Cross-Validation for Model Assessment and Selection with Extensions to Spatial Data

June 29th, 2018

Abstract: Cross-validation is a popular computational method used for model assessment and selection. With spatial data, however, many of the independence assumptions behind cross-validation break down. In this paper, we engage in an in-depth study of cross-validation methods for model assessment and selection on spatial data. We conclude with a simple simulation study to compare the performances of nonspatial and spatial cross-validation methods on simulated spatial data. Though there is more work to be done, our initial results indicate that spatial methods indeed outperform nonspatial ones when applied to model assessment and selection in spatial contexts.

Chapter 1

Introduction

Recent years have seen an explosion of interest in "machine learning": the production of models from data, broadly speaking. As the modern world continues to churn out data at unprecedented rates, researchers and practitioners alike are scrambling to develop models to make sense of all of it. Producing these models, however, is only half the battle; once produced, models need to be tested to see how well they can be expected to perform relative to other models. It has therefore become increasingly important to understand not only how to model data, but also how to assess and compare different statistical models to choose the best one for the problem at hand.

Statisticians have developed many methods to deal with this problem of *model selection*: how to choose the best model for the real-world mechanisms hidden in the data. One popular and intuitive method for doing so draws on tools from *model assessment*. The idea is to first estimate how the models in consideration would perform on new data, and to then choose the model estimated to have the best performance.

Though model assessment and selection techniques can be applied to nearly any kind of data, their applications to spatial data result in some unique challenges. This is primarily due to *spatial autocorrelation*, or, as famously stated by geographer Waldo Tobler, the first law of geography: "Everything is related to everything else, but near things are more related than distant things" [38]. The strong relationships between nearby observations in spatial data violate many of the independence assumptions underpinning popular model assessment and selection methods. To address the issues that arise, researchers have proposed a variety of spatial model assessment and selection methods.

This paper will focus on the use of *cross-validation* (CV) for model assessment and selection, with three primary goals:

1. To clearly explore the properties of CV in the context of model assessment and selection
2. To motivate and survey extensions of CV to spatial data
3. To show that spatial CV methods indeed outperform nonspatial ones at model assessment and selection using simulated data

The chapters are organized as follows. Chapter 2 introduces CV for model assessment. Chapter 3 motivates and explores extensions of CV to spatial settings. Chapter 4 applies the various CV methods to the problem of model selection in both spatial and nonspatial settings. Finally, Chapter 5 runs the abovementioned simulations to study how well spatial CV methods perform at model assessment and selection.

Chapter 2

Cross-Validation for Model Assessment

Model assessment methods aim to estimate how well a given model will perform on new data. This chapter will first develop some brief background on statistical modeling and cross-validation before diving into the details of cross-validation for model assessment.[1]

2.1 Background

This paper focuses on methods for assessing *supervised learning* models, fit on labeled *training sets* $\mathcal{T}$ of N observations $(\mathbf{x_i}, y_i)$. The $\mathbf{x_i}$ are vectors of *predictor variables*, and the y_i are *response variables*. Each y_i is assumed to be of the form $f(\mathbf{x_i}) + \epsilon_i$, where $f\colon \mathbf{x_i} \to y_i$ is some unknown *data-generating function* (or *signal*) and ϵ_i is the associated random *irreducible error* (or *noise*).

The goal of supervised learning is to use $\mathcal{T}$ to construct a model $\hat{f}\colon \mathbf{x} \to y$ that can input any $\mathbf{x}$ and output a corresponding prediction $\hat{y} = \hat{f}(\mathbf{x})$ of its true label $f(\mathbf{x})$. Model assessment methods estimate how close $\hat{f}(\mathbf{x})$ is expected to be to $f(\mathbf{x})$ for a random $\mathbf{x}$ from the population. One naive way to assess a model is to compute its *training error*:

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(\mathbf{x_i})).$$

Here, $L(y_i, \hat{y}_i)$ is a *loss function*, some nonnegative measurement of error, i.e., how much the predicted $\hat{y}_i = \hat{f}(\mathbf{x_i})$ differs from the actual y_i.[2]

Training error measures how well a model can predict the labels of the data it is trained on. Users, though, are generally more interested in knowing how well their models would predict the labels of unseen, unlabeled observations – the *out-of-sample error* of their models.

