MỤC LỤC

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

LỜI MỞ ĐẦU

Đối với các ngân hàng việc chấm điểm tín dụng và phân loại các khách hàng là một trong những khâu thiết yếu cho quy trình quản trị rủi ro của ngân hàng. Phương pháp truyền thống của việc ra quyết định có cho một cá nhân cụ thể vay hay không là dựa trên đánh giá cảm tính dựa trên kinh nghiệm cá nhân. Tuy nhiên, sự phát triển về quy mô của nền kinh tế đã tạo ra sức ép về nhu cầu vay, đi kèm với đó là sự cạnh tranh giữa các ngân hàng và công nghệ máy tính ngày càng phát triển đã khiến cho việc sử dụng các mô hình thống kê trong việc phân loại các khách hàng tín dụng là bắt buộc đối với các ngân hàng trên thế giới mà ở Việt Nam cũng không phải là ngoại lệ.

Vậy, phương pháp ước lượng nào có thể giúp chúng ta xây dựng được hệ thống chấm điểm tín dụng chính xác nhất? Đã có một số nghiên cứu mang tính chất so sánh hiệu năng giữa các mô hình (Baesens et al. 2003; Xiao, Zhao, and Fei 2006; Lessmann et al. 2015). Sự khác biệt về hiệu năng của các phương pháp khác nhau là có, tuy nhiên hầu như là không đáng kể, và không phải các mô hình hiệu quả hơn đều là các mô hình mới và tân tiến. Theo Thomas (2010), cách hiệu quả để xây dựng một hệ thống lượng định hiệu quả là phối hợp nhiều mô hình khác nhau thay vì tìm kiếm một mô hình toàn diện có thể áp dụng với tất cả các ngân hàng.

Trong bài này, chúng ta sẽ tiếp cận đến một số phương pháp phân loại các khách hàng tín dụng phổ biến hiện nay và rút ra một số kết luận về việc sử dụng các phương pháp khác nhau sao cho hợp lý. Bài viết này được bố cục như sau:

- **Chương 1** đưa ra một cái nhìn tổng quan về lĩnh vực quản trị rủi ro tín dụng trong ngân hàng và đưa ra một số vấn đề của việc chấm điểm tín dụng tại các ngân hàng Việt Nam.

- Các mô hình được thực hiện trong bài này sẽ được giới thiệu ở **Chương 2**, đi kèm với đó là một số chỉ tiêu sẽ được dùng để đánh giá mô hình trong bài này.

- Trong **Chương 3**, chúng ta sẽ ứng dụng các phương pháp được giới thiệu ở **Chương 2** trong một bộ số liệu mẫu về các khách hàng thẻ tín dụng trong một ngân hàng ở Đài Loan.
- Kết quả của các mô hình sẽ được thảo luận ở **Chương 4**, cùng với một số kết luận rút ra được sau khi áp dụng mô hình.

Đề tài này được soạn thảo bằng LaTeX kết hợp với `Sweave` và `knitr` (Xie 2015). Tất cả phân tích được thực hiện trên phần mềm thống kê R version 3.4.0 (2017-04-21) (R Core Team 2017), các phân tích cụ thể được thực hiện sử dụng các gói mở rộng `caret`(Jed Wing et al. 2016), `tidyverse` (Wickham 2017)... Mô hình logit được thực hiện với gói `glmnet`(Friedman, Hastie, and Tibshirani 2010). Mô hình SVM được thực hiện với gói `kernlab` (Karatzoglou et al. 2004), là một giao diện của phần mềm `LIBSVM` (Chang and Lin 2011) trong môi trường `R`.

Em xin cảm ơn giáo viên hướng dẫn, cô Nguyễn Thị Minh, cùng với các thầy cô giáo khác trong khoa đã tạo điều kiện cho em thực hiện đề tài này.

