Stockholms universitet

A Simulation Study Comparing Ridge Logistic Regression and Crammer–Singer Support Vector Machine for Multiclass Classification

Adam Carlén

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2026:17
Matematisk statistik
September 2026

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Bachelor thesis **2026:17**
https://www.math.su.se

A Simulation Study Comparing Ridge Logistic Regression and Crammer–Singer Support Vector Machine for Multiclass Classification

Adam Carlén*

September 2026

Abstract

This work compares ridge multinomial logistic regression and the Crammer–Singer support vector machine, two standard methods for multiclass classification. While the two are derived from quite different arguments, the first part of this work shows that they nonetheless lead to optimisation problems of the same form, the difference being mainly in the loss function used. Here we compare the two empirically on simulated multiclass problems for which the Bayes-optimal classifier is known and linear, spanning 300 conditions varying the number of classes, dimension, sample size, class separation and correlation. The two methods turned out to perform very similarly. The mean paired difference in test accuracy never exceeded 5.3 percentage points in magnitude, and stayed below one percentage point in the large majority of conditions. The one difference regarded as reliable within the present design is a small advantage for the Crammer–Singer support vector machine at strong correlation between variables. The overall conclusion made here is that for data of this kind, the choice between the logistic and the hinge loss matters little for classification accuracy.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: adam_carlen@hotmail.com. Supervisor: Martina Favero, Jan-Olov Persson.

Acknowledgements

I would like to thank my supervisors, Jan Olov Persson and Martina Favero, for their guidance and valuable feedback throughout the writing of this thesis. I would also like to thank my dear friend Lasse for always being willing to discuss ideas and brainstorm with me. Finally, I am very grateful to Olle and Simone for generously letting me use their computer on several occasions to run simulations, which was a great help in completing this work.

AI tools, specifically Claude (Anthropic), were used for assistance throughout the work on this project. The main uses were reviewing drafts written by the author, debugging code used for simulations in R, help producing the figures and tables, as well as LaTeX formatting and language proofreading. All theoretical content, simulation design, analysis and conclusions are the author's own.

Contents

1 Introduction

Classification is one of the basic tasks of statistical learning. From a set of observations whose class membership is known, one learns a rule that may then be used to assign a class to observations whose class is unknown. One of the most established methods for constructing such a rule is logistic regression, or, in the case of several classes, multinomial logistic regression, which postulates a parametric form for the conditional class probabilities and estimates its parameters by maximum likelihood. A more recent method is the support vector machine (Cortes and Vapnik, 1995), which at heart is a geometric method rather than probabilistic. A support vector machine instead seeks hyperplanes separating the classes of training data with the largest, if possible, margin.

Even though the two are motivated by quite different arguments, they arrive at similar objects. Both assign an observation $\boldsymbol{x}$ to the class that maximizes a linear function of that observation, that is,

$$\hat{h}(\boldsymbol{x}) = \arg\max_k \boldsymbol{w}_k^\top \boldsymbol{x},$$

where the coefficient vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ are what each method estimates from the training data. In practice, fitting logistic regression by maximum likelihood alone can produce unstable coefficient estimates, which is why it is common practice to add a penalty on the size of the coefficients. When doing so, the respective optimisation problems of the two methods take a strikingly similar form, namely a loss summed over the training observations plus a quadratic penalty on the coefficients, the only difference between them being which loss function is used. This brings us to the question raised in the present work, how much does the choice of loss matter for classification accuracy?

We approach this question by simulation. For a simulated classification problem, a mean vector for each class is first drawn at random from a normal distribution about the origin. Equally many observations are then drawn for each class, normally distributed about the respective class means, with shared covariance. For data of this kind the Bayes-optimal classifier is a linear rule of exactly the form above. Both methods are therefore in principle capable of expressing the optimal classifier, leaving the loss as the main structural difference between them. For the support vector machine, we use the multiclass formulation of Crammer and Singer (2001), which handles all K classes within a single optimisation problem, keeping the symmetry with multinomial logistic regression.

The goal of this work is to compare the test accuracy of the two methods on problems of this kind, and to describe how the comparison depends on the number of classes K, the number of variables p, the number of observations, how far apart the class means are drawn, and how strongly the variables are correlated. The conditions simulated are divided into two sets, by the ratio of training sample size to number of variables. The first one targeting conditions in which the training sample size is large relative to the number of variables, and the second one targeting conditions where the training sample size is comparable to or smaller than the number of variables. The accuracy attainable by the Bayes-optimal classifier is estimated and reported for every condition.

Section 2 develops the Bayes-optimal benchmark, the two methods, and the sense in which they differ mainly in their loss. Section 3 describes the data-generating process and the simulation procedure, Section 4 presents the results together with diagnostics for the hyperparameter tuning, Section 5 discusses them and Section 6 concludes.

2 Theory

This section develops the theoretical background for the two classifiers compared in the simulation study, and the Bayes-optimal benchmark against which both will be measured.

2.1 The multiclass classification problem

In this work we are concerned with data points of the form $(y, x_1, x_2, ..., x_p)$ where p is some positive integer. Here y denotes the class label, which is assumed to be known and thought of as the dependent variable, also called the response variable, while the vector $\boldsymbol{x} = (x_1, x_2, ..., x_p)$ denotes the independent variables, sometimes called features, or the feature vector. We will restrict our attention to the case where all feature variables take real values, so $\boldsymbol{x} \in \mathbb{R}^p$. If we further assume that each data point belongs to one of K classes, the multiclass classification problem entails to from a set of such data points, learn a function $h : \mathbb{R}^p \to \{1, 2, ..., K\}$ that generalises well to new data points with unknown label, i.e., points of the form $(\cdot, x_1, x_2, ..., x_p)$.

2.2 Bayes optimal classifier and Linear Discriminant Analysis

For the multiclass classification problem of K classes on feature space $\mathbb{R}^p$, a function $h : \mathbb{R}^p \to \{1, 2, ..., K\}$ is called a *classifier*. If we consider the data $(Y, \boldsymbol{X})$ as random with some joint probability distribution, the *misclassification rate* for a given classifier is defined as

$$\varepsilon(h) = \Pr(h(\boldsymbol{X}) \neq Y). \tag{1}$$

Among all possible classifiers, the *Bayes classifier*, given by the formula

$$h^*(\boldsymbol{x}) = \arg \max_{k \in \{1,...,K\}} \Pr(Y = k \mid \boldsymbol{X} = \boldsymbol{x}) \tag{2}$$

minimises the misclassification rate (James et al., 2021, Section 2.2.3). The misclassification rate of Bayes classifier, $\varepsilon^* := \varepsilon(h^*)$ is called the *Bayes error* for the classification problem of this joint distribution of $(Y, \boldsymbol{X})$.

To compute using the Bayes classifier one must have knowledge of the conditional probabilities $\Pr(Y = k \mid \boldsymbol{X} = \boldsymbol{x})$. However in the case of generative models, and as is the case in the simulation study in Section 3, it is typical to instead have knowledge of $\pi_k = \Pr(Y = k)$, as well as the density of $\boldsymbol{X} = \boldsymbol{x}$ given $Y = k$ which we denote by $f_k(\boldsymbol{x})$. Having these two, Bayes theorem may be employed to write

$$\Pr(Y = k | \boldsymbol{X} = \boldsymbol{x}) = \frac{\pi_k f_k(\boldsymbol{x})}{\sum_{i=1}^K \pi_i f_i(\boldsymbol{x})}. \tag{3}$$

In particular, since the simulation design is balanced we have that $\pi_k = \frac{1}{K}$ for each k. Moreover, each $f_k(\boldsymbol{x})$ is the density of a multivariate normal distribution with class specific mean $\boldsymbol{\mu}_k$ and common covariance matrix Σ

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\Big(-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\Big). \tag{4}$$

Inserting this in (3) gives

$$\Pr(Y = k | \boldsymbol{X} = \boldsymbol{x}) = \frac{\exp\Big(-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\Big)}{\sum_{i=1}^K \exp\Big(-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)\Big)}$$

where the denominator does not depend on k, thus the problem of maximising the quantity in equation (3) reduces to maximising

$$\exp\Big(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\Big) \tag{5}$$

over k. Taking the logarithm of (5) and expanding gives

$$\begin{aligned}-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) &= -\frac{1}{2}\Big(\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{x}-\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{\mu}_k-\boldsymbol{\mu}_k^\top\Sigma^{-1}\boldsymbol{x}+\boldsymbol{\mu}_k^\top\Sigma^{-1}\boldsymbol{\mu}_k\Big)\\ &= -\frac{1}{2}\Big(\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{x}-2\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{\mu}_k+\boldsymbol{\mu}_k^\top\Sigma^{-1}\boldsymbol{\mu}_k\Big)\end{aligned} \tag{6}$$

where in the last equality we use that Σ, and hence Σ^{-1}, is symmetric. Since the term $\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{x}$ does not depend on k we conclude that in our case

$$\arg\max_{k\in\{1,\dots,K\}}\Pr(Y=k\mid\boldsymbol{X}=\boldsymbol{x}) = \arg\max_{k\in\{1,\dots,K\}}\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{\mu}_k-\frac{1}{2}\boldsymbol{\mu}_k^\top\Sigma^{-1}\boldsymbol{\mu}_k. \tag{7}$$

The expression appearing inside the argmax on the right-hand side of (7) is called the *linear discriminant function*

$$\delta_k(\boldsymbol{x}) := \boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{\mu}_k-\tfrac{1}{2}\boldsymbol{\mu}_k^\top\Sigma^{-1}\boldsymbol{\mu}_k, \tag{8}$$

and the derivation above shows that the Bayes classifier reduces to

$$h^*(\boldsymbol{x}) = \arg\max_{k\in\{1,\dots,K\}}\delta_k(\boldsymbol{x}). \tag{9}$$

Moreover, the decision boundaries, say the boundary between the two different classes k and k' becomes

$$\{\boldsymbol{x}\in\mathbb{R}^p : \delta_k(\boldsymbol{x})=\delta_{k'}(\boldsymbol{x})\}, \tag{10}$$

that is, a hyperplane in $\mathbb{R}^p$. This resulting classifier is known as *linear discriminant analysis* (LDA) (James et al., 2021, §4.4.2). It is the Bayes-optimal rule for the generative model considered and will therefore serve as the theoretical benchmark in our simulation study in Section 3.

2.3 Multinomial logistic regression

2.3.1 Model and likelihood

Multinomial logistic regression is the canonical generalization of binary logistic regression to an arbitrary number of classes K, and is the form implemented in standard statistical software (Friedman et al., 2010). Unlike the setup in Section 2.2, where the conditional probabilities were derived from a generative model via Bayes theorem, we assume that the conditional probability for class $k\in\{1,2,...,K\}$ given a feature vector $\boldsymbol{x}\in\mathbb{R}^p$ takes the following specific parametric form:

$$\Pr(Y=k\mid\boldsymbol{X}=\boldsymbol{x};\boldsymbol{B}) \;=\; \frac{\exp(\boldsymbol{b}_k^\top\boldsymbol{x})}{\sum_{j=1}^K\exp(\boldsymbol{b}_j^\top\boldsymbol{x})}, \qquad \boldsymbol{B}=[\boldsymbol{b}_1,\dots,\boldsymbol{b}_K]\in\mathbb{R}^{p\times K}. \tag{11}$$

Here the parameter is the matrix $\boldsymbol{B}$. For convenience we set the first entry of $\boldsymbol{x}$ equal to 1 to accommodate for a constant or intercept. The relations in (11) are called the *softmax* relation and the resulting classifier assigns a point $\boldsymbol{x}$ to the class with the largest softmax probability. Specifically, since the denominator is constant in k and exp is monotone we have

$$\hat{h}(\boldsymbol{x}) = \arg\max_{k\in\{1,\dots,K\}}\boldsymbol{b}_k^\top\boldsymbol{x}, \tag{12}$$

and consequently the pairwise decision boundaries

$$\{\boldsymbol{x}\in\mathbb{R}^p : (\boldsymbol{b}_k-\boldsymbol{b}_{k'})^\top\boldsymbol{x}=0\}$$

are hyperplanes in $\mathbb{R}^p$. This mirrors the linear-discriminant form of the Bayes classifier in Section 2.2, though here the coefficients are estimated from data rather than derived from a generative model. The parameter matrix $\boldsymbol{B}$ is estimated from data using the maximum likelihood method (MLE). The remainder of this section first derives an expression for the log-likelihood function $\ell(\boldsymbol{B})$ (17), and then establishes that finding the MLE is equivalent to solving a certain convex optimisation problem (18).