Training error, however, is typically a poor estimate of out-of-sample error because of *overfitting*. Intuitively, overfitting occurs when models "memorize" training data instead of extracting general rules. Recall that labels $y_i = f(\mathbf{x_i}) + \epsilon_i$ are combinations of signal and random noise, and consider two random samples $\mathcal{T}$ and $\mathcal{V}$ with signal f and respective noise vectors $\epsilon_\mathcal{T}$ and $\epsilon_\mathcal{V}$. A model $\hat{f}_\mathcal{T}$ overfit to $\mathcal{T}$ would approximate $f + \epsilon_\mathcal{T}$ instead of f. Clearly, while $\hat{f}_\mathcal{T}$ would perform well on $\mathcal{T}$, it would perform much more poorly on $\mathcal{V}$ due to the differences between $\epsilon_\mathcal{T}$ and $\epsilon_\mathcal{V}$.

Supervised learning models tend to perform better on their training sets than on new observations, which may involve new values of $\mathbf{x_i}$ or additional noise. Because of this, we say that training error is too *optimistic* of an estimate of out-of-sample error. A simple way to address this problem

[1] The notation and definitions in this chapter generally follow from [16], unless otherwise noted.

[2] E.g., one common loss function for regression problems is squared error, $(y_i - \hat{y}_i)^2$.

is to use the *validation set* approach. Instead of training the model on all of the given data, the data are split into two sets: the model is trained on the training set, and used to predict on the validation set. The average loss between these predictions and the actual labels of the validation set produces an estimate of the model's out-of-sample error.

Notably, using a validation set means that the model is not trained on any of the data used to evaluate it. The latent assumption is that this means the training set $\mathcal{T}$ is independent from the validation set $\mathcal{V}$.[3] If this assumption holds, then $\mathcal{V}$ represents an independent, out-of-sample set of new data, so the model's error on $\mathcal{V}$ should be similar to its out-of-sample error.

While the validation-set approach is effective at model assessment, it has a major drawback. In many situations, it is costly to sacrifice training data to validate models. Setting aside data for validation means reducing the size of the training sample, which can lead to weaker predictive models and less certainty about relationships found in the data. To address this issue, techniques have been developed to estimate the out-of-sample error rates of models using all of the given data. Though there are many such methods, this paper will focus on cross-validation.

2.2 Cross-Validation

Cross-validation (CV) is a popular model-assessment method. Like the validation-set approach, CV estimates out-of-sample error by splitting the given data into training and validation sets. Rather than doing so only once, however, CV splits it multiple times and averages errors over all splits. In this manner, CV can use all of the given data to train models while still evaluating the models on out-of-sample observations.

K-fold Cross-Validation

While there are many CV methods,[4] the most popular CV method is *K-fold cross-validation*, shown in Algorithm 1. This paper will primarily focus on K-fold CV and the related *leave-one-out cross-validation* (LOOCV), which is K-fold CV when $K = N$, the number of observations in the data.

Algorithm 1 K-fold Cross-Validation

1: Randomly partition data $(\mathbf{x_i}, y_i)$ into K equally sized folds F_k
2: **for** each fold F_k **do**
3: `train` $\leftarrow \{(\mathbf{x_i}, y_i) \mid (\mathbf{x_i}, y_i) \notin F_k\}$
4: `validate` $\leftarrow \{(\mathbf{x_i}, y_i) \mid (\mathbf{x_i}, y_i) \in F_k\}$
5: Train model m_k on `train`
6: Record performance p_k of m_k on `validate`
7: **return** $\frac{1}{K} \sum_{k=1}^{K} p_k$

2.3 Conditional vs. Expected Test Error

Up to this point, we have seen how CV estimates out-of-sample error, generally speaking. Clearly distinguishing the specific type of error that CV estimates, though, reveals some important theoretical and practical complications.

[3] I.e., the occurrence of observations in $\mathcal{T}$ should not affect the occurrence of observations in $\mathcal{V}$.

[4] A more detailed survey of different CV methods can be found in [3].