CHƯƠNG I

TỔNG QUAN VỀ QUẢN TRỊ RỦI RO TÍN DỤNG ĐỐI VỚI KHÁCH HÀNG CÁ NHÂN

1.1 TỔNG QUAN VỀ RỦI RO TÍN DỤNG VÀ VẤN ĐỀ QUẢN TRỊ RỦI RO TRONG NGÂN HÀNG

1.1.1 Rủi ro và các khái niệm liên quan trong ngân hàng

1.1.1.1 Tín dụng

1.1.1.2 Rủi ro tín dụng trong ngân hàng

1.1.2 Quản trị rủi ro trong ngân hàng

1.1.2.1 Hoạt động quản trị rủi ro trong ngân hàng

1.1.2.2 Quản trị rủi ro tín dụng trong lịch sử

1.1.2.3

1.2 THỰC TRẠNG CỦA VIỆC CHẤM ĐIỂM TÍN DỤNG TẠI VIỆT NAM

1.3 KẾT LUẬN

CHƯƠNG II

CÁC MÔ HÌNH PHÂN LOẠI KHÁCH HÀNG VAY THẺ TÍN DỤNG

2.1 CÁC MÔ HÌNH PHÂN LOẠI

2.1.1 Mô hình logit

2.1.1.1 Khái niệm

Mô hình hồi quy Logistic (hay logit) được dùng để nghiên cứu mối quan hệ giữa xác suất của các biến nhị phân hoặc phân loại và các biến giải thích khác. Hướng tiếp cận của mô hình Logistic cho bài toán phân loại là bằng cách ước lượng giá trị xác suất $P(y = 1|X)$ như sau:

$$P(y = 1|X) = \frac{e^{\beta_0 + \sum \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n}}{1 + e^{\beta_0 + \sum \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n}}$$

Với y là biến dùng để phân loại, chỉ nhận hai giá trị 0 hoặc 1, X là các vector của biến độc lập, β_0, β_1, β_2, ..., β_n là các hệ số cần ước lượng.

Hay còn được viết dưới dạng:

$$\log \frac{P(y = 1|X)}{P(y = 0|X)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

Trong trường hợp các biến độc lập là biến phân loại không so sánh được (ví dụ: Giới tính, dân tộc, v..v..) chúng ta đưa các biến này vào mô hình bằng cách sử dụng một nhóm các biến giả tương ứng với từng giá trị khác nhau của biến phân loại.

2.1.1.2 Ước lượng mô hình logit

a, Ước lượng hợp lý tối đa

Các hệ số β thường được ước lượng bằng phương pháp ước lượng hợp lý tối đa (Hosmer Jr, Lemeshow, and Sturdivant 2013), sử dụng hàm hợp lý có điều kiện G đối với mỗi giá trị của X. Hàm hợp lý logarit cho N quan sát được viết như sau:

$$L(\theta) = \sum_{i=1}^{N} P_{g_i}(x_i;\theta)$$

, với $p_k(x_i;\theta) = P(G = k|X = x_i;\theta)$.

Trong trường hợp biến phụ thuộc Y chỉ có 2 giá trị: $(0,1)$, ta có thể mã hóa 2 nhóm của g_i thành$y_i = 1$ khi $g_i = 1$ và $y_i = 0$ khi $g_i = 2$. Khi đó logarit của hàm hợp lý có thể viết lại như sau:

$$L(\beta) = \sum_{i=1}^{N} \{y_i log(x_i;\beta) + (1-y_i) log(x_i;\beta)\}$$

Tối ưu hóa hàm L sẽ cho chúng ta ước lượng hợp lý tối đa cho các hệ số β trong mô hình.

b, Ràng buộc L1 hay mô hình Lasso

Một vấn đề mô hình logit hay gặp phải đó là hiện tượng đa cộng tuyến giữa các biến khi số lượng biến p tăng lên. Hậu quả của hiện tượng này là các ước lượng cho hệ số β thường là có sai số lớn, mặc dù ước lượng vẫn là không chệch. Nói cách khác, các giá trị β ước lượng được thường có hiệu quả kém khi áp dụng trên mẫu mới, mặc dù mô hình vẫn có độ chính xác cao khi áp dụng trên bộ số liệu mẫu dùng để ước lượng ra mô hình.

Để xử lý vấn đề này, chúng ta có thể áp dụng nhiều phương pháp để loại biến ra khỏi mô hình, hoặc sử dụng các phương pháp ước lượng khác mà các biến có ý nghĩa thống kê thấp bị loại ra khỏi mô hình trong quá trình ước lượng.

Phương pháp Lasso là một cải tiến của các mô hình tuyến tính, trong mô hình này, chúng ta áp dụng thêm ràng bộc L1 đối với hàm hợp lý tối đa. Áp dụng với mô hình Logit, thay vì tối ưu hàm hợp lý tối đa, chúng ta tối ưu:

$$\max_{\beta_0,\beta} \left\{ \sum_{i=1}^{N} [y_i log(x_i;\beta) + (1-y_i) log(x_i;\beta)] - \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Sử dụng các giá trị khác nhau của tham số λ, phương pháp lasso thu nhỏ giá trị ước lượng của các β so với phương pháp tối đa hóa hàm hợp lý truyền thống. Vì các giá trị thu nhỏ của β có thể giảm về 0 và loại ra khỏi mô hình nếu λ đủ lớn, phương pháp lasso có thể được dùng để thay thế cho việc chọn biến trong các mô hình đa biến.

Đồng thời đối với lasso, ta thường không ràng buộc các hệ số chặn, và các biến giải thích phải được chuẩn hóa để ràng buộc chung cho các β là có ý nghĩa. Vì hàm tối ưu của chúng ta là hàm lõm, lời giải có thể tính sử dụng các phương pháp phi tuyến.

2.1.1.3 Diễn giải kết quả ước lượng mô hình

2.1.2 Mô hình SVM (Support Vector Machine)

2.1.2.1 Khái niệm

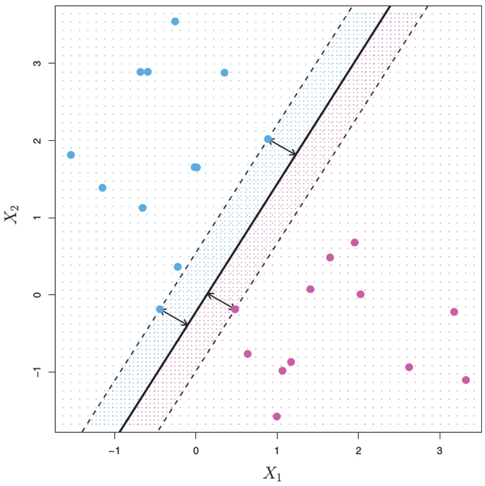
Thay vì đi tìm mô mô hình ước lượng tỷ lệ $P(Y=1)$ như trong mô hình logit. một hướng tiếp cận khác là đi tìm một siêu mặt phẳng có khả năng chia cắt không gian của bộ số liệu ra làm 2 phần. Nói cách khác là ước lượng một hàm:

$$f(x) = \beta_0 + \beta 1 x_1 + \beta_2 x2 ... \beta_p x_p$$

sao cho các quan sát thuộc 2 nhóm khác nhau sẽ được quyết định bằng dấu của $f(x)$, tức là năm ở 2 phía của siêu mặt phẳng:

$$\beta_0 + \beta 1 x_1 + \beta_2 x2 ... \beta_p x_p = 0$$

Mô hình véc tơ máy hỗ trợ (Support Vector Machine - SVM) là một trong những mô hình thuộc loại này. SVM phát triển từ những năm 1990 và nhanh chóng được mọi người đón nhận vì khả năng phân loại tốt trong nhiều trường hợp khác nhau.



Hình 2.1: Ví dụ về siêu mặt phẳng lề cực đại trong không gian 2 chiều.

a, Phân loại lề cực đại (Maximal Marginal Classifier)

Mô hình SVM được phát triển từ một phương pháp phân loại khá đơn giản gọi là phương pháp maximal margin classifier (Boser, Guyon, and Vapnik 1992). Thông thường, nếu như một bộ số liệu có thể được chia ra bởi một siêu mặt phẳng ngăn cách, chúng ta sẽ có thể tìm được vô siêu mặt phẳng như thế. Điều này là do các mặt phẳng có di chuyển nhẹ lên xuống hoặc quay mà không chạm tới các quan sát. Để xây dựng một mô hình phân loại dựa trên một siêu mặt phẳng phân loại, chúng ta phải có một phương pháp hợp lý để chọn mặt phẳng hợp lý trong số vô số siêu mặt phẳng này.

Phương pháp Maximal Marginal Classifier lựa chọn siêu mặt phẳng mà nằm xa nhất các quan sát trong bộ số liệu. Nếu chúng ta tính khoảng cách từ các quan sát tới siêu mặt phẳng đã cho, khoảng cách nhỏ nhất từ các quan sát đến siêu mặt phẳng này gọi là lề (margin) của siêu mặt phẳng. Siêu mặt phẳng lề cực đại mà chúng ta chọn trong phương pháp này là siêu mặt phẳng mà lề là lớn nhất.

Siêu mặt phẳng được ước lượng bằng cách giải phương trình:

$$\begin{cases} \max\limits_{\beta_0,\beta_1,\ldots,\beta_p} M \\ \sum_{j=1}^{p} \beta_j^2 = 1 \\ y_i(\beta_0 + \beta_1 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \qquad \forall i = 1,2,\ldots n \end{cases}$$

trong đó M là độ rộng của lề, chúng ta tìm giá trị tối đa cho giá trị này, dưới ràng buộc rằng

$$y_i(\beta_0 + \beta_1 x_{i2} + ... + \beta_p x_{ip}) \geq M \qquad \forall i = 1, 2, ...n$$

để đảm bảo rằng mỗi quan sát đều nằm trên đúng phía của siêu mặt phẳng phân loại, với điều kiện rằng M không âm. Khoảng cách từ quan sát thứ i đến siêu mặt phẳng có thể được tính bằng:

$$y_i(\beta_0 + \beta_1 x_{i2} + ... + \beta_p x_{ip})$$

b, Phương pháp phân loại hỗ trợ máy

Phương pháp phân loại lề cực đại không thể thực hiện được trong trường hợp không tồn tại một siêu mặt phẳng nào có thể chia tách bộ số liệu ra thành 2 nhóm.

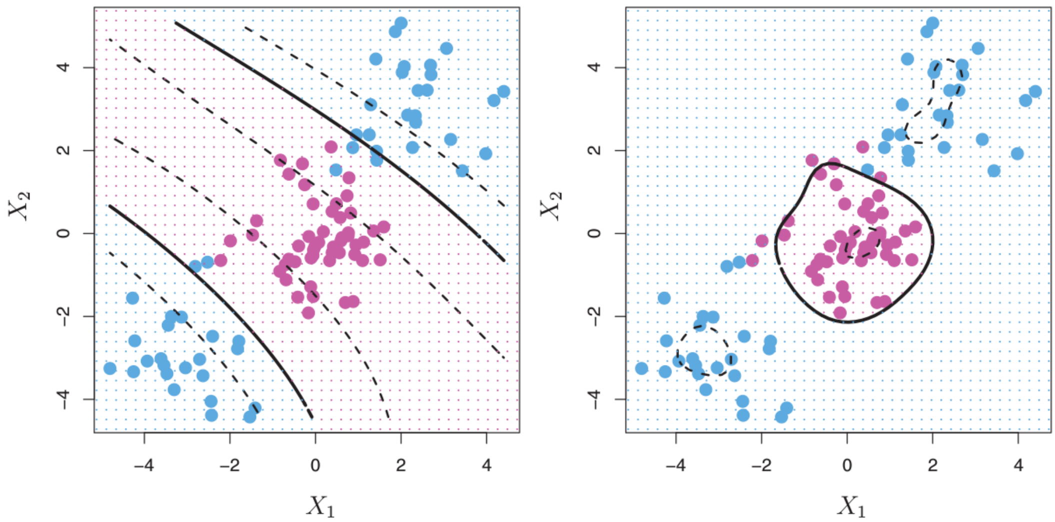
Một cải tiến của phương pháp này là phương pháp hỗ trợ máy, trong đó siêu mặt phẳng được ước lượng bằng phương trình:

$$\begin{cases} \max\limits_{\beta_0,\beta_1,...,\beta_p,\varepsilon_1,...,\varepsilon_n} M \\ \sum_{j=1}^{p} \beta_j^2 = 1 \\ y_i(\beta_0 + \beta_1 x_{i2} + ... + \beta_p x_{ip}) \geq M(1 - \varepsilon_i) \\ \varepsilon_i \geq 0, \quad \sum_{i=1}^{n} \varepsilon_i \leq C \end{cases}$$

Trong đó C là một tham số không âm do ta lựa chọn. ε_i là các hệ số không âm cho phép quan sát thứ i vi phạm quy tắc của siêu mặt phẳng biên cực đại, với C là giới hạn cho số sai phạm này. Nếu $0 < \varepsilon_i < 1$, điểm i nằm ở phía trong biên nhưng vẫn đúng phía của siêu mặt phẳng phân loại. Nếu $\varepsilon_i = 1$, điểm i nằm ở sai phía của siêu mặt phẳng phân loại.