Suppose we have N data points $(y_i, \boldsymbol{x}_i)$ indexed by $i \in \{1, 2, ..., N\}$. Defining for each i the indicator function

$$\mathbf{1}_i(k) = \begin{cases} 1 & \text{if } y_i = k, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

and using the shorthand $p_k(\boldsymbol{x}_i; \boldsymbol{B}) = \Pr(Y = k \mid \boldsymbol{X} = \boldsymbol{x}_i; \boldsymbol{B})$ we can write

$$p_{y_i}(\boldsymbol{x}_i; \boldsymbol{B}) = \prod_{\ell=1}^{K} p_\ell(\boldsymbol{x}_i; \boldsymbol{B})^{\mathbf{1}_i(\ell)} \tag{14}$$

for each i. Assuming the y values of our observations are conditionally independent, we are allowed to write their joint likelihood function as the product

$$\mathcal{L}(\boldsymbol{B}) = \prod_{i=1}^{N} \prod_{\ell=1}^{K} p_\ell(\boldsymbol{x}_i; \boldsymbol{B})^{\mathbf{1}_i(\ell)} \tag{15}$$

as well as the log-likelihood

$$\ell(\boldsymbol{B}) = \log \left[\prod_{i=1}^{N} \prod_{\ell=1}^{K} p_\ell(\boldsymbol{x}_i; \boldsymbol{B})^{\mathbf{1}_i(\ell)} \right] = \sum_{i=1}^{N} \sum_{\ell=1}^{K} \mathbf{1}_i(\ell) \log \left(p_\ell(\boldsymbol{x}_i; \boldsymbol{B}) \right). \tag{16}$$

Expanding $p_\ell(\boldsymbol{x}_i; \boldsymbol{B})$ in the summand using (11) gives

$$\log p_\ell(\boldsymbol{x}_i; \boldsymbol{B}) = \log \frac{\exp(\boldsymbol{b}_\ell^\top \boldsymbol{x}_i)}{\sum_{j=1}^{K} \exp(\boldsymbol{b}_j^\top \boldsymbol{x}_i)} = \boldsymbol{b}_\ell^\top \boldsymbol{x}_i - \log \sum_{j=1}^{K} \exp(\boldsymbol{b}_j^\top \boldsymbol{x}_i).$$

Substituting this into the log-likelihood yields

$$\begin{aligned} \ell(\boldsymbol{B}) &= \sum_{i=1}^{N} \sum_{\ell=1}^{K} \mathbf{1}_i(\ell) \left[\boldsymbol{b}_\ell^\top \boldsymbol{x}_i - \log \sum_{j=1}^{K} \exp(\boldsymbol{b}_j^\top \boldsymbol{x}_i) \right] \\ &= \sum_{i=1}^{N} \left[\sum_{\ell=1}^{K} \mathbf{1}_i(\ell)\, \boldsymbol{b}_\ell^\top \boldsymbol{x}_i \;-\; \log \sum_{j=1}^{K} \exp(\boldsymbol{b}_j^\top \boldsymbol{x}_i) \right] \end{aligned} \tag{17}$$

by rearranging and using the identity $\sum_{\ell=1}^{K} \mathbf{1}_i(\ell) = 1$ for each i. Thus, we have found that finding the likelihood maximiser, $\hat{\boldsymbol{B}}_{\text{MLE}}$ is equivalent to solving the following optimisation problem.

$$\hat{\boldsymbol{B}}_{\text{MLE}} = \arg \min_{\boldsymbol{B} \in \mathbb{R}^{p \times K}} -\ell(\boldsymbol{B}) = \arg \min_{\boldsymbol{B} \in \mathbb{R}^{p \times K}} \sum_{i=1}^{N} \left[\log \sum_{j=1}^{K} \exp(\boldsymbol{b}_j^\top \boldsymbol{x}_i) \;-\; \sum_{\ell=1}^{K} \mathbf{1}_i(\ell)\, \boldsymbol{b}_\ell^\top \boldsymbol{x}_i \right]. \tag{18}$$

The negative log-likelihood $-\ell(\boldsymbol{B})$ is convex in $\boldsymbol{B}$ so the minimisation problem is well posed (Hastie et al., 2009, §4.4.1). However it turns out there is no closed form for $\hat{\boldsymbol{B}}_{\text{MLE}}$, which is why in this work the minimizer is computed numerically using the R package `glmnet`.

2.3.2 Ridge (ℓ_2) penalisation

The maximum-likelihood method for estimating coefficients as described in 2.3.1 may exhibit high variance when the dimension p of the feature space is high compared to the number of observations N, and may also generalise poorly to new data (Hastie et al., 2009, §18.1). This is of particular interest in this work, since a subset of the simulation conditions (Set II, Section 3) targets precisely such a regime. To combat this, we follow the standard approach of *regularisation* (James et al., 2021, §6.2.1), by adding a quadratic penalty term to the negative log-likelihood in the optimisation problem (18). This yields the *ridge multinomial logistic regression* estimator

$$\hat{\boldsymbol{B}}_\lambda \;=\; \arg\min_{\boldsymbol{B}\in\mathbb{R}^{p\times K}} \Big\{ -\ell(\boldsymbol{B}) + \lambda \|\boldsymbol{B}\|_F^2 \Big\}. \tag{19}$$

Here $\|\boldsymbol{B}\|_F^2 = \sum_{k=1}^{K}\sum_{j=1}^{p} B_{jk}^2$ is the squared Frobenius norm and the parameter $\lambda \geq 0$ controls the strength of the regularisation term. The hyperparameter λ is selected by *cross-validation* as described in Section 3. An additional motivation for this particular choice of penalty is that the Crammer–Singer SVM introduced in Section 2.4 uses the same ℓ_2 regulariser, so that the comparison between the two methods is symmetric. As described in James et al. (2021, §6.2.1), standardisation of the feature vectors is necessary when fitting ridge- penalised models. In this work, the feature vectors $\boldsymbol{x}_i$ are standardised to zero mean and unit variance using training-set statistics prior to fitting as described in Section 3.

2.4 Multiclass support vector machines

2.4.1 The separable case

As in the linear discriminant analysis of Section 2.2 and the multinomial logistic regression of Section 2.3.1, fitting a multiclass support vector machine also produces a linear classifier. That is, linear in the sense that the classifier is of the form

$$\hat{h}(\boldsymbol{x}) = \arg\max_{k\in\{1,\ldots,K\}} \boldsymbol{w}_k^\top \boldsymbol{x}, \qquad \boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K] \in \mathbb{R}^{p\times K}. \tag{20}$$

where the object to estimate from data is the coefficient matrix $\boldsymbol{W}$. As before, we set the first entry of $\boldsymbol{x}$ equal to 1 to absorb an intercept for each class into $\boldsymbol{w}_k$. The resulting pairwise decision boundaries between two different classes k and k' are again hyperplanes,

$$\{\boldsymbol{x} \in \mathbb{R}^p : (\boldsymbol{w}_k - \boldsymbol{w}_{k'})^\top \boldsymbol{x} = 0\}. \tag{21}$$

Before stating the optimisation problem that defines the Crammer–Singer SVM, we develop the geometric picture that motivates it. As in the binary case, the SVM seeks separating hyperplanes with the largest margin (see e.g. James et al., 2021, §9.1–9.2). We therefore begin by considering the idealised setting in which the training classes are completely separable by such hyperplanes.

Given training data $(\boldsymbol{x}_i, y_i)_{i=1}^N$ with $y_i \in \{1, \ldots, K\}$, we define the *algebraic margin*,

$$m_i(\boldsymbol{W}) \;=\; \boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i - \max_{k\neq y_i} \boldsymbol{w}_k^\top \boldsymbol{x}_i, \tag{22}$$

and note that the sign of the margin corresponds to whether a data point will be correctly classified (positive), incorrectly classified (negative) or a tie (zero). A training data set is called *linearly separable* if there exists a $\boldsymbol{W}$ such that $m_i(\boldsymbol{W}) > 0$ for all i. The condition that $m_i(\boldsymbol{W}) > 0$ for all i can be strengthened if we observe that scaling $\boldsymbol{W}$ by a positive constant $c > 0$, yields the same classifier in (20), while margins change like $m_i(c\boldsymbol{W}) = c m_i(\boldsymbol{W})$. Now if $\boldsymbol{W}$ is a matrix that linearly separates data, and we define $m^* = \min_i m_i(\boldsymbol{W})$ and $\boldsymbol{W}' = \frac{1}{m^*}\boldsymbol{W}$, we get that

$$m_i(\boldsymbol{W}') = \frac{1}{m^*} m_i(\boldsymbol{W}) \geq 1 \tag{23}$$

since $m_i(\boldsymbol{W}) \geq m^*$ for each i. Importantly $\boldsymbol{W}'$ yields the same classifier as $\boldsymbol{W}$ but with every margin guaranteed to be at least 1.

The perpendicular distance (with sign) from a point $\boldsymbol{x}_i$ to the pairwise decision boundary between its true class y_i and some other class $r \neq y_i$ is given by

$$\gamma_{i,r}(\boldsymbol{W}) \;=\; \frac{(\boldsymbol{w}_{y_i} - \boldsymbol{w}_r)^\top \boldsymbol{x}_i}{\|\boldsymbol{w}_{y_i} - \boldsymbol{w}_r\|}, \tag{24}$$

which we shall call the *geometric margin* of observation i with respect to class r. In light of the derived constraint in (23) the numerator is at least 1, so we have

$$\gamma_{i,r}(\boldsymbol{W}) = \frac{(\boldsymbol{w}_{y_i} - \boldsymbol{w}_r)^\top \boldsymbol{x}_i}{\|\boldsymbol{w}_{y_i} - \boldsymbol{w}_r\|} \;\geq\; \frac{1}{\|\boldsymbol{w}_{y_i} - \boldsymbol{w}_r\|}. \tag{25}$$

We now seek to find a lower bound for every pairwise margin $\gamma_{i,r}(\boldsymbol{W})$ independent of i and r. By standard calculations one can show that the Frobenius-norm, $\|\boldsymbol{W}\|_F^2$ will do. That is,

$$\min_{i,\, r \neq y_i} \gamma_{i,r}(\boldsymbol{W}) \;\geq\; \frac{1}{\sqrt{2}\,\|\boldsymbol{W}\|_F}, \tag{26}$$

so minimising $\|\boldsymbol{W}\|_F$ maximises a simultaneous lower bound for every pairwise decision boundary, motivating the *hard-margin Crammer–Singer* optimisation problem (Crammer and Singer, 2001)

$$\hat{\boldsymbol{W}} \;=\; \arg\min_{\boldsymbol{W} \in \mathbb{R}^{p \times K}} \tfrac{1}{2}\|\boldsymbol{W}\|_F^2 \quad \text{subject to} \quad \boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i - \max_{k \neq y_i} \boldsymbol{w}_k^\top \boldsymbol{x}_i \geq 1 \quad \forall i. \tag{27}$$

In practice, it is not feasible to expect training data to be linearly separable, and classes typically have some overlap. Therefore, in the next section we shall introduce a modified version of (27) which relaxes this constraint, and which is the method used in the simulation study of Section 3.

2.4.2 The non separable case and Crammer–Singer SVM

In practice, training data are rarely linearly separable, and the constraints in the hard-margin problem (27) are typically infeasible. To address this, we introduce for each observation a slack variable $\xi_i \geq 0$ that measures the amount by which that observation is permitted to violate the unit margin,

$$m_i(\boldsymbol{W}) \;\geq\; 1 - \xi_i, \qquad \xi_i \geq 0. \tag{28}$$

The interpretation here is that if a slack variable equals zero, $\xi_i = 0$, the observations algebraic margin is at least one and so satisfies the constraint from the separable case (23). If $0 < \xi_i \leq 1$ the observation lies on the correct side of the true-class decision boundary but inside the margin, and if $\xi_i > 1$ the observation lies on the wrong side of the decision boundary and so will be misclassified.

Naturally we would like to minimize the total amount of slack needed, and thus we add the following penalty term to the minimisation objective from (27), obtaining the *soft-margin Crammer–Singer* primal (Crammer and Singer, 2001)

$$\begin{aligned} &\min_{\boldsymbol{W},\, \boldsymbol{\xi}} \tfrac{1}{2}\|\boldsymbol{W}\|_F^2 \;+\; C \sum_{i=1}^{N} \xi_i \\ &\text{subject to} \quad \boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i - \max_{k \neq y_i} \boldsymbol{w}_k^\top \boldsymbol{x}_i \geq 1 - \xi_i,\; \xi_i \geq 0 \quad \forall i. \end{aligned} \tag{29}$$

Here $C > 0$ controls the strength of the slack penalty.

The constrained problem in (29) can equivalently be written as a minimisation problem with a single objective, a fact that will turn out to be useful for the comparison with ridge logistic regression. Using the shorthand $[z]_+ = \max(z, 0)$, the estimated values ξ_i^* of the slack variables can be written $\xi_i^* = \big[\, 1 - m_i(\boldsymbol{W}) \,\big]_+$. Inserting this in (29) gives the equivalent unconstrained problem

$$\hat{\boldsymbol{W}}_C \;=\; \arg\min_{\boldsymbol{W} \in \mathbb{R}^{p \times K}} \left\{ \tfrac{1}{2}\|\boldsymbol{W}\|_F^2 \;+\; C \sum_{i=1}^{N} \big[\, 1 - m_i(\boldsymbol{W}) \,\big]_+ \right\}. \tag{30}$$

The term $[1 - m_i(\boldsymbol{W})]_+$ is called the multiclass *hinge loss*. It is zero whenever the algebraic margin is at least one, and otherwise grows linearly proportional to the amount by which the margin is violated. An observation with $m_i(\boldsymbol{W}) \geq 1$ thus contributes nothing to the objective, and in turn does not affect the resulting classifier. This shows that the resulting classifier is in fact only influenced by a (typically small) subset of the observations. These observations are called the *support vectors* of the resulting classifier, hence the name support vector machine.

The objective in (30) is convex in $\boldsymbol{W}$ but has no closed-form minimiser (see Boyd and Vandenberghe, 2004, §3), so the problem is solved numerically. Specifically, in the simulation study of Section 3, the optimisation problem problem (29) is solved using the R package `LiblineaR` (Helleputte et al., 2024; Fan et al., 2008), with `type = 4` and the hyperparameter C is selected by cross-validation as described in the same section.

2.5 Relationship between ridge LR and CS-SVM

Even though the two methods considered in this work were motivated quite differently, ridge multinomial logistic regression probabilistically via maximum likelihood, while Crammer–Singer SVM via a geometric margin maximisation argument, as we will show shortly they take a surprisingly similar form.

Specifically, dividing the CS-SVM objective in (30) by C and writing $\lambda_{\text{SVM}} = 1/(2C)$ for the resulting penalty coefficient gives

$$\hat{\boldsymbol{W}}_{\lambda_{\text{SVM}}} = \arg\min_{\boldsymbol{W}} \{\, L_{\text{hinge}}(\boldsymbol{W}) + \lambda_{\text{SVM}} \|\boldsymbol{W}\|_F^2 \,\},$$

where $L_{\text{hinge}}(\boldsymbol{W}) = \sum_i [1 - m_i(\boldsymbol{W})]_+$. The ridge LR estimator (19) has the same form, namely

$$\hat{\boldsymbol{B}}_{\lambda_{\text{LR}}} = \arg\min_{\boldsymbol{B}} \{\, L_{\text{LR}}(\boldsymbol{B}) + \lambda_{\text{LR}} \|\boldsymbol{B}\|_F^2 \,\},$$

with $L_{\text{LR}}(\boldsymbol{B}) = -\ell(\boldsymbol{B})$, the negative log-likelihood.