Given a training set $\mathcal{T} = (\mathbf{x_i}, y_i)$, there are two kinds of out-of-sample error that one might want to estimate. The first is *conditional test error*:

$$Err_{\mathcal{T}} = \mathbb{E}_{\mathbf{x_i}, y_i}[L(y_i, \hat{f}(\mathbf{x_i}))|\mathcal{T}].$$

Conditional test error is the error that a model trained on $\mathcal{T}$ is expected to have on points drawn from the same population as $\mathcal{T}$. The second kind of error is *expected test error*:

$$Err = \mathbb{E}_{\mathcal{T}}[\mathbb{E}_{\mathbf{x_i}, y_i}[L(y_i, \hat{f}(\mathbf{x_i}))|\mathcal{T}]],$$

which is the expected error of a model trained on a random training set of the same size as $\mathcal{T}$ drawn from the population [16].

Though these two errors look similar, they have distinct uses. For model assessment, practitioners want to estimate the predictive performances of the models trained on particular training sets $\mathcal{T}$. As such, they would like to use CV to estimate $Err_{\mathcal{T}}$. Unfortunately, CV provides better estimates of Err; in fact, simulation studies show that K-fold CV estimates of error are actually *negatively* correlated with $Err_{\mathcal{T}}$ (see Section 12.2 of [14], Section 7.12 of [16]). Intuitively, this is because K-fold CV partitions the original training set $\mathcal{T}$ into training and validation sets, so it only ever uses training sets sampled from $\mathcal{T}$, not $\mathcal{T}$ in its entirety.

[16] concludes that "estimation of test error for a particular training set is not easy in general, given just the data from that same training set. Instead, cross-validation and related methods may provide reasonable estimates of the *expected* error Err."[5] Though this conclusion is discouraging, nonparametric methods like CV are still the most intuitive methods to estimate test error for many models. As such, it remains important to study their properties; indeed, one could argue that the result described above makes it all the more important to precisely study how they behave.

2.4 The Bias-Variance Tradeoff in Model Assessment

While there is no best model-assessment method for in all situations, assessment methods can be compared in terms of the well-known *bias-variance tradeoff* in statistics. In the context of model assessment, the tradeoff is between the bias and variance of the error estimates produced by CV.

All things remaining equal, models tend to perform better when trained on more data. *Learning curves* visualize this improvement by plotting the validation accuracy of a model against the size N_t of its training set. Figure 2.1 shows how models typically have steep gains in accuracy as small training sets grow larger (to the left of the red line), with smaller gains as $N_t \to \infty$ [16]. The learning curve tells us that a model trained on all of the given data $\mathcal{T}$ will tend to perform better on out-of-sample data than a model trained on only some of the data, particularly if $\mathcal{T}$ is small to begin with. In holding out data to use for validation, then, CV pessimistically biases (i.e., inflates) the error-rate estimates for the model. The more data held out for validation, the more biased the error estimates will be.

Holding out less data for validation, though, increases the variance of the error estimates. When few observations are held out for validation in CV, the resampled training datasets are all similar to the original $\mathcal{T}$, so the trained models are all similar to each other. In other words, the trained models are all highly dependent on the particular $\mathcal{T}$ originally sampled from the population. If a new training set $\mathcal{T}'$ were drawn from the population, all of the trained models would

[5] A brief survey of some other literature on whether K-fold CV estimates conditional or expected error can be found in [5], which also concludes that K-fold CV serves as a better estimator of expected error.

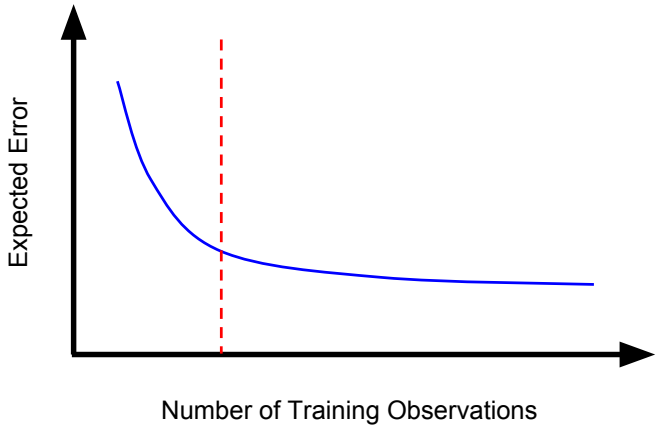

Figure 2.1: Hypothetical model learning curve

change significantly. This between-model dependence leads to large changes in out-of-sample error estimates for different draws from the population, i.e., high variance [16].[6]

The Bias-Variance Tradeoff in K-Fold Cross-Validation

In K-fold CV, bias and variance are controlled by the user-defined parameter K. