing should be repeated at least at 3-year intervals, with consideration of more frequent testing depending on initial results (e.g., those with prediabetes should be tested yearly) and risk status.

* At-risk BMI may be lower in some ethnic groups.

Table 1.1: ADA criteria for testing for diabetes in asymptomatic adult individuals, taken from [6].

Symptoms

1.2.1 Using Belgian health expenditure data

Strengths and advantages

Weaknesses and limitations

1.3 Belgian mutual health insurance

This project was done in close collaboration with the National Alliance of Christian Mutualities (NACM)[3]. NACM is the largest Belgian mutual health

[3] Landsbond der Christelijke Mutualiteiten (LCM).

insurer with over 60% and 40% market share in Flanders and Belgium, respectively.

The Belgian health care insurance is a broad solidarity-based form of social insurance. Mutual health insurers are the legally-appointed bodies for managing and providing the Belgian compulsory health care and disability insurance. Joining one of several mutual health insurers or, alternatively, the relief fund for sickness and disability insurance [4] is obliged for anyone who (i) starts working, (ii) is still studying at the age of 25 years or (iii) receives unemployment benefits. Among other things, mutual health insurers are responsible for refunding medical interventions, drug purchases and payments related to disability and pregnancy leave.

1.3.1 Health expenditure records

Our screening system was developed to be based exclusively on basic personal information (e.g. age, gender) and readily-available health expenditure records owned by NACM, without requiring any external input.

1.3.2 A note on privacy

1.4 Machine learning challenges and contributions

This Section highlights the main machine learning contributions made during this project. We will briefly describe every research question that was tackled, outline the approach and clarify how it fits into the overarching theme.

[4] Hulpkas voor Ziekte -en Invaliditeitsuitkering (HZIV).

1.4.1 Learning from positive and unlabeled data

1.4.2 Automated hyperparameter optimization

1.4.3 Open-source software

1.5 Structure of the thesis

This Section summarizes the content of all subsequent chapters and reiterates how every aspect is relevant to diabetes screening based on Belgian health expenditure records.

- *Chapter 3* introduces the EnsembleSVM software package, which provides efficient routines for ensemble learning using SVM base models.
- *Chapter 4* describes a novel algorithm to learn robust binary classifiers from positive and unlabeled data. The key design criterion is robustness to false positives, which was lacking in existing approaches. The implementation is based on EnsembleSVM (see Chapter 3).
- *Chapter 5* discusses the main optimization challenges posed by automated hyperparameter search and summarizes the current state-of-the-art.
- *Chapter 6* describes the Optunity software package, which provides metaheuristic optimization routines for automated hyperparameter optimization. Optunity is available on most commonly used machine learning platforms and tackles the challenges outlined in Chapter 5.
- *Chapter 7* presents a method to evaluate the performance of binary classifiers without negative labels. This method enables estimating most commonly used performance metrics in a semi-supervised setting, which was uncharted territory.
- *Chapter 8* integrates all machine learning aspects into a workflow to predict which patients are likely to start glucose-lowering pharmacotherapy, based exclusively on readily available, individual health expenditure records. This chapter combines all techniques described in previous chapters.

blah blah blah

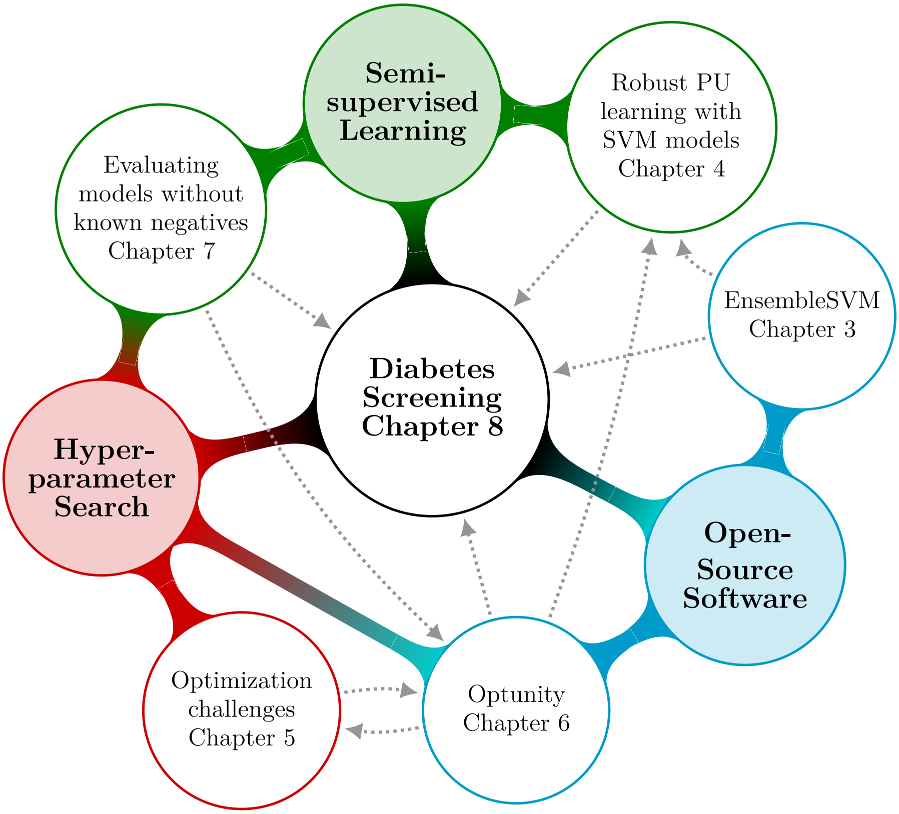

Semi-supervised Learning
Robust PU learning with SVM models Chapter 4
Evaluating models without known negatives Chapter 7
EnsembleSVM Chapter 3
Diabetes Screening Chapter 8
Hyper-parameter Search
Open-Source Software
Optimization challenges Chapter 5
Optunity Chapter 6

Figure 1.1: Dependencies of the key contributions made to machine learning research during this project. To enable diabetes screening based on Belgian health expenditure data (Chapter 8), we made contributions to semi-supervised learning (Chapters 4 and 7), automated hyperparameter search (Chapters 5 and 6) and open-source machine learning software (Chapters 3 and 6).

Chapter 2

Mortality in individuals treated with glucose lowering agents: a large, controlled cohort study

This chapter has been submitted to:
Claesen, M.⋆, Gillard, P.⋆, De Smet, F., Callens, M., De Moor, B. & Mathieu, C. (2015). **Mortality in individuals treated with glucose lowering agents: a large, controlled cohort study**, *Journal of Clinical Endocrinology and Metabolism.*
⋆: these authors have contributed equally to the manuscript.

Abstract

Context Several observational studies and meta-analyses have reported increased mortality of patients taking sulfonylurea and insulin. The impact of patient profiles and concomitant therapies often remains unclear.

Objective To quantify survival of patients after starting glucose-lowering agents (GLAs) and compare it to control subjects, matched for risk profiles and concomitant therapies.

Design Controlled, retrospective cohort study.

Setting The study is based on health expenditure records of the largest Belgian health mutual insurer, covering over 4.4 million people. Patients: 115,896 patients starting metformin, sulfonylurea or insulin (alone or in combination) between January 2003 and December 2007. Control subjects without GLA therapy were matched for age, gender, history of cardiovascular events and therapy with antihypertensives, statins and blood platelet aggregation inhibitors.

Main Outcome Measure 5-year survival after start of GLA.

Results Profiles of patients using different GLAs varied, with patients on sulfonylurea being oldest and patients on insulin having more frequently a history of cardiovascular disease. Excess mortality differed across GLA therapies compared to matched controls without GLAs, even after adjusting for observable characteristics. Only metformin monotherapy was not associated with increased 5-year mortality compared to matched controls, while individuals on combination of sulfonylurea and insulin had highest mortality risks. Age and concomitant use of statins strongly affect survival.

Conclusions Differences exist in 5-year survival of patients on GLA, at least partly driven by the risk profile of the individuals themselves. Metformin use was associated with lowest 5-year mortality risk and statins dramatically lowered 5-year mortality throughout all cohorts.

2.1 Introduction

Glucose lowering therapy in type 2 diabetes is challenging, due to the progressive nature of the disease by the underlying failure of the insulin-secreting beta-cells [155]. Algorithms and guidelines are proposed by international bodies, guiding clinicians through the maze of possibilities of glucose-lowering agents, but guidance is mostly based on evidence from the original UKPDS study, reported in the middle of the 1990's [87]. Evidence on the impact of glucose-lowering agents on the hardest endpoint, survival, is limited. In particular, sulfonylurea and insulin have been associated with higher mortality risks in cross-sectional studies or population studies [86, 163, 52, 104, 141, 190] with criticisms arising that comparing the mortality risk in these individuals to the global population is unfair as the profile of this population may be different, predisposing them to a higher mortality risk. On the other hand, many studies report a lower mortality risk in type 2 diabetes patients treated with metformin [86, 163, 52, 104, 141, 190, 13] but again, the profile of these people may be different by itself, thus influencing risk. Finally, in the high cardiovascular risk disease that is type 2 diabetes, use of statins has been debated frequently, with doubts being cast over the usefulness of these drugs in this population, in particular in the young or very old age groups.

This study investigated the survival of patients starting therapies involving various glucose-lowering agents (GLAs) compared to fully matched control subjects. We particularly analyzed the effect of age and concomitant use of statins.

This study was performed in collaboration with the largest mutual health insurance fund in Belgium (National Alliance of Christian Mutualities - NACM), which has access to a large database containing health expenditure records of 4.4 million people throughout the country. The Belgian health care insurance is a broad solidarity-based form of social insurance. Mutual health insurers like NACM are the legally-appointed bodies for managing and providing the Belgian compulsory health care and disability insurance. To implement its operations, NACM disposes of a large database containing health expenditure records of all its members. These records hold all financial reimbursements of drugs, procedures and contacts with health care professionals. Long-term follow-up and full matching of the people using GLAs to people identical in age, gender, concomitant medications and start of follow-up are possible. We performed a 5-year survival analysis to assess the excess mortality in patient cohorts defined by their GLA therapy compared to references without GLA therapy but with otherwise similar observable characteristics. Our analysis shows differences in 5-year survival in individuals treated with GLAs, at least partly driven by the risk profile of the individuals themselves.

2.2 Research design and methods

This study is based on records of the NACM, the largest Belgian mutual health insurer with over 4.4 million members (market shares of over 40% and 60% in Belgium and Flanders, respectively). All data extractions and analyses were performed at the Medical Management Department of the NACM under supervision of the Chief Medical Officer. NACM disposes of a longitudinal overview of its members' medical resource use, embedded in health expenditure records. Only 2% of the subpopulation under study left the NACM to switch to another mutual health insurer, emigration or employment by a foreign employer during the 5-year follow-up period, leading to a retention rate of 98% in our study. Patients that joined NACM after December 1999 were excluded from all analyses to minimize the chance of missing glucose-lowering therapy and/or cardiovascular events prior to starting follow-up.

Medication records were mapped onto the fifth level of the anatomical therapeutic chemical (ATC) classification system via the main chemical substances associated with each drug. The ATC system classifies drugs based on the targeted organ or system and their therapeutic and chemical characteristics [147]. Patients were partitioned into treatment groups based on ATC codes listed in their individual histories. Exact definitions of all pharmacological groups can be consulted in Table 2.1. In addition to pharmacotherapy, we considered a set of cardiovascular events prior to follow-up, which were identified via a combination of medicinal and surgical interventions (also described in Table 2.1). Based on usual prescription behavior in Belgium, exposure of oral glucose lowering drugs was assumed to be uninterrupted between the dates of the first record and up to six months after the final record in the insurance database.

2.2.1 Study cohort selection

The selection process is illustrated in Figure 2.1. 115,896 patients over 18 years old in whom glucose-lowering therapy was prescribed between 1st of January 2003 and 31st of December 2007 were eligible for the study. Eligible patients were assigned to study cohorts based on their glucose-lowering pharmacotherapy: more specifically metformin (MET), sulfonylurea (SU) and insulin (INS). Every combination of these three drug types defines a study cohort. Patients on DPP4 inhibitors or GLP-1 receptor agonists were not included as these were only introduced in Belgium around 2008.

Follow-up started on the first day of therapy intake, based on the patient's purchase of the prescribed agent(s). In each study and control group, subjects

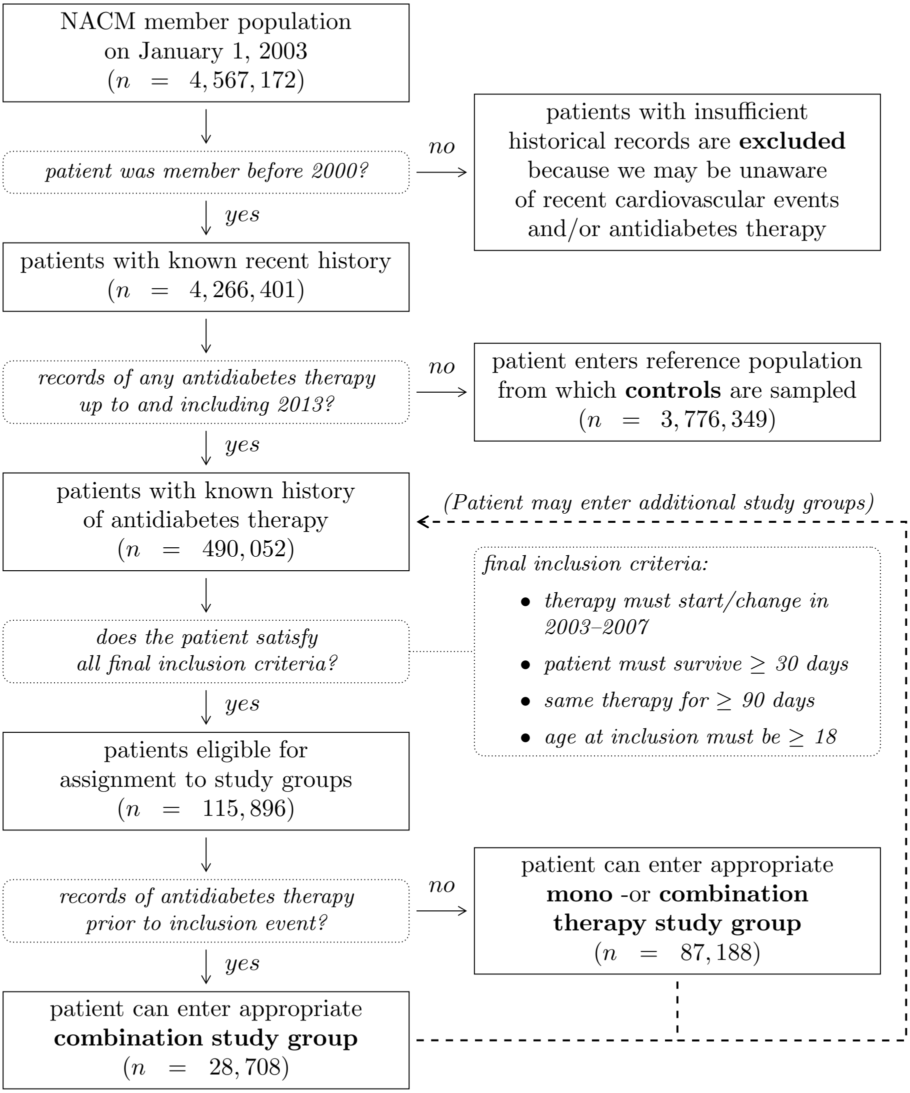

NACM member population
on January 1, 2003
($n = 4,567,172$)
patient was member before 2000?
no
patients with insufficient historical records are **excluded** because we may be unaware of recent cardiovascular events and/or antidiabetes therapy
yes
patients with known recent history
($n = 4,266,401$)
records of any antidiabetes therapy up to and including 2013?
no
patient enters reference population from which **controls** are sampled
($n = 3,776,349$)
yes
patients with known history of antidiabetes therapy
($n = 490,052$)
(Patient may enter additional study groups)
final inclusion criteria:
therapy must start/change in 2003–2007
patient must survive $\geq$ 30 days
same therapy for $\geq$ 90 days
age at inclusion must be $\geq$ 18
does the patient satisfy all final inclusion criteria?
yes
patients eligible for assignment to study groups
($n = 115,896$)
records of antidiabetes therapy prior to inclusion event?
no
patient can enter appropriate **mono** -or **combination therapy study group**
($n = 87,188$)
yes
patient can enter appropriate **combination study group**
($n = 28,708$)

Figure 2.1: Flowchart describing the selection protocol for study and control patients. Patients can move from the bottom right (monotherapy) to the bottom left group (combination therapy), but not vice versa. All listed counts are for unique patients.

Category	Definition
metformin	ATC codes: A10BA02, A10BD02, A10BD07, A10BD08
sulfonylurea	ATC codes: A10BB01, A10BB08, A10BB09, A10BB12, A10BD02
insulin	ATC codes: A10AB02, A10AB03, A10AB04, A10AB05, A10AB06, A10AB30, A10AC01, A10AC02, A10AC03, A10AC04, A10AC30, A10AD01, A10AD02, A10AD03, A10AD04, A10AD05, A10AD30, A10AE01, A10AE02, A10AE03, A10AE04, A10AE05, A10AE30, A10AF01
statins	ATC codes: all codes under C10AA
antihypertensives	ATC codes: all codes under C02, C03, C07, C08, C09
blood platelet aggregation inhibitors	ATC codes: all codes under B10AC
cardiovascular events	An arterial thrombotic event of the coronary, carotid, vertebral, aortic, iliac or lower extremity arteries necessitating an intervention for revascularization. Open and/or percutaneous interventions included thrombectomy, embolectomy, endarterectomy, artery bypass, vascular endoprosthese, endovascular dilatation, stenting or brinolysis. A positive history of a cardiovascular event was defined as having one or more of those events reimbursed by the NACM based on the Belgian nomenclature of health care provisions.

Table 2.1: Definitions of drug categories and cardiovascular events.

were followed until death or censoring over a maximum period of 5 years since inclusion. For control subjects, the start of follow-up was determined at random within the year of inclusion of the associated study patient to avoid bias related to the time of entry into the study.

Monotherapy study cohorts denote the first glucose-lowering therapy consisting of a single type of GLA, given to a patient without prior use of other GLAs (n=74,938), based on historical records from 1990 onwards. Patients are excluded from monotherapy cohorts if they transition to combination therapy within three months. Patients who started a combination therapy during the selection interval for at least 3 months (or until death) were included in the associated study cohorts, regardless of potential prior glucose-lowering therapy (n=47,149).

Patients could successively enter multiple study cohorts and be included in multiple cohorts during follow-up. For instance, a patient without prior GLA therapy who started metformin in 2003 and added sulfonylurea in 2005 is included in both the metformin monotherapy and the metformin and sulfonylurea combination therapy cohorts (5-year follow-up starting in 2003 and 2005, respectively), with some period of overlap (2005-2008).

Only patients with at least one month between the first and last purchase of associated GLAs were included in study cohorts, inducing an immortal time of one month. We accounted for potential bias by consistently matching control patients who survived for at least one month [162, 177]. Patients who started treatment and died during a single hospital admission were excluded from the analysis.

2.2.2 Control cohort selection

We compared study groups to controls with similar observable characteristics. Controls were sampled without replacement from the NACM population with matched characteristics to the study cohort, but without records of GLA therapy up to and including 2013. Unless stated otherwise, the control groups contained 5 subjects per subject in the study cohort, matched exactly on age at the start of follow-up, gender, cardiovascular history (had event/no event before the start of the follow-up), associated therapy (use of statins, antiplatelet and antihypertensive drugs) and the year of start of follow-up. Matching based on associated therapy was performed dichotomously (subject has/has not received the therapy for more than half of the individual's effective follow-up period).

2.2.3 Therapy changes within cohorts

The majority of patients remained on the same GLA therapy during the entire follow-up (Table 2.2.4). 15.4 to 28.5% of patients starting on monotherapy moved to a combination therapy by the end of the follow-up. Patients on combination therapy at start were still on the same regimen in 47.7 to 66.5% of cases: changes were often due to stopping of sulfonylurea (9 to 20%) or eliminating metformin from combination regimens that include insulin (15.2 to 18.7%).

2.2.4 Censoring

As we were primarily interested in prognoses for patients starting a certain therapy, no censoring was done based on therapy changes (such as adding additional GLAs) or poor compliance. Censoring based on therapy changes

	mono therapy			combination therapy			
→ end regimen	metformin	sulfonylurea	insulin	metf+sulf	metf+ins	sulf+ins	metf+sulf+ins
metformin	81.8%	2.0%	0.8%	10.9%	2.9%	0.3%	1.3%
sulfonylurea	5.2%	63.9%	2.4%	20.7%	1.3%	3.9%	2.7%
insulin	2.2%	0.9%	86.2%	0.7%	5.8%	2.6%	1.6%
metf+sulf	9.2%	4.5%	3.2%	66.5%	5.5%	1.7%	9.3%
metf+insulin	9.5%	0.5%	15.2%	1.9%	66.1%	1.3%	5.5%
sulf+insulin	0.9%	11.2%	20.1%	2.9%	3.2%	51.6%	10.2%
metf+sulf+insulin	2.3%	1.7%	11.7%	9.1%	20.6%	7.0%	47.7%

Table 2.2: Partitioning of final treatment regimen of patients starting a specific therapy (rows). The final treatment regimen (columns) is based on the last 9 months of individual followup. Patients are censored in all survival analyses after 9 consecutive months of renunciation from metformin, sulfonylurea and insulin.

would be informative and hence bias the survival estimates of interest. Patients that discontinued all GLA therapy for nine consecutive months are right censored, as this was considered to indicate that the patient was not using GLAs to manage glucose levels (e.g. using metformin for weight loss). Right censoring also occurred when subjects left the health insurer (lost to follow-up), which was rare (less than 2% of all patients in follow-up in each cohort). Switching health insurer was considered unrelated to a patient's medical condition and can therefore be considered non-informative.

2.2.5 Statistical analysis

Empirical survival curves were obtained using the Kaplan-Meier estimator. The associated 95% CIs were computed using the exponential Greenwood formula [109]. We used Cox proportional hazards (PH) models to quantify excess mortality between study and control cohorts while controlling for all observable patient characteristics. Adjusting for concomitant medication was particularly important, as controls were only matched in a binary fashion. Unless mentioned otherwise, the PH models contained the following set of predictors: continuous covariates describing age at start of follow-up and associated therapy (specifically statins, antiplatelet and antihypertensive drugs) and dichotomous factors for gender and the group a subject belonged to (study or control). Associated therapy-related predictors quantify the fraction of the subject's effective follow-up time during which he/she was exposed to the agent. Finally, an interaction term between age and gender is consistently included. The PH assumption was assessed via the Grambsch-Therneau test on scaled Schoenfeld residuals from the PH models [81]. The proportionality assumption was tested for each reported hazard ratio at the 1% significance level and rejections are indicated in all tables.

Software Statistical analyses were conducted in R using the survival package [185, 186]. Statistical plots were made in R using the ggplot2 package [203].

2.3 Results

2.3.1 Baseline cohort characteristics

An overview of the study cohorts and their baseline characteristics is given in Table 2.3.3. The study group with the youngest patient population was the group on insulin monotherapy without CV history ($p < 0.001$ compared to all other groups), followed by patients on metformin monotherapy without CV history ($p < 0.001$ compared to all remaining groups). The oldest patients were those who received sulfonylurea regardless of CV history ($p < 0.001$ compared to all other groups). Patients with a history of CV disease were consistently older than others ($p < 0.001$ in all pair-wise comparisons to groups without CV history) except in the sulfonylurea-insulin combination group. Patients without insulin in their GLA therapy were less likely to have a history of CV disease (less than 9% percent of the total group) than patients with insulin on board (more than 20% percent of total group) ($p < 0.001$).

The percentage of males and intake of associated therapies (statins, antiplatelet and antihypertensive therapies) were consistently higher in the patients with a history of CV disease than in those without, irrespective of the glucose lowering therapy ($p < 0.001$ for all groups). The majority of patients with a CV history were taking statins for over half the follow-up period, ranging from 58% in the SU + INS group to 79% in the metformin monotherapy group. In contrast, only a minority of patients without CV history were taking statins: ranging from 23% in the insulin monotherapy group to 47% in the MET + INS group.

2.3.2 Five-year survival in individuals on different glucose lowering agents

Compared to their associated matched controls, patients on metformin monotherapy showed no significant excess mortality during the follow-up. In contrast, patients started on SU, and certainly on insulin, did much worse than their respective controls (Figure 2.2). The excess mortality was highest in patients starting on insulin (23.8%), followed by SU (4.1%) and finally metformin (0.3%, though not statistically significant at the 5% significance level). Patients who started with bitherapy (MET + SU or MET + INS) or tritherapy (MET +SU +INS) also exhibited reduced 5-year survival compared to matched controls, with the highest difference in survival (12.9 and 15.6%) when insulin was part of the regimen from the start of follow-up.

Comparable differences were seen in survival of patients without a history of cardiovascular (CV) events, with the lowest survival rates in therapies involving both insulin and SU (up to 29% difference after 5 years) (Table 2.3). Patients with a history of CV events consistently exhibited lower survival than patients without a CV history, but excess mortality compared to matched controls was comparable for both subgroups. Of note, the survival of patients with a CV history on metformin monotherapy was not significantly different from the survival of the associated controls. The observed survival benefit of metformin monotherapy disappeared in combination therapy cohorts (Table 2.3).

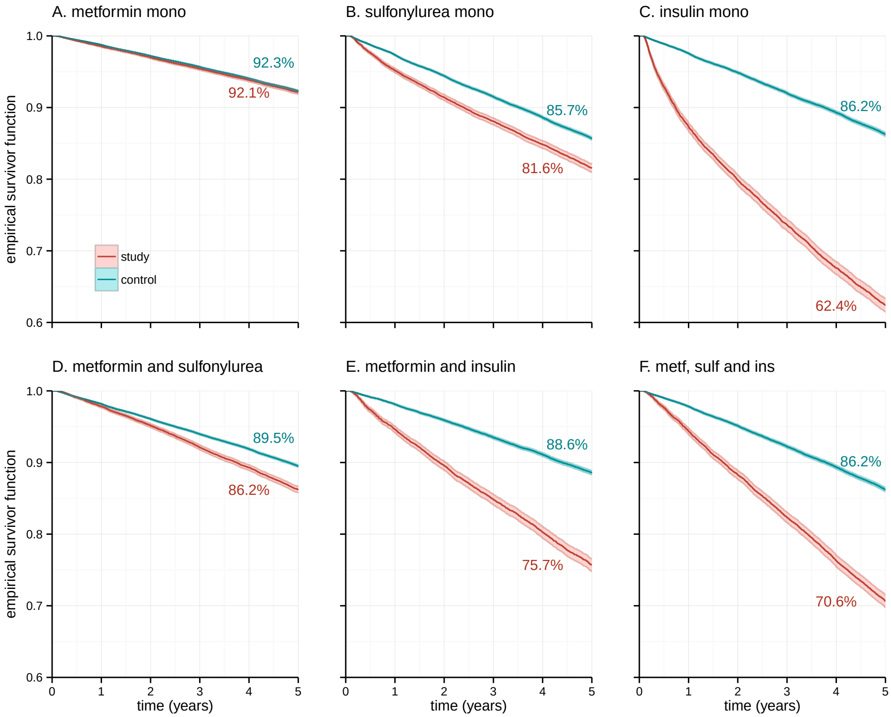


Figure 2.2: 5-year survival for increasing age per cohort.

	no history of cardiovasular disease		
	5-year survival (%)		hazard ratio
study cohort	study cohort	control	$\frac{study}{control}$
metformin	92.6 [92.3–92.9]	92.9 [92.8–93.0]	1.07 [1.02–1.11]
sulfonylurea	82.5 [81.9–83.1]	86.5 [86.3–86.8]	1.45 [1.40–1.52] ●
insulin	63.9 [62.9–64.9]	88.1 [87.8–88.3]	4.32 [4.14–4.51] ●
metf+sulf	87.0 [86.5–87.4]	90.3 [90.1–90.5]	1.40 [1.35–1.46]
metf+insulin	77.1 [76.1–78.0]	89.8 [89.5–90.1]	2.71 [2.56–2.87] ●
sulf+insulin	50.3 [48.8–51.7]	78.4 [77.8–78.9]	3.07 [2.92–3.23] ●
metf+sulf+insulin	71.5 [70.5–72.5]	87.5 [87.2–87.8]	2.71 [2.58–2.85]

	history of cardiovasular disease		
	5-year survival (%)		hazard ratio
study cohort	study cohort	control	$\frac{study}{control}$
metformin	86.7 [85.4–87.8]	85.2 [84.6–85.7]	0.92 [0.83–1.02]
sulfonylurea	72.5 [70.2–74.6]	77.2 [76.4–78.1]	1.35 [1.22–1.50]
insulin	56.1 [53.9–58.3]	78.6 [77.9–79.3]	2.69 [2.49–2.90]
metf+sulf	79.1 [77.4–80.7]	81.8 [81.1–82.5]	1.18 [1.07–1.30] ●
metf+insulin	69.2 [66.9–71.4]	82.7 [81.8–83.5]	2.07 [1.87–2.30]
sulf+insulin	48.8 [46.2–51.3]	74.3 [73.3–75.2]	2.66 [2.44–2.89]
metf+sulf+insulin	67.4 [65.5–69.3]	81.5 [80.8–82.2]	2.05 [1.89–2.23]

Table 2.3: Overview of survival for the study cohorts compared to a fully matched control cohort, stratified by cv history. The control group is sampled from the general population and matched for age, gender and use of statins, antihypertensives and antiplatelet drugs. For every patient in the study cohorts, 5 patients with completely matching profiles were used in control. ● indicates that the proportional hazards assumption was rejected for the associated hazards ratio ($p < 0.01$).

			gender	associated therapy		
	subjects	age	female	statins	antiplatelet	antihypertensive
study cohort	n	$mean \pm SD$	n (%)	n (%)	n (%)	n (%)
metformin	42,900	62.0 ± 12.3	21,759 (51)	19,747 (46)	7,725 (18)	33,785 (79)
no cv history	39,578	61.6 ± 12.4	20,913 (53)	17,127 (43)	5,579 (14)	30,592 (77)
cv history	3,322	66.8 ± 10.3	846 (25)	2,620 (79)	2,146 (65)	3,193 (96)
sulfonylurea	19,231	68.4 ± 12.6	10,100 (53)	7,479 (39)	3,825 (20)	15,507 (81)
no cv history	17,438	68.0 ± 12.8	9,576 (55)	6,325 (36)	2,739 (16)	13,786 (79)
cv history	1,793	71.8 ± 9.7	524 (29)	1,154 (64)	1,086 (61)	1,721 (96)
insulin	12,807	62.8 ± 17.8	5,818 (45)	3,842 (30)	4,270 (33)	10,214 (80)
no cv history	10,372	61.0 ± 18.7	5,125 (49)	2,395 (23)	2,410 (23)	7,827 (75)
cv history	2,435	70.6 ± 10.2	693 (28)	1,447 (59)	1,860 (76)	2,387 (98)
metf+sulf	25,218	65.8 ± 12.0	12,632 (50)	11,718 (46)	5,521 (22)	20,913 (83)
no cv history	22,830	65.4 ± 12.1	11,966 (52)	9,973 (44)	4,038 (18)	18,612 (82)
cv history	23,88	69.6 ± 9.4	666 (28)	1,745 (73)	1,483 (62)	2,301 (96)
metf+insulin	9,506	64.8 ± 13.9	4,880 (51)	4,891 (51)	3,562 (37)	8,305 (87)
no cv history	7,874	64.0 ± 14.5	4,330 (55)	3,716 (47)	2,333 (30)	6,710 (85)
cv history	1,632	68.7 ± 10.1	550 (34)	1,175 (72)	1,229 (75)	1,595 (98)
sulf+insulin	6,087	74.1 ± 11.0	3,201 (53)	2,285 (38)	2,730 (45)	5,580 (92)
no cv history	4,580	74.1 ± 11.6	2,639 (58)	1,415 (31)	1,584 (35)	4,108 (90)
cv history	1,507	74.1 ± 8.8	562 (37)	870 (58)	1,146 (76)	1,472 (98)
metf+sulf+insulin	10,653	69.1 ± 11.4	5,570 (52)	5,405 (51)	4,680 (44)	9,746 (91)
no cv history	8,380	68.7 ± 12.0	4,800 (57)	3,827 (46)	2,933 (35)	7,520 (90)
cv history	2,273	70.5 ± 9.1	770 (34)	1,578 (69)	1,747 (77)	2,226 (98)

Table 2.4: Baseline characteristics of the study cohorts. Individuals starting starting mono therapy are selected such that they have no prior history of diabetes-related drugs. Individuals starting combination therapy may have a prior history of diabetes-related drugs. All differences in use of associated therapy are statistically significant between study cohorts, except for use of antihypertensives in insulin and sulfonylurea mono cohorts (no significant difference). All associated therapy use is statistically significantly elevated in subgroups with prior cardiovascular events within each study cohort. All comparisons of study group characteristics use significance level $\alpha = 0.05$ and are computed using Tukey's test in conjunction with ANOVA to adjusts for multiple comparison.

2.3.3 Age-dependent 5-year survival of individuals on different glucose lowering agents

Figure 2.3.4 illustrates the 5-year survival of patients as a function of age at the start of follow-up. Compared to the general population, 5-year survival was lower at any age in all cohorts on glucose lowering monotherapy except the metformin monotherapy cohort, which exhibits comparable survival to the general population. At any certain age, survival was highest in patients on metformin, worse in patients on sulfonylurea, and worst in patients on insulin. In patients starting on combination therapy, survival was also lower at any age than associated controls. Again, if the regimen contains insulin, survival is worse at any age category, with or without sulfonylurea on board.

The differences in survival at any age were slightly reduced when comparing to fully matched controls, though they remain large and statistically significant (illustrations are given in Figure 2.3.4). This reduction in excess mortality appears to be mainly attributable to the fact that the fully matched control groups have a higher frequency of prior cardiovascular events than the unmatched general population.

Patients starting metformin monotherapy at a very young age (between 18 and 40 years; n=1,446; 83.3% male) had a 5-year survival rate of 99.2% [98.4%–99.6%] compared to 99.3% [99.1%–99.5%] for fully matched controls (p=0.644). Of note, all females in this study group (n = 242) survived the entire follow-up. In the age category 18 to 40 years, the 5-year survival rate of patients on insulin monotherapy (n = 1,873) was reduced compared to fully matched controls ($p < 0.001$), with survival rates of 94.7% [93.4%–95.6%] and 99.5% [99.3%–99.6%] respectively.

2.3.4 Statins and survival in individuals on different glucose lowering therapy

Survival was compared between patients with and without statins. Survival was consistently higher for patients that used statins in conjunction with GLA therapy (Table 2.5), irrespective of CV history. The observed mortality rate when using statins along with GLAs was 57 to 64% lower in patients without a history of CV disease and by 50 to 68% in patients with a CV history, compared to patients using only GLAs.

In a second analysis, cohorts in which all patients on glucose lowering therapy were taking statins were compared to the general age and gender matched population. Patients on metformin monotherapy (with and without a CV

	no history of cardiovasular disease		
	5-year survival (%)		hazard ratio
study cohort	without statins	with statins	$\frac{\mathit{statins}}{\mathit{no\ statins}}$
metformin	90.2 [89.8–90.6]	95.5 [95.2–95.9]	0.43 [0.39–0.47] •
sulfonylurea	76.9 [76.1–77.8]	91.7 [91.0–92.4]	0.36 [0.32–0.40] •
insulin	59.6 [58.4–60.8]	77.3 [75.4–79.0]	0.37 [0.33–0.41] •
metf+sulf	82.5 [81.8–83.2]	92.6 [92.0–93.1]	0.42 [0.38–0.46] •
metf+insulin	68.2 [66.7–69.6]	86.8 [85.7–87.9]	0.39 [0.35–0.44]
sulf+insulin	41.3 [39.6–43.1]	69.9 [67.4–72.3]	0.43 [0.38–0.48]
metf+sulf+insulin	61.5 [60.0–62.9]	83.4 [82.1–84.5]	0.43 [0.39–0.48] •

	history of cardiovasular disease		
	5-year survival (%)		hazard ratio
study cohort	without statins	with statins	$\frac{\mathit{statins}}{\mathit{no\ statins}}$
metformin	71.3 [67.5–74.6]	90.6 [89.4–91.7]	0.36 [0.29–0.45]
sulfonylurea	53.1 [48.8–57.2]	82.5 [80.1–84.6]	0.32 [0.26–0.40]
insulin	40.5 [37.0–43.9]	66.5 [63.7–69.1]	0.45 [0.39–0.53]
metf+sulf	62.8 [58.8–66.5]	85.0 [83.2–86.6]	0.42 [0.34–0.51] •
metf+insulin	46.7 [42.0–51.2]	77.9 [75.4–80.2]	0.40 [0.33–0.49]
sulf+insulin	32.3 [28.6–36.0]	60.8 [57.4–63.9]	0.50 [0.42–0.59]
metf+sulf+insulin	46.6 [42.8–50.3]	76.6 [74.4–78.6]	0.42 [0.35–0.49]

Table 2.5: Analysis of the effect of statins within each study cohort. Presented hazard ratios are associated to the fraction of follow-up on statins. Patients are classified as statin users if they were on statins for at least half the follow-up. The proportional hazards models used here control for age, gender, use of antihypertensive and antiplatelet drugs and an age-gender interaction. • indicates that the proportional hazards assumption was rejected for the associated hazards ratio ($p < 0.01$).

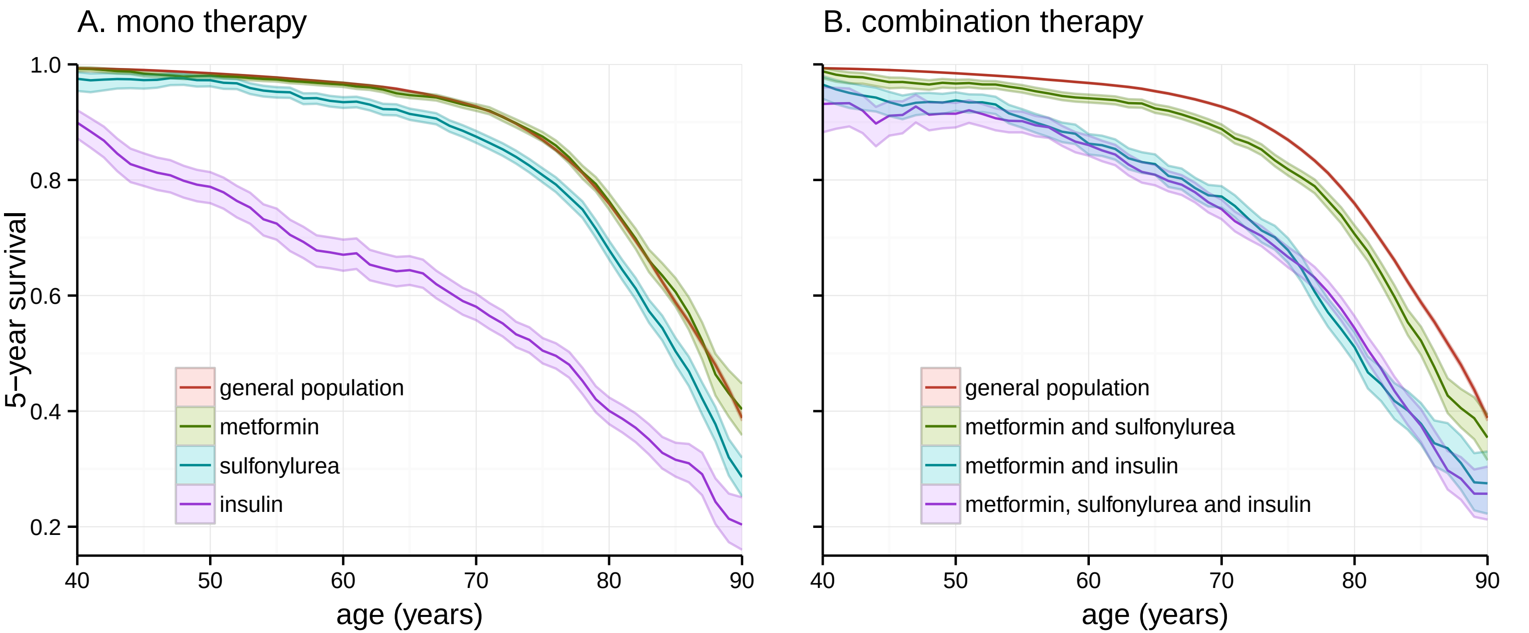


Figure 2.3: 5-year survival for increasing age per cohort.

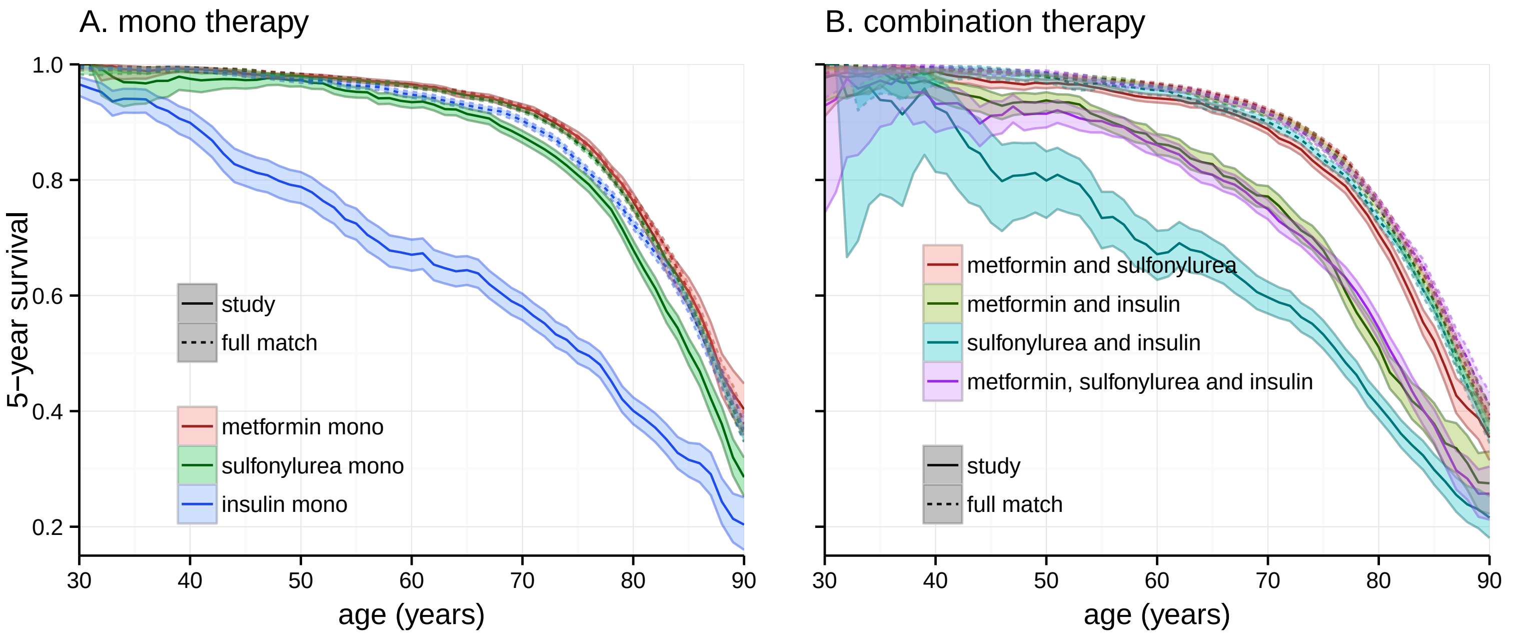


Figure 2.4: 5-year survival for increasing age per cohort with matched controls.

history) and sulfonylurea mono (without CV history) that were also taking statins exhibited higher survival rates than age and gender matched control groups without glucose lowering therapy (of which resp. only 32,4%, 24.7%, and 29.9% were taking statins during the majority of the follow-up). Patients on the combination of MET and SU and statins had the same survival rate than their controls, irrespective of CV history (Table 2.6).

study cohort	no history of CVD	history of CVD
metformin	0.66 [0.61–0.71] •	0.67 [0.58–0.78]
sulfonylurea	0.88 [0.80–0.97]	0.95 [0.81–1.12]
insulin	2.84 [2.53–3.19] •	2.12 [1.83–2.47]
metf+sulf	0.91 [0.84–0.99] •	0.96 [0.84–1.11] •
metf+insulin	1.91 [1.71–2.14]	1.67 [1.37–2.03]
sulf+insulin	2.40 [2.13–2.71]	2.43 [2.07–2.87]
metf+sulf+insulin	1.83 [1.66–2.01]	1.44 [1.23–1.69]

Table 2.6: Hazard ratios between statin users in the study group a control group matched for age and gender. The remaining characteristics of the control groups used here follow the distribution of the total NACM member population (after matching for age and gender). The proportional hazard models used to compute these hazard ratios control for age, gender, the use of antihypertensive and antiplatelet drugs and an age-gender interaction. • indicates that the proportional hazards assumption was rejected for the associated hazards ratio ($p < 0.01$).

2.4 Conclusions

The main objective of this large controlled cohort study was to investigate the survival of patients on various glucose lowering therapies in comparison to a reference population with similar observable characteristics. It was found that 5-year survival rates vary between glucose lowering therapies, at least partly driven by the risk profile of the individuals themselves, and substantially influenced by the intake of statins and the age at the start of GLA therapy.

Increased 5-year mortality rates were observed in patients on GLAs compared to matched references not on GLAs. This confirms the study of Bannister et al. [13] showing an increased mortality in patients on SU monotherapy and extends the evidence to other groups on insulin monotherapy and different combination therapies. Although we did not see a better survival rate in patients on metformin monotherapy, our data show that these patients have similar survival rates compared to matched controls, especially if a positive history of CV disease is present.

Our study confirms data from many other observational studies that patients on metformin monotherapy have a lower mortality risk than patients on other glucose lowering therapy [86, 163, 52, 104, 141, 190], characterized by reduced excess mortality compared to matched controls. This study does determine whether this excess mortality of patients on SU and insulin is mainly caused by the vulnerability of the background population or by negative properties of the therapies themselves. The extra mortality risk can, at least partially, be explained by the risk profile of the individuals themselves and not by using SU or insulin per se. First of all, this might reflect the progressive nature of type 2 diabetes such that patients with less pronounced hyperglycemia are started on metformin monotherapy whereas uncontrolled patients are started on insulin or combination therapies including SU. Age is another important independent predictor of mortality and can explain why younger patient groups (i.e. metformin mono) have better survival rates than older patient groups (i.e. SU mono). Age however does not explain the lower survival rates in younger patients on insulin monotherapy and survival differences throughout all age categories. A positive history of cardiovascular events also increases the background risk of our populations and explains the lower survival rates in any study cohort with a positive CV history. As in Morgan et al. [141], a combination of several other elements will probably play a role such as hypertension and factors that were not available in our study such as presence of chronic kidney disease and albuminuria, the level of glycemic control, smoking and heart failure.

Data from literature are conflicting concerning differences between agents. On the one hand several studies report no difference in survival when comparing metformin with SU [8, 97, 107] or SU with insulin [97]. Also in the ORIGIN trial, insulin glargine was not associated with higher mortality rates than controls [78] despite the lower use of metformin in the insulin glargine group. On the other hand, many clinical and observational studies have indicated an increased mortality risk associated with the use of SU and insulin compared to metformin [86, 163, 52, 104, 141, 190], although differences were shown to depend on the type of SU [170, 201], and the dose of insulin [76] used. In fact only a well-controlled RCT with sufficient power comparing different treatment strategies might answer this question, but it is very unlikely that these RCT's will ever be undertaken. Observational studies are in that view considered complementary as they do not omit patients on the basis of strict criteria and will usually have enough follow-up time to evaluate hard endpoints such as mortality risk.

This study is the first to show a beneficial impact of intake of statins on real-life survival data in a large population study of patients on glucose lowering therapy.

This is not unexpected, since available evidence from RCT's convincingly showed beneficial effect of statins on survival and prevention of cardiovascular events in secondary prevention (reviewed in [19]). Data in primary prevention are scarce with the only RCT in diabetic patients lacking power to show an overall mortality benefit [48]. Our trial shows a beneficial effect of statins in patients with diabetes, both in primary and secondary prevention, with mortality risks being 60 to 80% lower independent of the type of glucose lowering therapy or presence of a CV disease history. Of note, patients taking statins in combination with metformin or SU monotherapy even showed better survival than the general population.

An asset of this study was the use of health expenditure records to assess the survival of patients on various glucose lowering therapies in comparison with a similar reference cohort from the general population. Claims records constitute a valuable source of information for observational epidemiological studies by embedding long-term longitudinal medical information of a large number of patients. Additionally, claims records aggregate proxies of medical information from various caregivers into a complete patient-wide overview which is often unavailable to individual caregivers and other medical stakeholders.

Through exact pair-wise matching of the reference cohort and regression adjustment in the proportional hazards models we were able to exclude important observable confounders in comparisons of the study cohorts with their respective references [165]. Having access to a large population from which to sample control subjects allowed us to find references with exact matches on key confounding variables. Matching on these observable factors excludes the confounding effect and yields an efficiency gain [118]. Some residual confounding resulting from uncontrolled and unobservable factors may remain. Our study also has limitations. While there are considerable benefits in using claims data for epidemiological research, the absence of detailed clinical parameters prohibits causal inference because we could not control for level of glycemic control (e.g., fasting blood glucose or HbA1c), BMI, or other modifiable cardiovascular risk factors (e.g., smoking). However, we controlled for age, sex, concomitant medication, and presence of history of CV disease. Due to its observational nature our study remains susceptible to confounding by indication [85, 158, 25]. Therefore our study is not suitable to compare GLA therapies directly as the patients' underlying conditions yield indications for their treatment, preventing comparisons [85, 7].

We conclude that 5-year survival in patients on glucose lowering therapy is lower than in matched controls except for metformin monotherapy. Intake of metformin is associated with lowest 5-year mortality. In all groups, the intake of statins was associated with a reduced mortality rate.

Chapter 3

EnsembleSVM: A Library for Ensemble Learning Using Support Vector Machines

This chapter has been previously published as:
Claesen, M., De Smet, F., Suykens, J. A. K., & De Moor, B. (2014). **EnsembleSVM: A library for ensemble learning using support vector machines**. *Journal of Machine Learning Research*, 15(1), 141–145.

Abstract

EnsembleSVM is a free software package containing efficient routines to perform ensemble learning with support vector machine (SVM) base models. It currently offers ensemble methods based on binary SVM models. Our implementation avoids duplicate storage and evaluation of support vectors which are shared between constituent models. Experimental results show that using ensemble approaches can drastically reduce training complexity while maintaining high predictive accuracy. The EnsembleSVM software package is freely available online at http://esat.kuleuven.be/sista/ensemblesvm.

3.1 Introduction

Data sets are becoming increasingly large. Machine learning practitioners are confronted with problems where the main computational constraint is the amount of time available. Problems become particularly challenging when the training sets no longer fit into memory. Accurately solving the dual problem for SVM training with nonlinear kernels requires a run time which is at least quadratic in the size of the training set n, thus training complexity is $\Omega(n^2)$ [26, 126].

`EnsembleSVM` employs a divide-and-conquer strategy by aggregating many SVM models, trained on small subsamples of the training set. Through subdivision, total training time decreases significantly, even though more models need to be trained. For example, training p classifiers on subsamples of size n/p, results in an approximate complexity of $\Omega(n^2/p)$. This reduction in complexity helps in dealing with large data sets and nonlinear kernels.

Ensembles of SVM models have been used in various applications [199, 125, 140]. Collobert et al. [49] use ensembles for large scale learning and employ a neural network to aggregate base models. Valentini and Dietterich [192] provide an implementation which allows base models to use different kernels. For efficiency reasons, we require base models to share a single kernel function.

While other implementations mainly focus on improving predictive performance, our framework primarily aims to (i) make nonlinear large-scale learning feasible through complexity reductions and (ii) enable fast prototyping of novel ensemble algorithms.

3.2 Software Description

The `EnsembleSVM` software is freely available online under a LGPL license. `EnsembleSVM` provides ensembles of instance-weighted SVMs, as defined in Equation (3.1). The default approach we offer is bagging, which is commonly used to improve the performance of unstable classifiers [29]. In bagging, base models are trained on bootstrap subsamples of the training set and their predictions are aggregated through majority voting.

Base model flexibility is maximized by using instance-weighted binary support vector machine classifiers, as defined in Equation (3.1). This formulation lets users define misclassification penalties per training instance $C_i,\ i = 1, \ldots, n$ and encompasses popular approaches such as C-SVC and class-weighted SVM

[50, 148].

$$\min_{\mathbf{w},\xi,\rho} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} C_i\xi_i, \tag{3.1}$$

$$\text{subject to } y_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + \rho) \geq 1 - \xi_i, \qquad i = 1, \ldots, n$$

$$\xi_i \geq 0, \qquad i = 1, \ldots, n$$

When aggregating SVM models, the base models often share support vectors (SVs). The `EnsembleSVM` software intelligently caches distinct SVs to ensure that they are only stored and used for kernel evaluations once. As a result, `EnsembleSVM` models are smaller and faster in prediction than ensemble implementations based on wrappers.

3.2.1 Implementation

`EnsembleSVM` has been implemented in `C++` and makes heavy use of the standard library. The main implementation focus is training speed. We use facilities provided by the `C++11` standard and thus require a moderately recent compiler, such as `gcc` ≥ 4.7 or `clang` ≥ 3.2. A portable Makefile system based on GNU autotools is used to build `EnsembleSVM`.

`EnsembleSVM` interfaces with `LIBSVM` to train base models [35]. Our code must be linked to `LIBSVM` but does not depend on a specific version. This allows users to choose the desired version of the `LIBSVM` software in the back-end.

The `EnsembleSVM` programming framework is designed to facilitate prototyping of ensemble algorithms using SVM base models. We particularly provide extensive support to define novel aggregation schemes, should the available options be insufficient. Key components are extensively documented in the code and on a wiki, which serves as a high-level guideline.[1] Intuitive APIs are provided for convenient features such as thread pools, command line interface and deserialization to enable users to develop new tools efficiently.

[1] The `EnsembleSVM` development wiki is available at `https://github.com/claesenm/EnsembleSVM/wiki`.

The EnsembleSVM library was built with extensibility and user contributions in mind. Major API functions are well documented to lower the threshold for external development. The executable tools provided with EnsembleSVM are essentially wrappers for the library itself. The tools can be considered as use cases of the main API functions to help developers.

3.2.2 Tools

The main tools in this package are esvm-train and esvm-predict, used to train and predict with ensemble models. Both of these are pthread-parallelized. Additionally, the merge-models tool can be used to merge standard LIBSVM models into ensembles. Finally, esvm-edit provides facilities to modify the aggregation scheme used by an ensemble.

EnsembleSVM includes a variety of extra tools to facilitate basic operations such as stratified bootstrap sampling, cross-validation, replacing categorical features by dummy variables, splitting data sets and sparsifying standard data sets. We recommend retaining the original ratio of positives and negatives in the training set when subsampling.

3.3 Benchmark Results

To illustrate the potential of our software, EnsembleSVM 2.0 has been benchmarked with respect to LIBSVM 3.17. To keep the experiments simple, we use majority voting to aggregate predictions, even though more sophisticated methods are offered. For reference, we also list the best obtained accuracy with a linear model, trained using LIBLINEAR [71]. Linear methods are common in large-scale learning due to their speed, but may result in significantly decreased accuracy. This is why scalable nonlinear methods are desirable.

We used two binary classification problems, namely the covtype and ijcnn1 data sets.[2] Both data sets are balanced. Features were always scaled to $[0,1]$. We have used C-SVC as SVM and base models ($\forall\ i : C_i = C$). Reported numbers are averages of 5 test runs to ensure reproducibility. We used the RBF kernel, defined by the kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2}$. Optimal parameter selection was done through cross-validation.

The covtype data set is a common classification benchmark featuring 54 dimensions [22]. We randomly sampled balanced training and test sets of

[2] Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html and UCI.

100, 000 and 40, 000 instances respectively and classified class 2 versus all others. The ijcnn1 data set was used in a machine learning challenge during IJCNN 2001 [156]. It contains 35, 000 training instances in 22 dimensions.

data set	**test set accuracy**			**no. of SVs**		**time (s)**	
	LIBSVM	LIBLINEAR	ESVM	LIBSVM	ESVM	LIBSVM	ESVM
covtype	0.92	0.76	0.89	26516	50590	728	35
ijcnn1	0.98	0.92	0.98	3564	7026	9.5	0.3

Table 3.1: Summary of benchmark results per data set: test set accuracy, number of support vectors and training time. Accuracies are listed for a single LIBSVM model, LIBLINEAR model and an ensemble model.

Results in Table 3.1 show several interesting trends. Training EnsembleSVM models is orders of magnitude faster, because training SVMs on small subsets significantly reduces complexity. Subsampling induces smaller kernels per base model resulting in lower overall memory use.

Ensembles can end up with more support vectors than a single SVM. Due to our parallelized implementation, prediction with ensemble models was faster than with LIBSVM models in both experiments even though the ensembles comprise twice as many SVs.

The ensembles in these experiments are competitive with a traditional SVM even though we used simple majority voting. For covtype, ensemble accuracy is 3% lower than a single SVM and for ijcnn1 the ensemble is marginally better (0.2%). Linear SVM falls far short in terms of accuracy for both experiments, but is trained much faster ($<$ 2 seconds).

We obtained good results with very basic aggregation. Collobert et al. [49] illustrated that more sophisticated aggregation methods can improve the predictive performance of ensembles. Others have reported performance improvements over standard SVM for ensembles using majority voting [192, 199].

3.4 Conclusions

EnsembleSVM provides users with efficient tools to experiment with ensembles of SVMs. Experimental results show that training ensemble models is significantly faster than training standard LIBSVM models while maintaining competitive predictive accuracy.

Linear methods are frequently applied in large-scale learning, mainly due to their low training complexity. Linear methods are known to have competitive accuracy for high dimensional problems. As our benchmarks showed, the difference in accuracy may be large for low dimensional problems. As such, fast nonlinear methods remain desirable in large-scale learning, particularly for low dimensional tasks with many training instances. Our benchmarks illustrate the potential of the ensemble approaches offered by `EnsembleSVM`.

Ensemble performance may be improved by using more complex aggregation schemes. `EnsembleSVM` currently offers various aggregation schemes, both linear and nonlinear. Additionally, it facilitates fast prototyping of novel methods through its `Pipeline` framework.[3]

`EnsembleSVM` strives to provide high-quality, user-friendly tools and an intuitive programming framework for ensemble learning with SVM base models. The software will be kept up to date by incorporating promising new methods and ideas when they are presented in the literature. User requests and suggestions are welcome and appreciated.

[3] `https://github.com/claesenm/EnsembleSVM/wiki/Pipeline`

Chapter 4

SVM Ensemble Learning from Positive and Unlabeled Data

This chapter has been previously published as:
Claesen, M., De Smet, F., Suykens, J. A., & De Moor, B. (2015). **A robust ensemble approach to learn from positive and unlabeled data using SVM base models**. *Neurocomputing*, 160, 73-84.

Abstract

We present a novel approach to learn binary classifiers when only positive and unlabeled instances are available (PU learning). This problem is routinely cast as a supervised task with label noise in the negative set. We use an ensemble of SVM models trained on bootstrap resamples of the training data for increased robustness against label noise. The approach can be considered in a bagging framework which provides an intuitive explanation for its mechanics in a semi-supervised setting. We compared our method to state-of-the-art approaches in simulations using multiple public benchmark data sets. The included benchmark comprises three settings with increasing label noise: (i) fully supervised, (ii) PU learning and (iii) PU learning with false positives. Our approach shows a marginal improvement over existing methods in the second setting and a significant improvement in the third.

4.1 Introduction

Training binary classifiers on positive and unlabeled data is referred to as PU learning [128]. The absence of known negative training instances warrants appropriate learning methods. Inaccurate label information can be more problematic than attribute noise [213]. Specialised PU learning approaches are recommended when (i) negative labels cannot be acquired, (ii) the training data contains a large amount of false negatives or (iii) the positive set has many outliers.

Practical applications of PU learning typically feature large, imbalanced training sets with a small amount of labeled (positive) and a large amount of unlabeled training instances. The PU learning problem arises in various settings, including web page classification [210], intrusion detection [119] and bioinformatics tasks such as variant prioritization [179], gene prioritization [1, 139] and virtual screening of drug compounds [178].

Though these applications share a common underlying learning problem, the final evaluation criteria may be fundamentally different. For instance, in prioritization one wishes to obtain high precision since highly ranked targets may be subjected to further biological analysis. Intrusion detection, on the other hand, necessitates high recall to ensure that no anomalies go unnoticed.

Following Mordelet and Vert [138], we will use the term *contamination* to refer to the fraction of mislabeled instances in a given set. We will denote the positive and unlabeled training instances by $\mathcal{P}$ and $\mathcal{U}$, respectively. Contamination in $\mathcal{P}$ refers to false positives while contamination in $\mathcal{U}$ refers to the presence of positives in $\mathcal{P}$. Usually $\mathcal{U}$ contains mostly true negative instances (e.g. contamination below 0.5) and $\mathcal{P}$ is assumed to be uncontaminated.

The distributions of the positive and a contaminated unlabeled set overlap even when those of the positive and underlying negative sets do not, which makes classification more difficult compared to a traditional supervised setting. Elkan and Noto [67] and Blanchard et al. [23] report statistical approaches to estimate the contamination of the unlabeled set and additionally show that distinguishing positives from unlabeled instances is a valid proxy for distinguishing positives from negatives.

The assumption in PU learning that $\mathcal{P}$ is uncontaminated may be violated in applications due to various reasons [75]. Additionally, outliers in the positive set may have a similar effect on classification performance [153]. We propose a novel PU learning method that is less vulnerable to potential contamination in $\mathcal{P}$ called the robust ensemble of support vector machines (RESVM). RESVM is compared to other methods in a series of simulations based on several public data sets.

4.2 Related work

PU learning approaches can be split into two main conceptual categories: (i) approaches that account for the contamination of the unlabeled set explicitly by modeling the label noise and (ii) approaches that try to infer an uncontaminated (negative) subset $\hat{\mathcal{N}}$ from $\mathcal{U}$ and then train supervised algorithms to distinguish $\mathcal{P}$ from $\hat{\mathcal{N}}$. When *very* few labeled examples are available, the structure within the data is the main source of information which can be exploited by semi-supervised clustering techniques [5].

Accounting for the contamination of $\mathcal{U}$ in the modeling process This can be done by weighting individual data points, such as in weighted logistic regression [67, 121]. Another approach is by changing the penalties on misclassification during training, as is done in class-weighted SVM [128], bagging SVM [138] and RT-SVM [130].

Inferring an uncontaminated subset from $\mathcal{U}$ Another class of approaches tries to infer a negative set $\hat{\mathcal{N}}$ from $\mathcal{U}$. After the inferential step, binary classifiers are trained to distinguish $\mathcal{P}$ from $\hat{\mathcal{N}}$ in a supervised fashion. Examples of such two-step approaches include S-EM [129], mapping convergence (MC) [209] and ROC-SVM [122].

Class-weighted SVM and related approaches The approach we suggest belongs to the first class of methods and is closely related to class-weighted SVM and bagging SVM (which uses class-weighted SVM internally). We will discuss both of these approaches in more detail before moving on to the proposed method. We evaluated our method compared to both class-weighted SVM and bagging SVM.

4.2.1 Class-weighted SVM

Class-weighted SVM (CWSVM) is a supervised technique in which the penalty for misclassification differs per class. Liu et al. [128] first applied class-weighted SVM for PU learning by considering the unlabeled set to be negative with noise on its labels. CWSVM is trained to distinguish $\mathcal{P}$ from $\mathcal{U}$. During training, misclassification of positive instances is penalized more than misclassification of unlabeled instances to emphasize the higher degree of certainty on positive labels. In the context of PU learning, the optimization problem for training CWSVM can be written as:

$$\min_{\alpha,\xi,b} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) + C_{\mathcal{P}}\sum_{i\in\mathcal{P}}\xi_i + C_{\mathcal{U}}\sum_{i\in\mathcal{U}}\xi_i, \tag{4.1}$$

$$\text{s.t. } y_i(\sum_{j=1}^{N}\alpha_j y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) + \rho) \geq 1 - \xi_i, \qquad i = 1,\ldots,N,$$

$$\xi_i \geq 0, \qquad i = 1,\ldots,N,$$

with $\alpha \in \mathbb{R}^N$ the support values, $\mathbf{y} \in \{-1,+1\}^N$ the label vector, $\kappa(\cdot,\cdot)$ the kernel function, ρ the bias term and $\xi \in \mathbb{R}^N$ the slack variables. The misclassification penalties $C_{\mathcal{P}}$ and $C_{\mathcal{U}}$ require tuning (typically $C_{\mathcal{P}} > C_{\mathcal{U}}$ to emphasize known labels). SVM formulations with unequal penalties across classes have been used previously to tackle imbalanced data sets [148].

4.2.2 Bagging SVM

Mordelet and Vert introduce bagging SVM as a meta-algorithm which consists of aggregating classifiers trained to discriminate $\mathcal{P}$ from a small, random resample of $\mathcal{U}$ [138]. They posit that PU learning problems have a particular structure that leads to instability of classifiers, namely the sensitivity of classifiers to the contamination of the unlabeled set. Bagging is a common technique used to improve the performance of instable classifiers [29].

In bagging SVM, random resamples of $\mathcal{U}$ are drawn and CWSVM classifiers are trained to discriminate $\mathcal{P}$ from each resample. By resampling $\mathcal{U}$, the contamination is varied. This induces variability in the classifiers which the aggregation procedure can then exploit. The size of the bootstrap resample of $\mathcal{U}$ is a tuning parameter in bagging SVM. The ratio $C_{\mathcal{P}}/C_{\mathcal{U}}$ is fixed so that the following holds:

$$|\mathcal{P}| \times C_{\mathcal{P}} = n_{\mathcal{U}} \times C_{\mathcal{U}}, \tag{4.2}$$

with $|\mathcal{P}|$ the size of the positive set and $n_{\mathcal{U}}$ the size of resamples from the unlabeled set. This choice of weights is common in imbalanced settings [34, 53]. All base models in bagging SVM classify the full set of positives against a subset of unlabeled instances and use a high misclassification penalty on the positives similar to CWSVM.

4.3 Robust Ensemble of SVMs

We propose a new technique called the robust ensemble of SVMs (RESVM). RESVM is a bagging method using CWSVM base models as discussed in Section 8.4.3. Base model training sets are constructed by bootstrap resampling both $\mathcal{P}$ and $\mathcal{U}$ separately, both of which may be contaminated.

The key difference between RESVM and bagging SVM is that the former resamples $\mathcal{P}$ in addition to $\mathcal{U}$ to increase variability between base models. RESVM additionally features an extra degree of freedom to control the relative misclassification penalty between positive and unlabeled instances, which is fixed in bagging SVM. Mordelet and Vert [138] report no significant changes when varying the relative penalty in bagging SVM, though our experiments show that it is important in RESVM (see w_{pos} in Table 4.6).

Before elaborating on the details of RESVM, we briefly illustrate the effect of resampling contaminated sets. Subsequently we summarize the mechanisms of bagging and why they are advantageous when learning with label noise in the RESVM approach. Finally, we provide the full RESVM training approach and the way ensemble decision values are computed based on the base model decision values.

4.3.1 Bootstrap resampling contaminated sets

The RESVM approach resamples both $\mathcal{P}$ and $\mathcal{U}$, both of which are potentially contaminated. Resampling contaminated sets with replacement induces variability in contamination across the resampled sets (e.g. resamples of $\mathcal{U}$ and $\mathcal{P}$ that are used for training). The variability in contamination between resamples increases for increasing contamination of the original set. We assume contamination levels below 50%, e.g. less than half the instances in a given set are mislabeled. Due to the law of large numbers the contamination in bootstrap resamples of increasing size converges to the expected contamination, which equals that of the original set that is being resampled. As a result, the variability in contamination decreases for increasing resample size. Figure 4.1 illustrates this property empirically based on $20,000$ repeated measurements for each resample size: the expected value (mean) equals the original contamination, but the variability in resample contamination decreases for increasing resample size.

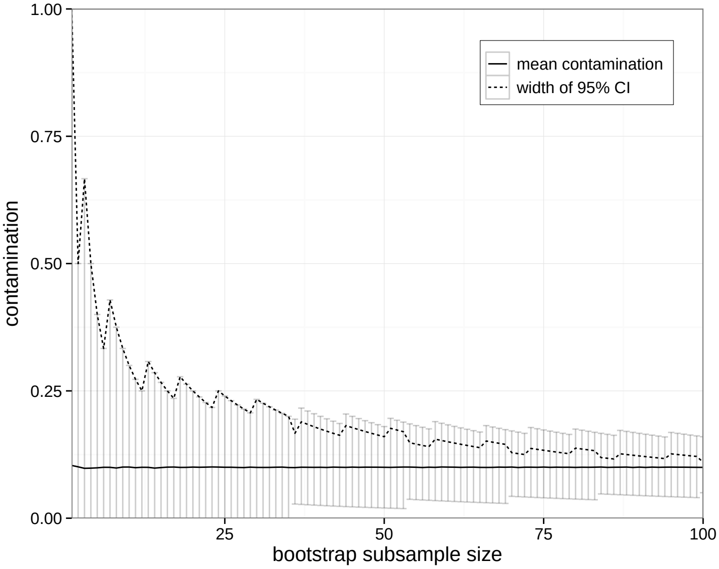


Figure 4.1: Contamination of bootstrap resamples for increasing size of resamples when the original sample has 10% contamination. Errorbars indicate the 95% confidence interval (CI) of contamination in resamples. The contamination varies greatly between small resamples as shown by the CIs.

4.3.2 Bagging predictors

Breiman [29] introduced bagging as a technique to construct strong ensembles by combining a set of base models and stated that “the essential problem in combining classifiers is growing a suitably diverse ensemble of base classifiers” which can be done in various ways [32]. In bagging, the ensemble models use majority voting to aggregate decisions of base models which are trained on bootstrap resamples of the training set. From a Bayesian point of view, bagging can be interpreted as a Monte Carlo integration over an approximated posterior distribution [160].

In his landmark paper, Breiman [29] noted that base model instability is an important factor in the success of bagging which led to the use of inherently instable methods like decision trees in early bagging approaches [61, 31]. The main mechanism of bagging is often said to be variance reduction [14, 30]. In more recent work, Grandvalet [82] explained that base model instability is not related to the intrinsic variability of a predictor but rather to the presence of influential instances in a data set for a given predictor (so-called *leverage points*).

The effect of bagging is explained as equalizing the influence of all training instances, which is beneficial when highly influential instances are harmful for the predictor's accuracy.

4.3.3 Justification of the RESVM algorithm

We have shown the effect of resampling contaminated sets and provided some basic insight into the mechanics of bagging. We will now link these two elements to justify bagging approaches in the context of contaminated training sets. Its usefulness can be considered by both the variance reduction argument of Bauer and Kohavi [14] and equalizing the influence of training points as described by Grandvalet [82].

Variance reduction Resampling a contaminated set yields different levels of contamination in the resamples as explained in Section 4.3.1. Varying the contamination between base model training sets induces variability between base models without increasing bias. This observation enables us to create a diverse set of base models by resampling both $\mathcal{P}$ and $\mathcal{U}$. The variance reduction of bagging is an excellent mechanism to exploit the variability of base models based on resampling [14, 30]. In the context of RESVM, a tradeoff takes place between increased variability (by training on smaller resamples, see Figure 4.1) and base models with increased stability (larger training sets for the SVM models).

Equalizing influence The influence of a training instance on an SVM model can be quantified in terms of its dual weight (the associated α value). Three distinct cases can be distinguished: (i) the training instance is correctly classified and not within the margin ($\alpha = 0$, not a SV), (ii) the training instance lies on the margin and is correctly classified ($\alpha \in [0, C]$, free SV) and (iii) the training instance is incorrectly classified or within the margin ($\alpha = C$, bounded SV), where C is the misclassification penalty associated to the training instance [27]. Instances that are misclassified during training become bounded SVs, which have the maximal α value and can therefore be considered leverage points of the SVM model. When learning with label noise, the mislabeled training instances are likely to end up as bounded SVs. In a best case scenario, the mislabeled training instances are classified in concordance to their true label by the SVM model (which means they must be a bounded SV as the training procedure identifies this as a misclassification). As such, mislabeled training instances act as leverage points for SVM models. Following Grandvalet [82], bagging equalizes the influence of training instances (e.g. lowers the influence of mislabeled leverage points in comparison to the rest of the data) which yields improved robustness against contamination in the context of RESVM.

4.3.4 RESVM training

RESVM uses CWSVM base models trained on resamples from the original training set, where both $\mathcal{P}$ and $\mathcal{U}$ are being resampled. The technique involves 5 hyperparameters: 3 to define the resampling strategy and 2 for the base models. Additional hyperparameters may be involved, for example γ for the RBF kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$.

The number of base models to include in the ensemble, $n_{nmodels}$, is the first hyperparameter. Using more base models improves the stability of the ensemble (up to a certain plateau) at a linear increase in computational cost for training and prediction. n_{models} is not a traditional hyperparameter in the sense that a good value can be determined during training, for example based on out-of-bag error estimates [12].[1]

By resampling $\mathcal{P}$, RESVM takes potential contamination of the labeled instances into account by design. Since the contamination between $\mathcal{P}$ and $\mathcal{U}$ can vary, the ability to vary the size of resamples from $\mathcal{P}$ and $\mathcal{U}$ separately is required. This results in two tuning parameters: n_{pos} and n_{unl}. In general, using small base model training sets results in increased base model variability which then necessitates using more base models in the ensemble to obtain a given level of stability. In our experiments, we have tuned n_{pos} and n_{unl} but it is also possible to obtain good values using out-of-bag techniques [133].[1]

RESVM additionally inherits at least 2 hyperparameters from its SVM base models, namely misclassification penalties for both classes and, if applicable, hyperparameters related to the kernel function. We define the CWSVM penalties in see Eq. (8.5) based on 2 hyperparameters $C_{\mathcal{U}}$ and w_{pos}:

$$C_{\mathcal{P}} = C_{\mathcal{U}} \times w_{pos} \times \frac{n_{unl}}{n_{pos}}. \tag{4.3}$$

w_{pos} enables reweighting labeled and unlabeled instances after equalizing class imbalance. In bagging SVM, w_{pos} is always fixed to 1.

[1] Note that the error estimates in out-of-bag techniques must account for potential contamination. See our discussion of hyperparameter tuning for a possible score function.

The RESVM training approach has been summarised in Algorithm 1. The algorithm uses 5 hyperparameters plus additional kernel parameters.

Algorithm 1: Training procedure for RESVM.

```
Data: $\mathcal{P}$: the set of positive instances.
      $\mathcal{U}$: the set of unlabeled instances.
Input: $n_{models}$: number of base models to include in the ensemble.
       $n_{unl}$: size of bootstrap resamples of $\mathcal{U}$.
       $n_{pos}$: size of bootstrap resamples of $\mathcal{P}$.
       $C_{\mathcal{U}}$: misclassification penalty for $\mathcal{U}$ in class-weighted SVM.
       $w_{pos}$: relative positive misclassification penalty coefficient.
       $\kappa(\cdot,\cdot)$: kernel function to be used by base models.
Output: $\Omega$: RESVM with $n_{models}$ base models.
begin
    $\Omega \leftarrow \emptyset$;
    $C_{\mathcal{P}} \leftarrow C_{\mathcal{U}} \times w_{pos} \times \frac{n_{unl}}{n_{pos}}$;
    for $i \leftarrow 1$ to $n_{models}$ do
        // create base model training set from $\mathcal{P}$ and $\mathcal{U}$.
        $\mathcal{P}^{(i)} \leftarrow$ sample $n_{pos}$ instances from $\mathcal{P}$ with replacement;
        $\mathcal{U}^{(i)} \leftarrow$ sample $n_{unl}$ instances from $\mathcal{U}$ with replacement;
        // train CWSVM base model $\psi^{(i)}$ and add to ensemble $\Omega$.
        $\psi^{(i)} \leftarrow$ train CWSVM for $\mathcal{P}^{(i)}$ vs. $\mathcal{U}^{(i)}$ (parameters $C_{\mathcal{P}}$, $C_{\mathcal{U}}$, $\kappa$);
        $\Omega \leftarrow \{\Omega, \psi^{(i)}\}$;
```

4.3.5 RESVM prediction

RESVM uses majority voting to aggregate base model predictions. By default, the returned label is the one predicted by most base models. The fraction of positive votes for a test instance $\mathbf{x}$ can be written as:

$$v(\mathbf{x}) = \frac{n_{models} + \sum_{i=1}^{n_{models}} \mathtt{sgn}(\psi^{(i)}(\mathbf{x}))}{2n_{models}}, \tag{4.4}$$

where $\mathtt{sgn}(\cdot)$ is the sign function and $\psi^{(i)}$ denotes the decision function of SVM base model i with codomain $\mathbb{R}$. $v(\cdot)$ has the interval $[0,1]$ as codomain.

The RESVM decision value for a test instance $\mathbf{x}$ is defined as the fraction of votes in favor of the positive class $v(\mathbf{x})$ unless the result is unanimous. In the case of a unanimous vote, the ensemble decision value is based on the decision values of its base models to increase the model's ability to differentiate. In case of a unanimous negative vote, the sum of the decision values of the base models is taken (each SVM base model decision value is negative in this case). In case of a unanimous positive vote, the sum of the decision values of the base models (all positive) plus one is taken. The decision value $d(\cdot)$ has codomain $\mathbb{R}$ and is computed as follows:

$$d(\mathbf{x}) = \begin{cases} v(\mathbf{x}) & \text{if } 0 < v(\mathbf{x}) < 1, \\ \sum_{i=1}^{n_{models}} \psi^{(i)}(\mathbf{x}) & \text{if } v(\mathbf{x}) = 0, \\ 1 + \sum_{i=1}^{n_{models}} \psi^{(i)}(\mathbf{x}) & \text{if } v(\mathbf{x}) = 1. \end{cases} \tag{4.5}$$

The resulting label for a given decision threshold T can be written as follows:

$$l(\mathbf{x}) = \mathtt{sgn}\big(d(\mathbf{x}) - T\big). \tag{4.6}$$

The default decision value threshold for positive classification is $T = 0.5$ (this is majority voting, e.g. positive iff more than half of all base models predict positive). Using the modified decision values $d(\mathbf{x})$ instead of the votes $v(\mathbf{x})$ does not affect the predicted labels for typical choices of the threshold T (e.g. $T \in (0, 1)$). It does, however, affect performance measures that use the entire range of decision values such as area under the PR curve. Using $d(\mathbf{x})$ enables us to rank different instances that received all positive or all negative votes by base models (e.g. $v(\mathbf{x}) = 1$ and $v(\mathbf{x}) = 0$, respectively).

4.4 Experimental setup

RESVM has been compared to class-weighted SVM (CWSVM) and bagging SVM (BAG) in a number of simulations to assess the merits of our modifications compared to conceptually comparable algorithms. In this Section we will summarize the experimental setup (training set construction, model selection and performance evaluation) and the data sets we used.

4.4.1 Simulation setup

Our experiments consist of repeated simulations on a variety of data sets under different settings. Briefly, in each iteration hyperparameters were optimized per approach based on cross-validation on the training set (using identical folds for all approaches). Subsequently, a model with the optimal parameters is trained on the full training set and used to predict an independent test set. An overview of the experiments is shown in Figure 6.2. Every experiment consists of 20 repetitions.

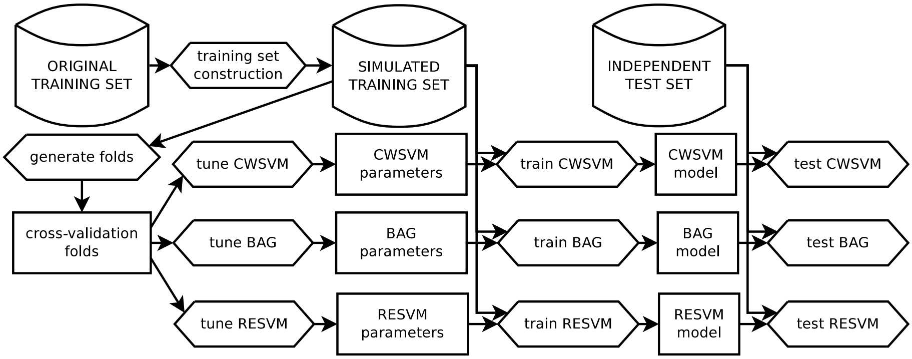


Figure 4.2: Overview of a single benchmark iteration.

To assess what situations are favorable per approach, we have investigated three different settings with distinct label noise configurations. For every data set, we performed 10 iterations per simulation in the following settings:

1. **supervised**: no contamination in $\mathcal{P}$ or $\mathcal{U}$ ($\mathcal{U}$ is the negative class).
2. **PU learning**: contamination in $\mathcal{U}$ but not in $\mathcal{P}$.
3. **semi-supervised**: contamination in both $\mathcal{P}$ and $\mathcal{U}$. The contamination levels in $\mathcal{P}$ and $\mathcal{U}$ were always chosen equal.

The contamination levels we used were chosen per data set based on when differences between the three approaches become visible. A summary is available in Table 4.1 in Section 8.4.2. When applicable, contamination was introduced by flipping class labels (e.g. true positives in $\mathcal{U}$ and true negatives in $\mathcal{P}$). This effectively changes the empirical densities of the classes in the training set (illustrated in Figure 4.3 in the next Section).

Every binary learning task was repeated 20 times to get reliable assessments of all methods. Each repetition involves redoing all steps shown in Figure 6.2, including resampling of training sets based on the known true positives and true negatives. Contamination was introduced at random where applicable by flipping class labels.

Hyperparameter selection In every iteration, hyperparameters were tuned per setting using 10-fold cross-validation over a grid of parameter tuples. To ensure a fair comparison, one set of folds is generated in each iteration and used by all methods. We ensured that the optimal values that were found during tuning in any setting were never on the edge of the search grid. The search resolution in comparable parameters between methods was always defined to be identical (for example γ in the case of an RBF kernel).

The same search grids were used in all three settings for a given data set to illustrate that a method can work well in a supervised setting with a given search grid but degrade when label noise is added. Since negative labels are unavailable in PU learning, we used the following score function in all learning settings which only requires positive labels for hyperparameter selection [121]:

$$\texttt{pu\_score} = \frac{\texttt{precision} \times \texttt{recall}}{Pr(y=1)} = \frac{\texttt{recall}^2}{Pr(\hat{y}=1)}, \tag{4.7}$$

where $Pr(y=1)$ is the fraction of known positive labels in the predicted set and $Pr(\hat{y}=1)$ is the fraction of positive predictions made by the classifier. Note that this score function is not ideal when $\mathcal{P}$ is contaminated, though we obtained good results even in that setting.

The following parameters were tuned per method: (CWSVM) $C_{\mathcal{P}}$ and $C_{\mathcal{U}}$, (BAG) $C_{\mathcal{U}}$ and $n_{\mathcal{U}}$ and (RESVM) $C_{\mathcal{U}}$, w_{pos}, n_{pos} and n_{unl}. In both ensemble approaches we consistently used 50 base models.

Performance assessment Models are trained with the optimal hyperparameters on the full training set and subsequently tested on the independent test set. We use the known test labels to compute the area under the Precision-Recall curve (AUC) for each model. We opted to use PR curves because they capture the performance of interest of models over their entire operating range and work well for imbalanced data [56].

We used statistical analyses to determine whether one approach trumps another while accounting for the variability between simulations. The nonparametric Wilcoxon signed-rank test is recommended for pairwise comparisons between learning algorithms [58]. In every setting per data set we performed a paired one-tailed Wilcoxon signed-rank test comparing the area under the PR curve of bagging SVM and RESVM with alternative hypothesis h_1 : $AUC^{RESVM} > AUC^{BAG}$ (pairs being iterations). Low p-values indicate a statistically significant improvement.

Implementation details We used the class-weighted SVM implementation available in LIBLINEAR [70] and LIBSVM [35] for models using the linear and RBF kernel, respectively. Bagging SVM and RESVM were implemented using the EnsembleSVM library [45].[2] The decision values of bagging SVM

[2] Python code for RESVM is available at `https://github.com/claesenm/resvm`.

used to compute PR curves were defined in the same way as for RESVM (see Section 4.3.5).

4.4.2 Data sets

We used a synthetic data set and 5 publicly available data sets:[3]

- `synthetic`: a 2-D binary data set. Positive instances are sampled from a standard normal distribution. Negative instances are sampled from a circle centered at the origin with radius 4 with 2-D noise superimposed from a standard normal distribution. Training and testing data was generated in every iteration. Figure 4.3 shows densities for all settings.
- `cancer`: the Wisconsin breast cancer data set related to breast cancer diagnosis. It consists of 10 features and 683 instances without an explicit train/test partitioning so we partitioned it at random in every iteration.
- `ijcnn1`: used for the IJCNN 2001 neural network competition [156], comprising 2 classes, 22 features and $49,990/91,701$ training/testing instances.
- `covtype`: a common classification benchmark about predicting forest cover types based on cartographic information [22]. We used a subsample of $100,000/40,000$ training/testing instances.
- `mnist`: a digit recognition task [120]. This data set contains 10 classes (one for each digit), 780 features, $60,000$ training instances and $10,000$ test instances with an almost uniform class distribution. We performed one-versus-all classification for each digit.
- `sensit`: SensIT Vehicle (combined), vehicle classification [63]. This data set contains 3 classes with an uneven distribution. We performed one-versus-all classification for each class. This data set has 100 features, $78,823$ training instances and $19,705$ testing instances.

Most data sets have a prespecified test set, except for `synthetic` and `cancer`. We used the prespecified test sets when available. We used the RBF kernel for all data sets except `mnist` (linear kernel). Note that both RESVM and bagging SVM models are always implicitly nonlinear due to their majority voting scheme, even when using linear base models.

[3] Public data at: `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

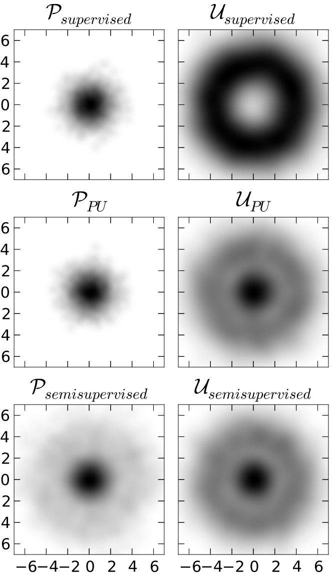


Figure 4.3: Empirical densities of the `synthetic` data used for training per problem setting (visualized in input space). The supervised densities (top row) are based on samples of the underlying positive and negative classes. The use of high contamination (30%) induces similar empirical densities for $\mathcal{P}$ and $\mathcal{U}$ in the semi-supervised setting (bottom row).

In every setting each original data set was resampled without replacement to construct training sets to use in the simulations. The resampled training sets are typically significantly smaller than what is available in the original data sets to show that some methods can obtain good models even with few training instances. An overview of the actual training sets we constructed is presented in Table 4.1.

		contamination	training set		test set	
data set	d	in percent	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{N}\rvert$
`synthetic`	2	30	100	200	5,000	5,000
`cancer`	10	30	50	200	100	100
`ijcnn1`	22	10	100	10,000	8,712	82,989
`covtype`	54	30	100	1,000	20,000	20,000
`mnist`	780	10	50	2,000	$\approx 1,000$	$\approx 9,000$
`sensit 1`	100	30	100	1,000	4,575	15,130
`sensit 2`	100	30	100	1,000	5,520	14,455
`sensit 3`	100	30	100	1,000	9,880	9,825

Table 4.1: Overview of the data sets used in simulations: number of features, contamination (when applicable), training set size as used in the experiments and test set size. The `mnist` data set consists of 10 classes and the test set is almost uniformly distributed. The `sensit` data set has 3 classes with uneven class distribution in the test set, so we treat it separately here.

4.5 Results and discussion

We will summarize all results of our simulation experiments comparing class-weighted SVM (CWSVM), bagging SVM (BAG) and the robust ensemble of SVMs (RESVM). First we will show the results of each setting separately. Subsequently we present an overview of the number of wins per setting for each method across all data sets. Section 4.5.6 shows the results of an experiment to assess the effect of contamination in $\mathcal{P}$ and $\mathcal{U}$ on all methods. Finally, we include an interesting observation regarding the optimal hyperparameters of RESVM that were found using cross-validation on the `mnist` data set per setting in Table 4.6.

4.5.1 Results for supervised classification

Table 4.2 summarizes our results in a fully supervised setting. In these experiments both $\mathcal{P}$ and $\mathcal{U}$ are uncontaminated. Based on the number of wins per simulation and the confidence intervals, we can conclude that all methods are competitive in this setting.

The confidence intervals show that all methods obtain comparable results for all simulations except `mnist` digit 8, where CWSVM performs poorly compared to the others. This performance difference could be caused by the fact we used linear class-weighted SVM while both ensemble methods implicitly yield nonlinear decision boundaries. A linear model may be too simple to properly distinguish this digit from the others.

The overall good results in the supervised setting confirm that the score function in Equation (4.7) is a good choice for tuning. In these supervised experiments we could have used a traditional score like accuracy, area under the ROC curve or F-measure, but these would no longer be useful in the other settings. The performance in these supervised experiments can be considered an objective baseline for comparison in the PU learning and semi-supervised setting since only levels of contamination are varied.

	area under PR curve				number of wins		
data	CWSVM	BAG	RESVM	p	CWSVM	BAG	RESVM
synthetic	98.1–98.7	98.7–98.8	98.7–98.8		2	12	6
cancer	98.4–98.8	98.4–98.7	98.3–98.7		8	12	0
ijcnn1	85.3–87.4	79.1–81.6	82.3–86.2	• • •	16	0	4
covtype	77.1–78.3	76.8–78.5	76.8–78.7		8	6	6
mnist (positive = x)							
0	96.9–97.5	96.9–97.4	96.9–97.4		7	8	5
1	98.1–98.3	98.3–98.5	98.2–98.5		0	8	12
2	87.3–89.1	88.5–89.8	89.6–90.5	•	2	6	12
3	83.7–85.9	86.9–88.7	88.8–90.1	• • •	0	5	15
4	88.8–90.2	89.8–91.1	90.8–92.2	• • •	1	3	16
5	78.7–80.9	79.2–81.0	81.4–83.2	• •	3	3	14
6	92.4–93.4	93.9–94.7	94.3–94.9		0	8	12
7	92.2–92.9	92.6–93.2	93.1–93.7	• • •	1	3	16
8	56.5–58.9	74.3–76.1	79.6–80.5	• • •	0	0	20
9	72.5–75.6	77.8–80.3	81.5–82.6	• • •	0	2	18
sensit (positive = x)							
1	80.5–81.4	79.8–80.7	80.5–81.3	•	10	2	8
2	65.7–75.4	72.6–74.0	73.5–74.9	• • •	15	0	5
3	35.5–56.1	92.3–92.7	91.7–92.3		0	15	5

Table 4.2: 95% CIs for mean test set performance in a fully supervised setup, the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $h_1 : AUC^{RESVM} > AUC^{BAG}$ and the number of times each approach had best test set performance. Test result encoding: • $p < 0.05$, • • $p < 0.01$ and • • • $p < 0.001$.

4.5.2 Results for PU learning

The results of our experiments in a PU learning setting are shown in Table 4.3. In the pure PU learning setting, $\mathcal{P}$ is uncontaminated but $\mathcal{U}$ is contaminated. Class-weighted SVM tends to suffer from the largest loss in performance between supervised learning and pure PU learning based on area under PR curves. Class-weighted SVM obtains less wins than it did in the supervised simulations (21 wins in PU learning compared to 73 in the supervised setting), except on the `cancer` data set. Bagging SVM and RESVM maintain strong performance. Bagging SVM obtains a comparable number of wins and RESVM gains many compared to the supervised setting.

On the `mnist` data, RESVM consistently exhibits the best performance (based on the Wilcoxon signed-rank test), though the effective improvement over bagging SVM is marginal. On `sensit` with classes 2 or 3 as positive, bagging SVM obtains the majority of wins though the confidence intervals of its area under the PR curve overlap completely with those of RESVM. On the other data sets, no worthwile differences were obtained between both ensemble methods.

4.5.3 Results of semi-supervised classification

In the semi-supervised setting we deliberately violated the assumption of an uncontaminated positive training set by contaminating $\mathcal{P}$ and $\mathcal{U}$. The results listed in Table 4.4 confirm that both class-weighted and bagging SVM are vulnerable to contamination in $\mathcal{P}$ and experience very large performance losses. We believe this is induced by using high misclassification penalties for training instances in $\mathcal{P}$ without any resampling to account for potential false positives. In bagging SVM this leads to a systematic bias in all base models. The resampling strategy of RESVM prevents systematic bias over all base models.

The results clearly show that RESVM is more robust to false positives, evidenced by a much lower drop in predictive performance for almost all data sets. The performance difference between bagging SVM and RESVM is statistically significant for all data sets except `covtype` and `sensit`. Surprisingly, CWSVM obtains 8 wins on `sensit` with class 2 as positive. RESVM shows the best and most consistent performance overall.

On the `mnist` data, RESVM not only achieved consistently higher area under the PR curve, but visual inspection showed that its PR curves almost always dominated the others over the entire range. This means that in this experiment, RESVM models are always better than the others regardless of design priorities (high precision versus high recall). As an illustration, Figure 4.4 shows the PR

	area under PR curve				number of wins		
data	CWSVM	BAG	RESVM	p	CWSVM	BAG	RESVM
synthetic	96.9–98.4	97.9–98.6	98.2–98.5		6	8	6
cancer	98.2–98.5	87.5–98.4	96.1–98.1		10	7	3
ijcnn1	71.2–76.5	73.4–78.2	72.6–80.7	•	1	5	14
covtype	65.2–67.9	70.2–72.2	71.4–73.0		0	6	14
mnist (positive = x)							
0	74.1–77.8	90.5–93.3	94.6–95.5	• • •	0	5	15
1	89.1–91.2	95.2–96.7	96.4–97.3	• •	0	5	15
2	55.2–60.1	75.5–80.0	84.2–86.1	• • •	0	0	20
3	54.6–60.2	74.5–80.3	83.6–86.2	• • •	0	2	18
4	57.8–62.5	73.9–80.3	83.9–85.9	• • •	0	2	18
5	53.3–56.7	63.8–70.3	69.1–72.6	•	0	7	13
6	66.9–71.0	85.9–89.7	90.6–92.5	• •	0	4	16
7	71.4–74.8	84.0–88.0	90.0–91.4	• • •	0	1	19
8	34.8–38.8	63.5–69.1	72.2–74.8	• • •	0	4	16
9	50.5–54.8	66.2–71.0	74.2–76.4	• • •	0	1	19
sensit (positive = x)							
1	61.6–73.0	70.6–75.3	72.5–76.2	•	2	7	11
2	58.6–68.1	68.5–70.5	67.8–70.0		2	10	8
3	33.2–50.2	90.2–91.8	89.7–91.1		0	14	6

Table 4.3: 95% CIs for mean test set performance in a PU learning setup, the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $h_1 : AUC^{RESVM} > AUC^{BAG}$ and the number of times each approach had best test set performance. Test result encoding: • $p < 0.05$, • • $p < 0.01$ and • • • $p < 0.001$.

and ROC curves of a representative simulation with digit 7 as positive. Since the PR curve of RESVM completely dominates the others we know that its ROC curve does too [56].

Finally, it is worth noting that the confidence intervals of RESVM tend to be narrower than those of both other approaches. Even though RESVM base models have more variability compared to bagging SVM base models, the overall performance of RESVM is more reliable. This constitutes an important practical advantage since assessing different models is not trivial outside of simulation studies (e.g. when no negative labels are available).

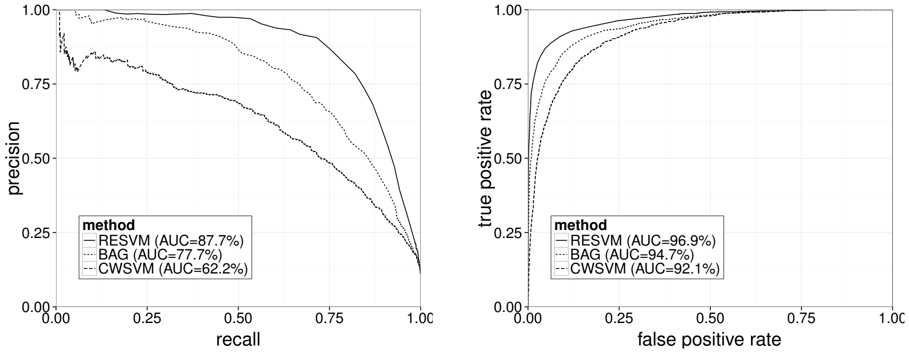
	area under PR curve				number of wins		
data	CWSVM	BAG	RESVM	p	CWSVM	BAG	RESVM
synthetic	83.6–90.0	91.9–94.9	96.4–97.4	• • •	3	2	15
cancer	62.5–80.2	91.1–96.7	96.2–97.6	•	1	8	11
ijcnn1	69.8–73.4	67.4–70.4	72.0–75.2	• • •	5	2	13
covtype	58.1–61.8	61.2–64.2	60.4–65.7		4	4	12
mnist (positive = x)							
0	59.9–64.1	72.8–81.1	91.4–93.4	• • •	0	0	20
1	80.3–82.7	90.6–93.4	96.1–97.4	• • •	0	0	20
2	42.3–48.0	55.1–63.7	79.8–83.0	• • •	0	0	20
3	43.8–47.6	59.9–66.0	78.1–81.1	• • •	0	0	20
4	52.4–56.2	66.4–72.8	79.7–83.4	• • •	0	0	20
5	40.5–45.2	56.0–61.1	65.8–69.4	• • •	0	2	18
6	52.4–57.3	72.9–79.3	87.9–90.9	• • •	0	0	20
7	58.7–61.6	69.9–77.3	87.9–90.2	• • •	0	1	19
8	29.7–33.9	48.3–55.3	68.0–71.0	• • •	0	0	20
9	42.1–44.9	52.5–59.0	68.7–72.7	• • •	0	0	20
sensit (positive = x)							
1	34.5–49.4	59.6–69.0	60.6–66.4		3	12	5
2	44.9–53.7	46.4–53.4	50.1–56.7	•	8	4	8
3	44.5–61.1	75.4–83.5	80.5–84.9	•	1	7	12

Table 4.4: 95% CIs for mean test set performance in a semi-supervised setup, the results of a paired one-tailed Wilcoxon signed-rank test comparing the AUC of BAG and RESVM with alternative hypothesis $h_1 : AUC^{RESVM} > AUC^{BAG}$ and the number of times each approach had best test set performance. Test result encoding: • $p < 0.05$, • • $p < 0.01$ and • • • $p < 0.001$.

4.5.4 A note on the number of repetitions per experiment

The tightness of the confidence intervals of generalization performance allow us to conclude that the number of repetitions (20) is sufficient to demonstrate the merits of RESVM (see Tables 4.2–4.4). Increasing the number of repetitions further would yield even narrower confidence intervals and increase the amount of statistically significant results in the Wilcoxon signed-rank test comparing bagging SVM and RESVM (due to increased power). All key conclusions remain valid if the number of repetitions would be increased.

Additional statistically significant results may only be obtained in experiments where the improvement offered by RESVM is too small to be of practical significance (as large improvements already yield significant test results). Failure



(a) Precision-Recall curves.

(b) ROC curves.

Figure 4.4: Performance in semi-supervised setting on `mnist`, digit 7 as positive.

to reject the null hypothesis ($h_0 : AUC^{BAG} \geq AUC^{RESVM}$) in our current results indicates that (i) bagging SVM is effectively better than RESVM, (ii) they are comparable or (iii) the performance improvement of RESVM is too small to yield a significant test result given the current sample size (number of repetitions). Increasing the number of repetitions can only lead to additional statistically significant results in the latter situation.

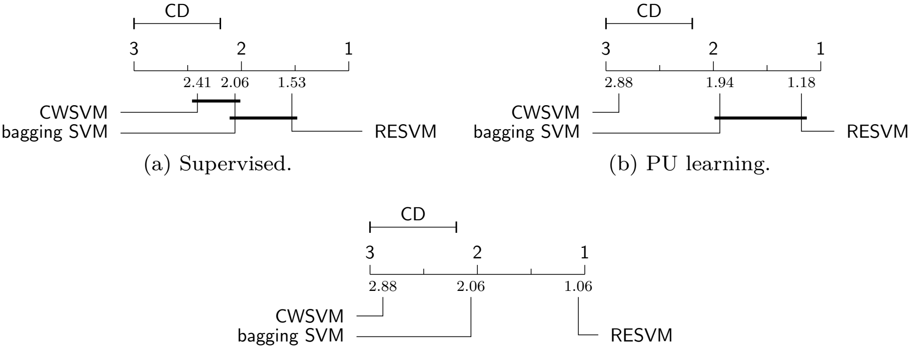
To illustrate our claims, we performed 100 repetitions for `covtype` in the semi-supervised setting. This yielded the following CIs and win counts: CWSVM 59.0–60.5% (8 wins), bagging SVM 62.3–63.5% (21 wins), RESVM 63.8–65.8% (71 wins). The p-value of the Wilcoxon signed-rank test becomes 2×10^{-5}, while the p-value was insignificant with 20 repetitions (Table 4.4).

4.5.5 Trend across data sets

In the previous tables we have shown the results per data set for each setting. In this section we summarize the results across all data sets, using critical difference diagrams [58] in Section 4.5.5 and an overview of win counts in Section 4.5.5.

Critical difference diagrams

In every setting, we compared the performance of the three learning approaches across all data sets using non-parametric statistical tests. For each data set, approaches were ranked based on their mean area under the PR curve across



(a) Supervised.

(b) PU learning.

(c) Semi-supervised.

Figure 4.5: Critical difference diagrams for each setting. Groups of algorithms that are not significantly different at the 5% significance level are connected.

all iterations. Multiclass data sets count once per class. Friedman tests per setting yielded significant evidence of differences between the three learning approaches at the $\alpha = 0.05$ level, though this was marginal in the supervised setting ($p = 0.034$). The Nemenyi post-hoc test [144] was used after each omnibus test to assess differences between all approaches. The critical difference diagrams in Figure 4.5 visualize the results.

Critical difference diagrams were introduced by Demšar [58] to visualize a comparison of multiple learning approaches over multiple data sets. These diagrams depict the average rank of each approach (lower is better) along with the critical difference (CD). The critical difference is the minimum difference in average ranks that yields a significant result in the Nemenyi post-hoc test. It depends on the significance level ($\alpha = 0.05$), the number of learning approaches (3) and the number of data sets (17).

From Figure 4.5 we can conclude that bagging SVM and RESVM are comparable in the PU learning setting (both significantly better than CWSVM). In the semi-supervised setting, bagging SVM is statistically significantly better than CWSVM and RESVM is significantly better than both other approaches across all data sets.

Win counts

The number of wins per method across all data sets are summarized in Table 4.5. The top half shows the total number of wins across all data sets, which weighs mnist and sensit heavier than the other data sets since we performed several one-vs-all experiments. Because RESVM consistently performed very strong on mnist, the top half is an overly optimistic representation.

The bottom half of Table 4.5 contains normalized results, where every data set contributes equally. Based on these numbers we can conclude that there is little difference between the three methods in a supervised setting. In the PU learning setting, ensemble methods become favorable over CWSVM (bagging SVM and RESVM being competitive). Finally, in the semi-supervised setting RESVM pulls far ahead of both other methods and obtains 65% of the normalized wins, which is over three times more than bagging SVM and over five times more than class-weighted SVM.

	CWSVM		bagging SVM		RESVM	
setting	count	win %	count	win %	count	win %
supervised	73	21	93	27	**174**	**51**
PU learning	21	6	88	26	**231**	**68**
semi-supervised	25	7	42	12	**273**	**80**
supervised	**44.8**	**37.3**	40.3	33.6	36.0	30.0
PU learning	18.3	15.3	39.4	32.8	**62.2**	**51.8**
semi-supervised	17.0	14.2	24.0	20.0	**79.0**	**65.8**

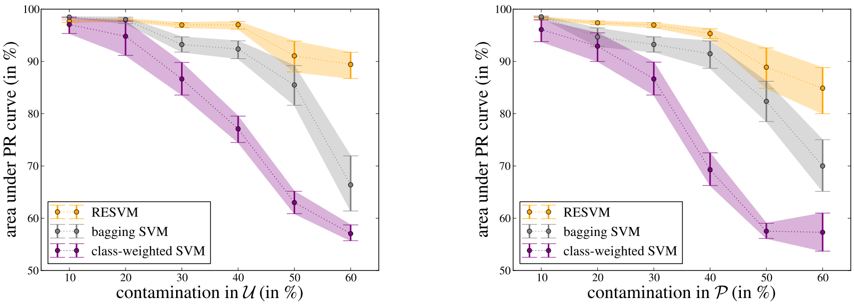
Table 4.5: Number of wins in simulations for each method per setting. The bottom half shows normalized number of wins, where wins in multiclass data sets (mnist and sensit) are divided by the number of classes.

4.5.6 Effect of contamination

In this Section we show the effect of different levels of contamination in $\mathcal{P}$ and $\mathcal{U}$ on the synthetic data set. In these simulations, we fixed the contamination level in one part of the training set ($\mathcal{P}$ or $\mathcal{U}$) and the contamination of other was varied. The fixed contamination was set to 30%. Twenty simulations were run per contamination setting.

In these experiments, we used random search to tune hyperparameters of each method [15] using the Optunity package.[4] Briefly, hyperparameters were searched by random sampling 100 tuples uniformly within a given box and subsequently the best tuple was selected as before. We ensured that the optimal hyperparameters were never too close to the edge of the feasible region (if so,

[4] Optunity is available at: http://www.optunity.net.



(a) Effect of contamination in $\mathcal{U}$.

(b) Effect of contamination in $\mathcal{P}$.

Figure 4.6: Effect of different levels of contamination in $\mathcal{U}$ and $\mathcal{P}$ on generalization performance. The plots show point estimates of the mean area under the PR curve across experiments and the associated 95% confidence intervals.

the box was expanded). Note that this approach of testing a fixed number of tuples favors methods with less hyperparameters. Even though RESVM has more hyperparameters than the other methods, good models can be obtained at the same search cost.

The results are shown in Figure 4.6. In general, contamination in $\mathcal{P}$ causes larger performance losses than the same level of contamination in $\mathcal{U}$ for all algorithms. As expected, the difference in sensitivity to contamination in $\mathcal{P}$ and $\mathcal{U}$ is smallest for RESVM in which $\mathcal{P}$ and $\mathcal{U}$ are resampled similarly. At high contamination levels, RESVM is the only method that still works well (even at 60%).

Figure 4.6a illustrates that RESVM and bagging SVM behave in a similar fashion at contamination levels of $\mathcal{U}$ up to 50% and both outperform class-weighted SVM. RESVM outperforms bagging SVM for contamination levels of 30–50% but the consistency (width of CI) and performance losses of both methods are comparable. Figure 4.6b shows the increased robustness of RESVM to contamination in $\mathcal{P}$ resulting in reduced loss of generalization performance for increasing contamination.

4.5.7 RESVM optimal parameters

As an illustration of the implicit mechanism of RESVM we show some of the optimal tuning parameters for every setting in Table 4.6. These parameters were obtained by performing 10-fold cross-validation on the training set.

	0	1	2	3	4	5	6	7	8	9	mean
$\mathbf{n_{pos}}$											
supervised	20	20	20	20	20	20	10	20	20	10	18
PU learn	10	10	10	10	10	15	10	10	10	10	10.5
semi-sup.	10	5	10	10	10	10	10	10	10	10	9.5
$\mathbf{n_{unl}/n_{pos}}$											
supervised	10	10	10	10	10	10	10	10	10	10	10
PU learn	5	5	5	5	5	5	5	5	5	5	5
semi-sup.	5	5	5	8	5	5	5	5	5	5	5.25
$\mathbf{w_{pos}}$											
supervised	1.6	1.6	1.6	3.2	3.2	3.2	3.2	1.6	3.2	2.4	2.48
PU learn	4.8	6.4	3.2	6.4	4.8	6.4	4.8	4.8	6.4	6.4	5.44
semi-sup.	12.8	6.4	4.8	2.1	4.8	6.4	4.8	3.2	3.2	3.2	5.17

Table 4.6: Medians of optimal hyperparameters per digit obtained via cross-validation and mean of all medians per setting. The normalized relative weight on positives versus unlabeled instances (w_{pos}) is associated with the relative size and contamination of the positive and unlabeled training sets.

An interesting observation is that the size of the training sets that are being used decreases for increasing contamination. Increasing label noise induces RESVM to favor smaller base model training sets for which the variability in contamination is larger (see Figure 4.1). Though this may appear counterintuitive, bagging approaches are known to exhibit a bias-variance tradeoff [14] for which using weaker base models with increased variability may yield better ensembles [111].

The optimal value of the misclassification penalty for positive training instances relative to unlabeled instances, w_{pos}, changes between learning settings (see Equation (4.3)). It exhibits expected behaviour: the maximum value is obtained when the certainty on $\mathcal{P}$ relative to $\mathcal{U}$ is largest (e.g. the pure PU learning setting). This parameter implicitly balances empirical certainty on $\mathcal{P}$ and $\mathcal{U}$ and is an important degree of freedom in RESVM. In bagging SVM, this parameter is implicitly fixed to 1 via Equation (8.6) [138]. Note that w_{pos} need not be larger than 1 (which would place extra emphasisis on the known labels after accounting for class imbalance). In highly imbalanced settings where $n_{unl} \gg n_{pos}$, the optimal value of w_{pos} may well be less than 1.

4.6 Conclusion

We have introduced a new approach for learning from positive and unlabeled data, called the robust ensemble of SVMs (RESVM). RESVM constructs an ensemble model using a bagging strategy in which the positive and unlabeled sets are resampled to obtain base model training sets. By resampling both $\mathcal{P}$ and $\mathcal{U}$, our approach is more robust against false positives than others.

The robustness of our approach to potential contamination in both $\mathcal{P}$ and $\mathcal{U}$ can be attributed to the synergy between our resampling scheme and voting aggregation. The resampling itself strongly resembles a typical bootstrap approach. RESVM uses class-weighted SVM base models though the resampling scheme is likely to work well with other types of base models.

RESVM was compared with class-weighted SVM and bagging SVM on several data sets under different label noise conditions. The trends across data sets show that bagging SVM and RESVM outperform class-weighted SVM in PU learning. In a pure PU learning setting the average improvement over existing methods is modest though RESVM classifiers exhibit lower variance in performance making it more reliable.

In the semi-supervised setting, label noise was introduced in $\mathcal{P}$ to highlight the improved robustness of RESVM compared to the other methods. Our experimental results show that RESVM remains very strong in the semi-supervised setting while both other approaches degrade dramatically. Statistical analysis showed that RESVM is significantly better than both other approaches across all data sets.

Visual inspection of the PR curves shows that in the majority of experiments the curve for RESVM not only has higher AUC but completely dominates the other curves. As such RESVM models are a good approach regardless of design priorities (high recall versus high precision).

A weakness of RESVM is its amount of hyperparameters (5 plus potential kernel parameters), though RESVM models are less sensitive to accurate tuning of these parameters than standard SVM. Our experiments indicated that although RESVM has more hyperparameters, good models can be obtained at the same search effort than the other approaches (e.g. testing the same number of hyperparameter tuples). An interesting question is whether prior knowledge regarding contamination of $\mathcal{P}$ and $\mathcal{U}$ can help in limiting the search scope for some of the hyperparameters (n_{pos}, n_{unl} and w_{pos} specifically).

Chapter 5

Hyperparameter Search in Machine Learning

This chapter has been previously published as:
Claesen, M., & De Moor, B. (2015). **Hyperparameter Search in Machine Learning**. In *Proceedings of the 11th Metaheuristics International Conference (MIC)*, Agadir, Morocco.
Manuscript available at `http://arxiv.org/abs/1502.02127`.

Abstract

We describe the hyperparameter search problem in the field of machine learning and discuss its main challenges from an optimization perspective. Machine learning methods attempt to build models that capture some element of interest based on given data. Most common learning algorithms feature a set of hyperparameters that must be determined before training commences. The choice of hyperparameters can significantly affect the resulting model's performance, but determining good values can be complex; hence a disciplined, theoretically sound search strategy is essential.

5.1 Introduction

Machine learning research focuses on the development of methods that are capable of capturing some element of interest from a given data set. Such elements include but are not limited to coherent structures within data (clustering) or the ability to predict certain target values based on given characteristics, which may be discrete (classification) or continuous (regression).

A large variety of learning methods exist, ranging from biologically inspired neural networks [21] over kernel methods [169] to ensemble models [31, 46]. A common trait in these methods is that they are parameterized by a set of hyperparameters λ, which must be set appropriately by the user to maximize the usefulness of the learning approach. Hyperparameters are used to configure various aspects of the learning algorithm and can have wildly varying effects on the resulting model and its performance.

Hyperparameter search is commonly performed manually, via rules-of-thumb [99, 98] or by testing sets of hyperparameters on a predefined grid [154]. These approaches leave much to be desired in terms of reproducibility and are impractical when the number of hyperparameters is large [43]. Due to these flaws, the idea of automating hyperparameter search is receiving increasing amounts of attention in machine learning, for instance via benchmarking suites [66] and various initiatives.[1] Automated approaches have already been shown to outperform manual search by experts on several problems [18, 15].

We briefly introduce some key challenges inherent to hyperparameter search in Section 5.2. The combination of all these hurdles make hyperparameter search a formidable optimization task. In Section 5.3 we give a succinct overview of the current state-of-the-art in terms of algorithms and available software.

5.1.1 Example: controlling model complexity

A key balancing act in machine learning is choosing an appropriate level of model complexity: if the model is too complex, it will fit the data used to construct the model very well but generalize poorly to unseen data (overfitting); if the complexity is too low the model won't capture all the information in the data (underfitting). This is often referred to as the bias-variance trade-off [77, 51], since a complex model exhibits large variance while an overly simple one is strongly biased. Most general-purpose methods feature hyperparameters to control this trade-off; for instance via regularization as in support vector machines and regularization networks [69, 95].

[1] Such as `http://www.automl.org/` and `https://www.codalab.org/competitions/2321`.

5.1.2 Formalizing hyperparameter search

The goal of many machine learning tasks can be summarized as training a model $\mathcal{M}$ which minimizes some predefined loss function $\mathcal{L}(\mathbf{X}^{(te)};\ \mathcal{M})$ on given test data $\mathbf{X}^{(te)}$. Common loss functions include mean squared error and error rate. The model $\mathcal{M}$ is constructed by a learning algorithm $\mathcal{A}$ using a training set $\mathbf{X}^{(tr)}$; typically involving solving some (convex) optimization problem. The learning algorithm $\mathcal{A}$ may itself be parameterized by a set of hyperparameters λ, e.g. $\mathcal{M} = \mathcal{A}(\mathbf{X}^{(tr)};\ \lambda)$. An example model $\mathcal{M}$ is a support vector machine classifier with Gaussian kernel [169], for which the training problem $\mathcal{A}$ is parameterized by the regularization constant C and kernel bandwidth σ, i.e. $\lambda = [C, \sigma]$.

The goal of hyperparameter search is to find a set of hyperparameters $\lambda^\star$ that yield an optimal model $\mathcal{M}^\star$ which minimizes $\mathcal{L}(\mathbf{X}^{(te)};\ \mathcal{M})$. This can be formalized as follows [43]:

$$\lambda^\star = \arg\min_{\lambda} \mathcal{L}\big(\mathbf{X}^{(te)};\ \mathcal{A}(\mathbf{X}^{(tr)};\ \lambda)\big) = \arg\min_{\lambda} \mathcal{F}(\lambda;\ \mathcal{A},\ \mathbf{X}^{(tr)}, \mathbf{X}^{(te)},\ \mathcal{L}). \quad (5.1)$$

The objective function $\mathcal{F}$ takes a tuple of hyperparameters λ and returns the associated loss. The data sets $\mathbf{X}^{(tr)}$ and $\mathbf{X}^{(te)}$ are given and the learning algorithm $\mathcal{A}$ and loss function $\mathcal{L}$ are chosen . Depending on the learning task, $\mathbf{X}^{(tr)}$ and $\mathbf{X}^{(te)}$ may be labeled and/or equal to each other. In supervised learning, a data set is often split into $\mathbf{X}^{(tr)}$ and $\mathbf{X}^{(te)}$ using hold-out or cross-validation methods [65, 115].

5.2 Challenges in hyperparameter search

The characteristics of the search problem depend on the learning algorithm $\mathcal{A}$, the chosen loss function $\mathcal{L}$ and the data set $\mathbf{X}^{(tr)}$, $\mathbf{X}^{(te)}$, as shown in Equation (6.1). Hyperparameter search is typically approached as a non-differentiable, single-objective optimization problem over a mixed-type, constrained domain. In this section we will discuss the origins and consequences of challenges in hyperparameter search.

5.2.1 Costly objective function evaluations

Each objective function evaluation requires evaluating the performance of a model trained with hyperparameters λ. Depending on the available computational resources, the nature of the learning algortihm $\mathcal{A}$ and size of the problem ($\mathbf{X}^{(tr)}$, $\mathbf{X}^{(te)}$) each evaluation may take considerable time. Training times in the order of minutes are considered fast, since days and even weeks are not unheard of [117, 57, 184]. Evaluation time is exacerbated when procedures that train multiple models are employed; for instance to reliably estimate generalization performance [65, 115]. This leads to an increasing need for efficient methods to optimize hyperparameters that require a minimal amount of objective function evaluations.

Additionally, the time required to train and test models can be contingent upon the choice of hyperparameters. Some hyperparameters have an obvious influence on train and/or test time, e.g. the architecture of neural networks [21] and size of ensembles [31, 46]. The influence of hyperparameters can also be subtle, for instance regularization and kernel complexity can significantly affect training time for support vector machines [27].

5.2.2 Randomness

The objective function often exhibits a stochastic component, which can be induced by various components of the machine learning pipeline, for example due to inherent randomness of the learning algorithm (initialization of a neural network, resampling in ensemble approaches, ...) or due to finite sample effects in estimating generalization performance. This stochasticity can sometimes be addressed via machine learning techniques; but unfortunately such solutions typically dramatically increase the time required per objective function evaluation, limiting their usefulness in some settings.

This inherent stochasticity directly implies that the empirical best hyperparameter tuple, obtained after a given set of evaluations, is not necessarily the true optimum of interest $\lambda^\star$. Fortunately, many search methods are designed to probe many tuples close to the empirical best. If the search region surrounding the empirical optimum is densely sampled, we can determine whether the empirical best was an outlier or not in a post-processing phase, for instance by assuming Lipschitz continuity or smoothness.

5.2.3 Complex search spaces

The number of hyperparameters is usually small (≤ 5), but it can range up to hundreds for complex learning algorithms [17] or when preprocessing steps are also subjected to optimization [102]. It has been demonstrated empirically that in many cases only a handful of hyperparameters significantly impact performance, though identifying the relevant ones in advance is difficult [15].

Hyperparameters are usually of continuous or integer type, leading to mixed-type optimization problems. Continuous hyperparameters are commonly related to regularization. Common integer hyperparameters are related to network architecture for neural networks [21], size of ensembles [31, 46] or the parameterization of kernels in kernel methods [169].

Some tasks feature highly complex search spaces, in which the very existence of certain hyperparameters are conditional upon the value of others [102, 18, 16]. A simple example is optimizing the architecture of neural networks [21], where the number of hidden layers is one hyperparameter and the size of each layer induces a set of additional hyperparameters, conditional upon the number of layers.

5.3 Current approaches

A wide variety of optimization methods have been used for hyperparameter search, including particle swarm optimization [135, 123], genetic algorithms [187], coupled simulated annealing [206] and racing algorithms [20]. Surprisingly, randomly sampling the search space was only established recently as a baseline for comparison of optimization methods [15]. Bayesian and related sequential model based optimization techniques using variants of the expected improvement criterion [105] are receiving a lot of attention currently [18, 101, 180, 10, 66], owing to their efficiency in terms of objective function evaluations.

Software packages are being released which implement various dedicated optimization methods for hyperparameter search. Such packages are usually intended to be used in synergy with machine learning libraries that provide learning algorithms [154]. Most of these packages focus on Bayesian methods [102, 180, 16], though metaheuristic optimization approaches are also offered [43]. The increased development of such packages testifies towards the growing interest in automated hyperparameter search.

5.4 Conclusion

A fully automated, self-configuring learning strategy can be considered the holy grail of machine learning. Though the current state-of-the-art still has a long way to go before this goal can be reached, it is evident that hyperparameter search is a crucial element in its pursuit. Automated hyperparameter search is a hot topic within the machine learning community which we believe can benefit greatly from the techniques and lessons learnt in metaheuristic optimization.

Chapter 6

Easy Hyperparameter Search Using Optunity

This chapter has been submitted to:
Claesen, M., Simm, J., Popovic, D., Moreau, Y., & De Moor, B. (2015). **Easy hyperparameter search using Optunity**, *Journal of Machine Learning Research.*

Abstract

OPTUNITY is a free software package dedicated to hyperparameter optimization. It contains various types of solvers, ranging from undirected methods to direct search, particle swarm and evolutionary optimization. The design focuses on ease of use, flexibility, code clarity and interoperability with existing software in popular machine learning environments. OPTUNITY is written in Python and contains interfaces to R, Octave and MATLAB. OPTUNITY uses a BSD license and is available at `http://www.optunity.net`.

6.1 Introduction

Many machine learning tasks involve training a model $\mathcal{M}$ which minimizes some loss function $\mathcal{L}(\mathcal{M} \mid \mathbf{X}^{(te)})$ on given test data $\mathbf{X}^{(te)}$. A model is obtained via a learning algorithm $\mathcal{A}$ which uses a training set $\mathbf{X}^{(tr)}$ and solves some optimization problem. The learning algorithm $\mathcal{A}$ may itself be parameterized by a set of hyperparameters λ, e.g. $\mathcal{M} = \mathcal{A}(\mathbf{X}^{(tr)} \mid \lambda)$. Hyperparameter search – also known as tuning – aims to find a set of hyperparameters λ^*, such that the learning algorithm yields an optimal model $\mathcal{M}^*$ that minimizes $\mathcal{L}(\mathcal{M} \mid \mathbf{X}^{(te)})$:

$$\lambda^* = \arg\min_{\lambda} \mathcal{L}\big(\mathcal{A}(\mathbf{X}^{(tr)} \mid \lambda) \mid \mathbf{X}^{(te)}\big) = \arg\min_{\lambda} \mathcal{F}(\lambda \mid \mathcal{A},\ \mathbf{X}^{(tr)}, \mathbf{X}^{(te)},\ \mathcal{L}). \tag{6.1}$$

In tuning, $\mathcal{F}$ is the objective function and the hyperparameters λ are optimization variables. The learning algorithm $\mathcal{A}$, loss function $\mathcal{L}$ and data sets $\mathbf{X}^{(tr)}$ and $\mathbf{X}^{(te)}$ are known.

Tuning hyperparameters is a recurrent task in machine learning which may significantly affect overall performance. Commonly tuned hyperparameters are related to kernels, regularization, learning rates and network architecture. Some specific challenges associated to hyperparameter optimization are discussed by Claesen and De Moor [42]. General machine learning packages provide only basic tuning methods like grid search [154]. In practice, the most common tuning approaches are grid search and manual tuning, though both are known to fail when the number of hyperparameters grows and manual search is additionally hard to reproduce [15].

6.2 Optunity

Optunity offers various optimizers and utility functions to enable efficient hyperparameter optimization using only a few of lines of code and minimal expertise. Our software is complementary to libraries that provide learning algorithms, such as scikit-learn [154]. The package uses a BSD license and is simple to deploy in any environment. Optunity supports Python, R, Octave and MATLAB on Linux, OSX and Windows.

6.2.1 Functional Overview

Optunity provides both simple routines for lay users and expert routines that enable fine-grained control of various aspects of the solving process. Basic tuning requires only an objective function, a maximum number of evaluations and box constraints on the hyperparameters to be optimized. Conditional search spaces in which the existence of some hyperparameters is contingent upon some discrete choice are also supported.

The objective function must be defined by the user. It takes a hyperparameter tuple λ and typically involves three steps: (i) training a model $\mathcal{M}$ with λ, (ii) use $\mathcal{M}$ to predict a test set and (iii) compute some score or loss based on the predictions.

Tuning involves a series of function evaluations until convergence or until a predefined maximum number of evaluations is reached. Optunity is capable of vectorizing evaluations in the working environment to speed up the process at the end user's volition.

Optunity also provides k-fold cross-validation to estimate the generalization performance of supervised modeling approaches. The implementation can account for strata and clusters.[1] Finally, a variety of common quality metrics is available. The snippet below shows how to tune an SVM classifier with RBF kernel using scikit-learn and Optunity:[2]

```
1 @optunity.cross_validated(x=data, y=labels, num_folds=10, num_iter=2)
2 def score(x_train, y_train, x_test, y_test, C, gamma):
3     model = sklearn.svm.SVC(C=10**C, gamma=10**gamma).fit(x_train, y_train)
4     decision_values = model.decision_function(x_test)
5     return optunity.metrics.roc_auc(y_test, decision_values)
6
7 hps, _, _ = optunity.maximize(score, num_evals=100, C=[-5, 2], gamma=[-5, 0])
8 svm = sklearn.svm.SVC(C=10**hps[’C’], gamma=10**hps[’gamma’])
9 svm.fit(data, labels)
```

The objective function as per Equation (6.1) is defined on lines 1 to 5, where $\lambda = (C, \gamma)$, $\mathcal{A}$ is the SVM training algorithm and $\mathcal{L}$ is area under the ROC curve. We use 2× iterated 10-fold cross-validation to estimate area under the ROC curve. Up to 100 hyperparameter tuples are tested in an exponential search space, bounded by $10^{-5} < C < 10^{2}$ and $10^{-5} < \gamma < 10^{0}$ on line 7. Finally, an SVM with optimized hyperparameters is trained on lines 8 and 9.

6.2.2 Available Solvers

Optunity provides a wide variety of solvers, ranging from basic, undirected methods like grid search, sobol sequences and random search [15] to evolutionary

[1] Instances in a stratum should be spread across folds. Clustered instances must remain in a single fold.

[2] We assume the correct imports are made and `data` and `labels` contain appropriate content.

methods such as particle swarm optimization [112], the covariance matrix adaptation evolutionary strategy (CMA-ES) [90], tree-structured Parzen estimator [18] and the Nelder-Mead simplex. The default solver is particle swarm optimization, which performs well for a large variety of tuning tasks involving various learning algorithms. Additional solvers will be incorporated in the future.

6.2.3 Software Design and Implementation

The design philosophy of Optunity prioritizes code clarity over performance. This is justified by the fact that objective function evaluations constitute the performance bottleneck.

In contrast to typical Python packages, we avoid dependencies to facilitate users working in non-Python environments (sometimes at the cost of performance). To prevent issues for users that are unfamiliar with Python, care is taken to ensure all code in Optunity works out of the box on any Python version above 2.7, without requiring tools like `2to3` to make explicit conversions. Optunity has optional dependencies on DEAP [74] and Hyperopt [16] for the CMA-ES and TPE solvers, respectively.

A key aspect of Optunity's design is interoperability with external environments. This requires bidirectional communication between Optunity's Python back-end ($\mathcal{O}$) and the external environment ($\mathcal{E}$) and roughly involves three steps: (i) $\mathcal{E} \rightarrow \mathcal{O}$ solver configuration, (ii) $\mathcal{O} \leftrightarrow \mathcal{E}$ objective function evaluations and (iii) $\mathcal{O} \rightarrow \mathcal{E}$ solution and solver summary. To this end, Optunity can do straightforward communication with any environment via sockets using JSON messages as shown in Figure 6.1. Only some information must be communicated, big objects like data sets are never exchanged. To port Optunity to a new environment, a thin wrapper must be implemented to handle communication.

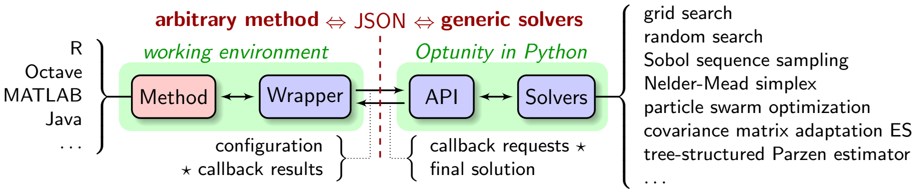


Figure 6.1: Integrating Optunity in non-Python environments.

6.2.4 Development and Documentation

Collaborative development is organized via GitHub.[3] The project's master branch is kept stable and is subjected to continuous integration tests using Travis CI. We recommend prospective users to clone the master branch for the most up-to-date stable version of the software. Bug reports and feature requests can be filed via issues on GitHub. Future development efforts will focus on wrappers for Java and C/C++. We additionally plan to incorporate Bayesian optimizers which have no reference implementation in other packages.

Code is documented using Sphinx and contains many doctests that can serve as both unit tests and examples of the associated functions. Our website contains API documentation, user documentation and a wide range of examples to illustrate all aspects of the software. The examples involve various packages and environments, including scikit-learn [154], OpenCV and Spark's MLlib [211].

6.3 Related Work

A number of software solutions exist for hyperparameter search. Hyperopt offers random search and sequential model-based optimization [16]. Some packages dedicated to Bayesian approaches include Spearmint [180], DiceKriging [164]. Finally, ParamILS is a command-line-only tuning framework providing iterated local search [102].

Optunity distinguishes itself from other packages by exposing a variety of fundamentally different solvers through a lightweight API. Optunity's client-server model facilitates integration in any language and environment and can even be used to run solvers remotely.

6.4 Solver Performance Benchmark

Optunity's particle swarm optimizer (PSO) was compared to Hyperopt's tree of Parzen estimators (TPE) in minimizing the Rastrigin function $f(x, y) = 20 + x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y)$, which has a known optimum $f(0, 0) = 0$. Optimization was done uniformly within box constraints $|x|, |y| < 5.12$. We used the Rastrigin function because it has many local minima, as is common

[3] We maintain the following subdomains for convenience: `http://`{builds, docs, git, issues}`.optunity.net`.

in hyperparameter search [42]. Figure 6.2 shows that both PSO and TPE outperform random search. TPE is marginally better than PSO initially (few evaluations) while PSO converges much faster later on. PSO has also been shown to be competitive to TPE in tuning RBMs by Papa et al. [150].

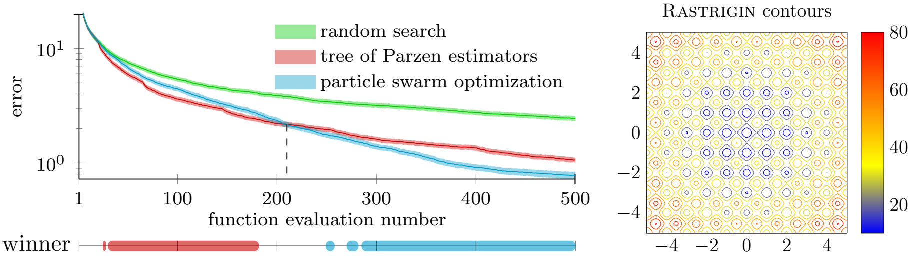


Figure 6.2: Evolution of average error per solver while minimizing the Rastrigin function (contours depicted on the right) based on 1,000 repetitions. The winner line shows the statistically significant best solver per evaluation count, if any ($\alpha = 5\%$).

Chapter 7

Assessing Binary Classifiers Using Only Positive and Unlabeled Data

This chapter has been submitted to:
Claesen, M., Davis, J., De Smet, F., & De Moor, B. (2015). **Assessing Binary Classifiers Using Only Positive and Unlabeled Data**, *Neural Information Processing Systems (NIPS).*

Abstract

Assessing the performance of a learned model is a crucial part of machine learning. However, in some domains only positive and unlabeled examples are available, which prohibits the use of most standard evaluation metrics. We propose an approach to estimate any metric based on contingency tables, including ROC and PR curves, using only positive and unlabeled data. Estimating these performance metrics is essentially reduced to estimating the fraction of (latent) positives in the unlabeled set, assuming known positives are a random sample of all positives. We provide theoretical bounds on the quality of our estimates, illustrate the importance of estimating the fraction of positives in the unlabeled set and demonstrate empirically that we are able to reliably estimate ROC and PR curves on real data.

7.1 Introduction

Model evaluation is a critical step in the learning process. Typically, evaluations either report summary metrics, such as accuracy, F1 score, or area under the receiver operator characteristic (ROC) curve or visually show a model's performance under different operating conditions by using ROC or precision-recall curves. All the aforementioned evaluation approaches require constructing contingency tables (also called confusion matrices), which show how a model's predicted labels relate to an example's ground truth label. Computing a contingency table requires labeled examples. However, for many problems only a few labeled examples and many unlabeled ones are available as acquiring labels can be time-consuming, costly, unreliable, and in some cases impossible.

The field semi-supervised learning [36] focuses on coping with partially labeled data. Positive and unlabeled (PU) learning is a special case of semi-supervised learning where each example's label is either positive or not known [127, 208, 59, 67, 173, 137, 40]. Both semi-supervised and PU learning tend to focus on developing learning algorithms that cope with partially labeled data during training as opposed to evaluating algorithms when the test set is partially labeled. What is less well studied is the effect of partially labeled data on evaluation. Currently, algorithms are evaluated assuming that the test data is fully labeled [80, 145, 39, 33, 37, 137, 40] and if the test data is only partially labeled, sometimes it is assumed that all unlabeled instances are negative when evaluating performance [139, 179, 174].

This paper describes how to incorporate the unlabeled data in the model evaluation process. We show how to compute contingency tables based on only positive and unlabeled examples where the unlabeled set contains both positive and negative examples, by looking at the ranking of examples produced by a model. Theoretically, we establish important relationships between contingency tables and rank distributions, which allow us to provide bounds on the false positive rate at each rank when the ranking contains examples whose ground truth label is unknown. Our findings have important implications for model selection as we show that naively assuming that all unlabeled examples are negative, as is sometimes done in PU learning, could lead to selecting the wrong model. We demonstrate the efficacy of our approach by estimating ROC and PR curves from real-world data.

7.2 Background and definitions

We first review the relevant background on model evaluation and issues caused by partial labeling.

7.2.1 Rank distributions and contingency tables

We focus on binary decision problems, where the goal is to classify examples as either positive or negative. Most learned models (e.g., SVM, logistic regression, naive Bayes) predict a numeric score for each example where higher values imply higher confidence that the instance belongs to the positive class. Typically, a *ranking* $\mathcal{R}$ is produced by sorting examples in descending order by their numeric score such that confident positive predictions are ranked close to the top of $\mathcal{R}$.[1]

Within a ranking $\mathcal{R}$, we treat $\mathcal{P} \subset \mathcal{R}$ as the subset of examples with positive labels, $\bar{\mathcal{P}} = \mathcal{R} - \mathcal{P}$ as the subset of examples with negative labels, and let $\mathrm{rank}(\mathcal{R}, x)$ denote the rank of an instance x in $\mathcal{R}$. Given a cutoff rank r, predictions can be made by assigning the positive class to the r top ranked instances and the negative class to the rest. This decision rule yields a *true positive rate (TPR)*, which is the fraction of positive examples that are correctly labeled as positive, and *false positive rate (FPR)*, which is the fraction of negative examples that are incorrectly labeled as positive:

$$\begin{aligned}\mathrm{TPR}(\mathcal{P}, r) &= \Pr(\mathrm{rank}(\mathcal{R}, x) \leq r \mid x \in \mathcal{P}),\\ &= |\{\, x \in \mathcal{P} \;:\; \mathrm{rank}(\mathcal{R}, x) \leq r\}| \;/\; |\mathcal{P}|, \qquad (7.1)\end{aligned}$$

$$\mathrm{FPR}(\mathcal{P}, r) = \Pr(\mathrm{rank}(\mathcal{R}, \bar{x}) \leq r \mid \bar{x} \in \bar{\mathcal{P}}) = \mathrm{TPR}(\mathcal{R} - \mathcal{P}, r). \qquad (7.2)$$

Given the number of positives $|\mathcal{P}|$ and negatives $|\mathcal{R} - \mathcal{P}|$, the contingency table for a rank r is:

$$\mathrm{TP}(\mathcal{P}, r) = \mathrm{TPR}(\mathcal{P}, r) \cdot |\mathcal{P}|, \;\; (7.3) \qquad \mathrm{FP}(\mathcal{P}, r) = \mathrm{FPR}(\mathcal{P}, r) \cdot |\mathcal{R} - \mathcal{P}|,$$

$$\mathrm{FN}(\mathcal{P}, r) = |\mathcal{P}| - \mathrm{TP}(\mathcal{P}, r), \;\; (7.4) \qquad \mathrm{TN}(\mathcal{P}, r) = |\mathcal{R} - \mathcal{P}| - \mathrm{FP}(\mathcal{P}, r). \;\; (7.5)$$

The rank distribution of a set of instances $\mathcal{P}$ within an overall ranking $\mathcal{R}$ is defined as the distribution of their corresponding ranks within $\mathcal{R}$. The rank cumulative distribution function (CDF) of a set of instances $\mathcal{P}$ is defined as the (empirical) CDF of their ranks, i.e. $\forall\ r \in \{1, \ldots, |\mathcal{R}|\}$:

$$\mathrm{F}(\mathcal{P}, r) = \Pr(\mathrm{rank}(\mathcal{R}, x) \leq r \mid x \in \mathcal{P}). \qquad (7.6)$$

[1] Which means a low value for rank in this work, though this is often referred to as *highly ranked* in literature.

The concept of rank CDF is illustrated in Figure 7.1. Note that $\mathrm{F}(\mathcal{P}, r) \equiv \mathrm{TPR}(\mathcal{P}, r)$ (Equations (7.1) and (7.6)), that is, the rank CDF of the set of positives $\mathcal{P}$ at rank r in an overall ranking $\mathcal{R}$ can be interpreted directly as a true positive rate, when labeling the r top ranked instances as positive.

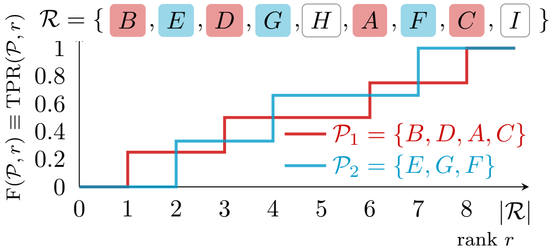


Figure 7.1: Rank CDF of two sets of positives $\mathcal{P}_1 = \{B, D, A, C\}$ and $\mathcal{P}_2 = \{E, G, F\}$ within an overall ranking $\mathcal{R} = \{B, E, D, G, H, A, F, C, I\}$, with $|\mathcal{P}_1| = 4$ and $|\mathcal{P}_2| = 3$. In practice $\mathcal{R}$ is obtained by sorting the data according to classifier score. The rank CDF of a set $\mathcal{S} \subseteq \mathcal{R}$ is based on the positions of elements of $\mathcal{S}$ in $\mathcal{R}$.

We use two convenience functions to partition sets of ranks:

$$\mathrm{head}(X, r) = \{\ \mathrm{rank}(\mathcal{R}, x) \leq r\ :\ x \in X\ \},$$

$$\mathrm{tail}(X, r) = \{\ \mathrm{rank}(\mathcal{R}, x) > r\ :\ x \in X\ \},$$

such that $\mathrm{head}(X, r)\ \cup\ \mathrm{tail}(X, r) = X$ and $|\,\mathrm{head}(X, r)| = \mathrm{F}(X, r) \cdot |X|$.

7.2.2 ROC and PR curves

Receiver operator characteristic (ROC) curves are used extensively for evaluating classifiers in machine learning [28] as they illustrate the performance of a model over its entire operating range. ROC curves depict how a model's true positive rate (shown on the y-axis) varies as a function of its false positive rate (shown on the x-axis). Each cutoff rank $r \in \{1, \ldots, |\mathcal{R}|\}$ corresponds to a single point (i.e., (FPR, TPR) pair) in ROC space (Eqs. (7.1) and (7.2)). An (empirical) ROC curve for a ranking $\mathcal{R}$ and set of positives $\mathcal{P} \subset \mathcal{R}$ is constructed by computing $\mathrm{FPR}(\mathcal{P}, r)$ and $\mathrm{TPR}(\mathcal{P}, r)$ at each rank r and interpolating by drawing a straight line between points corresponding to consecutive ranks. The area under an ROC curve (AUROC) is a commonly used summary statistic, typically ranging between 0.5 (random model) and 1 (perfect model). AUROC is a popular criterion in model selection and is often used as the optimization objective in hyperparameter search [28].

Precision-Recall (PR) curves [55] are an alternative to ROC curves that show how a model's precision (y-axis) varies as a function of recall (x-axis). Recall is equivalent to TPR and precision is the fraction of examples classified as positive that are truly positive ($\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$). PR curves are widely used when there is a skew in the class distributions [54, 40].

7.2.3 Evaluation with partially labeled data

In the partial labeling setting, $\mathcal{R}$ consists of disjoint sets of known positives $\mathcal{P}_L$, known negatives $\mathcal{N}_L$ and unlabeled instances $\mathcal{U}$. The unlabeled set $\mathcal{U}$ consists of latent positives $\mathcal{P}_U$ and latent negatives. The fraction of latent positives in the unlabeled set plays a crucial role in our work, denoted by β:

$$\beta = \Pr(x \in \mathcal{P}_U \mid x \in \mathcal{U}) = |\mathcal{P}_U| \ / \ |\mathcal{U}|. \tag{7.7}$$

Note that computing contingency tables requires fully labeled data. If only a few labeled instances of both classes are available, they can be used to compute rough estimates of predictive performance. However, if only positive labels are available, even a rough approximation of common metrics cannot be estimated directly as we do not know which unlabeled examples are positives and which are negative. A common approach to evaluate models in a PU learning context is to treat the full unlabeled set as negative [139, 179, 174], though we will show that this may lead to spurious results.

7.3 Relationship between the rank CDF of positives and contingency tables

The challenge of incorporating unlabeled data into an evaluation metric is knowing which unlabeled examples are latent positives and which are latent negatives. Our insight is that, if the known positives are sampled completely at random from all positives, the rank distribution of latent positives should follow the rank distribution of known positives. Thus if we know β, which is needed to compute the expected number of latent positives within the unlabeled data, this provides an avenue for building contingency tables that incorporate the unlabeled data. To do so, we first prove relationships between rank CDFs of sets of positives within an overall ranking at a given rank r and the corresponding contingency tables. Then, we use these relationships to prove bounds on the FPR at a given rank r when the ranking includes unlabeled examples, some of which are latent positives.

7.3.1 Rank distributions and contingency tables based on subsets of positives within a ranking

We begin by considering given sets of positives within an overall ranking. Proofs of all lemmas can be found in Appendix 7.A, along with figures to illustrate the associated property.

Lemma 1. *Given a rank r and two disjoint subsets of positives $\mathcal{P}_1$ and $\mathcal{P}_2$ within an overall ranking $\mathcal{R}$. If $|\mathcal{P}_1| = |\mathcal{P}_2|$ and $\mathrm{TPR}(\mathcal{P}_1, r) > \mathrm{TPR}(\mathcal{P}_2, r)$, then $\mathrm{FPR}(\mathcal{P}_1, r) < \mathrm{FPR}(\mathcal{P}_2, r)$.*

Lemma 2. *Given a rank r and two disjoint sets of positives $\mathcal{P}_1$, $\mathcal{P}_2$ in a ranking $\mathcal{R}$ and $\mathcal{P}_\Omega = \mathcal{P}_1 \cup \mathcal{P}_2$. If $\mathrm{TPR}(\mathcal{P}_1, r) < \mathrm{TPR}(\mathcal{P}_2, r)$ then $\mathrm{TPR}(\mathcal{P}_1, r) < \mathrm{TPR}(\mathcal{P}_\Omega, r) < \mathrm{TPR}(\mathcal{P}_2, r)$.*

Corollary 1. *Given a rank r and three sets of positives $\mathcal{P}_A$, $\mathcal{P}_B$ and $\mathcal{P}_C$ within a ranking $\mathcal{R}$ such that $\mathcal{P}_A \cap \mathcal{P}_B = \emptyset$ and $\mathcal{P}_A \cap \mathcal{P}_C = \emptyset$ and $|\mathcal{P}_B| = |\mathcal{P}_C|$, then*

$$\mathrm{TPR}(\mathcal{P}_B, r) < \mathrm{TPR}(\mathcal{P}_C, r) \quad \leftrightarrow \quad \mathrm{TPR}(\mathcal{P}_A \cup \mathcal{P}_B, r) < \mathrm{TPR}(\mathcal{P}_A \cup \mathcal{P}_C, r).$$

7.3.2 Contingency tables based on partially labeled data

Lemmas 1 and 2 describe relationships between rank distributions and contingency tables of different (but known) sets of positives within an overall ranking. We now show how to construct contingency tables corresponding to the greatest-lower and least-upper bound of the FPR at a given rank, accounting for the unknown set of latent positive example from partially labeled data, given β.

Theorem 1. *Given an overall ranking $\mathcal{R}$ consisting of disjoint sets of known positives $\mathcal{P}_L$, known negatives $\mathcal{N}_L$ and unlabeled instances $\mathcal{U}$, where $\mathcal{U}$ contains an unknown set of latent positives $\mathcal{P}_U \subset \mathcal{U}$ of known size $|\mathcal{P}_U| = \beta \cdot |\mathcal{U}|$. Given a rank r and an upper bound $\mathcal{T}_{ub}(r) \geq \mathrm{TPR}(\mathcal{P}_U, r)$, a tight lower bound on $\mathrm{FPR}(\mathcal{P}_\Omega, r)$ with $\mathcal{P}_\Omega = \mathcal{P}_L \cup \mathcal{P}_U$ can be found without explicitly identifying $\mathcal{P}_U$.*

Proof: *Step 1: assign a set of surrogate positives $\mathcal{P}_U^\star$:*[2]

$$\mathcal{P}_U^\star = \underset{\mathcal{P}_U^* \subset \mathcal{U}}{\arg\min} \; \mathrm{TPR}(\mathcal{P}_U^*, r), \tag{7.8}$$

$$\text{subject to } \; \mathrm{TPR}(\mathcal{P}_U^*, r) \geq \mathcal{T}_{ub}(r) \; \textit{and} \; |\mathcal{P}_U^*| = \beta \cdot |\mathcal{U}|,$$

[2] A surrogate positive is an example that we treat as if its ground truth label is positive (even though in reality its ground truth label is unknown) when constructing a contingency table.

then $\mathrm{TPR}(\mathcal{P}_U^\star, r) \geq \mathrm{TPR}(\mathcal{P}_U, r)$ *by construction. If* $|\,\mathrm{head}(\mathcal{U}, r)| < \beta\mathcal{T}_{ub}(r) \cdot |\mathcal{U}|$*, then no* $\mathcal{P}_U^\star$ *exists that satisfies the constraint* $\mathrm{TPR}(\mathcal{P}_U^\star, r) \geq \mathcal{T}_{ub}(r)$ *in Equation* (7.8).[3] *In this case, treat all instances in* $\mathrm{head}(\mathcal{U}, r)$ *as surrogate positive, which trivially implies* $\mathrm{TPR}(\mathcal{P}_U^\star, r) \geq \mathrm{TPR}(\mathcal{P}_U, r)$*.*

Step 2: define $\mathcal{P}_\Omega^\star = \mathcal{P}_L \cup \mathcal{P}_U^\star$*. Using Corollary 1 yields* $\mathrm{TPR}(\mathcal{P}_\Omega^\star, r) \geq \mathrm{TPR}(\mathcal{P}_\Omega, r)$*. Since* $|\mathcal{P}_\Omega^\star| = |\mathcal{P}_\Omega|$*, using Lemma 1 yields the lower bound on FPR, i.e.,* $\mathrm{FPR}(\mathcal{P}_\Omega^\star, r) \leq \mathrm{FPR}(\mathcal{P}_\Omega, r)$*.* ■

Applying Theorem 1 yields a nontrivial lower bound on $\mathrm{FPR}(\mathcal{P}_\Omega, r)$. In Lemma 3 we prove that $\mathrm{FPR}(\mathcal{P}_\Omega^\star, r)$ is the greatest achievable lower bound based on a given $\mathcal{U} \subset \mathcal{R}$.

Lemma 3. *Minimizing* $\mathrm{TPR}(\mathcal{P}_U^\star, r)$ *in Equation* (7.8) *of Theorem 1 ensures* $\mathrm{FPR}(\mathcal{P}_\Omega^\star, r)$ *is the greatest achievable lower bound on* $\mathrm{FPR}(\mathcal{P}_\Omega, r)$ *given* β*,* $\mathcal{T}_{ub}(r)$*,* $\mathcal{R}$ *and* $\mathcal{U}$*.*

Due to its symmetry, Theorem 1 can also be used to obtain the least achievable upper bound of $\mathrm{FPR}(\mathcal{P}_\Omega, r)$ given a ranking $\mathcal{R}$ and a bound $\mathcal{T}_{lb}(r) \leq \mathrm{TPR}(\mathcal{P}_U^\star, r)$ by assigning $\mathcal{P}_U^\star$ such that:

$$\mathcal{P}_U^\star = \underset{\mathcal{P}_U^* \subset\, \mathcal{U}}{\arg\max}\ \mathrm{TPR}(\mathcal{P}_U^*, r), \tag{7.9}$$

$$\text{subject to}\ \ \mathrm{TPR}(\mathcal{P}_U^*, r) \leq \mathcal{T}_{lb}(r) \text{ and } |\mathcal{P}_U^*| = \beta \cdot |\mathcal{U}|.$$

7.4 Efficiently computing the bounds

We now describe how to use Theorem 1 and Lemma 3 to compute the contingency tables corresponding to the greatest lower and least upper bound on $\mathrm{FPR}(\mathcal{P}_\Omega, r)$ from a finite sample. First, we explain how to compute contingency tables efficiently via Theorem 1. Second, we propose how to obtain the bounds on rank CDF ($\mathcal{T}_{lb}(r)$ and $\mathcal{T}_{ub}(r)$) that are needed to build the contingency table.

7.4.1 Computing the contingency table with greatest-lower bound on FPR at given rank r

Given β, $\mathcal{R}$ and the sets $\mathcal{P}_L$, $\mathcal{N}_L$, and $\mathcal{U}$, Theorem 1 enables computing contingency tables corresponding to the least upper and greatest lower bound

[3] An infeasibility implies that $\mathcal{T}_{ub}(r)$ and/or β are too high.

on FPR at a given cutoff rank r. We focus on building the contingency table corresponding to the lower bound on the FPR, the other is analogous.

We decompose the computation to consider the labeled and unlabeled instances separately:

$$\begin{bmatrix}\mathrm{TP}^r_\Omega & \mathrm{FP}^r_\Omega \\ \mathrm{FN}^r_\Omega & \mathrm{TN}^r_\Omega\end{bmatrix} = \begin{bmatrix}\mathrm{TP}^r_L = |\,\mathrm{head}(\mathcal{P}_L, r)| & \mathrm{FP}^r_L = |\,\mathrm{head}(\mathcal{N}_L, r)| \\ \mathrm{FN}^r_L = |\,\mathrm{tail}(\mathcal{P}_L, r)| & \mathrm{TN}^r_L = |\,\mathrm{tail}(\mathcal{N}_L, r)|\end{bmatrix} + \begin{bmatrix}\mathrm{TP}^r_U & \mathrm{FP}^r_U \\ \mathrm{FN}^r_U & \mathrm{TN}^r_U\end{bmatrix}.$$

Given that at rank r we can directly compute partial contingency tables for the labeled data based on $\mathcal{R}$, $\mathcal{P}_L$ and $\mathcal{N}_L$, we focus on computing the contingency table for the unlabeled instances.

Given $\mathcal{T}_{ub}(r)$, we can use Theorem 1 to determine the values in the contingency table for the unlabeled instances for the greatest lower bound on FPR. Doing so requires inferring a set of surrogate positives $\mathcal{P}^\star_U$ from the unlabeled data, which must be a solution to Equation (7.8). This requires θ surrogate positives in $\mathrm{head}(\mathcal{P}^\star_U, r)$ and the rest in $\mathrm{tail}(\mathcal{P}^\star_U, r)$, where θ is defined as:

$$\theta = \lceil \mathcal{T}_{ub}(r) \cdot |\mathcal{P}^\star_U| \rceil = \lceil \mathcal{T}_{ub}(r) \cdot \beta \cdot |\mathcal{U}| \rceil, \tag{7.10}$$

By rounding up in Equation (7.10), we ensure that $\mathrm{TPR}(\mathcal{P}^\star_U, r) \geq \mathcal{T}_{ub}(r)$ as required by Theorem 1.

In practice, two corner cases must be considered. One is if $|\,\mathrm{head}(\mathcal{U}, r)| < \theta$, then it is impossible to assign θ surrogates below rank r in $\mathcal{U}$. In this case, all of $\mathrm{head}(\mathcal{U}, r)$ is assigned as surrogate positives and the remaining surrogates are in $\mathrm{tail}(\mathcal{U}, r)$ (as discussed in Theorem 1). Two is if $|\,\mathrm{tail}(\mathcal{U}, r)| < |\mathcal{P}^\star_U| - \theta$, in which case all of $\mathrm{tail}(\mathcal{U}, r)$ is labeled positive and the remaining surrogate positives inevitably end up in $\mathrm{head}(\mathcal{U}, r)$. Hence, any set of surrogate positives $\mathcal{P}^\star_U$ that meets the following criteria solves Equation (7.8) and thus yields a valid bound:

$$\begin{aligned} |\mathcal{P}^\star_U| &= \beta \cdot |\mathcal{U}|, \\ |\,\mathrm{head}(\mathcal{P}^\star_U, r)| &= \begin{cases} \min\big(|\,\mathrm{head}(\mathcal{U}, r)|,\ \theta\big) & \text{if } |\mathcal{P}^\star_U| - \theta \leq |\,\mathrm{tail}(\mathcal{U}, r)|, \\ |\mathcal{P}^\star_U| - |\,\mathrm{tail}(\mathcal{U}, r)| & \text{if } |\mathcal{P}^\star_U| - \theta > |\,\mathrm{tail}(\mathcal{U}, r)|. \end{cases} \end{aligned} \tag{7.11}$$

Given a set of surrogate positives $\mathcal{P}^\star_U$, the partial contingency table of interest becomes:

$$\begin{bmatrix}\mathrm{TP}^r_U & \mathrm{FP}^r_U \\ \mathrm{FN}^r_U & \mathrm{TN}^r_U\end{bmatrix} = \begin{bmatrix}|\,\mathrm{head}(\mathcal{P}^\star_U, r)| & |\,\mathrm{head}(\mathcal{U} - \mathcal{P}^\star_U, r)| \\ |\,\mathrm{tail}(\mathcal{P}^\star_U, r)| & |\,\mathrm{tail}(\mathcal{U} - \mathcal{P}^\star_U, r)|\end{bmatrix}, \tag{7.12}$$

where $\mathcal{U} - \mathcal{P}^\star_U$ is the set of surrogate negatives and $|\mathcal{P}^\star_U|$ and $|\,\mathrm{head}(\mathcal{P}^\star_U, r)|$ are known via Eq. 7.11.

Note that computing the partial contingency table for the unlabeled data can be done very efficiently since it only requires set sizes as shown in Equation 7.12, without explicitly partitioning the unlabeled set $\mathcal{U}$. That is, we do not need to know which examples are in head$(\mathcal{P}^\star_U, r)$, tail$(\mathcal{P}^\star_U, r)$, head$(\mathcal{U} - \mathcal{P}^\star_U, r)$ and tail$(\mathcal{U} - \mathcal{P}^\star_U, r)$, we just need to know the number of examples each set contains.

The contingency table with least upper bound on $\mathrm{FPR}(\mathcal{P}_U, r)$ is obtained by replacing Eq. (7.10) by:

$$\theta = \left\lfloor \mathcal{T}_{lb}(r) \cdot |\mathcal{P}^\star_U| \right\rfloor = \left\lfloor \mathcal{T}_{lb}(r) \cdot \beta \cdot |\mathcal{U}| \right\rfloor. \tag{7.13}$$

7.4.2 Bounds on the rank distribution of $\mathcal{P}_U$

Applying Theorem 1 to build a contingency table at rank r requires a bound $\mathcal{T}_{ub}(r) \geq \mathrm{TPR}(\mathcal{P}_U, r)$ for estimating a lower bound on the FPR and a bound $\mathcal{T}_{lb}(r) \leq \mathrm{TPR}(\mathcal{P}_U, r)$ for estimating an upper bound on the FPR. To compute these bounds, we assume known and latent positives have similar rank distributions. This holds when known positives $\mathcal{P}_L$ are selected completely at random from all positives $\mathcal{P}_\Omega$, but is violated if the process of selecting examples for labeling is biased [39].

$\mathrm{TPR}(\mathcal{P}_\Omega, r)$ is estimated via the empirical CDF of $\mathcal{P}_L$, which only approximates the true CDF. To acccount for uncertainty, we construct confidence intervals (CIs) for the rank CDF. Our assumption implies that a CI of the CDF based on $\mathcal{P}_L$ is also a CI of the CDF of $\mathcal{P}_U$. A CI boundary is treated as a function mapping rank r to the estimated bound on the CDF. $\mathcal{T}_{lb}$ and $\mathcal{T}_{ub}$ denote these bounds:

$$0 \leq \mathcal{T}_{lb}(r) \leq \mathrm{TPR}(\mathcal{P}_L, r), \mathrm{TPR}(\mathcal{P}_U, r), \mathrm{TPR}(\mathcal{P}_\Omega, r) \leq \mathcal{T}_{ub}(r) \leq 1,\ \forall\ r. \tag{7.14}$$

We formalize the bounds of the CI of the CDF as functions of rank because an underlying set with that rank distribution does not necessarily exist in the overall ranking $\mathcal{R}$.

The confidence band on rank CDF can be computed based on the known positives in several ways. We use a standard bootstrap approach [64] in our experiments. Having many known positives yields a tight confidence band on rank CDF, which then translates to tight bounds on performance metrics.

7.5 Constructing ROC and PR curve estimates

Next, we describe how to estimate bounds on the true ROC and PR curves. Though we focus on these two criteria, our approach can be used to estimate any metric based on contingency tables.

ROC curves Given a ranking, instead of constructing a single ROC curve, our approach computes two curves: one corresponding to the upper bound and one corresponding to the lower bound on the CI on rank CDF of known positives $\mathcal{P}_L$, using the methodology outlined in Section 4 to compute two contingency tables for each rank r, corresponding to the greatest lower and least upper bound on $\text{FPR}(\mathcal{P}_\Omega, r)$. The set of contingency tables corresponding to greatest lower bounds on FPR at each rank form an upper bound on the ROC curve of all positives $\mathcal{P}_\Omega$, whereas the set of contingency tables corresponding to the least upper bound on FPR form a lower bound on the ROC curve of $\mathcal{P}_\Omega$.

It is important to understand how these estimates correspond to bounds in ROC space. By computing θ as in Equation (7.10) to obtain the greatest lower bound on $\text{FPR}(\mathcal{P}_U, r)$, the corresponding TPR is higher than $\text{TPR}(\mathcal{P}_U, r)$. As such, the upper bound on the ROC curve is shifted upwards and to the left. Conversely, the lower bound on the ROC curve (based on the least upper bound on FPR at each rank, i.e. θ as in Equation (7.13)) is shifted downward and to the right. This implies that the upper bound on the ROC curve completely dominates the curve of $\mathcal{P}_\Omega$ and the lower bound is completely dominated by the curve of $\mathcal{P}_\Omega$, provided that $\mathcal{T}_{lb}(r) \leq \text{TPR}(\mathcal{P}_U, r) \leq \mathcal{T}_{ub}(r),\ \forall r \in \{1, \ldots, |\mathcal{R}|\}$.

Convergence properties The convergence properties of our bounds are contingent on those of (a CI on) the empirical CDF: via the strong law of large numbers the empirical CDF $\hat{F}_n(x)$ is a consistent pointwise estimator of the true CDF $F(x)$, converging uniformly for increasing n [196].

Figure 7.2 shows the convergence of the bounds on area under the curve for the estimated lower and upper bound of the ROC curve for increasing amounts of known positives in simulated rankings. The range of bounds depends on the width of the CI on rank CDF, which in turn depends on the number of known positives (higher is better) and the size of the total data set (lower is better).

PR curves Given the contingency tables used to generate the least upper bound and greatest lower bound ROC curves, it is straightforward to construct the corresponding bounds in PR space. Each contingency table contains all the required information for generating a point in PR space.

A key result relating ROC and PR curves is that one curve dominates another in ROC space if and only if it also dominates in PR space [55]. Given this result, mapping the bounds we obtain for ROC curves to PR space directly yields (tight) bounds on the corresponding true PR curve. Since the upper bound in ROC space completely dominates the true curve, and the lower bound in ROC space is completely dominated by it, the same holds for the bounds on PR curves.

7.6 Discussion and Recommendations

Next, we discuss several issues related to using our approach in practice.

7.6.1 Determining $\hat{\beta}$ and its effect

Our approach requires having an estimate $\hat{\beta}$ of β. There are many problems where β is known from domain knowledge (e.g., calculated and published based on a data source you do not have access to), but explicit negatives are scarce or unavailable in the data under analysis. A real-world example where this is true is the task of predicting whether someone has diabetes from health insurance data [BLINDED for review]. Some individuals are coded as having diabetes, but many diabetics are undiagnosed and hence it is wrong to assume that all unlabeled patients do not have diabetes. However, the incidence rate of diabetes is known and published in the medical literature. This type of situation characterizes many medical problems. If β is not known from domain knowledge, then it could be estimated from data [67, 173, 168].

In either case, if $\hat{\beta}$ is not exact, the conditions of Lemma 1 are potentially violated where it is used within Theorem 1. The effects of set size on FPR is characterized in Lemma 4, which will help us understand the effect of over or under estimating β.

Lemma 4. *Given two sets of positive labels $\mathcal{P}_1$ and $\mathcal{P}_2$ within an overall ranking $\mathcal{R}$ and a rank r, such that $\mathrm{TPR}(\mathcal{P}_1, r) = \mathrm{TPR}(\mathcal{P}_2, r) = t$ and $|\mathcal{P}_1| > |\mathcal{P}_2|$, then:*

$$(a) \quad \mathrm{FPR}(\mathcal{P}_2, r) < t \rightarrow \mathrm{FPR}(\mathcal{P}_1, r) < \mathrm{FPR}(\mathcal{P}_2, r),$$

$$(b) \quad \mathrm{FPR}(\mathcal{P}_2, r) > t \rightarrow \mathrm{FPR}(\mathcal{P}_1, r) > \mathrm{FPR}(\mathcal{P}_2, r).$$

(a) corresponds to a ranking and cutoff that is better than random (i.e. $\mathrm{TPR}(\mathcal{P}, r) > \mathrm{FPR}(\mathcal{P}, r)$).

Lemma 4 has a large practical impact. If the ranking of $\mathcal{P}_L$ is better than random, then over and under estimating $\hat{\beta}$ is useful to obtain a (loose) upper/lower bound on performance curves, respectively. In other words, given bounds or a CI on β, that is $\hat{\beta}_{lo} \leq \beta \leq \hat{\beta}_{up}$, we can use $\hat{\beta}_{lo}$ and $\hat{\beta}_{up}$ to estimate a lower and upper bound on the true ROC or PR curve. Bounds computed based on a CI for β constitute a CI for the performance metric (at the same confidence level), assuming the rank CDF of $\mathcal{P}_U$ is contained by the confidence band on the rank CDF. Tighter bounds on β translate directly to tighter bounds on performance estimates. Finally, treating the full unlabeled set as negative results underestimates performance, since $\hat{\beta} = 0 < \beta$. The effect of varying $\hat{\beta}$ is shown in Figure 7.3.

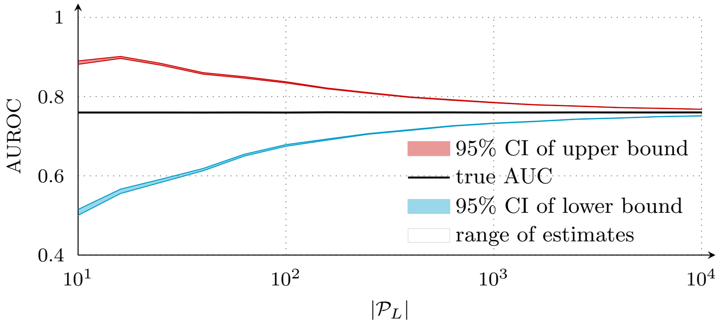


Figure 7.2: The effect of $|\mathcal{P}_L|$ on estimated AUC. Based on $|\mathcal{U}| = 100,000$, $\mathcal{N}_L = \emptyset$ and $\hat{\beta} = \beta = 0.2$. Bounds on rank CDF were obtained via bootstrap. The depicted confidence intervals are based on 200 repeated experiments.

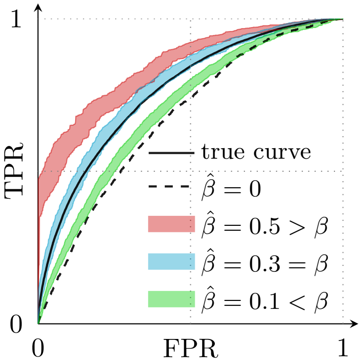


Figure 7.3: The effect of $\hat{\beta}$ on estimated ROC curves, based on 2,000 known positives, 100,000 unlabeled instances and $\beta = 0.3$.

7.6.2 Model selection

Often evaluation metrics are used to select the best model from a set of candidates. If model A's ROC (PR) curve dominates model B's ROC (PR) curve, then for all β model A is better than model B (leaving aside significance testing). However, in most cases one model does not dominate another model and there exists a point where the two curves cross. Surprisingly, the ordering in terms of both AUROC and AUPR are dependent on $\hat{\beta}$ when this happens. This means that the ordering of models according to these metrics can switch when $\hat{\beta}$ changes. Figure 7.4 depicts an example that illustrates this. This demonstrates that $\hat{\beta}$ can play a crucial role in model selection. In the likely event that the curves cross, it is important to look at the range of possible values for $\hat{\beta}$ that represent different operating conditions when selecting among different models.

A more formal explanation of why this occurs can be made based on partial derivatives of each entry of the partial contingency table and TPR, FPR and precision based on unlabeled instances to $\hat{\beta}$:[4]

$$\frac{\partial\, \mathrm{TPR}_U^r}{\partial \hat{\beta}} = 0, \tag{7.15}$$

$$\frac{\partial\, \mathrm{FPR}_U^r}{\partial \hat{\beta}} = \frac{|\,\mathrm{head}(\mathcal{U}, r)| - \mathcal{T}(r) \cdot |\mathcal{U}|}{(1-\hat{\beta})^2}, \tag{7.16}$$

$$\frac{\partial\, \mathrm{PRE}_U^r}{\partial \hat{\beta}} = \frac{\mathcal{T}(r) \cdot |\mathcal{U}|}{|\,\mathrm{head}(\mathcal{U}, r)|} \geq 0. \tag{7.17}$$

The partial derivative of TPR is exactly 0 because our approach is based on rank CDFs (that is TPR at each rank). Interestingly, the partial derivatives of FPR and precision to $\hat{\beta}$ are dependent on the value of the rank CDF $\mathcal{T}(r)$ that is being used to infer surrogate positives. Since TPR is not a function of $\hat{\beta}$ and the partial derivatives of FPR/precision to $\hat{\beta}$ are functions of $\mathcal{T}(r)$, distinct segments of an ROC/PR curve are moved differently when $\hat{\beta}$ changes, inducing a non-uniform scaling of AUC across the TPR range. Such scaling potentially changes the ordering of models based on AUC.

7.6.3 Empirical quality of the estimates

We illustrate the quality of our estimated bounds on ROC and PR curves using a model trained in a PU learning setting [BLINDED for review] on the `covtype` data set [22]. The model was evaluated on a fully labeled test set of $20,000$ positive and $20,000$ negative examples. To estimate performance, we randomly selected 5% of positive examples to serve as our labeled set and treated all other examples as unlabeled, which yields $|\mathcal{P}_L| = 1,000$, $|\mathcal{U}| = 39,000$ and $\beta \approx 49\%$. We present ROC and PR curves with bounds for $\hat{\beta} = \beta$, $\hat{\beta} = 0$, and a confidence interval $\hat{\beta}_{lo} = 0.8\beta \leq \hat{\beta} \leq \hat{\beta}_{up} = 1.2\beta$. Finally, as we have the ground truth, we present true curves as a reference.[5]

[4] We made some simplifications, the details are described in Appendix 7.B.

[5] Python code to reproduce all results (and modify the configuration) is available as supplementary material.

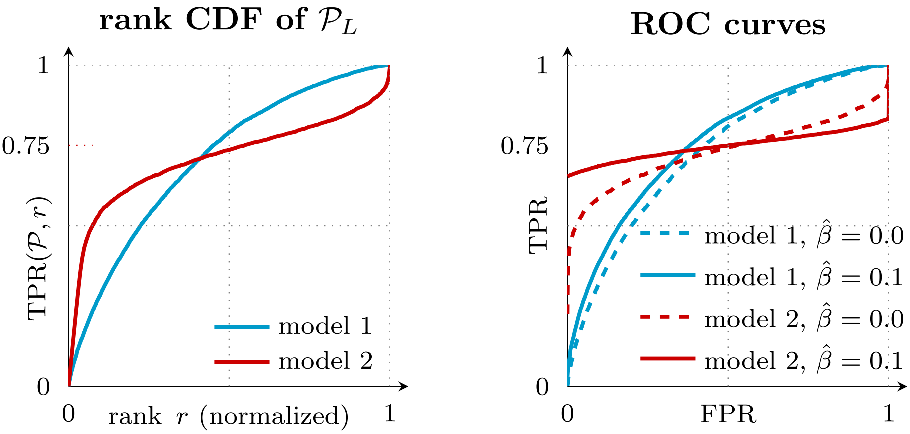


Figure 7.4: The effect of $\hat{\beta}$ on ROC curves. Setup: $|\mathcal{U}| = 45,000$, $|\mathcal{P}_L| = 5,000$.

Corresponding AUROC (best in bold):

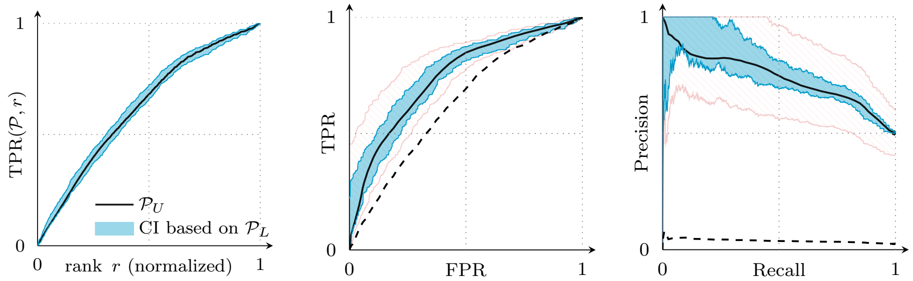
estimated $\hat{\beta}$	model 1	model 2
0.0	72.5%	**73.2%**
0.1	**75.5%**	74.7%

Figure 7.5 presents the rank CDF and estimated bounds on ROC and PR curves. Figure 7.5a shows the true rank CDF of $\mathcal{P}_U$ along with an estimated 95% CI on the rank CDF using the $\mathcal{P}_L$ via a standard bootstrap approach with 2,000 resamples. In this case, the CI contains the true rank CDF of latent positives.[6] Figures 7.5b and 7.5c show that the bounds closely approximate the true performance curves. The estimated bounds are wider in PR space than in ROC space, particularly at low recall. Note that estimated PR curves are sensitive to the estimation error in $\hat{\beta}$, as precision is directly affected by class balance, limiting their usefulness if only a rough estimate of β is available.

7.6.4 Relative importance of known negatives compared to known positives

As our approach can incorporate known negatives, a natural question is how their presence influences the estimates. In practice, a test set is of fixed size, so known negatives essentially reduce the size of the unlabeled subset, which in turn reduces the number of degrees of freedom in assigning surrogate positives.

[6] The rank CDF of $\mathcal{P}_U$ is unknown in practice, but assumed to be comparable to the rank CDF of $\mathcal{P}_L$.



(a) Rank CDF.

(b) ROC curves.

(c) PR curves.

Figure 7.5: Results for `covtype` showing rank CDF, ROC and PR curves, with $\beta \approx 49\%$. Performance curve legend:
— true curve, - - - $\hat{\beta} = 0$, 🟦 $\hat{\beta} = \beta$ and ▨ $0.8\beta \leq \hat{\beta} \leq 1.2\beta$.

Using the same setup as in Subsection 7.6.3, we varied the proportion of known positives and negatives and found known negatives provide some benefit, though this is small in practice. However, our approach can also be reversed given a large amount of negatives, that is flip known class labels, use $\bar{\beta} = 1 - \beta$ and adjust the resulting contingency tables accordingly, which can improve performance bounds. The benefits of known negatives are further discussed in Appendix 7.C.

7.7 Conclusion

We presented an approach to construct contingency tables corresponding to a lower and upper bound on FPR using only partially labeled, which enables computing many commonly used performance metrics in a semi-supervised setting. Our approach relies on knowing the fraction of latent positives in the unlabeled data, and we discussed its effect on determing the bounds and model selection. We have seen that our approach can yield good estimates in practice.

Appendix

7.A Proofs

Lemma 1. *Given a rank r and two disjoint subsets of positives $\mathcal{P}_1$ and $\mathcal{P}_2$ within an overall ranking $\mathcal{R}$. If $|\mathcal{P}_1| = |\mathcal{P}_2|$ and $\mathrm{TPR}(\mathcal{P}_1, r) > \mathrm{TPR}(\mathcal{P}_2, r)$, then $\mathrm{FPR}(\mathcal{P}_1, r) < \mathrm{FPR}(\mathcal{P}_2, r)$ (see Figure 7.A.1).*

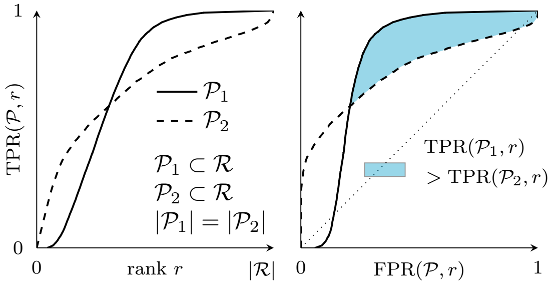


Figure 7.A.1: Illustration of Lemma 1: higher TPR at a given rank r implies lower FPR at r for two positive sets of the same size.

Proof: *The numerator of FPR is the number of false positives, this is the number of positive predictions minus the number of true positives. Via Equations* (7.2) *and* (7.3)*, this is r and $\mathrm{TPR}(\mathcal{P}, r) \cdot |\mathcal{P}|$, respectively:*

$$\mathrm{FPR}(\mathcal{P}, r) = \frac{r - \mathrm{TPR}(\mathcal{P}, r) \cdot |\mathcal{P}|}{|\mathcal{R}| - |\mathcal{P}|}. \tag{7.18}$$

Since $|\mathcal{P}_1| = |\mathcal{P}_2|$, the denominators of $\mathrm{FPR}(\mathcal{P}_1, r)$ and $\mathrm{FPR}(\mathcal{P}_2, r)$ are equal, so $\mathrm{TPR}(\mathcal{P}_1, r) > \mathrm{TPR}(\mathcal{P}_2, r) \leftrightarrow \mathrm{FPR}(\mathcal{P}_1, r) < \mathrm{FPR}(\mathcal{P}_2, r)$. ■

Lemma 2. *Given a rank r and two disjoint sets of positives $\mathcal{P}_1$ and $\mathcal{P}_2$ in a ranking $\mathcal{R}$ and $\mathcal{P}_\Omega = \mathcal{P}_1 \cup \mathcal{P}_2$. If $\mathrm{TPR}(\mathcal{P}_1, r) = t_1 < \mathrm{TPR}(\mathcal{P}_2, r) = t_2$ then $\mathrm{TPR}(\mathcal{P}_1, r) < \mathrm{TPR}(\mathcal{P}_\Omega, r) < \mathrm{TPR}(\mathcal{P}_2, r)$ (see Figure 7.A.2).*

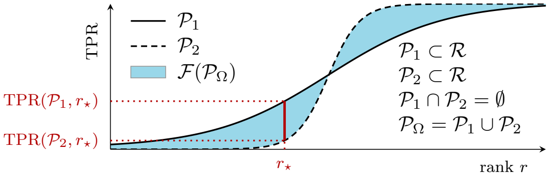


Figure 7.A.2: Illustration of Lemma 2: $\mathcal{F}(\cdot)$ denotes feasible region. The rank distribution of the union $\mathcal{P}_\Omega$ of two sets of positives $\mathcal{P}_1$ and $\mathcal{P}_2$ lies between their respective rank distributions.

Proof: *write* $\mathrm{TPR}(\mathcal{P}_\Omega, r)$ *in terms of* t_1 *and* t_2*:*

$$\mathrm{TPR}(\mathcal{P}_\Omega, r) = \frac{t_1 \cdot |\mathcal{P}_1| + t_2 \cdot |\mathcal{P}_2|}{|\mathcal{P}_1| + |\mathcal{P}_2|}. \tag{7.19}$$

since $t_1 < t_2$*, we get* $t_1 < \mathrm{TPR}(\mathcal{P}_\Omega, r) < t_2$*.* ■

Lemma 3. *Minimizing* $\mathrm{TPR}(\mathcal{P}_U^\star, r)$ *in Equation* (7.8) *of Theorem 1 ensures* $\mathrm{FPR}(\mathcal{P}_\Omega^\star, r)$ *is the greatest achievable lower bound on* $\mathrm{FPR}(\mathcal{P}_\Omega, r)$ *given* β*,* $\mathcal{T}_{ub}(r)$*,* $\mathcal{R}$ *and* $\mathcal{U}$*.*

Proof *(by contradiction): suppose another set of surrogate positives* $\mathcal{P}_U^\bullet \subset \mathcal{U}$ *exists with* $|\mathcal{P}_U^\bullet| = \beta \cdot |\mathcal{U}|$*, such that* $\mathcal{P}_U^\bullet \neq \mathcal{P}_U^\star$*, and* $\mathrm{TPR}(\mathcal{P}_U^\bullet, r) \geq \mathcal{T}_{ub}(r)$ *and for* $\mathcal{P}_\Omega^\bullet = \mathcal{P}_L \cup \mathcal{P}_U^\bullet$*:*

$$\mathrm{FPR}(\mathcal{P}_\Omega^\star, r) < \mathrm{FPR}(\mathcal{P}_\Omega^\bullet, r) \leq \mathrm{FPR}(\mathcal{P}_\Omega, r).$$

Via Corollary 1 this implies $\mathrm{TPR}(\mathcal{P}_U^\bullet, r) < \mathrm{TPR}(\mathcal{P}_U^\star, r)$*, which contradicts the definition of* $\mathcal{P}_U^\star$ *(Eq.* (7.8)*).* ■

Corollary 1. *Given a rank* r *and three sets of positives* $\mathcal{P}_A$*,* $\mathcal{P}_B$ *and* $\mathcal{P}_C$ *within a ranking* $\mathcal{R}$ *such that* $\mathcal{P}_A \cap \mathcal{P}_B = \emptyset$ *and* $\mathcal{P}_A \cap \mathcal{P}_C = \emptyset$ *and* $|\mathcal{P}_B| = |\mathcal{P}_C|$*, then*

$$\mathrm{TPR}(\mathcal{P}_B, r) = t_B < \mathrm{TPR}(\mathcal{P}_C, r) = t_C \quad \leftrightarrow \quad \mathrm{TPR}(\mathcal{P}_A \cup \mathcal{P}_B, r) < \mathrm{TPR}(\mathcal{P}_A \cup \mathcal{P}_C, r).$$

Proof: *all terms are equal for* $\mathrm{TPR}(\mathcal{P}_A \cup \mathcal{P}_B, r)$ *and* $\mathrm{TPR}(\mathcal{P}_A \cup \mathcal{P}_C, r)$ *except* $t_B < t_C$ *in Eq.* (7.19)*.*

Lemma 4. *Given two sets of positive labels* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *within an overall ranking* $\mathcal{R}$ *and a rank* r*, such that* $\mathrm{TPR}(\mathcal{P}_1, r) = \mathrm{TPR}(\mathcal{P}_2, r) = t$ *and* $|\mathcal{P}_1| > |\mathcal{P}_2|$*, then:*

$$(a) \qquad \mathrm{FPR}(\mathcal{P}_2, r) < t \rightarrow \mathrm{FPR}(\mathcal{P}_1, r) < \mathrm{FPR}(\mathcal{P}_2, r),$$

$$(b) \qquad \mathrm{FPR}(\mathcal{P}_2, r) > t \rightarrow \mathrm{FPR}(\mathcal{P}_1, r) > \mathrm{FPR}(\mathcal{P}_2, r).$$

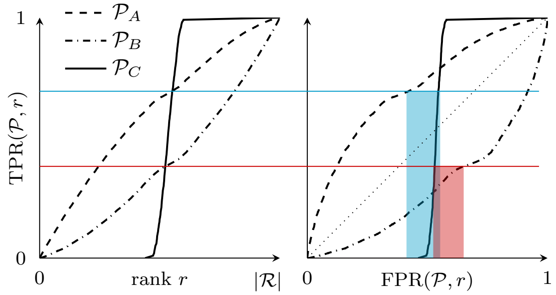


Figure 7.A.3: Illustration of Lemma 4, with $\mathcal{P}_A \subset \mathcal{R}$, $\mathcal{P}_B \subset \mathcal{R}$, $\mathcal{P}_C \subset \mathcal{R}$, $|\mathcal{P}_A| > |\mathcal{P}_C|$ and $|\mathcal{P}_B| > |\mathcal{P}_C|$. If two sets of positives $\mathcal{P}_1$ and $\mathcal{P}_2$ achieve a given TPR at the same rank r, e.g. $\text{TPR}(\mathcal{P}_1, r) = \text{TPR}(\mathcal{P}_2, r)$ and $|\mathcal{P}_1| > |\mathcal{P}_2|$ then $\text{FPR}(\mathcal{P}_1, r) < \text{FPR}(\mathcal{P}_2, r)$ if $\text{FPR}(\mathcal{P}_2, r) < \text{TPR}(\mathcal{P}_2, r)$ and otherwise $\text{FPR}(\mathcal{P}_1, r) > \text{FPR}(\mathcal{P}_2, r)$.

(a) corresponds to a ranking and cutoff that is better than random (i.e. $\text{TPR}(\mathcal{P}, r) > \text{FPR}(\mathcal{P}, r)$*) whereas (b) corresponds to a ranking and cutoff that is worse than random.*

Proof*: take the derivative of FPR to $|\mathcal{P}|$ while fixing r, based on Equation* (7.18)*:*

$$\frac{d\,\text{FPR}(\mathcal{P}, r)}{d|\mathcal{P}|} = \frac{r - t \cdot |\mathcal{R}|}{(|\mathcal{R}| - |\mathcal{P}|)^2},$$

$$= \frac{r - t \cdot |\mathcal{P}| - t \cdot |\mathcal{R} - \mathcal{P}|}{(|\mathcal{R}| - |\mathcal{P}|)^2}. \tag{7.20}$$

$r - t \cdot |\mathcal{P}|$ *is the number of negatives in the top ranking (false positives) and* $t \cdot |\mathcal{R} - \mathcal{P}|$ *is the number of false positives at* $\text{FPR} = t$*. The derivative is negative if the* FPR *is below t and vice versa, therefore if the ranking is better than random (*$\text{TPR} = t > \text{FPR}$*), increasing $|\mathcal{P}|$ leads to a lower* FPR *at rank r and vice versa.* ■

7.B Effect of $\hat{\beta}$ on contingency table entries and common performance metrics

To study the effect of imprecise estimates of β, we start by computing partial derivatives of each entry of the partial contingency table based on unlabeled instances to $\hat{\beta}$ (see Section 7.4.1). Subsequently, we will compute partial derivatives of TPR, FPR and precision to $\hat{\beta}$ to describe the effect of estimating β on (area under) ROC and PR curves.

For ease of notation, we base all subsequent calculations on $\tilde{\theta} = \hat{\beta}\mathcal{T}(r) \cdot |\mathcal{U}| \approx \theta$ which ignores the discrete effect of rounding in the real definition of θ (Eq. 7.10). We additionally assume it is possible to assign the desired amount $\tilde{\theta}$ of surrogate positives in head$(\mathcal{U}, r)$, which holds for ranks r that are not too close to the top or bottom of $\mathcal{R}$, given reasonable values of $\hat{\beta}$ and CDF bounds $\mathcal{T}(r)$.[7] If this does not hold, that is when there is clipping in Eq. 7.11, then (small) changes in $\hat{\beta}$ do not affect TP_U^r and hence the partial derivatives of *all* entries in the contingency table to $\hat{\beta}$ are effectively 0.

Given these simplifications, the partial contingency table based on unlabeled instances becomes:

$$\begin{aligned}
\mathrm{TP}_U^r &= \tilde{\theta} = \hat{\beta}\mathcal{T}(r) \cdot |\mathcal{U}| \\
\mathrm{FN}_U^r &= |\mathcal{P}_U^\star| - \mathrm{TP}_U^r = \hat{\beta} \cdot |\mathcal{U}| - \hat{\beta}\mathcal{T}(r) \cdot |\mathcal{U}| = \hat{\beta}\big(1 - \mathcal{T}(r)\big) \cdot |\mathcal{U}| \\
\mathrm{FP}_U^r &= |\operatorname{head}(\mathcal{U}, r)| - \mathrm{TP}_U^r = |\operatorname{head}(\mathcal{U}, r)| - \hat{\beta}\mathcal{T}(r) \cdot |\mathcal{U}|, \\
\mathrm{TN}_U^r &= |\mathcal{U}| - |\mathcal{P}_U^\star| - \mathrm{FP}_U^r = |\mathcal{U}| - \hat{\beta} \cdot |\mathcal{U}| - |\operatorname{head}(\mathcal{U}, r)| + \hat{\beta}\mathcal{T}(r) \cdot |\mathcal{U}|, \\
&= \big(1 - \hat{\beta} + \hat{\beta}\mathcal{T}(r)\big) \cdot |\mathcal{U}| - |\operatorname{head}(\mathcal{U}, r)|.
\end{aligned}$$

The partial derivatives of each entry of the partial contingency table then become:

$$\begin{aligned}
\frac{\partial\,\mathrm{TP}_U^r}{\partial\hat{\beta}} &= \mathcal{T}(r) \cdot |\mathcal{U}| \geq 0, & \frac{\partial\,\mathrm{FP}_U^r}{\partial\hat{\beta}} &= -\mathcal{T}(r) \cdot |\mathcal{U}| \leq 0, \\
\frac{\partial\,\mathrm{FN}_U^r}{\partial\hat{\beta}} &= \big(1 - \mathcal{T}(r)\big) \cdot |\mathcal{U}| \geq 0, & \frac{\partial\,\mathrm{TN}_U^r}{\partial\hat{\beta}} &= \big(\mathcal{T}(r) - 1\big) \cdot |\mathcal{U}| \leq 0.
\end{aligned}$$

[7] $\mathcal{T}(r)$ represents a bound on rank CDF, that is either $\mathcal{T}_{lb}(r)$ or $\mathcal{T}_{ub}(r)$ as used in the manuscript.

Partial derivatives for TPR, TPR and precision are a little more involved:

$$\begin{aligned}\frac{\partial\,\mathrm{TPR}_U^r}{\partial\hat{\beta}} &= \frac{\frac{\partial\,\mathrm{TP}_U^r}{\partial\hat{\beta}}|\mathcal{P}_U^\star| - \mathrm{TP}_U^r\,\frac{\partial|\mathcal{P}_U^\star|}{\partial\hat{\beta}}}{|\mathcal{P}_U^\star|^2} = \frac{\mathcal{T}(r)\hat{\beta}\cdot|\mathcal{U}|^2 - \mathcal{T}(r)\hat{\beta}|\mathcal{U}|^2}{\hat{\beta}^2|\mathcal{U}|^2} \\ &= \frac{\mathcal{T}(r) - \mathcal{T}(r)}{\hat{\beta}} = 0\end{aligned} \tag{7.21}$$

$$\begin{aligned}\frac{\partial\,\mathrm{FPR}_U^r}{\partial\hat{\beta}} &= \frac{\frac{\partial\,\mathrm{FP}_U^r}{\partial\hat{\beta}}\cdot(|\mathcal{U}| - |\mathcal{P}_U^\star|) - \mathrm{FP}_U^r\,\frac{\partial(|\mathcal{U}|-|\mathcal{P}_U^\star|)}{\partial\hat{\beta}}}{(|\mathcal{U}| - |\mathcal{P}_U^\star|)^2} \\ &= \frac{-\mathcal{T}(r)\cdot|\mathcal{U}|\cdot(|\mathcal{U}| - |\mathcal{P}_U^\star|) + \mathrm{FP}_U^r\cdot|\mathcal{U}|}{(|\mathcal{U}| - |\mathcal{P}_U^\star|)^2} \\ &= \frac{-\mathcal{T}(r)(1-\hat{\beta})\cdot|\mathcal{U}|^2 + \mathrm{FP}_U^r\cdot|\mathcal{U}|}{(1-\hat{\beta})^2\cdot|\mathcal{U}|^2} \\ &= \frac{-\mathcal{T}(r)}{1-\hat{\beta}} + \frac{(|\operatorname{head}(\mathcal{U},r)| - \hat{\beta}\mathcal{T}(r)\cdot|\mathcal{U}|)\cdot|\mathcal{U}|}{(1-\hat{\beta})^2\cdot|\mathcal{U}|^2} \\ &= \frac{-\mathcal{T}(r)}{(1-\hat{\beta})^2} + \frac{|\operatorname{head}(\mathcal{U},r)|}{(1-\hat{\beta})^2\cdot|\mathcal{U}|} = \frac{|\operatorname{head}(\mathcal{U},r)| - \mathcal{T}(r)\cdot|\mathcal{U}|}{(1-\hat{\beta})^2}\end{aligned} \tag{7.22}$$

$$\begin{aligned}\frac{\partial\,\mathrm{PRE}_U^r}{\partial\hat{\beta}} &= \frac{\frac{\partial\,\mathrm{TP}_U^r}{\partial\hat{\beta}}\cdot(\mathrm{TP}_U^r + \mathrm{FP}_U^r) - \mathrm{TP}_U^r\,\frac{\partial(\mathrm{TP}_U^r + \mathrm{FP}_U^r)}{\partial\hat{\beta}}}{(\mathrm{TP}_U^r + \mathrm{FP}_U^r)^2} \\ &= \frac{\mathcal{T}(r)\cdot|\mathcal{U}|\cdot(\mathrm{TP}_U^r + \mathrm{FP}_U^r)}{(\mathrm{TP}_U^r + \mathrm{FP}_U^r)^2} \\ &= \frac{\mathcal{T}(r)\cdot|\mathcal{U}|}{(\mathrm{TP}_U^r + \mathrm{FP}_U^r)} = \frac{\mathcal{T}(r)\cdot|\mathcal{U}|}{|\operatorname{head}(\mathcal{U},r)|} \geq 0\end{aligned} \tag{7.23}$$

Both $\partial\,\mathrm{FPR}_U^r\,/\partial\hat{\beta}$ and $\partial\,\mathrm{PRE}_U^r\,/\partial\hat{\beta}$ are a function of $\mathcal{T}(r)$, while $\partial\,\mathrm{FPR}_U^r\,/\partial\hat{\beta} = 0$. This implies that the ordering of rankings in terms of area under the ROC curve can change when the estimate of β changes, as proven by example in Figure 7.4.

7.C The effect of the fraction of known positives, known negatives and $\hat{\beta}$

Known negatives can be incorporated in our approach as described in Section 7.4.1. Given a fixed ranking $\mathcal{R}$, having known negatives essentially reduces the size of the unlabeled subset $\mathcal{U}$, which in turn reduces the number of degrees of freedom in assigning surrogate positives. As such, known negatives provide some benefit, though this is small in practice. Table 7.C.1 illustrates the effect of increasing amounts of known positives and known negatives: known positives significantly tighten bounds on AUROC, while known negatives only do so marginally (cfr. bounds with 10% known positives and 40/60/80% known negatives).

However, when the number of known negatives is large, it may be useful to reverse our approach, i.e., start from the rank distribution of known negatives. To do so, we can essentially flip all known class labels, use $\bar{\beta} = 1 - \beta$ and adjust the resulting contingency tables accordingly.

Table 7.C.2 shows bounds when based on known positives or known negatives (whichever are tightest). It is important to see that $|\mathcal{N}_L| > |\mathcal{P}_L|$ does not guarantee that performance bounds based on known negatives are tighter, because β also affects the bounds. When computing performance bounds based on known negatives, overestimating $\hat{\beta}$ leads to underestimated bounds (since we use $\bar{\beta} = 1 - \hat{\beta}$) and vice versa. The effect of errors in $\hat{\beta}$ is opposite in bounds based on $\mathcal{N}_L$.

Hence, bounds on performance metrics can be computed based primarily on known positives $\mathcal{P}_L$ *or* known negatives $\mathcal{N}_L$. The width of the bounds depends on the combination of $|\mathcal{P}_L|$ (or $|\mathcal{N}_L|$) and β (or $\bar{\beta}$) in a nontrivial way: depending on β, it is possible to obtain wider bounds based on known negatives, even if $|\mathcal{N}_L| > |\mathcal{P}_L|$ (or vice versa). In practice, we can estimate metrics based on $\mathcal{P}_L$ and $\mathcal{N}_L$ separately and then use whichever yields the tightest bounds, as shown in Table 7.C.2.

configuration			bounds on area under the ROC curve (true AUROC=76.8%)		
$\frac{\lvert\mathcal{P}_L\rvert}{\lvert\mathcal{P}_\Omega\rvert}$	$\frac{\lvert\mathcal{N}_L\rvert}{\lvert\mathcal{N}_\Omega\rvert}$	β	$\hat{\beta}\ /\ \beta = 0.8$	$\hat{\beta}\ /\ \beta = 1.0$	$\hat{\beta}\ /\ \beta = 1.2$

Table 7.C.1: Estimated bounds on AUROC under different configurations. The total data set comprises $2,000$ positives and $10,000$ negatives. We varied the fraction of known positives and known negatives, which also implies changing β. All entries in the table are in percentages. We used three estimates for $\hat{\beta}$, namely an underestimate, the correct value and an overestimate (left to right). Legend: - - - true AUROC, 🟦 bounds based on known positives.

configuration			bounds on area under the ROC curve (true AUROC=76.8%)		
$\frac{\lvert\mathcal{P}_L\rvert}{\lvert\mathcal{P}_\Omega\rvert}$	$\frac{\lvert\mathcal{N}_L\rvert}{\lvert\mathcal{N}_\Omega\rvert}$	β	$\hat{\beta}\ /\ \beta = 0.8$	$\hat{\beta}\ /\ \beta = 1.0$	$\hat{\beta}\ /\ \beta = 1.2$

Table 7.C.2: Estimated bounds on AUROC under different configurations. The total data set comprises $2,000$ positives and $10,000$ negatives. We varied the fraction of known positives and known negatives, which also implies changing β. All entries in the table are in percentages. We used three estimates for $\hat{\beta}$, namely an underestimate, the correct value and an overestimate (left to right). In this table, we computed bounds based on known positives and known negatives (separately) and report the tightest confidence interval each time. Legend: - - - true AUROC, bounds based on 🟦 known positives and 🟥 known negatives.

Chapter 8

Building Classifiers to Predict the Start of Glucose-Lowering Pharmacotherapy Using Belgian Health Expenditure Data

This chapter has been submitted to:
Claesen, M., De Smet, F., Gillard, P., Mathieu, C.& De Moor, B. (2015). **Building Classifiers to Predict the Start of Glucose-Lowering Pharmacotherapy Using Belgian Health Expenditure Data**. *Journal of Machine Learning Research: special issue on Learning from Electronic Health Data.*

Abstract

Early diagnosis is important for type 2 diabetes (T2D) to improve patient prognosis, prevent complications and reduce long-term treatment costs. We present a novel risk profiling approach based exclusively on health expenditure data that is available to Belgian mutual health insurers. We used expenditure data related to drug purchases and medical provisions to construct models that predict whether a patient will start glucose-lowering pharmacotherapy in the coming years, based on that patient's recent medical expenditure history. The design and implementation of the modeling strategy are discussed in detail and several learning methods are benchmarked for our application. Our best performing model obtains between 74.9% and 76.8% area under the ROC curve, which is comparable to state-of-the-art risk prediction approaches for T2D based on questionnaires. In contrast to other methods, our approach can be implemented on a population-wide scale at virtually no extra operational cost. Possibly, our approach can be further improved by additional information about some risk factors of T2D that is unavailable in health expenditure data.

8.1 Introduction

Type 2 diabetes mellitus (T2D) is a chronic metabolic disorder characterized by hyperglycemia and is considered one of the main threats to human health [214]. In developed countries, T2D makes up about 85% of diabetes mellitus patients and occurs when either insufficient insulin is produced, the body becomes resistant to insulin or both [204]. Prediabetes and less severe cases of T2D are initially managed by lifestyle changes, specifically increasing physical exercise, dietary change and smoking cessation [188, 60, 6]. If this yields insufficient glycemic control, pharmacotherapy with glucose-lowering agents (GLAs) like metformin or insulin is started [189, 6].

Several studies have indicated that one third to one half of T2D patients are undiagnosed [92, 113, 166]. Additionally, patients often remain undiagnosed for extended periods of time, with average diagnose-free intervals ranging from 4 to 7 years [94]. The prognosis of untreated patients can deteriorate rapidly as prolonged hyperglycemia can cause serious damage to many of the body's systems. Timely diagnosis of T2D proves challenging in contemporary medicine, as many patients already present signs of complications of the disease at the time of clinical diagnosis of T2D [93, 159, 116, 11, 91, 100].

Earlier diagnosis and subsequent treatment is believed to prevent or delay complications and improve prognosis [152, 68]. When impaired glucose tolerance is diagnosed early, initial treatment can often be limited to lifestyle changes [149, 188, 60]. Compared to pharmacotherapy, lifestyle changes are simple, fully manageable by the patient and far less likely to cause serious treatment-induced complications like hypoglycemia [175, 212]. Complementary to health benefits, early diagnosis of T2D poses a health economical advantage, as patients that do not require acute or intensive long-term treatment are far less demanding on the health care system.

Universal screening for T2D is cost-prohibitive [200, 68], but many organizations advise opportunistic screening of high-risk subgroups [204, 3, 68, 6]. Several risk profiling strategies have been developed to aid in the timely diagnosis of T2D [9, 183, 124, 134, 38, 96, 172]. Risk profiling is typically done by assessing some of the key risk factors for T2D, which include obesity [136], genetic predisposal [176, 103], lifestyle [161] and various clinical parameters. Existing risk profiling approaches are implemented via questionnaires, potentially augmented with clinical information that is available to the patient's general practitionner [84, 182, 124, 79, 171, 96]. Commonly required information includes BMI, family history, exercise and smoking habits and various clinical parameters.

In this work, we present an alternative approach for risk profiling which only requires data that is already available to Belgian mutual health insurers. This work was done in collaboration with the National Alliance of Christian Mutualities (NACM). NACM is the largest Belgian mutual health insurer with over four million members. Our approach does not require any questionnaires or additional clinical information and predicts whether a patient will start taking GLAs in the next few years. Interestingly, our approach works well despite the fact that Belgian health insurer data contains little direct information regarding key risk factors of T2D, that is weight, lifestyle and family history are all unavailable.

8.2 Existing Type 2 Diabetes Risk Profiling Approaches

The Cambridge Risk Score (CRS) was developed to assess the probability of undiagnosed T2D based on data that is routinely available in primary care records, including age, sex, medication use, family history of diabetes, BMI and smoking status [84], The CRS has been shown to be useful on multiple occasions [84, 151, 182], though its AUC seems to depend heavily on the population in which it is used, ranging between 67% [182] and 80% [84]. The information used in the CRS is comparable to another approach which obtained AUCs ranging between 70% and 78% [9].

The FINDRISC score is based on a 10-year follow-up using age, BMI, waist circumference, history of antihypertensive drugs and high blood glucose, physical activity and diet with reported AUCs of 85% and 87% in predicting drug-treated diabetes [124]. The strongest reported predictors in this study were BMI, waist circumference, history of high blood glucose and physical activity. Glümer et al. [79] developed a risk score based on age, sex, BMI, known hypertension, physical activity and family history of diabetes with AUC ranging from 72% to 87.6%. The German diabetes risk score reached AUCs ranging from 75% to 83% on validation data and is based on age, waist circumference, height, history of hypertension, physical activity, smoking, and diet [171].

Heikes et al. [96] developed a decision tree for risk prediction achieving 82% AUC in a cross-validation setting, based on weight, age, family history and various clinical parameters. Various other approaches based on routine clinical information have demonstrated similarly accurate predictions of type 2 diabetes [183, 134].

8.3 Health Expenditure Data

The Belgian health care insurance is a broad solidarity-based form of social insurance. Mutual health insurers such as NACM are the legally-appointed bodies for managing and providing the Belgian compulsory health care and disability insurance, among other things. To implement their operations, Belgian mutual health insurers dispose of large databases containing health expenditure records of all their respective members.

These expenditure records hold all financial reimbursements of drugs, procedures and contacts with health care professionals. Each record comprises a timestamp, financial details and a description of the claim. The financial aspect is irrelevant from a medical point of view, but the type of resource-use as indicated by the description can contain medical information about the patient. These types belong to one of two main categories:

1. **Drug purchases** are recorded per package. The coding of packages contains information about the active substances in the drug along with the volume of the package.

2. **Medical provisions** are identified by a national encoding along with an identifier of the associated medical caregiver. Each provision has a distinct code number.

In addition to resource-use data, some biographical information is available about each patient including age, gender, place of residence and social parameters. In the remainder of this Section we will elaborate on expenditure records related to drugs and provisions as used in our models. Subsequently we will briefly summarize the main strengths and limitations of using health expenditure data for predictive modeling.

8.3.1 Records Related to Drug Purchases

Expenditure records concerning drug purchases contain information about the active substances in the drug and the purchased volume. We mapped all active substances onto the anatomical therapeutic chemical (ATC) classification system maintained by the WHO [202]. The ATC classification system divides active substances into different groups based on the organ or system on which they act and their therapeutic, pharmacological and chemical properties. Each drug is classified in groups at 5 levels in the ATC hierarchy: fourteen main groups (1st level), pharmacological/therapeutic subgroups (2nd level), chemical subgroups (3rd and 4th level) and the chemical substance (5th level).

After mapping records onto the ATC classification system, a patient's medication history consists of specific ATC codes (5th level) along with the associated number of defined daily doses (DDD). In the period of interest, purchases of 4,580 distinct active substances were recorded in the NACM database. Table 8.1 shows an example of the classification of active substance on all levels in the ATC system.

level	ATC code	description
1	A	alimentary tract and metabolism
2	A10	drugs used in diabetes
3	A10B	blood glucose lowering drugs, excluding insulins
4	A10BA	biguanides
5	A10BA02	metformin

Table 8.1: Example of the ATC classification system: classification of metformin per level.

8.3.2 Records Related to Medical Provisions

Expenditure records concerning medical provisions can be considered tuples containing time-stamped identifiers of the patient, physician and medical provision. A single patient-physician interaction may yield multiple such records, one for each specific provision that occurred.

In the Belgian health care system, medical provisions are encoded via the Belgian nomenclature of medical provisions [195], which is maintained by the National Institute for Health and Disability Insurance (NIHDI).[1] This nomenclature is an unstructured list of unique codes (numbers) for each provision that is being refunded. Nomenclature numbers are added when new provisions are defined or when revisions are made. A single provision may correspond to multiple numbers for various reasons.

8.3.3 Advantages of Health Expenditure Data

The key benefit of expenditure databases is that they centralize structured medical information across all medical stakeholders to yield a comprehensive, longitudinal overview of each patient's medical history. Other health data sources are commonly fragmented, e.g. medical records maintained by the patient's general practitioner or hospital often contain only a subset of the patient's medical history. This fragmentation hampers the identification of patterns that may indicate elevated risk for diseases like type 2 diabetes. The NACM database comprises claims records of over four million Belgians, which enables complex modeling. Additionally, claims data have few omissions due to the financial incentive for patients and medical stakeholders (e.g., hospitals) to claim refunds. While other health data sources may contain more detailed information, the strength of NACM's data is in its volume, both in terms of number of patients and the amount of information that is recorded per individual. Finally, as most people tend to stay affiliated with the same mutual health insurer, their expenditure records provide long-term information.

8.3.4 Limitations of Health Expenditure Data

Belgian health expenditure data is strictly limited to what is required for mutual health insurers to implement their operations, which are mainly administrative in nature. Detailed health information such as diagnoses and test results are not directly available. In some other countries, health insurers dispose of more detailed information, such as ICD-10 codes which include diagnoses and symptoms [205]. Including such information is out of scope of this work as we focus exclusively on data that is already available to Belgian mutual health insurers. Biographical information about patients does not contain

[1] The website of NIHDI is available at `http://www.riziv.fgov.be`.

direct information about some important risk factors such as lifestyle, family history and BMI, though this may be partially embedded indirectly in medical resource-use.

8.4 Methods

In this Section we define the prediction task and describe all its aspects: the overall setup (Section 8.4.1), the data and its representation (Section 8.4.2) and the learning algorithms (Section 8.4.3). Briefly, our aim is to predict which patients will start glucose-lowering pharmacotherapy within the next 4 years, based on expenditure records of the previous 4 years.

Our key hypothesis is that patients with increased risk for T2D or those that are already afflicted but not diagnosed have a different medical expenditure history than patients without impaired glycemic control. We essentially use the start of GLA therapy as a proxy for diagnosis of (advanced) type 2 diabetes. This is reasonable since most patients that start GLA therapy above 40 years old have T2D [204].

We posed this task as a binary classification problem. Our classifiers produce a numeric level of confidence that a given patient will start glucose-lowering pharmacotherapy. When predicting a population, the outputs can be used to rank patients according to decreasing confidence that the patients will start glucose-lowering therapy. Highly ranked patients represent a high-risk subgroup which can be targetted for clinical screening. Briefly, we used nested cross-validation to obtain unbiased estimates of the predictive performance of each vectorization and learning approach. Predictive performance of all models was quantified via (area under) receiver operating characteristic (ROC) curves.

Data Our work is based on a subset of the expenditure records of NACM. All data extractions and analyses were performed at the Medical Management Department of the NACM under supervision of the Chief Medical Officer. The other research partners received no personally identifiable information (including small cells) from NACM. The patient selection protocol and vector representations are described in detail in Section 8.4.2.

Class definitions The positive class was defined as patients that require GLAs for long-term glycemic control.[2] The negative class is then defined as patients

[2] GLAs are defined as any drug in ATC category `A10`, which includes metformin, sulfonylurea and insulin.

that do not need GLAs. Expenditure records related to GLAs were used to identify a set of known positives. However, the absence of such records in a patient's resource use history is not proof that this patient has no need for GLAs. This subtle difference is crucial, because it is well known that patients with impaired glycemic control or T2D often remain undiagnosed and hence untreated for a very long time [92, 113, 6]. As we cannot identify negatives, we had to build models from positive and unlabeled data.

PU learning Learning binary classifiers from positive and unlabeled data (PU learning) is a well-studied branch of semi-supervised learning [121, 67, 138, 47]. PU learning is more challenging than fully supervised binary classification, since it requires special learning approaches and quality metrics for hyperparameter optimization that account for the lack of known negatives. We benchmarked three PU learning methods, which are discussed in more detail in Section 8.4.3.

Software The entire data analysis pipeline was implemented using open-source software. For general data transformations and preprocessing we used *SciPy* and *NumPy* [106, 197]. The learning algorithms we used are available in *scikit-learn* and *EnsembleSVM* [154, 46] . Finally, we used *Optunity* for automated hyperparameter optimization [44].

8.4.1 Experimental Setup

We gathered all expenditure records during the 4-year interval of 2008 up to 2012. The selection protocol and representations of patients' medical resource-use are discussed in detail in Section 8.4.2. All vector representations of patients include age (in years), an indicator variable for gender and positive entries related to the patient's medical resource-use. A patient vector $\mathbf{p}$ can be written in the following general form, where d_{meds} and d_{provs} denote the number of features in the vectorization of medication and provision use, respectively:

$$\mathbf{p} \in \mathbb{R}_+^{2+d_{\text{meds}}+d_{\text{provs}}} = \begin{bmatrix} age & gender & medication & provisions \\ \mathbb{R}_+ & \{0,1\} & \mathbb{R}_+^{d_{\text{meds}}} & \mathbb{R}_+^{d_{\text{provs}}} \end{bmatrix}. \tag{8.1}$$

In Sections 8.4.2 and 8.4.2 we explain how records related to medication purchases and provisions were represented in vector form. All entries in the vector representations were consistently normalized to the interval $[0,1]$ by dividing feature-wise by the 99^{th} percentile and subsequently clipping where necessary. These normalized vector representations are used as inputs for the learning algorithms described in Section 8.4.3.

Figure 8.1 summarizes the full machine learning pipeline, which starts from expenditure records and ends with models to predict whether a patient will start glucose-lowering pharmacotherapy along with an estimate of their generalization performance. We used nested cross-validation to estimate generalization performance of different learning configurations [198]. The outer 3-fold cross-validation is used to estimate generalization performance of the full learning approach. Internally, twice iterated 10-fold cross-validation was used to find optimal hyperparameters for every learning method.

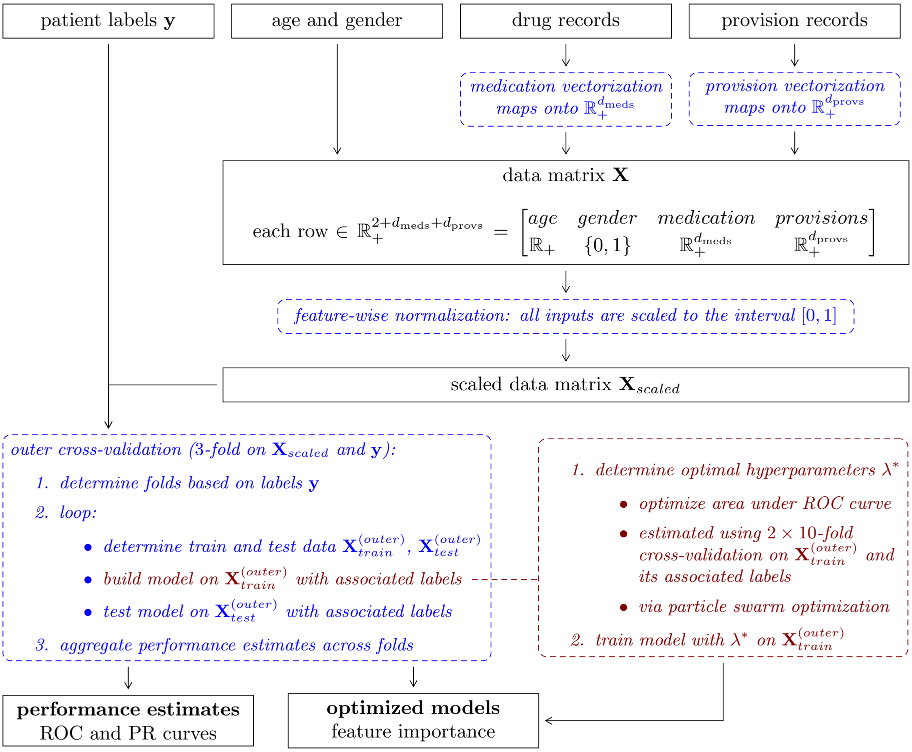

patient labels $\mathbf{y}$
age and gender
drug records
provision records
medication vectorization maps onto $\mathbb{R}_+^{d_{\text{meds}}}$
provision vectorization maps onto $\mathbb{R}_+^{d_{\text{provs}}}$
data matrix $\mathbf{X}$
each row $\in \mathbb{R}_+^{2+d_{\text{meds}}+d_{\text{provs}}} = \begin{bmatrix} age & gender & medication & provisions \\ \mathbb{R}_+ & \{0,1\} & \mathbb{R}_+^{d_{\text{meds}}} & \mathbb{R}_+^{d_{\text{provs}}} \end{bmatrix}$
feature-wise normalization: all inputs are scaled to the interval $[0,1]$
scaled data matrix $\mathbf{X}_{scaled}$
outer cross-validation (3-fold on $\mathbf{X}_{scaled}$ and $\mathbf{y}$):
1. determine folds based on labels $\mathbf{y}$
2. loop:
• determine train and test data $\mathbf{X}_{train}^{(outer)}$, $\mathbf{X}_{test}^{(outer)}$
• build model on $\mathbf{X}_{train}^{(outer)}$ with associated labels
• test model on $\mathbf{X}_{test}^{(outer)}$ with associated labels
3. aggregate performance estimates across folds
1. determine optimal hyperparameters λ^*
• optimize area under ROC curve
• estimated using 2×10-fold cross-validation on $\mathbf{X}_{train}^{(outer)}$ and its associated labels
• via particle swarm optimization
2. train model with λ^* on $\mathbf{X}_{train}^{(outer)}$
performance estimates
ROC and PR curves
optimized models
feature importance

Figure 8.1: Overview of the full learning approach: data set vectorization, normalization and the nested cross-validation setup. Per iteration, hyperparameter optimization and model training is done based exclusively on $\mathbf{X}_{train}^{(outer)}$.

Model evaluation Models are compared based on area under the ROC curve. ROC curves visualize a classifier's performance spectrum by depicting its true positive rate (TPR)[3] as a function of its false positive rate (FPR)[4] while varying the decision threshold to decide on positives. Area under the ROC curve (AUROC) is a useful summary statistic of a classifier's performance. AUROC is equal to the probability that the classifier ranks a random positive higher than a random negative and is known to be equivalent to the Wilcoxon test of ranks [89].

Hyperparameter search We used Optunity's particle swarm optimizer to identify suitable hyperparameters for each approach based on the given training set as defined by the outer cross-validation procedure [44]. Every tuple of hyperparameters was evaluated using twice iterated 10-fold cross-validation on the training set. Per technique, the hyperparameters that maximized cross-validated area under the ROC curve were selected and used to train a model on the full training set.

Computing ROC curves Full label knowledge is required to compute ROC curves. In previous work, we introduced a method to compute bounds on ROC curves based on positive and unlabeled data [41]. Briefly, it is based on the positions of known positives in a ranking produced by a given classifier and requires two things:

- The rank distributions of labeled and latent positives must be comparable. This holds when known and latent positives follow the same distribution in input space (ie. the vector representation of patients). This is a fair assumption in our application, since we specifically ignore records after the start of glucose-lowering pharmacotherapy while identifying the set of positives (see Section 8.4.2), so the medication regimen of known positives has not yet diverged from the regimen of untreated patients.
- An estimate $\hat{\beta}$ of the fraction of latent positives in the unlabeled set is needed, that is the fraction of members that have never used GLAs but are likely to start glucose-lowering pharmacotherapy. In the period 2010–2014 roughly 8% of members of NACM aged 40 or higher started using GLAs. Underestimating $\hat{\beta}$ results in an underestimated ROC curve and vice versa [41]. We opted to be conservative and used $\hat{\beta}_{lo} = 5\%$ to estimate lower bounds and $\hat{\beta}_{up} = 10\%$ for upper bounds.

[3] TPR measures the fraction of true positives that are correctly identified by the classifier.

[4] FPR measures the fraction of true negatives that are incorrectly identified by the classifier.

We consistently used the *lower* bounds for hyperparameter search. All our performance reports contain lower and upper bounds, based on $\hat{\beta}_{lo}$ and $\hat{\beta}_{up}$, respectively.

Diagnosing overfitting In addition to measuring performance, we diagnosed overfitting via the concept of rank distributions as defined by Claesen et al. [41]. The rank distribution of a subset of test instances is defined as the distribution of the positions of these test instances in a ranking of the full test set based on a model's predicted decision values. We diagnose overfitting based on the rank distributions of known positive training instances ($\mathcal{P}_{train}$) and known positives in the independent test fold ($\mathcal{P}_{test}$) after predicting the full data set. If the model overfits, the rank distribution of $\mathcal{P}_{train}$ is inconsistent with the rank distribution of $\mathcal{P}_{test}$. Specifically, ranks in $\mathcal{P}_{test}$ are worse than those in $\mathcal{P}_{train}$ when the model overfits. This can be quantified via the Mann-Whitney U test [132] based on ranks of $\mathcal{P}_{train}$ and $\mathcal{P}_{test}$ after predicting the full data set (that is all outer folds). The Mann-Whitney U test is expected to yield a non-significant result when the rank distributions of $\mathcal{P}_{train}$ and $\mathcal{P}_{test}$ are comparable. We report the average p-values of the test across outer cross-validation folds for each model (low p-values indicate overfitting).

8.4.2 Data Set Construction

We constructed a data set containing records of patients born before 1973 (e.g. 40 or more years old in 2012). Patients with records of glucose-lowering agents (GLAs) during less than 30 days were discarded. Patients with records of glucose-lowering therapy prior to 2012 were discarded. Patients that joined NACM after 2005 were also discarded, as we cannot determine whether these patients used GLAs in the recent past.

All patients that started glucose-lowering pharmacotherapy in 2012 or later are included as known positives ($n = 31,066$), along with unlabeled patients that were sampled at random from the remaining NACM members ($n = 79,243$). Known positives have a minimum of 30 days between the first and last purchase of GLAs to avoid contaminating the data set with false positives, for instance due to insulin use in surgical and medical ICUs [194, 193]. It must be noted that some false positives remain, that is patients that use GLAs but not for glycemic control.

In Sections 8.4.2 and 8.4.2 we describe the vector representations of records regarding medication and medical provisions, respectively.

Representation of Medication Records

The simplest way to represent medication purchases during a time interval is by having one input dimension per active substance (level 5 ATC codes) and counting the purchased volume in terms of DDDs. This representation is easy to construct but fails to capture any similarity between active substances, such as the system or organ on which they act.

Imposing structure We can directly use the hierarchical structure of the ATC system to define a measure of similarity between drugs. To impose structure between drugs we included input dimensions related to more generic levels of the ATC hierarchy (levels 1 to 4). On more generic levels we summed all DDD counts of active substances per category (level 5). This redundancy allowed us to express similarity between different active substances with a standard inner product. By normalizing every feature to the unit interval, we obtained the desired effect that patients with comparable drug use on ATC level 5 are more similar than patients that only share coefficients on more generic levels. Figure 8.2 illustrates this vector representation of trees and the effect of normalization.

Summary All vectorizations related to drug purchases are described in Table 8.1.

vectorization	description	d_{meds}
ATC 5	counts of DDDs per medication class in ATC level 5	4,580
ATC 1–4	counts of DDDs per medication class in ATC levels 1–4	1,257
ATC 1–5	counts of DDDs per medication class in ATC levels 1–5	5,837

Table 8.1: Summary of vectorization schemes used for records of drug purchases.

Representation of Provision Records

When considering a specific time period, we can describe records by a (sparse) three-dimensional tensor containing frequency counts as illustrated in Figure 8.3. We filtered all provisions with a description containing *diabetes*, *insulin* and *glucose* and provisions not recorded with a physician identifier. After filtering, 5,799 distinct provision codes remain (denoted by $\#provisions$).

Each patient is modelled by a histogram of their provisions in the period of interest. This essentially means we compute the sum over the *physician-*

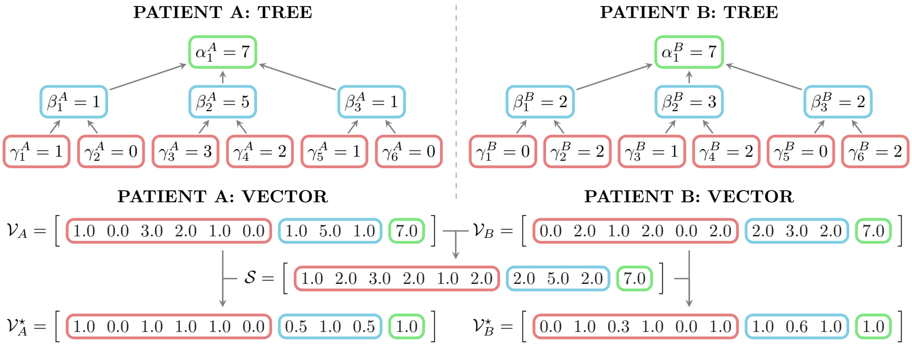


Figure 8.2: Visualization and vectorization of trees. In the tree representation, the value of internal nodes is the sum of the values of its children. The unnormalized vector representations $\mathcal{V}_A$ and $\mathcal{V}_B$ contain the values per node in the tree representation in some fixed order. Inner products between unnormalized representations $\mathcal{V}_A$ and $\mathcal{V}_B$ are mainly influenced by the top level nodes, since those have the largest value by construction. This undesirable effect can be fixed through feature-wise scaling. The scaling vector $\mathcal{S}$ was constructed using node-wise maxima. The normalized vector representations $\mathcal{V}_A^\star$ and $\mathcal{V}_B^\star$ are obtained by dividing the vector representations $(\mathcal{V}_A, \mathcal{V}_B)$ element-wise by entries in the scaling vector $\mathcal{S}$. $\mathcal{V}_A^\star$ and $\mathcal{V}_B^\star$ are used as input to classifiers in the remainder of this work. As desired, the inner product of normalized vector representations is increasingly influenced by similarities at higher depths in the tree representations.

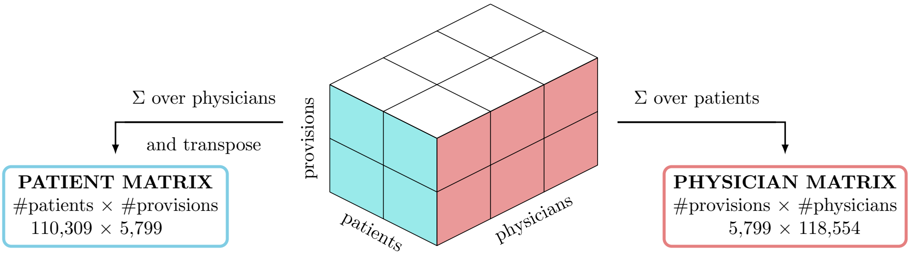


Figure 8.3: Tensor formulation of medical provisions with three components: patients, physicians and provisions. Each entry in the tensor is the frequency of the given tuple. This provision tensor is very sparse. The patient matrix is obtained by summing counts over all physicians (transposed). The physician matrix is obtained by summing counts over all patients. These matrices capture complementary information.

component of the tensor representation to obtain a matrix, in which rows and columns represent patients and provisions, respectively. Unfortunately, the encoding of provisions has no medically relevant structure in contrast to the ATC hierarchy for drugs as discussed in Section 8.4.2.

Imposing structure In order to define a reasonable similarity measure between patients, we first had to impose a structure onto the nomenclature that captures similarity between provisions. To structure provisions, we should not use information originating from the patient matrix, as this may cause information leaks (since the patient matrix is used directly in our models for prediction). Instead, we used the complementary physician matrix as a basis to define similarity between provisions, which essentially serves as a proxy for the medical specializations to which each provision belongs.

First, we computed cosine similarity between provisions based on the physician matrix. We used cosine similarity because it is known to work well for text mining with bag-of-words representations, which is comparable to our use case as it also features sparse, high dimensional input spaces. The cosine similarity κ_{cos} between two row vectors $\mathbf{u}$ and $\mathbf{v}$ is defined as:

$$\kappa_{cos}(\mathbf{u},\mathbf{v}) = \frac{\langle \mathbf{u},\mathbf{v}\rangle}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} = \frac{\mathbf{u}\mathbf{v}^T}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}. \tag{8.2}$$

Using cosine similarity we can construct a pair-wise similarity matrix $\mathbf{S}_{prov}$ between provisions based on the rows of the physician matrix $\mathbf{x}_i, i = 1..\#provisions$:

$$\mathbf{S}_{prov} = \big(\kappa_{cos}(\mathbf{x}_i,\mathbf{x}_j)\big)_{ij} \in \mathbb{R}^{\#provisions \times \#provisions}. \tag{8.3}$$

$\mathbf{S}_{prov}$ expresses similarity between provision codes based on the physicians that provide them and can be regarded as a proxy for the medical subdomain each provision frequently occurs in. In our context, its entries range from 0 (completely orthogonal) to +1 (exact similarity). To impose sparsity we set all entries of $\mathbf{S}_{prov}$ below 0.05 to 0. Its structure is visualized in Figure 8.4, which clearly indicates that our approach successfully identifies some coherent groups of provisions.

Finally, the structured representation of provisions $\mathbf{P}_{struct}$ is defined as the matrix product between the patient matrix $\mathbf{P}_{flat}$ and the provision similarity matrix $\mathbf{S}_{prov}$:

$$\mathbf{P}_{struct} = \mathbf{P}_{flat} \times \mathbf{S}_{prov} \in \mathbb{R}^{\#patients \times \#provisions}. \tag{8.4}$$

$\mathbf{P}_{struct}$ approximately captures which provisions occur in a patient's history with redundancy based on medical specializations.

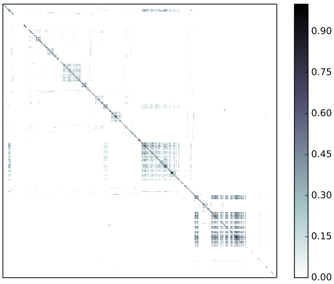


Figure 8.4: Structure of the provision similarity matrix $\mathbf{S}_{prov}$ based on providing physicians.

Summary All vectorizations related to medical provisions are described in Table 8.2.

vectorization	symbol	description	d_{provs}
PROVS FLAT	$\mathbf{P}_{flat}$	entries taken from the patient matrix	$5,799$
PROVS STRUCT	$\mathbf{P}_{struct}$	captures similarity between provisions	$5,799$
PROVS BOTH	$\mathbf{P}_{flat} \mid \mathbf{P}_{struct}$	concatenation of flat & structured	$11,598$

Table 8.2: Summary of vectorization schemes used for records of medical provisions.

8.4.3 Learning Methods

Having only positive and unlabeled data (PU learning) presents additional challenges for learning algorithms. Two broad classes of approaches exist to tackle these problems: (i) two-phase methods that first attempt to identify likely negatives from the unlabeled set and then train a supervised model on the positives and inferred negatives [129, 209] and (ii) approaches that treat the unlabeled set as negatives with label noise [67, 121, 138, 47].

We have tested three approaches from the latter category in this work, namely class-weighted SVM [128], bagging SVM [138] and the robust ensemble of SVM models [47]. All of these approaches are based on support vector machines. We used the linear kernel on vector representations of patients as described in Section 8.4.2.[5] We will briefly introduce each method in the following subsections.

Class-weighted SVM

Class-weighted SVM (CWSVM) uses a misclassification penalty per class. CWSVM was first applied in a PU learning context by Liu et al. [128], by considering the unlabeled set to be negative with noise on its labels. A CWSVM is trained to distinguish positives ($\mathcal{P}$) from unlabeled instances ($\mathcal{U}$), leading to the following optimization problem:

$$\min_{\alpha,\xi,b} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) + C_{\mathcal{P}}\sum_{i\in\mathcal{P}}\xi_i + C_{\mathcal{U}}\sum_{i\in\mathcal{U}}\xi_i, \tag{8.5}$$

$$\text{s.t. } y_i(\sum_{j=1}^{N}\alpha_j y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) + b) \geq 1 - \xi_i, \qquad i = 1,\dots,N,$$

$$\xi_i \geq 0, \qquad i = 1,\dots,N,$$

where $\alpha \in \mathbb{R}^N$ are the support values, $\mathbf{y} \in \{-1,+1\}^N$ is the label vector, $\kappa(\cdot,\cdot)$ is the kernel function, b is the bias term and $\xi \in \mathbb{R}^N$ are the slack variables for soft-margin classification. The misclassification penalties $C_{\mathcal{P}}$ and $C_{\mathcal{U}}$ require tuning. We used the implementation available in scikit-learn [154] based on LIBSVM [35].

Bagging SVM

In bagging SVM, random resamples are drawn from the unlabeled set and CWSVM classifiers are trained to discriminate all positives from each resample [138]. Resampling the unlabeled set induces variability in the base models which is exploited via bagging. Base model predictions are aggregated via majority voting.

Bagging SVM with linear base models has two hyperparameters, namely the size of resamples of the unlabeled set $n_{\mathcal{U}}$ and the misclassification penalty on unlabeled instances $C_{\mathcal{U}}$. The misclassification penalty on positives $C_{\mathcal{P}}$ is fixed via the following rule:

$$C_{\mathcal{P}} = \frac{n_{\mathcal{U}} \times C_{\mathcal{U}}}{|\mathcal{P}|}, \tag{8.6}$$

[5] Though it must be noted that the ensemble methods are always implicitly nonlinear.

where $|\mathcal{P}|$ denotes the number of known positives. The heuristic rule in Equation 8.6 is common in imbalanced settings [34, 53]. We implemented bagging SVM using the EnsembleSVM library [46].

Robust Ensemble of SVM models

The robust ensemble of SVM models (RESVM) is a modified version of bagging SVM in which both the positive and unlabeled sets are resampled when constructing base model training sets [47]. The extra resampling induces additional variability between base models which improves performance when combined with a majority vote aggregation scheme. Claesen et al. [47] demonstrated that resampling the positive set provides robustness against false positives, which makes RESVM appealing for our application since our data set is known to contain a small fraction of false positives (as explained in Section 8.4.2).

When using linear base models, the RESVM approach has four hyperparameters that must be tuned, namely resample sizes and misclassification penalties per class. This approach was implemented based on EnsembleSVM [46].

8.5 Results and Discussion

Section 8.5.1 shows the predictive performance per learning configuration and compares these performances to the current state-of-the-art in large-scale risk assessment for T2D. Section 8.5.2 shows performance curves of the best configuration, which enable us to determine suitable cutoffs to identify target groups in practice. Finally, Section 8.5.3 describes a simple approach to assess which features contribute most to risk according to our best models.

8.5.1 Benchmark of learning methods

Table 8.1 summarizes the performance of each learning configuration. The AGE,GENDER feature set provides a baseline for comparison, all other feature sets include these as well. As shown in the results, this two-dimensional representation already carries some information.

Based on Table 8.1 we can conclude that a patient's medication history is highly informative to predict the start of GLA therapy. Using features based on ATC level 5, the RESVM model obtained an AUC between 72.55% and 74.43%. By adding redundancy as described in Section 8.4.2 the performance based on

	RESVM		bagging SVM		class-weighted SVM	
features	AUROC (%)	p	AUROC (%)	p	AUROC (%)	p
AGE, GENDER	55.74–56.64	$*$	58.61–59.67	$*$	**60.96–62.21**	0.04
ATC 5	**72.55–74.43**	0.17	70.83–72.62	0.09	71.89–73.74	0.01
ATC 1–4	**73.12–75.07**	0.07	69.57–71.24	$*$	73.05–74.91	0.04
ATC 1–5	**74.34–76.27**	0.13	71.50–73.27	0.05	72.13–73.94	$*$
PROVS FLAT	58.45–59.51	$*$	60.74–61.92	$*$	**63.01–64.31**	$*$
PROVS STRUCT	57.40–58.39	0.02	59.53–60.58	0.01	**62.53–63.81**	0.01
PROVS BOTH	58.89–59.75	$*$	61.72–62.87	$*$	**63.45–64.75**	$*$
ATC | PROVS	**74.89–76.82**	0.04	69.72–71.40	$*$	73.77–75.64	$*$

Table 8.1: Average bounds on area under the ROC curve and p-value of the Mann-Whitney U test over all folds for different feature sets per learning approach in a long-term prediction setup. The lower and upper bounds on AUC were computed with $\hat{\beta}_{lo} = 0.05$ and $\hat{\beta}_{up} = 0.10$, respectively. The ATC | PROVS feature set is the concatenation of the best performing sets per aspect, namely ATC 1–5 and PROVS BOTH. Stars ($*$) denote p-values below 0.005.

medication history alone was further increased to between 74.34% and 76.27% for the best learning approach (RESVM).

Predictive performance based on provisions alone turned out fairly poor, showing only a mild improvement compared to models based exclusively on age and gender for all learning algorithms. Interestingly, the best approach for representations based on provisions was class-weighted SVM, with RESVM being worst of all three learning methods. It appears that for these representations, large training sets are more important than base model variability: class-weighted SVM uses the full training set, bagging SVM uses all positives and a subset of unlabeled instances per base model and RESVM uses (small) subsets of both positives and unlabeled instances per base model.

The best representation included age, gender, and structured information about the drugs and provision history of each patient. The best learning method on this representation was RESVM, achieving an AUC between 74.89% and 76.82%. In Section 8.5.1 we compare the performance of our approach to competing screening methods.

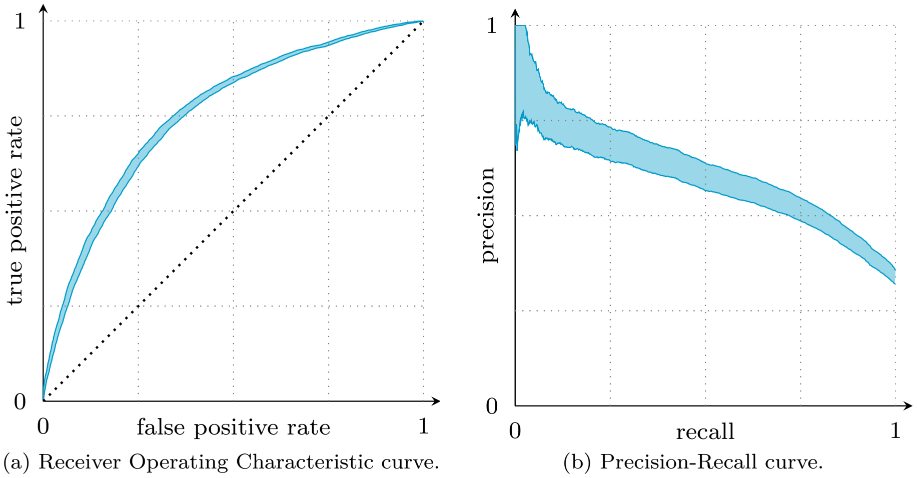
Finally, RESVM appears most resistant to overfitting in the hyperparameter optimization stage as it consistently exhibits the highest average p-values in our diagnostic test (higher is better, see Section 8.4.1). We believe this to be attributable to the use of small resamples of both positives and unlabeled instances when training base models in RESVM, since this makes it unlikely to obtain a structural overfit of the ensemble model on the full training set. In contrast, bagging SVM is far more prone to overfitting because every base model is trained on all positives.

Comparison to State-of-the-art

Our best approach obtained cross-validated AUC between 74.89% and 76.82% (exact numbers are unknown due to the lack of known negatives). This is comparable to many competing approaches, based on questionnaires and some clinical information such as the Cambridge Risk Score (AUC 67%–80%, [182, 84]), the Danish risk score (AUC 72%–87.6%, [79]), the German diabetes risk score (AUC 75%–83%, [171]) and a Dutch approach (AUC 74%, [9]). Approaches using detailed clinical information generally perform better, but are more expensive to maintain [183, 134, 124, 96]. The key advantage of our approach is the fact it is easy to implement on a population wide scale at virtually no operational cost.

The target class we used in this work is stricter than in the risk prediction methods mentioned in Section 8.2, namely patients that require GLAs for glycemic control versus patients with impaired glycemic control, respectively (except for Lindström and Tuomilehto [124], which also predicted drug-treated T2D). It is reasonable to assume that our models generally rank patients with impaired glycemic control but without a need for GLAs higher than patients without impaired glycemic control. In our performance assessment both of these patient groups are essentially treated as negatives, in contrast to the screening programmes mentioned previously which treat patients with impaired glycemic control as positives. Hence, we believe the performance of our models would appear higher when evaluated against a target class comprising all patients with impaired glycemic control, as is done in the evaluation of other screening approaches. Unfortunately, we are unable to accurately identify patients with impaired glycemic control but without need for GLAs.

All competing methods use either clinical information or direct knowledge of risk factors that is unavailable to us. Furthermore, some of the characteristics that are lacking in our data have been reported to be the most informative to assess risk for T2D [124, 183, 134]. We obtained generalization performances that are comparable to existing approaches, despite these missing predictors. Finally, our approach is the only one that is based exclusively on existing data that is always available, without requiring additional patient contacts or clinical tests.



(a) Receiver Operating Characteristic curve.

(b) Precision-Recall curve.

Figure 8.1: Performance curves for the best model: RESVM classifier based on ATC | PROVS vectorization. The lower and upper bounds are estimated using $\hat{\beta}_{lo} = 5\%$ and $\hat{\beta}_{up} = 10\%$, respectively.

8.5.2 Performance Curves for RESVM

The RESVM model based on ATC | PROVS vectorization had the best overall performance. Figure 8.1 shows bounds on the ROC and PR curves for this model. These bounds were computed using the technique described by Claesen et al. [41]. The true curve is unknown because we do not dispose of negative labels.

ROC curves enable us to determine a cutoff to use in practice, based on a suitable balance between true and false positive rate (sensitivity and 1-specificity, respectively). Determining a suitable balance requires a tradeoff between the relative importance of identifying undiagnosed patients (true positives) vis-à-vis increased amounts of screening tests on patients that are in fact healthy (false positives).

It should be noted that precision depends on class balance, and therefore the PR curve shown in Figure 8.1b is not representative for screening an overall population, since the overall population has a higher fraction of negatives than our custom data set (i.e. precision would be lower in practice). In contrast, the bounds in ROC space are representative because ROC curves are insensitive to changes in class distribution [72].

8.5.3 Feature Importance Analysis for the RESVM Model

The RESVM model is implicitly nonlinear due to its majority voting rule to aggregate base model decisions, which poses problems in assessing the importance of each predictor. However, our use of linear base models enables a simple approximation. The decision value for base model $i \in \{1, \dots, n_{\text{models}}\}$ for a test instance $\mathbf{z}$ can be written as follows:

$$f_i(\mathbf{z}) = \langle \mathbf{w}_i, \mathbf{z} \rangle + \rho_i = \mathbf{w}_i^T \mathbf{z} + \rho_i,$$

where $\mathbf{w}_i$ is the separating hyperplane and ρ_i is a bias term. A simple linear approximation of such ensemble models can be computed as the average of all base model hyperplanes:

$$\bar{\mathbf{w}} = \sum_{i=1}^{n_{\text{models}}} \mathbf{w}_i / n_{\text{models}}.$$

Feature importance can then be determined based on the coefficients in $\bar{\mathbf{w}}$. The range of every feature is comparable, since we normalized all features to the unit interval $[0, 1]$. This allows us to conclude that the features with largest (positive) coefficients in $\bar{\mathbf{w}}$ contribute most to risk according to our model.

Via this approach, the risk associated to use of cardiovascular medication (ATC main category C) far outweights all other ATC main categories. This is not surprising, as diabetes is known to be strongly related to cardiovascular problems [110, 88, 100]. The relative importance of features and associated medical implications will be discussed in detail in a forthcoming medical paper.

8.6 Conclusion

In this work we have demonstrated the ability to predict clinical outcomes based solely on readily available health expenditure data. We successfully built proof-of-concept classifiers to predict the start of glucose-lowering pharmacotherapy in patients above 40. Our experiments show that accurate predictions can be made based on historical medication purchases. These predictions can be further improved by incorporating information about medical provisions and the use of appropriate vectorization schemes.

Since adult patients starting glucose-lowering pharmacotherapy are mainly afflicted with type 2 diabetes (T2D), our models can be used for T2D risk assessment. Our approach presents a novel method for case finding which can be easily incorporated in modern healthcare, since all required data is already available. The associated operational costs are very low as the entire workflow can be fully automated without any need for patient contacts or medical tests. As such, our work provides an efficient and cost-effective method to identify a high risk subgroup, which can then be screened using decisive clinical tests.

Interestingly, our approach works well even though health expenditure data contains very limited direct information on some important known risk factors. In that sense, our approach is fundamentally different from the current state-of-the-art which mainly focuses on quantifying known risk factors directly, either by asking the patient or through clinical tests. The performance of our approach is expected to improve further when additional information about these risk factors can be obtained, e.g. family history and lifestyle.

Chapter 9

This is conclusion

. . .

Appendix A

This is myappendix

. . .

Bibliography

[1] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Frederik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, Peter Carmeliet, and Yves Moreau. Gene prioritization through genomic data fusion. *Nature Biotechnology*, 24(5):537–544, May 2006. pages 40

[2] K George MM Alberti, Paul Zimmet, Jonathan Shaw, IDF Epidemiology Task Force Consensus Group, et al. The metabolic syndrome – a new worldwide definition. *The Lancet*, 366(9491):1059–1062, 2005. pages 4

[3] KGMM Alberti, Mayer B Davidson, Ralph A DeFronzo, Allan Drash, Saul Genuth, Maureen I Harris, Richard Kahn, Harry Keen, William C Knowler, Harold Lebovitz, et al. Report of the expert committee on the diagnosis and classification of diabetes mellitus. *Diabetes Care*, 21:S5, 1998. pages 101

[4] Kurt George Matthew Mayer Alberti and PZ ft Zimmet. Definition, diagnosis and classification of diabetes mellitus and its complications. part 1: diagnosis and classification of diabetes mellitus. provisional report of a who consultation. *Diabetic medicine*, 15(7):539–553, 1998. pages 1, 2

[5] Carlos Alzate and Johan A. K. Suykens. A semi-supervised formulation to binary kernel spectral clustering. In *2012 IEEE World Congress on Computational Intelligence (IEEE WCCI/IJCNN 2012)*, Brisbane, Australia, June 2012. pages 41

[6] American Diabetes Association et al. Standards of medical care in diabetes–2014. *Diabetes care*, 37(Supplement 1):S14–S80, 2014. pages xxi, 7, 8, 100, 101, 106

[7] Peter C Austin, Muhammad M Mamdani, Therese A Stukel, Geoffrey M Anderson, and Jack V Tu. The use of the propensity score for estimating treatment effects: administrative versus clinical data. *Statistics in medicine*, 24(10):1563–1578, 2005. pages 32

[8] Laurent Azoulay, Verena Schneider-Lindner, Sophie Dell'Aniello, Alicia Schiffrin, and Samy Suissa. Combination therapy with sulfonylureas and metformin and the prevention of death in type 2 diabetes: a nested case-control study. *Pharmacoepidemiology and drug safety*, 19(4):335–342, 2010. pages 31

[9] Caroline A Baan, Johannes B Ruige, Ronald P Stolk, JC Witteman, Jacqueline M Dekker, Robert J Heine, and EJ Feskens. Performance of a predictive model to identify undiagnosed diabetes in a health care setting. *Diabetes Care*, 22(2):213–219, 1999. pages 101, 117

[10] François Bachoc. Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69, 2013. pages 69

[11] David J Ballard, Linda L Humphrey, L Joseph Melton, Peter P Frohnert, Chu-Pin Chu, W Michael O'Fallon, and Pasquale J Palumbo. Epidemiology of persistent proteinuria in type II diabetes mellitus: population-based study in Rochester, Minnesota. *Diabetes*, 37(4):405–412, 1988. pages 100

[12] Robert E Banfield, Lawrence O Hall, Kevin W Bowyer, and W Philip Kegelmeyer. A comparison of decision tree ensemble creation techniques. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):173–180, 2007. pages 46

[13] Christian A Bannister, Sarah E Holden, S Jenkins-Jones, C Ll Morgan, JP Halcox, G Schernthaner, J Mukherjee, and CJ Currie. Can people with type 2 diabetes live longer than those without? a comparison of mortality in people initiated with metformin or sulphonylurea monotherapy and matched, non-diabetic controls. *Diabetes, Obesity and Metabolism*, 16(11):1165–1173, 2014. pages 15, 30

[14] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999. pages 44, 45, 62

[15] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305, 2012. pages 60, 66, 68, 69, 72, 73

[16] James Bergstra, Dan Yamins, and David D Cox. Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference*, pages 13–20. SciPy, 2013. pages 69, 74, 75

[17] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of The 30th International Conference on Machine Learning*, pages 115–123, 2013. pages 68

[18] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011. pages 66, 69, 74

[19] D John Betteridge. Lipid control in patients with diabetes mellitus. *Nature Reviews Cardiology*, 8(5):278–290, 2011. pages 32

[20] Mauro Birattari, Zhi Yuan, Prasanna Balaprakash, and Thomas Stützle. F-race and iterated f-race: An overview. In *Experimental methods for the analysis of optimization algorithms*, pages 311–336. Springer, 2010. pages 69

[21] Christopher M Bishop et al. Neural networks for pattern recognition. 1995. pages 66, 68, 69

[22] Jock A. Blackard and Denis J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, December 1999. pages 36, 51, 89

[23] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11:2973–3009, 2010. pages 40

[24] Michael Bodmer, Christian Meier, Stephan Krähenbühl, Susan S Jick, and Christoph R Meier. Metformin, sulfonylureas, or other antidiabetes drugs and the risk of lactic acidosis or hypoglycemia a nested case-control analysis. *Diabetes Care*, 31(11):2086–2091, 2008. pages 5

[25] Jaclyn LF Bosco, Rebecca A Silliman, Soe Soe Thwin, Ann M Geiger, Diana SM Buist, Marianne N Prout, Marianne Ulcickas Yood, Reina Haque, Feifei Wei, and Timothy L Lash. A most stubborn bias: no adjustment method fully resolves confounding by indication in observational studies. *Journal of clinical epidemiology*, 63(1):64–74, 2010. pages 32

[26] Léon Bottou and Chih-Jen Lin. Support Vector Machine Solvers. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large Scale Kernel Machines*, pages 301–320, Cambridge, MA, USA, 2007. MIT Press. pages 34

[27] Léon Bottou and Chih-Jen Lin. Support vector machine solvers. *Large scale kernel machines*, pages 301–320, 2007. pages 45, 68