Both methods are of the form "method-specific loss function" + "Frobenius-norm penalty", they differ only in the choice of loss (hinge vs. logistic), and in this work the regularisation strength for both methods is selected by cross-validation (Section 3). Moreover, both methods produce linear classifiers of the form (20). Recalling from Section 2.2 that the Bayes-optimal classifier is linear under the Gaussian data-generating process with shared covariance used in this work, we see that both ridge LR and CS-SVM are in principle capable of recovering the Bayes-optimal classifier. Consequently, systematic differences in test accuracy may therefore be attributed to the loss function, and this is the question this work investigates empirically.

2.6 Cross-validation for hyperparameter tuning

Both methods introduced above depend on a regularisation hyperparameter, λ_{LR} for ridge logistic regression, or C (equivalently λ_{SVM}) for Crammer–Singer SVM, to be determined outside their respective optimisation problems. One standard tool for this is *K-fold cross-validation* (see

e.g. Hastie et al., 2009, §7.10), which estimates the expected prediction error of a fitted model at a given hyperparameter value from given training data.

Given a training set of N observations, and a hyperparameter value α, K-fold cross-validation partitions the observations into K roughly equal sized folds. For each fold $k = 1, \ldots, K$, the model is fit on the remaining $K - 1$ folds and then used to predict the held-out fold. Taking the average of the resulting misclassification rates across the K folds gives the cross-validation error estimate at that hyperparameter value α. Typically, and in Section 3 of this work, a range of values for α are evaluated this way and the value with the smallest average misclassification rate is selected. The final classifier is refit on the full training set at the selected hyperparameter value.

3 Method

3.1 Data-generating process

The data-generating process models a balanced K-class classification problem with real valued feature vectors, $\boldsymbol{x} \in \mathbb{R}^p$, where n_{pc} denotes number of observations per class, giving total sample size of $n = K \cdot n_{pc}$. First, class means $\boldsymbol{\mu}_k \in \mathbb{R}^p$ for $k = 1, \ldots, K$ are drawn from

$$\boldsymbol{\mu}_k \sim \mathcal{N}\left(0, \frac{\delta^2}{p}\boldsymbol{I}_p\right) \tag{31}$$

where $\boldsymbol{I}_p$ denotes the identity matrix of order p and $\delta > 0$ is a separation parameter controlling how far apart the class means are on average (see Section 3.5 for details). Given the class means, data points are drawn from

$$\boldsymbol{X}_i \mid Y_i = k \;\sim\; \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}), \qquad \Pr(Y_i = k) = \tfrac{1}{K}, \tag{32}$$

where the shared covariance of the classes is given by

$$\Sigma_{ij} \;=\; \rho^{|i-j|}, \qquad 0 \le \rho < 1, \tag{33}$$

that is, an AR(1) correlation structure. (Note that for $\rho = 0$ we have $\Sigma = \boldsymbol{I}_p$, so $\boldsymbol{X}_i \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{I}_p)$ is included as a special case).

3.2 Estimating the error rate of the Bayes-optimal classifier

The derivations of Section 2.2 show that, under the data generation process described in Section 3.1, the Bayes-optimal classifier is given by the linear discriminant rule (9) of Section 2.2. For a single instance of data simulated as above, once class means $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ are drawn (see Step 2 of Section 3.3), the Bayes-optimal classifier $h^*_{\boldsymbol{\mu}}$ is completely determined. The misclassification rate, $\varepsilon(h^*_{\boldsymbol{\mu}}) = \Pr(h^*_{\boldsymbol{\mu}}(\boldsymbol{X}) \neq Y)$, for this classifier could in principle be calculated, however it turns out there is no closed form expression for this when $K \geq 3$ (Genz and Bretz, 2009). We therefore turn to estimating it as follows(Step 3 of Section 3.3).

For drawn means $\boldsymbol{\mu}$, we generate $N = 10{,}000$ new observations $(\tilde{\boldsymbol{X}}_j, \tilde{Y}_j)$, $j = 1, \ldots, N$, as in Section 3.1, with an equal number of observations for each class. Note that this is separate from the data used to fit the respective models. Each data point is then classified using the known $h^*_{\boldsymbol{\mu}}$, and we record the fraction that are misclassified,

$$\widehat{\varepsilon}_{\boldsymbol{\mu}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}\{\, h^*_{\boldsymbol{\mu}}(\tilde{\boldsymbol{X}}_j) \neq \tilde{Y}_j \,\}. \tag{34}$$

By the law of large numbers (Gut, 2013), $\widehat{\varepsilon}_{\boldsymbol{\mu}}$ converges to $\varepsilon(h^*_{\boldsymbol{\mu}})$ as N grows, and the choice of N is motivated in Section 3.5.

3.3 Procedure

A replication of the simulation process takes as input one fixed condition $\{K, p, n_{pc}, \delta, \rho\}$ and then performs the following steps.

1. A seed for the replication is deterministically computed as a function of a base seed B, via $B + (i-1) \cdot R + r$, where i denotes the condition index and r denotes the replication index (more in Appendix A).
2. Class means $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$ are generated in accordance with eq (31).
3. Compute estimated Bayes error $\hat{\varepsilon}^*_{\boldsymbol{\mu}}$ as described in Section 3.2.
4. Exactly n_{pc} observations are drawn for each class in accordance with eq (32), making a balanced experiment design of $n = K \cdot n_{pc}$ data points in total.
5. The data is split into a training set and a test set 70/30 respectively. This is done stratified by class label, meaning exactly $\lfloor 0.7 \cdot n_{pc} \rfloor$ data points of each label are selected (uniformly randomly) for the training set and the remaining is held out as the test set. In particular, this means that the training data as well as the test data are also balanced.
6. Column means and standard deviations are computed on the training set only and then applied to both training and test sets.
7. Each method is tuned independently on the standardised training data using 5-fold cross-validation with misclassification rate as the selection criterion (As described in Section 2.6). The CV folds are drawn at random (unstratified) and are drawn independently for the two methods. The exact settings used can be found in Appendix A.
8. Each method is refit on the full training set at the tuned hyperparameter. For LR this means refitting at the CV-selected λ and for SVM at the CV-selected C.
9. The fitted classifiers from step 8 are evaluated on the held-out test set, and the following quantities are recorded for the replication: the test accuracies acc_{LR} and acc_{SVM}, their difference $\Delta_{\text{acc}} = \text{acc}_{\text{LR}} - \text{acc}_{\text{SVM}}$ (positive values indicate LR advantage), the CV-selected hyperparameters λ_{opt} and C_{opt} from step 7, the corresponding boundary-hit indicators for CV-tuning, and the Bayes error from step 3.

The above procedure is repeated $R = 50$ times per condition and the following aggregated quantities are recorded:

- Mean accuracies $\overline{\text{acc}}_{\text{LR}}$ and $\overline{\text{acc}}_{\text{SVM}}$ and their standard deviations
- Mean difference in accuracy $\overline{\Delta}_{\text{acc}}$ and its standard deviation $\text{SD}(\Delta_{\text{acc}})$
- Mean estimated Bayes error $\overline{\hat{\varepsilon}^*}$ and its standard deviation
- Mean CV-selected hyperparameters $\overline{\lambda}_{\text{opt}}$ and $\overline{C}_{\text{opt}}$ and their standard deviations
- boundary hit diagnostics for CV-tuning

3.4 Experimental conditions

In total 300 conditions $\{K, p, n_{pc}, \delta, \rho\}$ were simulated, each replicated 50 times giving 15,000 fits per method. These are split into two sets, **Set I** and **Set II** as summarized in Table 1. The two sets are disjoint in $\{K, p, n_{pc}, \delta, \rho\}$-space and within each set, every combination of parameter values is simulated, yielding 192 conditions in Set I and 108 in Set II. The number of classes is restricted to $K = 3$ and $K = 5$. The separation parameter δ is varied across low to high separation, and the covariance parameter ρ from 0, which means no correlation to 0.9, very high correlation. The main difference between the sets are the n_{pc} and p values, where the ratio $n_{\text{train}}/p := (K \cdot \lfloor 0.7 \cdot n_{pc} \rfloor)/p$ is of particular interest as discussed in Section 3.5. Set I targets conventional regimes, with lower dimensional feature space and/or a sufficient number of data points n_{train} relative to p. That is, large n_{train}/p value. Set II targets regimes with high dimensional feature space, and/or scarce data. That is, low n_{train}/p train value. Specifically for Set I the n_{train}/p range is $[4.2, 140]$ whereas for Set II the range is $[0.26, 3.5]$.

Parameter	Set I	Set II
K	{3, 5}	{3, 5}
p	{10, 20, 50, 100}	{200, 400}
n_{pc}	{200, 400}	{50, 100, 200}
δ	{1, 1.5, 2, 3}	{1, 2, 3}
ρ	{0, 0.5, 0.9}	{0, 0.5, 0.9}
n_{train}/p range	[4.2, 140]	[0.26, 3.5]
# conditions	192	108
# total fits	9 600	5 400

Table 1: Parameter values for the two sets of experimental conditions. Set I targets the conventional regime with $n_{\text{train}} > p$ throughout; Set II targets the high-dimensional regime with n_{train} comparable to or smaller than p.

3.5 Some Motivations

Separation parameter δ and scaling. The purpose of the separation parameter δ is to control the distance between the means $\boldsymbol{\mu_k}$ of the classes. Here high δ indicates high separation and in turn an easy classification problem, typically reflected by a Bayes optimal benchmark value closer to 1, and low δ indicating low separation, a hard problem and Bayes optimal close to chance, $1/K$. Since means are redrawn randomly as $\boldsymbol{\mu}_k \sim \mathcal{N}\left(0, \frac{\delta^2}{p}\boldsymbol{I}_p\right)$ for each replication, exact distance can not be controlled over replications. What can be controlled however is the *expected* distance between classes, $\mathbb{E}\left[\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_r\|^2\right]$, as demonstrated by the calculations below. The scaling δ^2/p in the variance of the class means serves to compensate for changing geometry as the dimension p varies, keeping the expected distance constant in p.

Take two distinct classes $k \neq r$. Since $\boldsymbol{\mu}_k, \boldsymbol{\mu}_r \sim \mathcal{N}(\mathbf{0}, \frac{\delta^2}{p} I_p)$ are drawn independently, each coordinate difference $\mu_{k,i} - \mu_{r,i}$ has mean zero and variance $\frac{\delta^2}{p} + \frac{\delta^2}{p} = \frac{2\delta^2}{p}$, so

$$\mathbb{E}\left[\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_r\|^2\right] = \sum_{i=1}^{p} \mathbb{E}\left[(\mu_{k,i} - \mu_{r,i})^2\right] = \sum_{i=1}^{p} \frac{2\delta^2}{p} = 2\delta^2. \tag{35}$$

The Bayes-optimal benchmark While the separation parameter δ discuss above is the most direct way of influencing the classification problem difficulty, it is ultimately determined jointly by the parameters that specify class distributions, δ, ρ and K. For example, the correlation coefficient ρ also shifts difficulty. At fixed δ, stronger correlation makes the problem easier on

average, visible in the Bayes-optimal accuracy rising with ρ in the results tables Section 4 as well as in Appendix B. An alternative experiment design could have treated problem difficulty as a parameter in its own right, rather than emerging from δ, ρ and K. This was not done here, as it would remove the ability to vary δ, ρ and K independently, which are quantities of interest in this study. Instead, they are varied directly, and Bayes-optimal accuracy is computed and reported for every condition, making resulting differences in difficulty visible and aiding the interpretation of results across conditions.

The benchmark is estimated as described in Section 3.2. The number of draws, $N = 10{,}000$, was chosen so that this estimate is precise for our purposes, with a standard error of at most 0.005 as shown below. Writing $I_j = \mathbf{1}\{h^*_{\boldsymbol{\mu}}(\tilde{\boldsymbol{X}}_j) \neq \tilde{Y}_j\}$, the estimate $\widehat{\varepsilon}_{\boldsymbol{\mu}} = \frac{1}{N}\sum_{j=1}^{N} I_j$ is an average of N independent indicators. Each I_j is Bernoulli with error probability $\varepsilon_j \in [0,1]$, so $\mathrm{Var}(I_j) = \varepsilon_j(1-\varepsilon_j) \leq \frac{1}{4}$. We get

$$\mathrm{Var}(\widehat{\varepsilon}_{\boldsymbol{\mu}}) = \frac{1}{N^2}\sum_{j=1}^{N}\mathrm{Var}(I_j) \;\leq\; \frac{1}{4N}, \qquad \mathrm{SE}(\widehat{\varepsilon}_{\boldsymbol{\mu}}) \;\leq\; \frac{1}{2\sqrt{N}} \;=\; \frac{1}{2\sqrt{10{,}000}} \;=\; 0.005. \tag{36}$$

The benchmark is therefore accurate to within about half a percentage point in the worst case, and more precise still when the problem is easy or hard. This is far tighter than the test-set accuracy estimates against which it is compared, whose test sets contain at most a few hundred observations.

The ratio $n_{\mathbf{train}}/p$. As discussed above, the scaling δ^2/p is intended to keep problem difficulty comparable when varying the feature dimension p. Changing p however changes the *estimation* problem. Both methods estimate a coefficient vector for each class over all p features, and how well this can be done depends on how much training data is available, n_{train} , *relative* to p. This, captured by the fraction n_{train}/p, which is why conditions are divided as the two sets (Table 1), and for the same reason the tables of Section 4 are organised by this ratio.

4 Results

In this section we present the findings of the simulation described in Section 3. For each condition, one replication of the simulation records the quantity paired difference in test accuracy on the held out test data, $\mathrm{acc}_r^{\mathrm{LR}} - \mathrm{acc}_r^{\mathrm{SVM}}$, $r \in \{1, ..., 50\}$. Thus, positive values favour ridge multinomial logistic regression and negative values favour the Crammer–Singer SVM. Over the 50 replications performed, the mean is formed

$$\bar{\Delta}_{\mathrm{acc}} = \frac{1}{50}\sum_{r=1}^{50}\left[\mathrm{acc}_r^{\mathrm{LR}} - \mathrm{acc}_r^{\mathrm{SVM}}\right]$$

which is the main quantity of study in this work, and the main quantity reported in each cell of the tables in this section, as well as in Appendix B. The same tables also report the mean test accuracies of the two fitted methods and standard error of $\bar{\Delta}_{\mathrm{acc}}$. Tables also report *mean* Bayes-optimal benchmark accuracy over the 50 replications, estimating the average difficulty across the redrawn class means. At each replication this is computed as described in Section 3.2, and over the 50 replications the variation was at most about one percentage point, so the reported mean is precise to roughly $0.01/\sqrt{50} \approx 0.001$, well under half a percentage point.

The main tables appearing in this section are collapsed by the ratio n_{train}/p, while the full tables for all 300 conditions can be found in Appendix B. When a displayed value is the average of

two original cells, the reported quantity is the average of the two original cell means and the standard error is computed by follows

$$\widehat{\mathrm{SE}}(\bar{\Delta}_{\mathrm{coll}}) = \tfrac{1}{2}\sqrt{\frac{s_1^2}{50} + \frac{s_2^2}{50}},$$

where s_1 and s_2 are standard deviations of the two original cells. Cells are highlighted in the tables when two conditions are both satisfied: the absolute mean gap is at least one percentage point, $|\bar{\Delta}_{\mathrm{acc}}| \geq 0.01$, and the approximate 95% t-interval $\bar{\Delta}_{\mathrm{acc}} \pm t_{0.975,\,49} \cdot \mathrm{SE}$ excludes zero. The highlighting rule thus requires both that the observed difference is large enough to be of practical interest and that it is unlikely to be due to sampling noise. One percentage point moreover corresponds to roughly twice the size of the largest observed standard errors, and thus approximately to the smallest differences that can be separated from sampling noise in the noisiest conditions.

4.1 Summary across conditions

Figure 1 gives a visual overview of the mean paired difference appearing the collapsed tables. The four panels are separated by the two sets Set I and Set II and the two class counts $K = 3$ and $K = 5$. Within each panel, the horizontal axis shows the correlation parameter ρ, colour encodes the value of separation parameter δ, size encodes n_{train}/p and filled points denote cells that satisfy the highlighting criteria.

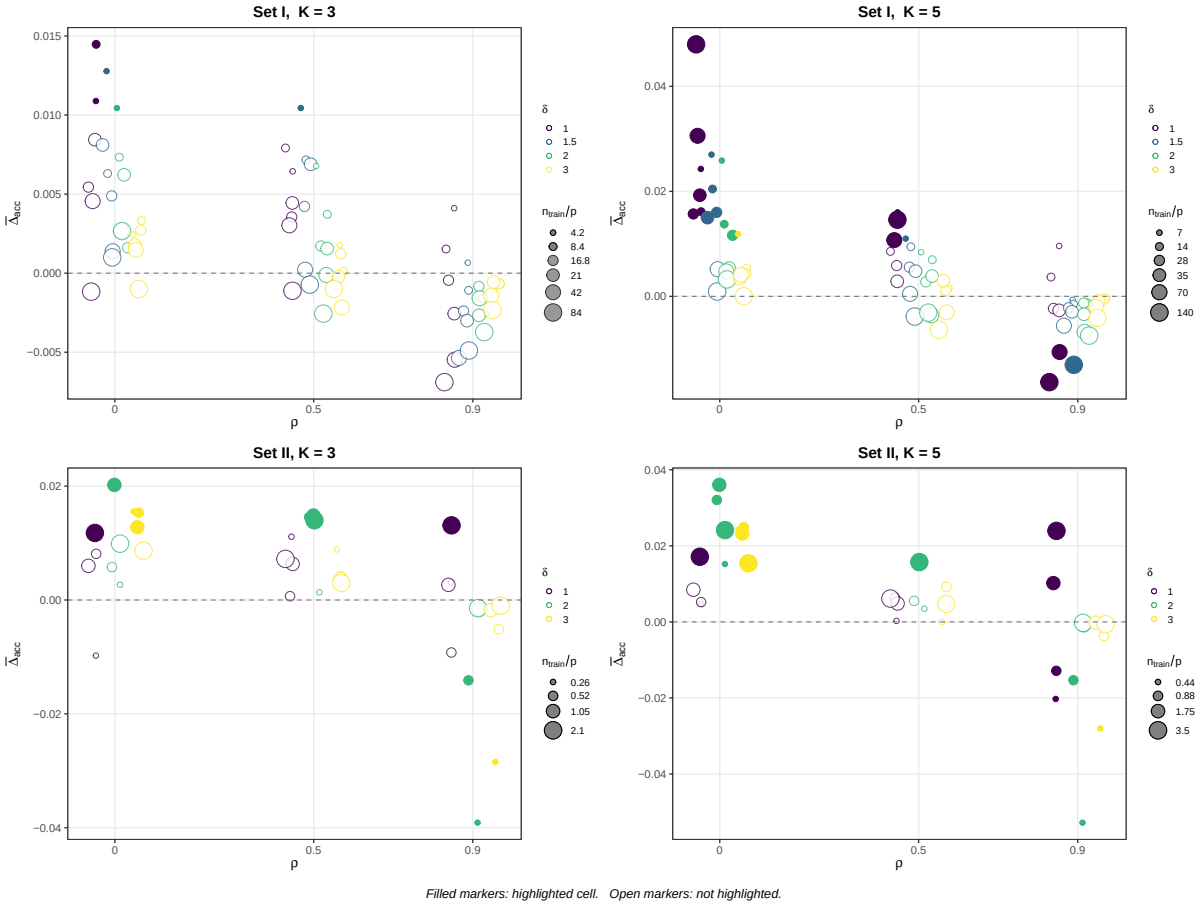

Figure 1: Per-cell mean paired difference $\bar{\Delta}_{\text{acc}}$ across all collapsed conditions, organised by feature correlation ρ. The four panels separate Set I and Set II and the two class counts $K = 3$ and $K = 5$. Within each panel, colour encodes the class-separation parameter δ, marker size encodes the ratio n_{train}/p, and filled markers denote cells satisfying the dual highlighting criterion ($|\bar{\Delta}_{\text{acc}}| \geq 0.01$ and the approximate 95% t_{49}-interval excluding zero). Open markers denote cells not meeting this criterion. The dashed line marks zero, where the two methods have equal mean test accuracy.

Across most conditions the methods perform very similarly, with a majority of points close to zero. Also relatively few points satisfy the highlighting criterion. Still, notable differences appear. The largest per-condition mean paired differences observed in the study are

$$\max_{\text{conditions}} \bar{\Delta}_{\text{acc}} = +0.048 \qquad \text{and} \qquad \min_{\text{conditions}} \bar{\Delta}_{\text{acc}} = -0.053,$$

occurring at $K = 5$ but at opposite values of ρ. At $\rho = 0$, when a difference is observed it tends to favour LR, while at $\rho = 0.9$ cells favouring Crammer–Singer SVM appear. Still, at $\rho = 0.9$ some cells still favour LR.

One important thing to note regarding this presentation is the cells indicating positive values (LR wins) for low values of the correlation coefficient ρ, which should be interpreted with caution in light of the tuning diagnostics discussed in Section 4.4.

4.2 Set I

Set I contains 192 simulation conditions and covers the range $n_{\text{train}}/p \in [4.2, 140]$. Thus, both methods have more training data relative to the feature dimension compared to Set II. Tables 2 and 3 present the collapsed per-cell results for $K = 3$ and $K = 5$ respectively.

For $K = 3$, differences are small throughout and few cells are highlighted, with LR marginally ahead at low ρ. For high ρ the LR advantage completely disappears while a small SVM advantage emerges, although below the highlighting threshold. For $K = 5$ we see bigger effects. At $\rho = 0$ LR leads in every cell, with a majority of cells highlighted. Also, the gap grows with n_{train}/p, reaching the largest LR win in the entire study (+0.048 at $n_{\text{train}}/p = 140$). At $\rho = 0.9$ the sign seemingly reverses and on the contrary we see SVM ahead, although with fewer cells highlighted. Also, we again see the difference grow with n_{train}/p. Throughout, both methods remain reasonably close to the Bayes-optimal benchmark, with the largest gaps occurring in the most difficult conditions ($K = 5$, low δ).

Table 2: Set I, $K = 3$, collapsed by n_{train}/p. Each cell reports the mean test accuracy of ridge LR / the mean test accuracy of CS-SVM, followed in parentheses by the mean paired difference $\bar{\Delta}_{\text{acc}} = \text{acc}_{\text{LR}} - \text{acc}_{\text{SVM}}$ (positive values favour LR) and its standard error, all over $R = 50$ replications. The final column gives the mean Bayes-optimal *accuracy* $1 - \hat{\varepsilon}^*$. Bold: $|\bar{\Delta}_{\text{acc}}| \geq 0.01$ and the t_{49}-interval excludes zero.

(δ, ρ)	n_{train}/p = 4.2	n_{train}/p = 8.4	n_{train}/p = 16.8	n_{train}/p = 21	n_{train}/p = 42	n_{train}/p = 84	Bayes-opt
(1, 0.0)	**0.565/0.554 (+0.011, 0.0038)**	**0.585/0.570 (+0.014, 0.0023)**	0.608/0.603 (+0.005, 0.0022)	0.600/0.592 (+0.008, 0.0032)	0.611/0.606 (+0.005, 0.0014)	0.618/0.619 (-0.001, 0.0016)	0.63
(1, 0.5)	0.608/0.601 (+0.006, 0.0028)	0.668/0.660 (+0.008, 0.0021)	0.686/0.683 (+0.004, 0.0021)	0.687/0.683 (+0.004, 0.0031)	0.672/0.669 (+0.003, 0.0017)	0.672/0.674 (-0.001, 0.0018)	0.70
(1, 0.9)	0.926/0.922 (+0.004, 0.0021)	0.946/0.945 (+0.002, 0.0013)	0.951/0.951 (+0.000, 0.0010)	0.941/0.944 (-0.003, 0.0021)	0.924/0.929 (-0.005, 0.0013)	0.917/0.924 (-0.007, 0.0012)	0.95
(1.5, 0.0)	**0.712/0.699 (+0.013, 0.0033)**	0.729/0.723 (+0.006, 0.0018)	0.738/0.733 (+0.005, 0.0019)	0.740/0.732 (+0.008, 0.0029)	0.746/0.744 (+0.001, 0.0014)	0.734/0.733 (+0.001, 0.0010)	0.75
(1.5, 0.5)	**0.776/0.765 (+0.010, 0.0034)**	0.811/0.804 (+0.007, 0.0016)	0.815/0.811 (+0.004, 0.0016)	0.826/0.819 (+0.007, 0.0025)	0.813/0.813 (+0.000, 0.0014)	0.821/0.821 (-0.001, 0.0018)	0.84
(1.5, 0.9)	0.990/0.990 (+0.001, 0.0010)	0.993/0.994 (-0.001, 0.0005)	0.993/0.995 (-0.002, 0.0005)	0.990/0.993 (-0.003, 0.0009)	0.976/0.981 (-0.005, 0.0008)	0.981/0.986 (-0.005, 0.0012)	0.99
(2, 0.0)	**0.831/0.820 (+0.010, 0.0030)**	0.846/0.838 (+0.007, 0.0014)	0.861/0.860 (+0.002, 0.0014)	0.852/0.846 (+0.006, 0.0025)	0.844/0.841 (+0.003, 0.0012)	0.839/0.836 (+0.003, 0.0011)	0.86
(2, 0.5)	0.884/0.878 (+0.007, 0.0030)	0.903/0.899 (+0.004, 0.0013)	0.918/0.916 (+0.002, 0.0015)	0.909/0.908 (+0.002, 0.0021)	0.898/0.898 (+0.000, 0.0012)	0.876/0.879 (-0.003, 0.0014)	0.92
(2, 0.9)	0.997/0.999 (-0.003, 0.0006)	0.998/1.000 (-0.002, 0.0003)	0.998/0.999 (-0.001, 0.0003)	0.996/0.998 (-0.003, 0.0009)	0.995/0.996 (-0.002, 0.0005)	0.991/0.995 (-0.004, 0.0007)	1.00
(3, 0.0)	0.957/0.954 (+0.002, 0.0013)	0.962/0.959 (+0.003, 0.0010)	0.963/0.960 (+0.003, 0.0007)	0.943/0.941 (+0.002, 0.0013)	0.947/0.946 (+0.001, 0.0009)	0.938/0.939 (-0.001, 0.0013)	0.96
(3, 0.5)	0.979/0.977 (+0.002, 0.0012)	0.982/0.982 (+0.000, 0.0006)	0.985/0.984 (+0.001, 0.0007)	0.975/0.975 (+0.000, 0.0009)	0.972/0.974 (-0.002, 0.0009)	0.975/0.976 (-0.001, 0.0010)	0.99
(3, 0.9)	0.999/1.000 (-0.001, 0.0004)	0.999/1.000 (-0.001, 0.0002)	0.999/1.000 (-0.001, 0.0003)	0.999/1.000 (-0.001, 0.0002)	0.998/1.000 (-0.001, 0.0003)	0.997/1.000 (-0.002, 0.0006)	1.00

Table 3: Set I, $K = 5$, collapsed by n_{train}/p. Each cell reports the mean test accuracy of ridge LR / the mean test accuracy of CS-SVM, followed in parentheses by the mean paired difference $\bar{\Delta}_{\text{acc}} = \text{acc}_{\text{LR}} - \text{acc}_{\text{SVM}}$ (positive values favour LR) and its standard error, all over $R = 50$ replications. The final column gives the mean Bayes-optimal *accuracy* $1 - \hat{\varepsilon}^*$. Bold: $|\bar{\Delta}_{\text{acc}}| \geq 0.01$ and the t_{49}-interval excludes zero.

(δ, ρ)	n_{train}/p = 7	n_{train}/p = 14	n_{train}/p = 28	n_{train}/p = 35	n_{train}/p = 70	n_{train}/p = 140	Bayes-opt
(1, 0.0)	**0.405/0.381 (+0.024, 0.0031)**	**0.438/0.422 (+0.016, 0.0015)**	**0.471/0.455 (+0.016, 0.0018)**	**0.459/0.440 (+0.019, 0.0034)**	**0.472/0.441 (+0.031, 0.0031)**	**0.459/0.411 (+0.048, 0.0045)**	0.48
(1, 0.5)	**0.479/0.463 (+0.016, 0.0030)**	0.528/0.520 (+0.009, 0.0016)	0.558/0.553 (+0.006, 0.0018)	0.543/0.540 (+0.003, 0.0022)	**0.545/0.534 (+0.011, 0.0018)**	**0.540/0.525 (+0.015, 0.0026)**	0.57
(1, 0.9)	0.892/0.882 (+0.010, 0.0022)	0.909/0.906 (+0.004, 0.0011)	0.919/0.921 (-0.002, 0.0013)	0.893/0.895 (-0.003, 0.0015)	**0.869/0.880 (-0.011, 0.0014)**	**0.851/0.867 (-0.016, 0.0015)**	0.92
(1.5, 0.0)	**0.593/0.566 (+0.027, 0.0032)**	**0.616/0.595 (+0.020, 0.0017)**	**0.629/0.613 (+0.016, 0.0016)**	**0.635/0.620 (+0.015, 0.0020)**	0.617/0.612 (+0.005, 0.0012)	0.629/0.628 (+0.001, 0.0014)	0.64
(1.5, 0.5)	**0.679/0.668 (+0.011, 0.0031)**	0.714/0.704 (+0.009, 0.0014)	0.737/0.731 (+0.006, 0.0016)	0.728/0.723 (+0.005, 0.0024)	0.706/0.706 (+0.000, 0.0012)	0.719/0.723 (-0.004, 0.0015)	0.75
(1.5, 0.9)	0.986/0.986 (-0.001, 0.0008)	0.987/0.988 (-0.001, 0.0006)	0.988/0.990 (-0.002, 0.0004)	0.977/0.980 (-0.003, 0.0010)	0.962/0.968 (-0.006, 0.0010)	**0.956/0.969 (-0.013, 0.0012)**	0.99
(2, 0.0)	**0.739/0.714 (+0.026, 0.0029)**	**0.755/0.741 (+0.014, 0.0013)**	**0.769/0.757 (+0.012, 0.0013)**	0.746/0.740 (+0.005, 0.0019)	0.749/0.745 (+0.005, 0.0010)	0.736/0.733 (+0.003, 0.0012)	0.77
(2, 0.5)	0.835/0.826 (+0.008, 0.0022)	0.849/0.842 (+0.007, 0.0013)	0.862/0.859 (+0.003, 0.0011)	0.850/0.846 (+0.004, 0.0019)	0.838/0.842 (-0.004, 0.0011)	0.822/0.825 (-0.003, 0.0013)	0.87
(2, 0.9)	0.998/0.999 (-0.001, 0.0004)	0.998/0.999 (-0.001, 0.0002)	0.998/0.999 (-0.001, 0.0003)	0.992/0.996 (-0.003, 0.0006)	0.984/0.991 (-0.007, 0.0007)	0.983/0.991 (-0.007, 0.0010)	1.00
(3, 0.0)	**0.925/0.913 (+0.012, 0.0017)**	0.931/0.926 (+0.005, 0.0010)	0.931/0.927 (+0.004, 0.0008)	0.916/0.912 (+0.003, 0.0013)	0.900/0.896 (+0.004, 0.0009)	0.901/0.900 (+0.000, 0.0010)	0.93
(3, 0.5)	0.967/0.966 (+0.002, 0.0012)	0.975/0.973 (+0.002, 0.0006)	0.973/0.972 (+0.001, 0.0006)	0.964/0.961 (+0.003, 0.0011)	0.952/0.955 (-0.003, 0.0008)	0.941/0.947 (-0.006, 0.0010)	0.97
(3, 0.9)	1.000/1.000 (+0.000, 0.0002)	1.000/1.000 (+0.000, 0.0001)	0.999/1.000 (-0.001, 0.0002)	0.998/0.999 (-0.001, 0.0003)	0.997/0.999 (-0.002, 0.0004)	0.994/0.998 (-0.004, 0.0009)	1.00

4.3 Set II

Set II contains 108 simulation conditions and covers the range $n_{\text{train}}/p \in [0.26, 3.5]$, a regime in which the training sample is comparable to or smaller than the feature dimension. Tables 4 and 5 present the collapsed per-cell results for $K = 3$ and $K = 5$ respectively.

The structure is similar for $K = 3$ and $K = 5$. At $\rho = 0$, we again find LR ahead as in Set I, but without a clear dependence on n_{train}/p. At $\rho = 0.9$ we find both LR and SVM wins, with about twice as many SVM wins for highlighted cells. Also notably, the SVM wins are concentrated at small n_{train}/p, in contrast with Set I, where the SVM advantage at $\rho = 0.9$ grew with n_{train}/p. In contrast to Set I, both methods often perform far below the Bayes-optimal benchmark in Set II, particularly at small n_{train}/p, reflecting the difficulty of the high dimensional/data scarce regime.

Caution should be exercised when reading and interpreting the apparently stronger results of Set II. As discussed in Section 4.4, in some regions of this regime the SVM cost parameter tuning may have been acting as a confounding variable. In particular several of the strongest LR-positive cells overlap with this. However, a relatively clean finding in Set II is the SVM-ahead cells at high ρ *and* small n_{train}/p, which are not affected by this.

Table 4: Set II, $K = 3$, collapsed by n_{train}/p. Each cell reports the mean test accuracy of ridge LR / the mean test accuracy of CS-SVM, followed in parentheses by the mean paired difference $\bar{\Delta}_{\text{acc}} = \text{acc}_{\text{LR}} - \text{acc}_{\text{SVM}}$ (positive values favour LR) and its standard error, all over $R = 50$ replications. The final column gives the mean Bayes-optimal *accuracy* $1 - \hat{\varepsilon}^*$. Bold: $|\bar{\Delta}_{\text{acc}}| \geq 0.01$ and the t_{49}-interval excludes zero.

$(\delta,\ \rho)$	n_{train}/p = 0.26	n_{train}/p = 0.52	n_{train}/p = 1.05	n_{train}/p = 2.1	Bayes-opt
(1, 0.0)	0.411/0.420 (-0.010, 0.0065)	0.441/0.433 (+0.008, 0.0047)	0.474/0.468 (+0.006, 0.0035)	**0.517/0.505 (+0.012, 0.0039)**	0.63
(1, 0.5)	0.411/0.400 (+0.011, 0.0069)	0.445/0.444 (+0.001, 0.0046)	0.499/0.493 (+0.006, 0.0033)	0.556/0.549 (+0.007, 0.0038)	0.71
(1, 0.9)	0.442/0.439 (+0.003, 0.0070)	0.612/0.621 (-0.009, 0.0050)	0.796/0.793 (+0.003, 0.0026)	**0.888/0.875 (+0.013, 0.0022)**	0.97
(2, 0.0)	0.628/0.626 (+0.003, 0.0075)	0.707/0.701 (+0.006, 0.0044)	**0.759/0.738 (+0.020, 0.0031)**	0.806/0.796 (+0.010, 0.0023)	0.86
(2, 0.5)	0.579/0.577 (+0.001, 0.0057)	**0.680/0.666 (+0.015, 0.0039)**	**0.783/0.768 (+0.015, 0.0030)**	**0.848/0.834 (+0.014, 0.0036)**	0.94
(2, 0.9)	**0.739/0.778 (-0.039, 0.0047)**	**0.939/0.953 (-0.014, 0.0020)**	0.992/0.994 (-0.002, 0.0007)	0.997/0.998 (-0.001, 0.0009)	1.00
(3, 0.0)	**0.841/0.826 (+0.016, 0.0048)**	**0.902/0.887 (+0.015, 0.0027)**	**0.939/0.926 (+0.013, 0.0016)**	0.954/0.945 (+0.009, 0.0017)	0.97
(3, 0.5)	0.837/0.828 (+0.009, 0.0050)	0.892/0.888 (+0.004, 0.0031)	0.949/0.946 (+0.004, 0.0019)	0.973/0.970 (+0.003, 0.0013)	0.99
(3, 0.9)	**0.925/0.953 (-0.028, 0.0046)**	0.993/0.998 (-0.005, 0.0010)	0.998/1.000 (-0.002, 0.0005)	0.999/1.000 (-0.001, 0.0004)	1.00

Table 5: Set II, $K = 5$, collapsed by n_{train}/p. Each cell reports the mean test accuracy of ridge LR / the mean test accuracy of CS-SVM, followed in parentheses by the mean paired difference $\bar{\Delta}_{\text{acc}} = \text{acc}_{\text{LR}} - \text{acc}_{\text{SVM}}$ (positive values favour LR) and its standard error, all over $R = 50$ replications. The final column gives the mean Bayes-optimal *accuracy* $1 - \hat{\varepsilon}^*$. Bold: $|\bar{\Delta}_{\text{acc}}| \geq 0.01$ and the t_{49}-interval excludes zero.

(δ, ρ)	n_{train}/p = 0.44	n_{train}/p = 0.88	n_{train}/p = 1.75	n_{train}/p = 3.5	Bayes-opt
(1, 0.0)	0.279/0.275 (+0.005, 0.0044)	0.289/0.284 (+0.005, 0.0036)	0.324/0.315 (+0.008, 0.0027)	**0.371/0.354 (+0.017, 0.0040)**	0.49
(1, 0.5)	0.273/0.273 (+0.000, 0.0049)	0.306/0.300 (+0.006, 0.0034)	0.349/0.344 (+0.005, 0.0026)	0.424/0.418 (+0.006, 0.0034)	0.59
(1, 0.9)	**0.325/0.345 (-0.020, 0.0070)**	**0.540/0.553 (-0.013, 0.0034)**	**0.741/0.731 (+0.010, 0.0018)**	**0.851/0.827 (+0.024, 0.0021)**	0.95
(2, 0.0)	**0.493/0.477 (+0.015, 0.0051)**	**0.569/0.537 (+0.032, 0.0040)**	**0.654/0.618 (+0.036, 0.0023)**	**0.703/0.679 (+0.024, 0.0024)**	0.79
(2, 0.5)	0.474/0.470 (+0.003, 0.0058)	0.579/0.574 (+0.006, 0.0030)	**0.697/0.681 (+0.015, 0.0027)**	**0.774/0.758 (+0.016, 0.0029)**	0.90
(2, 0.9)	**0.710/0.763 (-0.053, 0.0049)**	**0.941/0.956 (-0.015, 0.0019)**	0.992/0.993 (-0.001, 0.0005)	0.998/0.998 (+0.000, 0.0004)	1.00
(3, 0.0)	**0.765/0.741 (+0.024, 0.0051)**	**0.842/0.817 (+0.025, 0.0030)**	**0.895/0.872 (+0.023, 0.0019)**	**0.914/0.898 (+0.015, 0.0018)**	0.94
(3, 0.5)	0.739/0.739 (+0.000, 0.0051)	0.855/0.846 (+0.009, 0.0025)	0.928/0.923 (+0.005, 0.0014)	0.959/0.954 (+0.005, 0.0014)	0.99
(3, 0.9)	**0.942/0.970 (-0.028, 0.0032)**	0.995/0.999 (-0.004, 0.0007)	0.999/1.000 (+0.000, 0.0002)	0.999/1.000 (-0.001, 0.0002)	1.00

4.4 Hyperparameter tuning diagnostics

During the tuning of hyperparameters in step 7 of the simulation process (3.3), the selection of a value at the edge of the searched grid may indicate suboptimal tuning. Theoretically a value outside the grid may have yielded better accuracy. To monitor this potential failure mode, boundary-hit diagnostics were recorded.

Boundary hits for LR were a non problem, occurring in less than 1% of all the 15000 fits. In Set I, SVM boundary hits occurred in 2658 of 9600 replication-level fits (27.7%), with 85% being lower boundary hits ($C = 0.001$), while 15% being upper boundary hits ($C = 100$). In relation to the correlation coefficient ρ, the rate was highest at $\rho = 0$ (48.2%), compared with 14.2% at $\rho = 0.5$ and 20.6% at $\rho = 0.9$. In Set II, boundary hits occurred in 2523 of 5400 replication-level fits (46.7%), entirely at the lower boundary. Moreover, we observe a clear dependence on the correlation coefficient ρ. Namely, 80.7% at $\rho = 0$, 47.7% at $\rho = 0.5$, and 11.8% at $\rho = 0.9$.

The tuning concerns are primarily with the SVM cost grid, and suboptimal SVM tuning only hurts SVM accuracy. In light of this we take particular interest in highlighted cells with potentially problematically high boundary hit rate. Figures 2 and 3 plot every highlighted cell with its mean SVM boundary-hit fraction on the horizontal axis and $\bar{\Delta}_{\text{acc}}$ on the vertical axis.

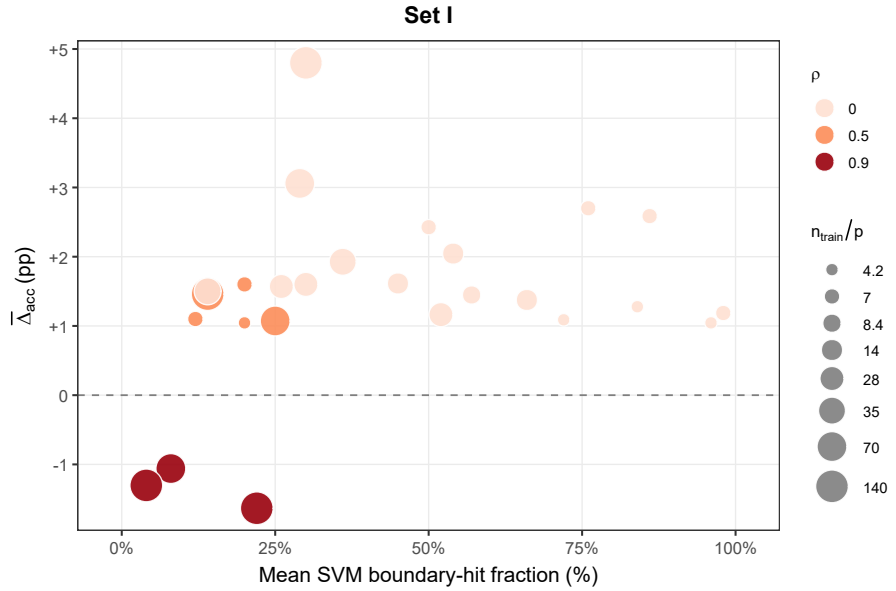

Figure 2: Highlighted cells in Set I: mean paired difference $\bar{\Delta}_{\text{acc}}$ (pp) against mean SVM boundary-hit fraction. Colour encodes the correlation parameter ρ (light to dark red: $\rho \in \{0, 0.5, 0.9\}$); point size encodes n_{train}/p. All points with negative $\bar{\Delta}_{\text{acc}}$ (SVM ahead) cluster near zero boundary-hit fraction.

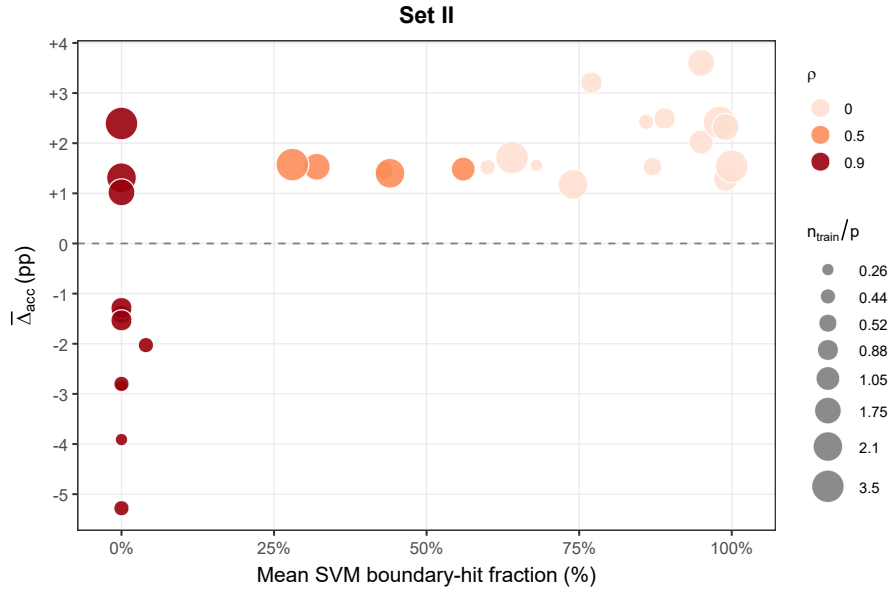

Figure 3: Highlighted cells in Set II: same encoding as Figure 2. The SVM-ahead cells (negative $\bar{\Delta}_{\text{acc}}$) are exclusively at $\rho = 0.9$ and have near-zero boundary-hit fractions, while the strongest LR-ahead cells at $\rho = 0$ coincide with high boundary-hit rates.

From these plots a few things become clear. Almost all cells indicating SVM wins (negative $\bar{\Delta}_{\text{acc}}$) have near zero boundary hits regardless of set, while the structure for cells indicating LR is more varied and differs between the two sets. In Set I, the cells indicating LR win span a range of SVM hit rates, from 10% to over 90%, with almost all cases above 25% concentrated at $\rho = 0$, indicating a somewhat dependence on ρ. Set II exhibits an even clearer pattern regarding ρ dependence, where at $\rho = 0.9$ there are close to no SVM boundary hits, while at $\rho = 0$ sit

almost exclusively above 60%. In conclusion the tuning issues may be a confounder of the results for cells at $\rho = 0$, artificially putting LR ahead, while this is not likely to be the case for the highlighted SVM wins at $\rho = 0.9$, making the results appearing in these cells the cleanest finding of this study. The full tables for this can be found in Appendix B Tables 18–21.

5 Discussion

5.1 Overall similarity of the two methods

The main finding of this study is the overall similarity of the two methods. Across the 300 conditions the mean paired difference, $\bar{\Delta}_{\text{acc}}$, never exceeds roughly 5.3 percentage points in magnitude, and only in 79 of 300 conditions does it exceed one percentage point while remaining distinguishable from sampling noise (see Tables of Appendix B). Thus, for data of the kind generated here, the choice between the logistic and hinge losses cannot be considered a major determinant of classification accuracy. The scope within which this overall similarity should be interpreted is specified in Section 5.4. Another overall observation is that, in Set I both methods track the Bayes optimal benchmark relatively well, while, in Set II both occasionally fall substantially short of it, particularly in the harder regimes such as small n_{train}/p and/or small separation δ.

5.2 Tendency for SVM advantage at high correlation

The results contain some systematic differences between the methods that we deem reliable within this study. At high values of the correlation coefficient ρ, in particular at $\rho = 0.9$, we observe some tendencies of SVM advantage, although the picture breaks down by the Sets I and II as well as value of K. For Set I we see a slight SVM advantage of up to about $1 - 1.5$ percentage points, however with no highlighted cells at $K = 3$. For Set I, in the uncollapsed tables of Appendix B, at $\rho = 0.9$ a total of 4 cells are highlighted, all favouring SVM. The tendency is also that the advantage increases with the ratio n_{train}/p. For Set II we see a clearer SVM advantage as well as a reversed dependency on the ratio n_{train}/p, where the SVM wins are concentrated at low values. Notably, the single largest paired difference observed anywhere in the study, $\bar{\Delta}_{\text{acc}} = -0.0528$ (the 5.3 percentage point bound cited at the start of this section), occurs among these low-n_{train}/p Set II cells, at $K = 5$, $\delta = 2$, $\rho = 0.9$. It is however noteworthy that a small number of LR wins are also observed in these conditions, concentrated at relatively high values of the ratio n_{train}/p. In total, reading of the un-collapsed tables of Appendix B, at $\rho = 0.9$, out of the in total 18 highlighted cells, 15 favor SVM. Importantly, what makes these findings reliable is that the SVM boundary-hit fraction (Section 4.4) is negligible, below 5%, in 16 of these 18 cells, so in these cells the cost parameter was selected at an interior value of the searched grid and can be ruled out as a confounder. The two exceptions are both in Set I at $K = 5$, $\delta = 1$, and both favouring SVM, occurring at $n_{\text{train}}/p = 70$ with a boundary-hit fraction of 14%, and at $n_{\text{train}}/p = 140$ with a boundary-hit fraction of 22%, making two exceptions where a degree of SVM under-tuning cannot be fully ruled out.

5.3 Hyperparameter tuning as a possible confounder at low correlation

The remaining systematic differences between the methods observed is a tendency for LR advantage at $\rho = 0$(and somewhat at $\rho = 0.5$). At $\rho = 0$, reading of the uncollapsed tables of Appendix B, a total of 44 cells are highlighted, all favouring logistic regression. As already mentioned, these observations should be interpreted far more cautiously than the results discuss in the previous section since, as documented in Section 4.4, these cells largely coincide with high SVM boundary hit rates. Across the 44 cells the boundary-hit fraction ranges from 14% to 100%, with the majority of cells above 60%. Since suboptimal SVM tuning may affect, and in

that case only degrade, SVM accuracy, suboptimal tuning as a confounder driving the apparent tendency for LR advantage can not be ruled out for these conditions. Thus, in the present study we do not deem these results reliable, and we refrain from attributing any observed tendency for difference in accuracy to the loss function.

The origin of the problem is implementational, and could in principle be remedied. The SVM cost parameter was selected from a fixed and comparatively coarse set of six candidate values, whereas the logistic-regression penalty was tuned over a considerably finer sequence of one hundred candidate values, which `glmnet` constructs from the training data itself, so that the searched range adapts to each fit and extends to a far lower floor (full details on both in Appendix A). The fact that the cross-validation optimum for SVM with these settings in `LiblineaR` falls at the edge in these regimes became apparent from the diagnostics only after running the full simulations. Repeating the experiment for the affected conditions with an extended search turned out not to be computationally feasible within the scope of this project. A natural direction for further work is therefore to repeat the SVM tuning over a finer set of cost values extending beyond the present limits, hopefully reaching an acceptable boundary hit rate. Doing so could establish whether any genuine difference remains at $\rho = 0$.

5.4 Further limitations

The main limitation to the interpretation of the results is the tuning issues presented in Section 4.4 and discussed in Section 5.3. Further limitations are collected below, some being deliberate design choices limiting the generality of the study rather than the validity of any findings. Motivations for some of these design choices are given in Section 3.5.

The most important of these is the restriction to the single data generating process (described in Section 3.1). Conclusions here are statements about data generated by this process, and no claim is made about how the comparison would behave under other data generating processes. Investigating this for other data generating processes is therefore a natural direction for future work.

There is also a limit to how small differences this study can detect. With $R = 50$ replications per condition, mean differences smaller than about one percentage point can in general not be separated from sampling noise (this is also the size threshold of the highlighting rule in Section 4). Moreover, for conditions with small sample sizes, the test sets become particularly small (≤ 100 in some cases), adding further variability. Small differences are therefore hardest to detect precisely in the Set II where data by construction is relatively scarce. It is also noteworthy that a small amount of variability is also introduced by the cross validation folds being drawn independently for the two methods.

The present study compares test accuracy only, while other properties of the methods might also be of interest. For example, as multinomial logistic regression is a probabilistic model, it produces not only a classifier but predicted class probabilities, whereas Crammer–Singer SVM does not. In an application where predicted probabilities are useful this might be an advantage favouring logistic regression. Another comparison omitted here is runtime, or computational cost, which may be an important factor in deciding which method to use.

The number of classes was restricted to $K \in \{3, 5\}$. The observed differences were generally somewhat larger at $K = 5$ than at $K = 3$, both in magnitude and in the number of highlighted cells, which makes extending the comparison to larger K a natural direction for further work.

A final remark, rather than a limitation, concerns how the highlighted cells should be read. With this many approximate 95% confidence intervals examined simultaneously, a few are expected

to exclude zero by pure chance. Therefore no individual cell should be read and interpreted in isolation. Structural tendencies reported here do not rest on any single cell, but on patterns spanning neighbouring highlighted cells favouring the same method throughout, a pattern unlikely to be produced by chance alone.

6 Conclusion

The present work compares test accuracy of ridge multinomial logistic regression and the Crammer–Singer support vector machine on a family of simulated multiclass problems. The simulation design is such that the Bayes-optimal classifier is known and linear, allowing a comparison to a Bayes-optimal benchmark, and the two linear methods to in principle have sufficient capacity to attain it. The setup also leaves the methods respective loss functions as the main structural difference. The intention of this design is that any systematic difference in test accuracy can be attributed to the choice of loss. Simulations span 300 conditions across two regimes, one in which the training sample is large relative to the feature dimension, and one in which it is comparable to or smaller than it.

The main finding is that the two methods perform very similarly. The mean paired difference $\bar{\Delta}_{\text{acc}}$ never exceeded 5.3 percentage points in magnitude anywhere in the study, and stayed below one percentage point in a large majority of conditions. For data generated by the process studied here, the choice between logistic and hinge loss should therefore not be considered to have a major influence on test accuracy.

Despite the overall similarity, one systematic difference stands out. At strong feature correlation ($\rho = 0.9$) the Crammer–Singer SVM has a small but consistent advantage, in 15 of the 18 conditions meeting a highlighting criterion (see Section 4), and reaching $\bar{\Delta}_{\text{acc}} = -0.053$ at most. This observation is regarded as reliable within the present design (Section 5.2).

A second apparent difference is not regarded as reliable within this study. At $\rho = 0$ and to a lesser degree at $\rho = 0.5$, logistic regression led in a substantial number of conditions. These results however coincide with a problem in the tuning of the SVM which can only have degraded SVM accuracy (Section 5.3). How much of the observed difference this accounts for cannot be determined within the present design, and so nothing can be concluded about the loss functions here.

Conclusions here concern data generated by the simulation process of Section 3.1, in which classes are normally distributed with randomly drawn means, and a shared AR(1) covariance structure, with parameter ranges given in Table 1. Moreover, conclusions depend on the specific software implementations used and their respective tuning protocols. The most straightforward extension of this work would be to repeat the SVM tuning over a finer and wider set of hyperparameter values, in an attempt to resolve the tuning issue discussed above and establish whether any genuine difference remains at $\rho = 0$.

References

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. ISBN 978-0-521-83378-3. URL `https://stanford.edu/~boyd/cvxbook/`.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001. URL `https://jmlr.org/papers/v2/crammer01a.html`.

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. URL `https://jmlr.org/papers/v9/fan08a.html`.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. doi: 10.18637/jss.v033.i01.

Alan Genz and Frank Bretz. *Computation of Multivariate Normal and t Probabilities*, volume 195 of *Lecture Notes in Statistics*. Springer, Berlin, Heidelberg, 2009. ISBN 978-3-642-01688-2. doi: 10.1007/978-3-642-01689-9.

Allan Gut. *Probability: A Graduate Course*. Springer Texts in Statistics. Springer, New York, 2 edition, 2013. ISBN 978-1-4614-4707-8. doi: 10.1007/978-1-4614-4708-5.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, 2 edition, 2009. ISBN 978-0-387-84857-0. doi: 10.1007/978-0-387-84858-7.

Thibault Helleputte, Jérôme Paul, and Pierre Gramme. *LiblineaR: Linear Predictive Models Based on the LIBLINEAR C/C++ Library*, 2024. URL `https://cran.r-project.org/package=LiblineaR`. R package version 2.10-24.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer Texts in Statistics. Springer, New York, 2 edition, 2021. ISBN 978-1-0716-1417-4. doi: 10.1007/978-1-0716-1418-1.

A Implementation details

Ridge multinomial logistic regression is fit with `glmnet::cv.glmnet` using `family="multinomial"`, `alpha=0`, `type.measure="class"`, `standardize=FALSE`, and `nfolds=5`. We use a dense 100-point λ path with `lambda.min.ratio=1e-6`; the lower-than-default floor is required for the ridge coordinate-descent solver to reach useful λ values in the $p > n$ regime. The final model is selected by `lambda.min`. The Crammer–Singer SVM is fit with `LiblineaR::LiblineaR` using `type=4` and `bias=1`, tuned by 5-fold CV over the cost grid $C \in 10^{\{-3,-2,\ldots,2\}}$ via the package's internal `cross` argument, with the cost maximising CV accuracy selected for refitting. Each condition is replicated $R = 50$ times. Replications within a condition are executed in parallel using `foreach`/`doParallel`, and the per-replication seed `base_seed + (i-1)*R + r` depends only on the condition index i and the replication index r, ensuring full reproducibility. Intermediate results are checkpointed after every condition via `saveRDS`.

B Tables

This appendix reports the full results for all 300 conditions. Tables 6–17 give the uncollapsed counterparts of Tables 2–5, one table per combination of set, number of classes K and feature dimension p. Each cell reports the mean test accuracy of ridge LR / the mean test accuracy of CS-SVM, followed in parentheses by the mean paired difference $\bar{\Delta}_{\text{acc}} = \text{acc}_{\text{LR}} - \text{acc}_{\text{SVM}}$ (positive values favour LR) and its standard error, all over $R = 50$ replications. The final column gives the mean Bayes-optimal *accuracy* $1 - \hat{\varepsilon}^*$. Bold: $|\bar{\Delta}_{\text{acc}}| \geq 0.01$ and the t_{49}-interval excludes zero.

Tables 18–21 list, for each set and each K, those cells of the collapsed tables that meet the highlighting criterion, together with the mean SVM boundary-hit fraction discussed in Section 4.4, that is, the proportion of the $R = 50$ replications in which cross-validation selected the SVM cost at an endpoint of the searched grid $C \in \{10^{-3}, \ldots, 10^{2}\}$. Where a collapsed cell averages two conditions with different p, the boundary-hit fraction is reported separately for each.

Table 6: Set I, $K = 3$, $p = 10$. Uncollapsed. Cell format and bold rule as in Table 2.

(δ, ρ)	$n_{\text{train}}/p = 42$	$n_{\text{train}}/p = 84$	Bayes-opt
(1, 0.0)	0.604/0.600 (+0.004, 0.0022)	0.618/0.619 (-0.001, 0.0016)	0.62
(1, 0.5)	0.669/0.667 (+0.002, 0.0029)	0.672/0.674 (-0.001, 0.0018)	0.68
(1, 0.9)	0.913/0.921 (-0.007, 0.0021)	0.917/0.924 (-0.007, 0.0012)	0.93
(1.5, 0.0)	0.757/0.755 (+0.002, 0.0024)	0.734/0.733 (+0.001, 0.0010)	0.75
(1.5, 0.5)	0.804/0.804 (+0.001, 0.0020)	0.821/0.821 (-0.001, 0.0018)	0.82
(1.5, 0.9)	0.962/0.970 (-0.007, 0.0014)	0.981/0.986 (-0.005, 0.0012)	0.98
(2, 0.0)	0.836/0.833 (+0.003, 0.0021)	0.839/0.836 (+0.003, 0.0011)	0.84
(2, 0.5)	0.885/0.887 (-0.002, 0.0021)	0.876/0.879 (-0.003, 0.0014)	0.89
(2, 0.9)	0.992/0.994 (-0.002, 0.0009)	0.991/0.995 (-0.004, 0.0007)	1.00
(3, 0.0)	0.942/0.940 (+0.001, 0.0016)	0.938/0.939 (-0.001, 0.0013)	0.94
(3, 0.5)	0.962/0.965 (-0.003, 0.0016)	0.975/0.976 (-0.001, 0.0010)	0.98
(3, 0.9)	0.997/0.999 (-0.002, 0.0005)	0.997/1.000 (-0.002, 0.0006)	1.00

Table 7: Set I, $K = 3$, $p = 20$. Uncollapsed. Cell format and bold rule as in Table 2.

(δ, ρ)	n_{train}/p = 21	n_{train}/p = 42	Bayes-opt
(1, 0.0)	0.600/0.592 (+0.008, 0.0032)	0.618/0.613 (+0.005, 0.0018)	0.62
(1, 0.5)	0.687/0.683 (+0.004, 0.0031)	0.676/0.671 (+0.004, 0.0019)	0.70
(1, 0.9)	0.941/0.944 (-0.003, 0.0021)	0.934/0.938 (-0.004, 0.0013)	0.95
(1.5, 0.0)	0.740/0.732 (+0.008, 0.0029)	0.734/0.733 (+0.001, 0.0016)	0.75
(1.5, 0.5)	0.826/0.819 (+0.007, 0.0025)	0.822/0.822 (+0.000, 0.0019)	0.84
(1.5, 0.9)	0.990/0.993 (-0.003, 0.0009)	0.989/0.992 (-0.003, 0.0007)	0.99
(2, 0.0)	0.852/0.846 (+0.006, 0.0025)	0.851/0.849 (+0.003, 0.0014)	0.86
(2, 0.5)	0.909/0.908 (+0.002, 0.0021)	0.912/0.910 (+0.002, 0.0013)	0.92
(2, 0.9)	0.996/0.998 (-0.003, 0.0009)	0.998/0.999 (-0.001, 0.0005)	1.00
(3, 0.0)	0.943/0.941 (+0.002, 0.0013)	0.953/0.952 (+0.002, 0.0009)	0.96
(3, 0.5)	0.975/0.975 (+0.000, 0.0009)	0.981/0.983 (-0.002, 0.0007)	0.99
(3, 0.9)	0.999/1.000 (-0.001, 0.0002)	0.999/1.000 (-0.001, 0.0002)	1.00

Table 8: Set I, $K = 3$, $p = 50$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 8.4$	n_{train}/p $= 16.8$	Bayes-opt
(1, 0.0)	**0.582/0.568 (+0.013, 0.0039)**	0.608/0.603 (+0.005, 0.0022)	0.63
(1, 0.5)	**0.669/0.659 (+0.010, 0.0037)**	0.686/0.683 (+0.004, 0.0021)	0.72
(1, 0.9)	0.945/0.941 (+0.003, 0.0022)	0.951/0.951 (+0.000, 0.0010)	0.97
(1.5, 0.0)	0.729/0.724 (+0.005, 0.0031)	0.738/0.733 (+0.005, 0.0019)	0.76
(1.5, 0.5)	0.809/0.802 (+0.007, 0.0026)	0.815/0.811 (+0.004, 0.0016)	0.84
(1.5, 0.9)	0.991/0.993 (-0.001, 0.0008)	0.993/0.995 (-0.002, 0.0005)	1.00
(2, 0.0)	0.844/0.837 (+0.007, 0.0024)	0.861/0.860 (+0.002, 0.0014)	0.87
(2, 0.5)	0.901/0.898 (+0.003, 0.0023)	0.918/0.916 (+0.002, 0.0015)	0.93
(2, 0.9)	0.997/1.000 (-0.002, 0.0006)	0.998/0.999 (-0.001, 0.0003)	1.00
(3, 0.0)	0.963/0.961 (+0.002, 0.0016)	0.963/0.960 (+0.003, 0.0007)	0.96
(3, 0.5)	0.980/0.979 (+0.000, 0.0010)	0.985/0.984 (+0.001, 0.0007)	0.99
(3, 0.9)	0.999/1.000 (-0.001, 0.0003)	0.999/1.000 (-0.001, 0.0003)	1.00

Table 9: Set I, $K = 3$, $p = 100$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 4.2$	n_{train}/p $= 8.4$	Bayes-opt
(1, 0.0)	**0.565/0.554 (+0.011, 0.0038)**	**0.588/0.572 (+0.016, 0.0024)**	0.63
(1, 0.5)	0.608/0.601 (+0.006, 0.0028)	0.666/0.660 (+0.006, 0.0022)	0.71
(1, 0.9)	0.926/0.922 (+0.004, 0.0021)	0.947/0.948 (+0.000, 0.0013)	0.97
(1.5, 0.0)	**0.712/0.699 (+0.013, 0.0033)**	0.730/0.722 (+0.007, 0.0019)	0.76
(1.5, 0.5)	**0.776/0.765 (+0.010, 0.0034)**	0.813/0.806 (+0.007, 0.0019)	0.85
(1.5, 0.9)	0.990/0.990 (+0.001, 0.0010)	0.995/0.996 (-0.001, 0.0004)	1.00
(2, 0.0)	**0.831/0.820 (+0.010, 0.0030)**	0.847/0.840 (+0.007, 0.0015)	0.87
(2, 0.5)	0.884/0.878 (+0.007, 0.0030)	0.905/0.900 (+0.004, 0.0014)	0.93
(2, 0.9)	0.997/0.999 (-0.003, 0.0006)	0.999/1.000 (-0.001, 0.0003)	1.00
(3, 0.0)	0.957/0.954 (+0.002, 0.0013)	0.962/0.957 (+0.005, 0.0011)	0.97
(3, 0.5)	0.979/0.977 (+0.002, 0.0012)	0.984/0.984 (+0.000, 0.0008)	0.99
(3, 0.9)	0.999/1.000 (-0.001, 0.0004)	0.999/1.000 (-0.001, 0.0002)	1.00

Table 10: Set I, $K = 5$, $p = 10$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 70$	n_{train}/p $= 140$	Bayes-opt
(1, 0.0)	**0.464/0.426 (+0.038, 0.0052)**	**0.459/0.411 (+0.048, 0.0045)**	0.47
(1, 0.5)	**0.537/0.521 (+0.015, 0.0031)**	**0.540/0.525 (+0.015, 0.0026)**	0.55
(1, 0.9)	**0.836/0.850 (-0.014, 0.0025)**	**0.851/0.867 (-0.016, 0.0015)**	0.87
(1.5, 0.0)	0.609/0.604 (+0.005, 0.0021)	0.629/0.628 (+0.001, 0.0014)	0.63
(1.5, 0.5)	0.682/0.684 (-0.002, 0.0020)	0.719/0.723 (-0.004, 0.0015)	0.72
(1.5, 0.9)	0.943/0.951 (-0.008, 0.0019)	**0.956/0.969 (-0.013, 0.0012)**	0.97
(2, 0.0)	0.737/0.732 (+0.004, 0.0015)	0.736/0.733 (+0.003, 0.0012)	0.74
(2, 0.5)	0.823/0.829 (-0.006, 0.0019)	0.822/0.825 (-0.003, 0.0013)	0.83
(2, 0.9)	**0.975/0.986 (-0.010, 0.0014)**	0.983/0.991 (-0.007, 0.0010)	0.99
(3, 0.0)	0.889/0.885 (+0.005, 0.0015)	0.901/0.900 (+0.000, 0.0010)	0.90
(3, 0.5)	0.945/0.949 (-0.003, 0.0014)	0.941/0.947 (-0.006, 0.0010)	0.95
(3, 0.9)	0.995/0.999 (-0.003, 0.0007)	0.994/0.998 (-0.004, 0.0009)	1.00

Table 11: Set I, $K = 5$, $p = 20$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 35$	n_{train}/p $= 70$	Bayes-opt
(1, 0.0)	**0.459/0.440 (+0.019, 0.0034)**	**0.479/0.456 (+0.023, 0.0033)**	0.49
(1, 0.5)	0.543/0.540 (+0.003, 0.0022)	0.554/0.548 (+0.006, 0.0019)	0.57
(1, 0.9)	0.893/0.895 (-0.003, 0.0015)	0.903/0.910 (-0.007, 0.0012)	0.92
(1.5, 0.0)	**0.635/0.620 (+0.015, 0.0020)**	0.625/0.620 (+0.005, 0.0012)	0.64
(1.5, 0.5)	0.728/0.723 (+0.005, 0.0024)	0.731/0.728 (+0.003, 0.0012)	0.74
(1.5, 0.9)	0.977/0.980 (-0.003, 0.0010)	0.982/0.985 (-0.004, 0.0007)	0.99
(2, 0.0)	0.746/0.740 (+0.005, 0.0019)	0.762/0.757 (+0.005, 0.0012)	0.77
(2, 0.5)	0.850/0.846 (+0.004, 0.0019)	0.853/0.855 (-0.001, 0.0011)	0.87
(2, 0.9)	0.992/0.996 (-0.003, 0.0006)	0.993/0.996 (-0.003, 0.0005)	1.00
(3, 0.0)	0.916/0.912 (+0.003, 0.0013)	0.910/0.907 (+0.004, 0.0009)	0.92
(3, 0.5)	0.964/0.961 (+0.003, 0.0011)	0.958/0.961 (-0.003, 0.0007)	0.97
(3, 0.9)	0.998/0.999 (-0.001, 0.0003)	0.999/1.000 (-0.001, 0.0002)	1.00

Table 12: Set I, $K = 5$, $p = 50$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p = 14	n_{train}/p = 28	Bayes-opt
(1, 0.0)	**0.434/0.418 (+0.016, 0.0024)**	**0.471/0.455 (+0.016, 0.0018)**	0.49
(1, 0.5)	**0.530/0.519 (+0.010, 0.0028)**	0.558/0.553 (+0.006, 0.0018)	0.58
(1, 0.9)	0.901/0.900 (+0.001, 0.0018)	0.919/0.921 (-0.002, 0.0013)	0.94
(1.5, 0.0)	**0.608/0.588 (+0.020, 0.0029)**	**0.629/0.613 (+0.016, 0.0016)**	0.65
(1.5, 0.5)	0.713/0.706 (+0.008, 0.0023)	0.737/0.731 (+0.006, 0.0016)	0.76
(1.5, 0.9)	0.985/0.986 (-0.001, 0.0011)	0.988/0.990 (-0.002, 0.0004)	0.99
(2, 0.0)	**0.749/0.731 (+0.018, 0.0021)**	**0.769/0.757 (+0.012, 0.0013)**	0.78
(2, 0.5)	0.845/0.837 (+0.008, 0.0023)	0.862/0.859 (+0.003, 0.0011)	0.88
(2, 0.9)	0.996/0.998 (-0.002, 0.0005)	0.998/0.999 (-0.001, 0.0003)	1.00
(3, 0.0)	0.927/0.922 (+0.005, 0.0017)	0.931/0.927 (+0.004, 0.0008)	0.94
(3, 0.5)	0.973/0.972 (+0.001, 0.0010)	0.973/0.972 (+0.001, 0.0006)	0.98
(3, 0.9)	0.999/1.000 (-0.001, 0.0002)	0.999/1.000 (-0.001, 0.0002)	1.00

Table 13: Set I, $K = 5$, $p = 100$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 7$	n_{train}/p $= 14$	Bayes-opt
(1, 0.0)	**0.405/0.381 (+0.024, 0.0031)**	**0.441/0.425 (+0.016, 0.0016)**	0.49
(1, 0.5)	**0.479/0.463 (+0.016, 0.0030)**	0.527/0.520 (+0.007, 0.0017)	0.59
(1, 0.9)	0.892/0.882 (+0.010, 0.0022)	0.917/0.911 (+0.006, 0.0011)	0.95
(1.5, 0.0)	**0.593/0.566 (+0.027, 0.0032)**	**0.624/0.603 (+0.021, 0.0018)**	0.65
(1.5, 0.5)	**0.679/0.668 (+0.011, 0.0031)**	**0.714/0.703 (+0.011, 0.0016)**	0.76
(1.5, 0.9)	0.986/0.986 (-0.001, 0.0008)	0.989/0.990 (-0.002, 0.0005)	1.00
(2, 0.0)	**0.739/0.714 (+0.026, 0.0029)**	0.761/0.751 (+0.009, 0.0014)	0.78
(2, 0.5)	0.835/0.826 (+0.008, 0.0022)	0.852/0.846 (+0.006, 0.0014)	0.89
(2, 0.9)	0.998/0.999 (-0.001, 0.0004)	0.999/0.999 (-0.001, 0.0002)	1.00
(3, 0.0)	**0.925/0.913 (+0.012, 0.0017)**	0.934/0.929 (+0.005, 0.0008)	0.94
(3, 0.5)	0.967/0.966 (+0.002, 0.0012)	0.976/0.974 (+0.002, 0.0007)	0.99
(3, 0.9)	1.000/1.000 (+0.000, 0.0002)	1.000/1.000 (+0.000, 0.0001)	1.00

Table 14: Set II, $K = 3$, $p = 200$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 0.52$	n_{train}/p $= 1.05$	n_{train}/p $= 2.1$	Bayes-opt
(1, 0.0)	0.444/0.433 (+0.011, 0.0074)	0.476/0.471 (+0.006, 0.0052)	**0.517/0.505 (+0.012, 0.0039)**	0.63
(1, 0.5)	0.445/0.442 (+0.004, 0.0072)	0.505/0.497 (+0.008, 0.0054)	0.556/0.549 (+0.007, 0.0038)	0.71
(1, 0.9)	0.608/0.619 (-0.010, 0.0092)	0.790/0.790 (+0.000, 0.0042)	**0.888/0.875 (+0.013, 0.0022)**	0.97
(2, 0.0)	0.703/0.706 (-0.003, 0.0075)	**0.756/0.739 (+0.016, 0.0052)**	0.806/0.796 (+0.010, 0.0023)	0.86
(2, 0.5)	**0.671/0.653 (+0.018, 0.0061)**	**0.781/0.765 (+0.016, 0.0049)**	**0.848/0.834 (+0.014, 0.0036)**	0.94
(2, 0.9)	**0.943/0.955 (-0.012, 0.0033)**	0.993/0.995 (-0.002, 0.0011)	0.997/0.998 (-0.001, 0.0009)	1.00
(3, 0.0)	**0.903/0.892 (+0.012, 0.0048)**	0.936/0.929 (+0.007, 0.0024)	0.954/0.945 (+0.009, 0.0017)	0.97
(3, 0.5)	0.880/0.871 (+0.008, 0.0050)	0.950/0.945 (+0.006, 0.0032)	0.973/0.970 (+0.003, 0.0013)	0.99
(3, 0.9)	0.991/0.997 (-0.006, 0.0017)	0.997/1.000 (-0.002, 0.0010)	0.999/1.000 (-0.001, 0.0004)	1.00

Table 15: Set II, $K = 3$, $p = 400$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 0.26$	n_{train}/p $= 0.52$	n_{train}/p $= 1.05$	Bayes-opt
(1, 0.0)	0.411/0.420 (-0.010, 0.0065)	0.439/0.433 (+0.006, 0.0057)	0.471/0.465 (+0.006, 0.0047)	0.63
(1, 0.5)	0.411/0.400 (+0.011, 0.0069)	0.444/0.446 (-0.002, 0.0056)	0.493/0.489 (+0.004, 0.0038)	0.71
(1, 0.9)	0.442/0.439 (+0.003, 0.0070)	0.615/0.623 (-0.008, 0.0036)	0.801/0.796 (+0.005, 0.0029)	0.97
(2, 0.0)	0.628/0.626 (+0.003, 0.0075)	**0.710/0.696 (+0.015, 0.0046)**	**0.762/0.738 (+0.024, 0.0035)**	0.86
(2, 0.5)	0.579/0.577 (+0.001, 0.0057)	**0.690/0.679 (+0.011, 0.0047)**	**0.785/0.770 (+0.014, 0.0034)**	0.94
(2, 0.9)	**0.739/0.778 (-0.039, 0.0047)**	**0.936/0.952 (-0.016, 0.0024)**	0.992/0.993 (-0.001, 0.0007)	1.00
(3, 0.0)	**0.841/0.826 (+0.016, 0.0048)**	**0.901/0.882 (+0.019, 0.0026)**	**0.942/0.923 (+0.019, 0.0020)**	0.97
(3, 0.5)	0.837/0.828 (+0.009, 0.0050)	0.905/0.905 (+0.000, 0.0036)	0.948/0.946 (+0.002, 0.0020)	0.99
(3, 0.9)	**0.925/0.953 (-0.028, 0.0046)**	0.994/0.998 (-0.004, 0.0009)	0.999/1.000 (-0.001, 0.0004)	1.00

Table 16: Set II, $K = 5$, $p = 200$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 0.88$	n_{train}/p $= 1.75$	n_{train}/p $= 3.5$	Bayes-opt
(1, 0.0)	0.287/0.288 (-0.001, 0.0054)	0.322/0.315 (+0.007, 0.0037)	**0.371/0.354 (+0.017, 0.0040)**	0.49
(1, 0.5)	0.309/0.298 (+0.011, 0.0058)	0.350/0.345 (+0.005, 0.0044)	0.424/0.418 (+0.006, 0.0034)	0.58
(1, 0.9)	**0.533/0.545 (-0.012, 0.0059)**	0.730/0.723 (+0.007, 0.0030)	**0.851/0.827 (+0.024, 0.0021)**	0.95
(2, 0.0)	**0.559/0.535 (+0.023, 0.0063)**	**0.654/0.617 (+0.037, 0.0038)**	**0.703/0.679 (+0.024, 0.0024)**	0.79
(2, 0.5)	0.574/0.568 (+0.007, 0.0049)	**0.696/0.681 (+0.016, 0.0044)**	**0.774/0.758 (+0.016, 0.0029)**	0.89
(2, 0.9)	**0.940/0.956 (-0.016, 0.0031)**	0.990/0.991 (-0.001, 0.0007)	0.998/0.998 (+0.000, 0.0004)	1.00
(3, 0.0)	**0.836/0.812 (+0.024, 0.0052)**	**0.893/0.876 (+0.017, 0.0032)**	**0.914/0.898 (+0.015, 0.0018)**	0.94
(3, 0.5)	0.846/0.839 (+0.007, 0.0044)	0.931/0.924 (+0.007, 0.0020)	0.959/0.954 (+0.005, 0.0014)	0.99
(3, 0.9)	0.994/0.998 (-0.004, 0.0013)	0.999/0.999 (+0.000, 0.0004)	0.999/1.000 (-0.001, 0.0002)	1.00

Table 17: Set II, $K = 5$, $p = 400$. Uncollapsed. Cell format and bold rule as in Table 2.

$(\delta,\ \rho)$	n_{train}/p $= 0.44$	n_{train}/p $= 0.88$	n_{train}/p $= 1.75$	Bayes-opt
(1, 0.0)	0.279/0.275 (+0.005, 0.0044)	**0.291/0.280 (+0.011, 0.0048)**	0.326/0.316 (+0.010, 0.0038)	0.49
(1, 0.5)	0.273/0.273 (+0.000, 0.0049)	0.303/0.302 (+0.001, 0.0035)	0.347/0.342 (+0.005, 0.0029)	0.59
(1, 0.9)	**0.325/0.345 (-0.020, 0.0070)**	**0.548/0.562 (-0.014, 0.0035)**	**0.752/0.739 (+0.013, 0.0022)**	0.95
(2, 0.0)	**0.493/0.477 (+0.015, 0.0051)**	**0.580/0.539 (+0.041, 0.0049)**	**0.654/0.619 (+0.035, 0.0028)**	0.79
(2, 0.5)	0.474/0.470 (+0.003, 0.0058)	0.585/0.580 (+0.004, 0.0035)	**0.697/0.682 (+0.015, 0.0031)**	0.90
(2, 0.9)	**0.710/0.763 (-0.053, 0.0049)**	**0.942/0.957 (-0.015, 0.0020)**	0.994/0.994 (+0.000, 0.0005)	1.00
(3, 0.0)	**0.765/0.741 (+0.024, 0.0051)**	**0.848/0.822 (+0.026, 0.0032)**	**0.898/0.868 (+0.029, 0.0019)**	0.95
(3, 0.5)	0.739/0.739 (+0.000, 0.0051)	**0.864/0.853 (+0.011, 0.0023)**	0.925/0.922 (+0.003, 0.0018)	0.99
(3, 0.9)	**0.942/0.970 (-0.028, 0.0032)**	0.997/1.000 (-0.003, 0.0007)	0.999/1.000 (+0.000, 0.0002)	1.00

Table 18: Highlighted cells and SVM boundary-hit fractions: Set I, $K = 3$.

$(\delta,\ \rho)$	n_{train}/p	$\bar{\Delta}_{\text{acc}}$	SE	SVM boundary-hit fraction
(1, 0)	4.2	+0.011	0.0038	72%
(1, 0)	8.4	+0.014	0.0023	$p=50$: 44% / $p=100$: 70%
(1.5, 0)	4.2	+0.013	0.0033	84%
(1.5, 0.5)	4.2	+0.010	0.0034	20%
(2, 0)	4.2	+0.010	0.0030	96%

Table 19: Highlighted cells and SVM boundary-hit fractions: Set I, $K = 5$.

$(\delta,\ \rho)$	n_{train}/p	$\bar{\Delta}_{\text{acc}}$	SE	SVM boundary-hit fraction
(1, 0)	7.0	+0.024	0.0031	50%
(1, 0)	14.0	+0.016	0.0015	$p=50$: 46% / $p=100$: 44%
(1, 0)	28.0	+0.016	0.0018	26%
(1, 0)	35.0	+0.019	0.0034	36%
(1, 0)	70.0	+0.031	0.0031	$p=10$: 34% / $p=20$: 24%
(1, 0)	140.0	+0.048	0.0045	30%
(1, 0.5)	7.0	+0.016	0.0030	20%
(1, 0.5)	70.0	+0.011	0.0018	$p=10$: 24% / $p=20$: 26%
(1, 0.5)	140.0	+0.015	0.0026	14%
(1, 0.9)	70.0	−0.011	0.0014	$p=10$: 14% / $p=20$: 2%
(1, 0.9)	140.0	−0.016	0.0015	22%
(1.5, 0)	7.0	+0.027	0.0032	76%
(1.5, 0)	14.0	+0.020	0.0017	$p=50$: 40% / $p=100$: 68%
(1.5, 0)	28.0	+0.016	0.0016	30%
(1.5, 0)	35.0	+0.015	0.0020	14%
(1.5, 0.5)	7.0	+0.011	0.0031	12%
(1.5, 0.9)	140.0	−0.013	0.0012	4%
(2, 0)	7.0	+0.026	0.0029	86%
(2, 0)	14.0	+0.014	0.0013	$p=50$: 34% / $p=100$: 98%
(2, 0)	28.0	+0.012	0.0013	52%
(3, 0)	7.0	+0.012	0.0017	98%

Table 20: Highlighted cells and SVM boundary-hit fractions: Set II, $K = 3$.

$(\delta,\ \rho)$	n_{train}/p	$\bar{\Delta}_{\text{acc}}$	SE	SVM boundary-hit fraction
(1, 0)	2.1	+0.012	0.0039	74%
(1, 0.9)	2.1	+0.013	0.0022	0%
(2, 0)	1.05	+0.020	0.0031	$p=200$: 94% / $p=400$: 96%
(2, 0.5)	0.53	+0.015	0.0039	$p=200$: 42% / $p=400$: 44%
(2, 0.5)	1.05	+0.015	0.0030	$p=200$: 48% / $p=400$: 64%
(2, 0.5)	2.1	+0.014	0.0036	44%
(2, 0.9)	0.26	−0.039	0.0047	0%
(2, 0.9)	0.53	−0.014	0.0020	$p=200$: 0% / $p=400$: 0%
(3, 0)	0.26	+0.016	0.0048	68%
(3, 0)	0.53	+0.015	0.0027	$p=200$: 84% / $p=400$: 90%
(3, 0)	1.05	+0.013	0.0016	$p=200$: 98% / $p=400$: 100%
(3, 0.9)	0.26	−0.028	0.0046	0%

Table 21: Highlighted cells and SVM boundary-hit fractions: Set II, $K = 5$.

$(\delta,\, \rho)$	n_{train}/p	$\bar{\Delta}_{\text{acc}}$	SE	SVM boundary-hit fraction
(1, 0)	3.5	+0.017	0.0040	64%
(1, 0.9)	0.44	−0.020	0.0070	4%
(1, 0.9)	0.88	−0.013	0.0034	$p=200$: 0% / $p=400$: 0%
(1, 0.9)	1.75	+0.010	0.0018	$p=200$: 0% / $p=400$: 0%
(1, 0.9)	3.5	+0.024	0.0021	0%
(2, 0)	0.44	+0.015	0.0051	60%
(2, 0)	0.88	+0.032	0.0040	$p=200$: 70% / $p=400$: 84%
(2, 0)	1.75	+0.036	0.0023	$p=200$: 90% / $p=400$: 100%
(2, 0)	3.5	+0.024	0.0024	98%
(2, 0.5)	1.75	+0.015	0.0027	$p=200$: 18% / $p=400$: 46%
(2, 0.5)	3.5	+0.016	0.0029	28%
(2, 0.9)	0.44	−0.053	0.0049	0%
(2, 0.9)	0.88	−0.015	0.0019	$p=200$: 0% / $p=400$: 0%
(3, 0)	0.44	+0.024	0.0051	86%
(3, 0)	0.88	+0.025	0.0030	$p=200$: 84% / $p=400$: 94%
(3, 0)	1.75	+0.023	0.0019	$p=200$: 98% / $p=400$: 100%
(3, 0)	3.5	+0.015	0.0018	100%
(3, 0.9)	0.44	−0.028	0.0032	0%