c, Các kernel và mô hình véc tơ hỗ trợ máy

Mô hình véc tơ hỗ trợ máy (Support Vector Machine) là một mở rộng của mô hình phân loại vectơ hỗ trợ, giúp xây dựng các mô hình phân loại phi tuyến. Mô hình này sử dụng một phép chiếu Φ, chiếu các quan sát từ một không gian không phân biệt tuyến tính lên một chiều không gian mới mà ở đó các quan sát trở nên phân biệt tuyến tính.



Hình 2.2: Ví dụ về phân loại sử dụng mô hình SVM sử dụng kernel đa thức với d = 3 (trái) và kernel tròn (phải)

Trong thực tế, việc thực hiện phép chiếu Φ này có thể trở nên rất khó khăn khi kích cỡ của bộ số liệu lớn. Schölkopf and Burges (1999) chỉ ra rằng đối với một số phép chiếu, để ước lượng được mô hình véc tơ hỗ trợ, chúng ta chỉ cần tính được tích vô hướng của các quan sát trong bộ dữ liệu, thủ thuật này được gọi là thủ thuật kernel. Mô hình véc tơ hỗ trợ sử dụng thủ thuật kernel này được gọi là mô hình véc tơ hỗ trợ máy (SVM). Một cách tổng quát, mô hình SVM ước lượng:

$$f(x) = \beta_0 + \sum \alpha_i K(x, x_i)$$

trong đó $\beta 0$, α_i là các hệ số cần ước lượng, $K(x, x_i)$ được gọi là hàm kernel, x_i và x lần lượt là véc tơ của quan sát thứ i trong bộ số liệu và vectơ của quan sát mới cần phân loại.

Một số hàm kernel phổ biến được sử dụng rộng rãi là:

- $K(x, x_i) = x_i^T x$: Hàm kernel tuyến tính, đây thực chất là hàm kernel của mô hình phân loại vec tơ hỗ trợ bình thường.
- $K(x, x_i) = (1 + \sum_{j=1}^{p} x_i x)^d$: Hàm kernel đa thức với tham số d là bậc của đa thức ước lượng.
- $K(x, x_i) = e^{-\sigma \|x - x_i\|^2}$: Hàm kernel tròn với tham số σ quyết định bán kính của đường tròn phân chia các quan sát

2.1.2.2 Quy trình xây dựng một mô hình véc tơ máy hỗ trợ

Trong xây dựng một mô hình SVM, việc lựa chọn kernel thích hợp và việc điều chỉnh để có được tham số thích hợp cho kernel đó là yếu tố vô cùng quan trọng. Hsu, Chang, and Lin (2003) đưa ra một quy trình để có thể đạt được kết quả chấp nhận được đối với hầu hết các trường hợp. Quy trình đó như sau:

- Thực hiện chuẩn hoá bộ số liệu
- Ưu tiên sử dụng kernel tròn (RBF): $K(x, x_i) = e^{-\sigma\|x-x_i\|^2}$
- Sử dụng kiểm định chéo để tìm tham số tốt nhất cho C và σ.
- Sử dụng tham số C và σ tốt nhất ước lượng được để ước lượng trên toàn bộ bộ số liệu dùng để xây dựng mô hình.
- Kiểm tra hiệu quả mô hình.

Trong bài này ta sẽ đi theo quy trình này để xây dựng mô hình SVM, sử dụng phần mềm `LIBSVM` Chang and Lin (2011), một phần mềm phổ biến được sử dụng rộng rãi để xây dựng các mô hình SVM.

CHƯƠNG III

TÌNH HUỐNG NGHIÊN CỨU

3.1 SỐ LIỆU VÀ CÁC BIẾN SỐ

3.2 ỨNG DỤNG MÔ HÌNH LOGIT

3.3 ỨNG DỤNG MÔ HÌNH SVM

CHƯƠNG IV

KẾT LUẬN

PHỤ LỤC A

THÔNG TIN VỀ PHIÊN LÀM VIỆC TRÊN R

```
## R version 3.4.0 (2017-04-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.2 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8    LC_NUMERIC=C
##  [3] LC_TIME=vi_VN           LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=vi_VN       LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=vi_VN          LC_NAME=C
##  [9] LC_ADDRESS=C            LC_TELEPHONE=C
## [11] LC_MEASUREMENT=vi_VN    LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets
## [6] methods   base
##
## other attached packages:
##  [1] glmnet_2.0-5    foreach_1.4.3   Matrix_1.2-8
##  [4] ggfortify_0.4.1 GGally_1.3.0    caret_6.0-76
##  [7] ggplot2_2.2.1   lattice_0.20-35 broom_0.4.2
## [10] dplyr_0.5.0     tidyr_0.6.1     readr_1.1.0
## [13] knitr_1.15.1
##
```

```
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.10       RColorBrewer_1.1-2
##  [3] nloptr_1.0.4       compiler_3.4.0
##  [5] plyr_1.8.4         iterators_1.0.8
##  [7] tools_3.4.0        lme4_1.1-13
##  [9] evaluate_0.10      tibble_1.3.0
## [11] gtable_0.2.0       nlme_3.1-131
## [13] mgcv_1.8-17        psych_1.7.3.21
## [15] DBI_0.6-1          parallel_3.4.0
## [17] SparseM_1.77       gridExtra_2.2.1
## [19] stringr_1.2.0      MatrixModels_0.4-1
## [21] hms_0.3            stats4_3.4.0
## [23] grid_3.4.0         nnet_7.3-12
## [25] reshape_0.8.6      R6_2.2.0
## [27] foreign_0.8-67     minqa_1.2.4
## [29] reshape2_1.4.2     car_2.1-4
## [31] magrittr_1.5       splines_3.4.0
## [33] scales_0.4.1       codetools_0.2-15
## [35] ModelMetrics_1.1.0 MASS_7.3-47
## [37] assertthat_0.2.0   pbkrtest_0.4-7
## [39] mnormt_1.5-5       colorspace_1.3-2
## [41] quantreg_5.33      stringi_1.1.5
## [43] lazyeval_0.2.0     munsell_0.4.3
```

TÀI LIỆU THAM KHẢO

1. R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

2. Wickham, Hadley (2017). *tidyverse: Easily Install and Load 'Tidyverse' Packages*. R package version 1.1.1. URL: https://CRAN.R-project.org/package=tidyverse.

3. Jed Wing, Max Kuhn. Contributions from et al. (2016). *caret: Classification and Regression Training*. R package version 6.0-73. URL: https://CRAN.R-project.org/package=caret.

4. Lessmann, Stefan et al. (2015). "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research". In: *European Journal of Operational Research* 247.1, pp. 124–136.

5. Xie, Yihui (2015). *Dynamic Documents with R and knitr*. 2nd. ISBN 978-1498716963. Boca Raton, Florida: Chapman and Hall/CRC. URL: http://yihui.name/knitr/.

6. Hosmer Jr, David W, Stanley Lemeshow, and Rodney X Sturdivant (2013). *Applied logistic regression*. Vol. 398. John Wiley & Sons.

7. Chang, Chih-Chung and Chih-Jen Lin (2011). "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27.

8. Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1, pp. 1–22. URL: http://www.jstatsoft.org/v33/i01/.

9. Thomas, Lyn C (2010). "Consumer finance: Challenges for operational research". In: *Journal of the Operational Research Society* 61.1, pp. 41–52.

10. Xiao, Wenbing, Qian Zhao, and Qi Fei (2006). "A comparative study of data mining methods in consumer loans credit scoring management". In: *Journal of Systems Science and Systems Engineering* 15.4, pp. 419–435.

11. Karatzoglou, Alexandros et al. (2004). "kernlab – An S4 Package for Kernel Methods in R". In: *Journal of Statistical Software* 11.9, pp. 1–20. URL: http://www.jstatsoft.org/v11/i09/.

12. Baesens, Bart et al. (2003). "Benchmarking state-of-the-art classification algorithms for credit scoring". In: *Journal of the operational research society* 54.6, pp. 627–635.

13. Hsu, Chih-Wei, Chih-Chung Chang, Chih-Jen Lin, et al. (2003). "A practical guide to support vector classification". In:

14. Schölkopf, Bernhard and Christopher JC Burges (1999). *Advances in kernel methods: support vector learning*. MIT press.

15. Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, pp. 144–152.