[28] AP Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997. pages 80

[29] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. pages 34, 42, 44

[30] Leo Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3):229–242, 2000. pages 44, 45

[31] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. pages 44, 66, 68, 69

[32] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005. pages 44

[33] B Calvo, P Larrañaga, and JA Lozano. Learning Bayesian classifiers from positive and unlabeled examples. *Pattern Recognition Letters*, 28(16):2375–2384, 2007. pages 78

[34] Gavin C Cawley. Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 1661–1668. IEEE, 2006. pages 42, 115

[35] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. pages 35, 50, 114

[36] O Chapelle, B Schölkopf, A Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006. pages 78

[37] O Chapelle, V Sindhwani, and SS Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008. pages 78

[38] G Charlone, L Torsten, C Bendix, et al. A Danish diabetes risk score for targeted screening. *Diabetes Care*, 27:727–733, 2004. pages 101

[39] NV Chawla and GI Karakoulas. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, 23:331–366, 2005. pages 78, 85

[40] M Claesen, F De Smet, J Suykens, and B De Moor. A robust ensemble approach to learn from positive and unlabeled data using SVM base models. *Neurocomputing*, 160(0):73 – 84, 2015. pages 78, 81

[41] Marc Claesen, Jesse Davis, Frank De Smet, and Bart De Moor. Assessing binary classifiers using only positive and unlabeled data. *arXiv preprint arXiv:1504.06837*, 2015. pages 108, 109, 118

[42] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. In *Proceedings of the 11th Metaheuristics International Conference (MIC'2015)*, 2015. arXiv preprint arXiv:1502.02127, available at `http://arxiv.org/abs/1502.02127`. pages 72, 76

[43] Marc Claesen, Jaak Simm, Dusan Popovic, Yves Moreau, and Bart De Moor. Easy hyperparameter search using Optunity. *CoRR*, abs/1412.1114, 2014. `http://www.optunity.net`. pages 66, 67, 69

[44] Marc Claesen, Jaak Simm, Dusan Popovic, Yves Moreau, and Bart De Moor. Easy hyperparameter search using Optunity. *arXiv preprint arXiv:1412.1114*, 2014. pages 106, 108

[45] Marc Claesen, Frank De Smet, Johan A.K. Suykens, and Bart De Moor. EnsembleSVM: A library for ensemble learning using support vector machines. *Journal of Machine Learning Research*, 15:141–145, 2014. pages 50

[46] Marc Claesen, Frank De Smet, Johan A.K. Suykens, and Bart De Moor. EnsembleSVM: A library for ensemble learning using support vector machines. *Journal of Machine Learning Research*, 15:141–145, 2014. pages 66, 68, 69, 106, 115

[47] Marc Claesen, Frank De Smet, Johan A.K. Suykens, and Bart De Moor. A robust ensemble approach to learn from positive and unlabeled data using SVM base models. *Neurocomputing*, 160(0):73 – 84, 2015. pages 106, 113, 115

[48] Helen M Colhoun, D John Betteridge, Paul N Durrington, Graham A Hitman, H Andrew W Neil, Shona J Livingstone, Margaret J Thomason, Michael I Mackness, Valentine Charlton-Menys, John H Fuller, et al. Primary prevention of cardiovascular disease with atorvastatin in type 2 diabetes in the collaborative atorvastatin diabetes study (cards): multicentre randomised placebo-controlled trial. *The Lancet*, 364(9435):685–696, 2004. pages 32

[49] Ronan Collobert, Samy Bengio, and Yoshua Bengio. A parallel mixture of SVMs for very large scale problems. *Neural Computation*, 14(5):1105–1114, 2002. pages 34, 37

[50] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995. pages 35

[51] Felipe Cucker and Steve Smale. Best choices for regularization parameters in learning theory: on the bias-variance problem. *Foundations of Computational Mathematics*, 2(4):413–428, 2002. pages 66

[52] Craig J Currie, Chris D Poole, Marc Evans, John R Peters, and Christopher Ll Morgan. Mortality and other important diabetes-related outcomes with insulin vs other antihyperglycemic therapies in type 2 diabetes. *Mortality*, 98(2), 2013. pages 15, 31

[53] Anneleen Daemen, Olivier Gevaert, Fabian Ojeda, Annelies Debucquoy, Johan A.K. Suykens, Christine Sempoux, Jean-Pascal Machiels, Karin Haustermans, and Bart De Moor. A kernel-based integration of genome-wide data for clinical decision support. *Genome Medicine*, 1(4):39, 2009. pages 42, 115

[54] J Davis and P Domingos. Deep transfer via second-order Markov logic. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 217–224, 2009. pages 81

[55] J Davis and M Goadrich. The relationship between Precision-Recall and ROC curves. In *Proc. of the 23rd International Conference on Machine learning*, pages 233–240. ACM, 2006. pages 81, 86

[56] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM. pages 50, 56

[57] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012. pages 68

[58] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006. pages 50, 58, 59

[59] F Denis, R Gilleron, and F Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348(1):70–83, 2005. pages 78

[60] Diabetes Prevention Program Research Group et al. Reduction in the incidence of type 2 diabetes with lifestyle intervention or metformin. *The New England journal of medicine*, 346(6):393, 2002. pages 4, 100, 101

[61] Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000. pages 44

[62] Daniel J Drucker, Jacques Philippe, Svetlana Mojsov, William L Chick, and Joel F Habener. Glucagon-like peptide i stimulates insulin gene expression and increases cyclic amp levels in a rat islet cell line. *Proceedings of the National Academy of Sciences*, 84(10):3434–3438, 1987. pages 5

[63] Marco F Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004. pages 51

[64] B Efron and RJ Tibshirani. *An introduction to the bootstrap*. CRC press, 1994. pages 85

[65] Bradley Efron and Gail Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48, 1983. pages 67, 68

[66] Katharina Eggensperger, Matthias Feurer, Frank Hutter, James Bergstra, Jasper Snoek, Holger Hoos, and Kevin Leyton-Brown. Towards an empirical foundation for assessing Bayesian optimization of hyperparameters. In *NIPS workshop on Bayesian Optimization in Theory and Practice*, 2013. pages 66, 69

[67] C Elkan and K Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 213–220, New York, NY, USA, 2008. ACM. pages 40, 41, 78, 87, 106, 113

[68] Michael M Engelgau, KM Narayan, and William H Herman. Screening for type 2 diabetes. *Diabetes Care*, 23(10):1563–1580, 2000. pages 100, 101

[69] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000. pages 66

[70] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. pages 50

[71] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, June 2008. pages 36

[72] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006. pages 118

[73] Internation Diabetes Federation. Idf worldwide definition of the metabolic syndrome, 2010. pages 4

[74] Félix-Antoine Fortin, De Rainville, Marc-André Gardner Gardner, Marc Parizeau, Christian Gagné, et al. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13(1):2171–2175, 2012. pages 74

[75] B. Frenay and M. Verleysen. Classification in the presence of label noise: A survey. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(5):845–869, May 2014. pages 40

[76] J-M Gamble, SH Simpson, DT Eurich, SR Majumdar, and JA Johnson. Insulin use and increased risk of mortality in type 2 diabetes: a cohort study. *Diabetes, Obesity and Metabolism*, 12(1):47–53, 2010. pages 31

[77] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992. pages 66

[78] Hertzel C Gerstein, Jackie Bosch, Gilles R Dagenais, Rafael Díaz, Hyejung Jung, Aldo P Maggioni, Janice Pogue, Jeffrey Probstfield, Ambady Ramachandran, Matthew C Riddle, et al. Basal insulin and cardiovascular and other outcomes in dysglycemia. *N Engl J Med*, 367(4):319–328, 2012. pages 31

[79] Charlotte Glümer, Bendix Carstensen, Annelli Sandbæk, Torsten Lauritzen, Torben Jørgensen, and Knut Borch-Johnsen. A Danish diabetes risk score for targeted screening – the Inter99 study. *Diabetes Care*, 27(3):727–733, 2004. pages 101, 102, 117

[80] SA Goldman and Y Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 327–334, 2000. pages 78

[81] Patricia M Grambsch and Terry M Therneau. Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, 81(3):515–526, 1994. pages 21

[82] Yves Grandvalet. Bagging equalizes influence. *Machine Learning*, 55(3):251–270, 2004. pages 44, 45

[83] Simon J Griffin, Knut Borch-Johnsen, Melanie J Davies, Kamlesh Khunti, Guy EHM Rutten, Annelli Sandbæk, Stephen J Sharp, Rebecca K Simmons, Maureen Van den Donk, Nicholas J Wareham, et al. Effect of early intensive multifactorial therapy on 5-year cardiovascular outcomes in individuals with type 2 diabetes detected by screening (addition-europe): a cluster-randomised trial. *The Lancet*, 378(9786):156–167, 2011. pages 7

[84] SJ Griffin, PS Little, CN Hales, AL Kinmonth, and NJ Wareham. Diabetes risk score: towards earlier detection of type 2 diabetes in general practice. *Diabetes/metabolism research and reviews*, 16(3):164–171, 2000. pages 101, 117

[85] Diederick E Grobbee and Arno W Hoes. Confounding and indication for treatment in evaluation of drug treatment for hypertension. *BMJ: British Medical Journal*, 315(7116):1151, 1997. pages 32

[86] UK Prospective Diabetes Study (UKPDS) Group et al. Effect of intensive blood-glucose control with metformin on complications in overweight patients with type 2 diabetes (ukpds 34). *The Lancet*, 352(9131):854–865, 1998. pages 15, 31

[87] UK Prospective Diabetes Study (UKPDS) Group et al. Intensive blood-glucose control with sulphonylureas or insulin compared with conventional treatment and risk of complications in patients with type 2 diabetes (ukpds 33). *The Lancet*, 352(9131):837–853, 1998. pages 15

[88] Scott M Grundy, Ivor J Benjamin, Gregory L Burke, Alan Chait, Robert H Eckel, Barbara V Howard, William Mitch, Sidney C Smith, and James R Sowers. Diabetes and cardiovascular disease a statement for healthcare professionals from the American Heart Association. *Circulation*, 100(10):1134–1146, 1999. pages 119

[89] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982. pages 108

[90] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001. pages 74

[91] Maureen I Harris and Richard C Eastman. Early detection of undiagnosed diabetes mellitus: a US perspective. *Diabetes/metabolism research and reviews*, 16(4):230–236, 2000. pages 100

[92] Maureen I Harris, Katherine M Flegal, Catherine C Cowie, Mark S Eberhardt, David E Goldstein, Randie R Little, Hsiao-Mei Wiedmeyer,

and Danita D Byrd-Holt. Prevalence of diabetes, impaired fasting glucose, and impaired glucose tolerance in US adults: the Third National Health and Nutrition Examination Survey, 1988–1994. *Diabetes Care*, 21(4):518–524, 1998. pages 100, 106

[93] Maureen I Harris, Ronald Klein, Catherine C Cowie, Michael Rowland, and Danita D Byrd-Holt. Is the risk of diabetic retinopathy greater in non-hispanic blacks and mexican americans than in non-hispanic whites with type 2 diabetes?: A US population study. *Diabetes Care*, 21(8):1230–1235, 1998. pages 100

[94] Maureen I Harris, Ronald Klein, Tim A Welborn, and Matthew W Knuiman. Onset of NIDDM occurs at least 4–7 yr before clinical diagnosis. *Diabetes Care*, 15(7):815–819, 1992. pages 100

[95] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. In *Journal of Machine Learning Research*, pages 1391–1415, 2004. pages 66

[96] Kenneth E Heikes, David M Eddy, Bhakti Arondekar, and Leonard Schlessinger. Diabetes risk calculator a simple tool for detecting undiagnosed diabetes and pre-diabetes. *Diabetes Care*, 31(5):1040–1045, 2008. pages 101, 102, 117

[97] Bianca Hemmingsen, Søren S Lund, Christian Gluud, Allan Vaag, Thomas Almdal, Christina Hemmingsen, and Jørn Wetterslev. Targeting intensive glycaemic control versus targeting conventional glycaemic control for type 2 diabetes mellitus. *The Cochrane Library*, 2011. pages 31

[98] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade*, pages 599–619. Springer, 2012. pages 66

[99] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003. pages 66

[100] Frank B Hu, Meir J Stampfer, Steven M Haffner, Caren G Solomon, Walter C Willett, and JoAnn E Manson. Elevated risk of cardiovascular disease prior to clinical diagnosis of type 2 diabetes. *Diabetes Care*, 25(7):1129–1134, 2002. pages 100, 119

[101] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pages 507–523. Springer, 2011. pages 69

[102] Frank Hutter, Holger H Hoos, Kevin Leyton-Brown, and Thomas Stützle. ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009. pages 68, 69, 75

[103] InterAct Consortium et al. The link between family history and risk of type 2 diabetes is not explained by anthropometric, lifestyle or genetic risk factors: the EPIC-InterAct study. *Diabetologia*, 56(1):60–69, 2013. pages 101

[104] Jeffrey A Johnson, Sumit R Majumdar, Scot H Simpson, and Ellen L Toth. Decreased mortality associated with the use of metformin compared with sulfonylurea monotherapy in type 2 diabetes. *Diabetes care*, 25(12):2244–2248, 2002. pages 15, 31

[105] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998. pages 69

[106] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. [Online; accessed 2015-04-16]. pages 106

[107] Kristijan H Kahler, Mangala Rajan, George G Rhoads, Monika M Safford, Kitaw Demissie, Shou-En Lu, and Leonard M Pogach. Impact of oral antihyperglycemic therapy on all-cause mortality among patients with diabetes in the veterans health administration. *Diabetes Care*, 30(7):1689–1693, 2007. pages 31

[108] Richard Kahn, Peter Alperin, David Eddy, Knut Borch-Johnsen, John Buse, Justin Feigelman, Edward Gregg, Rury R Holman, M Sue Kirkman, Michael Stern, et al. Age at initiation and frequency of screening to detect type 2 diabetes: a cost-effectiveness analysis. *The Lancet*, 375(9723):1365–1374, 2010. pages 7

[109] John D Kalbfleisch and Ross L Prentice. *The statistical analysis of failure time data*, volume 360. John Wiley & Sons, 2011. pages 21

[110] WB Kannel and DL McGee. Diabetes and cardiovascular disease: The Framingham study. *JAMA*, 241(19):2035–2038, 1979. pages 119

[111] Maarten Keijzer and Vladan Babovic. Genetic programming, ensemble methods and the bias/variance tradeoff – introductory investigations. In Riccardo Poli, Wolfgang Banzhaf, William B. Langdon, Julian Miller, Peter Nordin, and Terence C. Fogarty, editors, *Genetic Programming*, volume 1802 of *Lecture Notes in Computer Science*, pages 76–90. Springer Berlin Heidelberg, 2000. pages 62

[112] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010. pages 74

[113] Hilary King, Ronald E Aubert, and William H Herman. Global burden of diabetes, 1995–2025: prevalence, numerical estimates, and projections. *Diabetes Care*, 21(9):1414–1431, 1998. pages 100, 106

[114] Dmitri Kirpichnikov, Samy I McFarlane, and James R Sowers. Metformin: an update. *Annals of internal medicine*, 137(1):25–33, 2002. pages 4

[115] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1137–1145, 1995. pages 67, 68

[116] Eva M Kohner, Stephen J Aldington, Irene M Stratton, Susan E Manley, Rury R Holman, David R Matthews, and Robert C Turner. United Kingdom Prospective Diabetes Study, 30: diabetic retinopathy at diagnosis of non–insulin-dependent diabetes mellitus and associated risk factors. *Archives of Ophthalmology*, 116(3):297–303, 1998. pages 100

[117] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. pages 68

[118] Lawrence L Kupper, John M Karon, David G Kleinbaum, Hal Morgenstern, and Donald K Lewis. Matching in epidemiologic studies: validity and efficiency considerations. *Biometrics*, pages 271–291, 1981. pages 32

[119] Ar Lazarevic, Aysel Ozgur, Levent Ertoz, Jaideep Srivastava, and Vipin Kumar. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the Third SIAM International Conference on Data Mining*, 2003. pages 40

[120] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. pages 51

[121] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pages 448–455, 2003. pages 41, 50, 106, 113

[122] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, pages 587–592, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. pages 41

[123] Shih-Wei Lin, Kuo-Ching Ying, Shih-Chieh Chen, and Zne-Jung Lee. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert systems with applications*, 35(4):1817–1824, 2008. pages 69

[124] Jaana Lindström and Jaakko Tuomilehto. The diabetes risk score a practical tool to predict type 2 diabetes risk. *Diabetes Care*, 26(3):725–731, 2003. pages 101, 102, 117

[125] Bin Linghu and Bing-Yu Sun. Constructing effective SVM ensembles for image classification. In *Knowledge Acquisition and Modeling (KAM), 2010 3rd International Symposium on*, pages 80–83, 2010. pages 34

[126] Nikolas List and Hans Ulrich Simon. SVM-optimization and steepest-descent line search. In *Proceedings of the 22nd Annual Conference on Computational Learning Theory*, 2009. pages 34

[127] B Liu, Y Dai, X Li, WS Lee, and PS Yu. Building text classifiers using positive and unlabeled examples. In *3rd IEEE International Conference on Data Mining.*, pages 179–186, 2003. pages 78

[128] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 179–186, Washington, DC, USA, 2003. IEEE Computer Society. pages 40, 41, 113, 114

[129] Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 387–394, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. pages 41, 113

[130] Zhigang Liu, Wenzhong Shi, Deren Li, and Qianqing Qin. Partially supervised classification – based on weighted unlabeled samples support vector machine. In *Proceedings of the First international conference on Advanced Data Mining and Applications*, ADMA'05, pages 118–129, Berlin, Heidelberg, 2005. Springer-Verlag. pages 41

[131] Julie A Lovshin and Daniel J Drucker. Incretin-based therapies for type 2 diabetes mellitus. *Nature Reviews Endocrinology*, 5(5):262–269, 2009. pages 5

[132] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947. pages 109

[133] Gonzalo Martínez-Muñoz and Alberto Suárez. Out-of-bag estimation of the optimal sample size in bagging. *Pattern Recognition*, 43(1):143–152, 2010. pages 46

[134] Marguerite J McNeely, Edward J Boyko, Donna L Leonetti, Steven E Kahn, and Wilfred Y Fujimoto. Comparison of a clinical model, the oral glucose tolerance test, and fasting glucose for prediction of type 2 diabetes risk in Japanese Americans. *Diabetes Care*, 26(3):758–763, 2003. pages 101, 102, 117

[135] Michael Meissner, Michael Schmuker, and Gisbert Schneider. Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. *BMC bioinformatics*, 7(1):125, 2006. pages 69

[136] Ali H Mokdad, Earl S Ford, Barbara A Bowman, William H Dietz, Frank Vinicor, Virginia S Bales, and James S Marks. Prevalence of obesity, diabetes, and obesity-related health risk factors, 2001. *JAMA*, 289(1):76–79, 2003. pages 101

[137] F Mordelet and J-P Vert. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209, 2014. pages 78

[138] Fantine Mordelet and Jean-Philippe Vert. A bagging SVM to learn from positive and unlabeled examples. *arXiv preprint arXiv:1010.0772*, 2010. pages 40, 41, 42, 43, 62, 106, 113, 114

[139] Fantine Mordelet and Jean-Philippe Vert. ProDiGe: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC bioinformatics*, 12(1):389, 2011. pages 40, 78, 81

[140] Fantine Mordelet and Jean-Philippe P. Vert. ProDiGe: Prioritization Of Disease Genes with multitask machine learning from positive and unlabeled examples. *BMC bioinformatics*, 12(1):389+, 2011. pages 34

[141] C Ll Morgan, J Mukherjee, S Jenkins-Jones, SE Holden, and CJ Currie. Association between first-line monotherapy with sulphonylurea versus metformin and risk of all-cause mortality and cardiovascular events: a retrospective, observational study. *Diabetes, Obesity and Metabolism*, 16(10):957–962, 2014. pages 15, 31

[142] David M Nathan, Mayer B Davidson, Ralph A Defronzo, Robert J Heine, Robert R Henry, Richard Pratley, and Bernard Zinman. Impaired fasting glucose and impaired glucose tolerance implications for care. *Diabetes care*, 30(3):753–759, 2007. pages 2, 4

[143] Michael A Nauck, Tina Vilsbøll, Baptist Gallwitz, Alan Garber, and Sten Madsbad. Incretin-based therapies viewpoints on the way to consensus. *Diabetes care*, 32(suppl 2):S223–S231, 2009. pages 5

[144] Peter Nemenyi. Distribution-free multiple comparisons. In *BIOMETRICS*, volume 18, page 263. INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1962. pages 59

[145] K Nigam, AK McCallum, S Thrun, and T Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000. pages 78

[146] WHO Expert Committee on Diabetes Mellitus. Second report. 1980. Technical Report Series 646. pages 3

[147] World Health Organization et al. *Guidelines for ATC classification and DDD assignment*. World Health Organization, 2015. pages 16

[148] Edgar Osuna, Robert Freund, and Federico Girosi. Support Vector Machines: Training and Applications. Technical Report AIM-1602, 1997. pages 35, 42

[149] Xiao-Ren Pan, Guang-wei Li, Ying-Hua Hu, Ji-Xing Wang, Wen-Ying Yang, Zuo-Xin An, Ze-Xi Hu, Jina-Zhong Xiao, Hui-Bi Cao, Ping-An Liu, et al. Effects of diet and exercise in preventing NIDDM in people with impaired glucose tolerance: the Da Qing IGT and Diabetes Study. *Diabetes Care*, 20(4):537–544, 1997. pages 101

[150] João P Papa, Gustavo H Rosa, Aparecido N Marana, Walter Scheirer, and David D Cox. Model selection for Discriminative Restricted Boltzmann Machines through meta-heuristic techniques. *Journal of Computational Science*, 9:14–18, 2015. pages 76

[151] PJ Park, SJ Griffin, L Sargeant, and NJ Wareham. The performance of a risk score in predicting undiagnosed hyperglycemia. *Diabetes Care*, 25(6):984–988, 2002. pages 101

[152] Stephen G Pauker. Deciding about screening. *Annals of internal medicine*, 118(11):901–902, 1993. pages 100

[153] Mykola Pechenizkiy, Alexey Tsymbal, Seppo Puuronen, and Oleksandr Pechenizkiy. Class noise and supervised learning in medical domains: The effect of feature extraction. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pages 708–713. IEEE, 2006. pages 40

[154] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. pages 66, 69, 72, 75, 106, 114

[155] Marc Prentki and Christopher J Nolan. Islet β cell failure in type 2 diabetes. *Journal of Clinical Investigation*, 116(7):1802, 2006. pages 15

[156] Danil Prokhorov. IJCNN 2001 neural network competition. *Slide presentation in IJCNN*, 2001. pages 37, 51

[157] Peter Proks, Frank Reimann, Nick Green, Fiona Gribble, and Frances Ashcroft. Sulfonylurea stimulation of insulin secretion. *Diabetes*, 51(suppl 3):S368–S376, 2002. pages 5

[158] Bruce M Psaty, Thomas D Koepsell, Danyu Lin, Noel S Weiss, David S Siscovick, Frits R Rosendaal, Marco Pahor, and Curt D Furberg. Assessment and control for confounding by indication in observational studies. *Journal of the American Geriatrics Society*, 47(6):749–754, 1999. pages 32

[159] Ulla Rajala, Mauri Laakso, Qing Qiao, and Sirkka Keinänen-Kiukaanniemi. Prevalence of retinopathy in people with diabetes, impaired glucose tolerance, and normal glucose tolerance. *Diabetes Care*, 21(10):1664–1669, 1998. pages 100

[160] J Sunil Rao and Robert Tibshirani. The out-of-bootstrap method for model averaging and selection. *University of Toronto*, 1997. pages 44

[161] Jared P Reis, Catherine M Loria, Paul D Sorlie, Yikyung Park, Albert Hollenbeck, and Arthur Schatzkin. Lifestyle factors and risk for new-onset diabetes: a population-based cohort study. *Annals of internal medicine*, 155(5):292–299, 2011. pages 101

[162] Kenneth J Rothman, Sander Greenland, and Timothy L Lash. *Modern epidemiology*. Lippincott Williams & Wilkins, 2008. pages 19

[163] Christianne L Roumie, Adriana M Hung, Robert A Greevy, Carlos G Grijalva, Xulei Liu, Harvey J Murff, Tom A Elasy, and Marie R Griffin. Comparative effectiveness of sulfonylurea and metformin monotherapy on cardiovascular events in type 2 diabetes mellitus: a cohort study. *Annals of internal medicine*, 157(9):601–610, 2012. pages 15, 31

[164] Olivier Roustant, David Ginsbourger, Yves Deville, et al. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. 2012. pages 75

[165] Donald B Rubin. Using multivariate matched sampling and regression adjustment to control bias in observational studies. *Journal of the American Statistical Association*, 74(366a):318–328, 1979. pages 32

[166] Robert J Rubin, William M Altman, and Daniel N Mendelson. Health care expenditures for people with diabetes mellitus, 1992. *The Journal of Clinical Endocrinology & Metabolism*, 78(4):809A–809F, 1994. pages 100

[167] Alan R Saltiel and Jerrold M Olefsky. Thiazolidinediones in the treatment of insulin resistance and type ii diabetes. *Diabetes*, 45(12):1661–1669, 1996. pages 4

[168] B Schölkopf, JC Platt, J Shawe-Taylor, AJ Smola, and RC Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. pages 87

[169] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002. pages 66, 67, 69

[170] Tina Ken Schramm, Gunnar Hilmar Gislason, Allan Vaag, Jeppe Nørgaard Rasmussen, Fredrik Folke, Morten Lock Hansen, Emil Loldrup Fosbøl, Lars Køber, Mette Lykke Norgaard, Mette Madsen, et al. Mortality and cardiovascular risk associated with different insulin secretagogues compared with metformin in type 2 diabetes, with or without a previous myocardial infarction: a nationwide study. *European heart journal*, 32(15):1900–1908, 2011. pages 31

[171] Matthias B Schulze, Kurt Hoffmann, Heiner Boeing, Jakob Linseisen, Sabine Rohrmann, Matthias Möhlig, Andreas FH Pfeiffer, Joachim Spranger, Claus Thamer, Hans-Ulrich Häring, et al. An accurate risk score based on anthropometric, dietary, and lifestyle factors to predict the development of type 2 diabetes. *Diabetes Care*, 30(3):510–515, 2007. pages 101, 102, 117

[172] Peter EH Schwarz, Jiang Li, Manja Reimann, Alta E Schutte, Antje Bergmann, Markolf Hanefeld, Stefan R Bornstein, Jan Schulze, Jaakko Tuomilehto, and Jaana Lindstrom. The Finnish Diabetes Risk Score is associated with insulin resistance and progression towards type 2 diabetes. *The Journal of Clinical Endocrinology & Metabolism*, 94(3):920–926, 2009. pages 101

[173] C Scott and G Blanchard. Novelty detection: Unlabeled data definitely help. In *Proc. of the 12th International Conference on Artificial Intelligence and Statistics*, pages 464–471, 2009. pages 78, 87

[174] K Sechidis, B Calvo, and G Brown. Statistical hypothesis testing in positive unlabelled data. In *Machine Learning and Knowledge Discovery in Databases*, pages 66–81. Springer, 2014. pages 78, 81

[175] Holbrooke S Seltzer. Drug-induced hypoglycemia. a review of 1418 cases. *Endocrinology and metabolism clinics of North America*, 18(1):163–183, 1989. pages 101

[176] Iris Shai, Rui Jiang, JoAnn E Manson, Meir J Stampfer, Walter C Willett, Graham A Colditz, and Frank B Hu. Ethnicity, obesity, and risk of type 2 diabetes in women a 20-year follow-up study. *Diabetes care*, 29(7):1585–1590, 2006. pages 101

[177] Salimah Z Shariff, Meaghan S Cuerden, Arsh K Jain, and Amit X Garg. The secret of immortal time bias in epidemiologic studies. *Journal of the American Society of Nephrology*, 19(5):841–843, 2008. pages 19

[178] Brian K. Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, December 2004. pages 40

[179] Alejandro Sifrim, Dusan Popovic, Léon-Charles Tranchevent, Amin Arderschirdavani, Ryo Sakai, Peter Konings, Joris Vermeesch, Jan Aerts, Bart De Moor, and Yves Moreau. eXtasy: Variant prioritization by genomic data fusion. *Nature Methods*, 10:1083–1084, 2013. pages 40, 78, 81

[180] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012. pages 69, 75

[181] P Sonksen and J Sonksen. Insulin: understanding its action in health and disease. *British journal of anaesthesia*, 85(1):69–79, 2000. pages 2

[182] Annemieke MW Spijkerman, Matthew F Yuyun, Simon J Griffin, Jacqueline M Dekker, Giel Nijpels, and Nicholas J Wareham. The performance of a risk score as a screening test for undiagnosed hyperglycemia in ethnic minority groups data from the 1999 health survey for England. *Diabetes Care*, 27(1):116–122, 2004. pages 101, 117

[183] Michael P Stern, Ken Williams, and Steven M Haffner. Identification of persons at high risk for type 2 diabetes mellitus: do we need the oral glucose tolerance test? *Annals of Internal Medicine*, 136(8):575–581, 2002. pages 101, 102, 117

[184] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014. pages 68

[185] R Core Team. R: A language and environment for statistical computing. r foundation for statistical computing, vienna, austria, 2012, 2014. pages 21

[186] Terry Therneau. A package for survival analysis in s. r package version 2.37-4. *URL http://CRAN. R-project. org/package= survival. Box*, 980032:23298–0032, 2013. pages 21

[187] Jinn-Tsong Tsai, Jyh-Horng Chou, and Tung-Kuan Liu. Tuning the structure and parameters of a neural network by using hybrid taguchi-genetic algorithm. *Neural Networks, IEEE Transactions on*, 17(1):69–80, 2006. pages 69

[188] Jaakko Tuomilehto, Jaana Lindström, Johan G Eriksson, Timo T Valle, Helena Hämäläinen, Pirjo Ilanne-Parikka, Sirkka Keinänen-Kiukaanniemi, Mauri Laakso, Anne Louheranta, Merja Rastas, et al. Prevention of type 2 diabetes mellitus by changes in lifestyle among subjects with impaired glucose tolerance. *New England Journal of Medicine*, 344(18):1343–1350, 2001. pages 4, 100, 101

[189] Robert C Turner, Carole A Cull, Valeria Frighi, Rury R Holman, UK Prospective Diabetes Study (UKPDS) Group, et al. Glycemic control with diet, sulfonylurea, metformin, or insulin in patients with type 2 diabetes mellitus: progressive requirement for multiple therapies (UKPDS 49). *JAMA*, 281(21):2005–2012, 1999. pages 100

[190] Ioanna Tzoulaki, Mariam Molokhia, Vasa Curcin, Mark P Little, Christopher J Millett, Anthea Ng, Robert I Hughes, Kamlesh Khunti, Martin R Wilkins, Azeem Majeed, et al. Risk of cardiovascular disease and all cause mortality among patients with type 2 diabetes prescribed oral antidiabetes drugs: retrospective cohort study using uk general practice research database. *Bmj*, 339, 2009. pages 15, 31

[191] N Unwin, J Shaw, P Zimmet, and KGMM Alberti. Impaired glucose tolerance and impaired fasting glycaemia: the current status on definition and intervention. *Diabetic medicine*, 19(9):708–723, 2002. pages 4

[192] Giorgio Valentini and Thomas G Dietterich. Low bias bagged support vector machines. In *ICML*, pages 752–759, 2003. pages 34, 37

[193] Greet Van den Berghe, Alexander Wilmer, Greet Hermans, Wouter Meersseman, Pieter J Wouters, Ilse Milants, Eric Van Wijngaerden, Herman Bobbaers, and Roger Bouillon. Intensive insulin therapy in the medical ICU. *New England Journal of Medicine*, 354(5):449, 2006. pages 109

[194] Greet Van den Berghe, Pieter Wouters, Frank Weekers, Charles Verwaest, Frans Bruyninckx, Miet Schetz, Dirk Vlasselaers, Patrick Ferdinande, Peter Lauwers, and Roger Bouillon. Intensive insulin therapy in critically ill patients. *New England journal of medicine*, 345(19):1359–1367, 2001. pages 109

[195] R Van den Oever and C Volckaert. Financing health care in Belgium. the nomenclature: from fee-for-service to budget-financing. *Acta chirurgica Belgica*, 108(2):157, 2008. pages 104

[196] AW Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge university press, 2000. pages 86

[197] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011. pages 106

[198] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006. pages 107

[199] Shi-jin Wang, Avin Mathew, Yan Chen, Li-feng Xi, Lin Ma, and Jay Lee. Empirical analysis of support vector machine ensemble classifiers. *Expert Systems with Applications*, 36(3, Part 2):6466 – 6476, 2009. pages 34, 37

[200] Nicholas J Wareham and Simon J Griffin. Should we screen for type 2 diabetes? evaluation against national screening committee criteria. *BMJ: British Medical Journal*, 322(7292):986, 2001. pages 101

[201] S Wheeler, K Moore, CW Forsberg, K Riley, JS Floyd, NL Smith, and EJ Boyko. Mortality among veterans with type 2 diabetes initiating metformin, sulfonylurea or rosiglitazone monotherapy. *Diabetologia*, 56(9):1934–1943, 2013. pages 31

[202] WHO Collaborating Centre for Drug Statistics Methodology. *Guidelines for ATC classification and DDD assignment*. World Health Organization, 2015. pages 103

[203] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer Science & Business Media, 2009. pages 21

[204] World Health Organization et al. Prevention of diabetes mellitus: report of a WHO study group [meeting held in geneva from 16 to 20 november 1992]. (WHO technical report number 844), 1994. pages 100, 101, 105

[205] World Health Organization et al. International classification of diseases (ICD). 2012. pages 104

[206] Samuel Xavier-de Souza, Johan AK Suykens, Joos Vandewalle, and Désiré Bollé. Coupled simulated annealing. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(2):320–335, 2010. pages 69

[207] Hannele Yki-Järvinen. Thiazolidinediones. *New England Journal of Medicine*, 351(11):1106–1118, 2004. pages 4

[208] H Yu, J Han, and KC-C Chang. PEBL: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004. pages 78

[209] Hwanjo Yu. Single-class classification with mapping convergence. *Machine Learning*, 61(1-3):49–69, November 2005. pages 41, 113

[210] Hwanjo Yu, Jiawei Han, and Kevin C. Chang. PEBL: positive example based learning for web page classification using SVM. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–248, New York, NY, USA, 2002. ACM Press. pages 40

[211] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 1–7, 2010. pages 75

[212] Nicola N Zammitt and Brian M Frier. Hypoglycemia in type 2 diabetes pathophysiology, frequency, and effects of different treatment modalities. *Diabetes Care*, 28(12):2948–2961, 2005. pages 101

[213] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004. pages 40

[214] Paul Zimmet, KGMM Alberti, and Jonathan Shaw. Global and societal implications of the diabetes epidemic. *Nature*, 414(6865):782–787, 2001. pages 100

This is curriculum

. . .

List of publications

Input file chapters/publications/publications.tex does not exist. Make sure its starts with “\chapter{List of publications}”. To not include this chapter in the table of contents, use the starred version of the \chapter command…

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
IMINDS-STADIUS
Kasteelpark Arenberg 10, bus 2446
B-3001 Leuven
marc.claesen@esat.kuleuven.be
http://esat.kuleuven.be/stadius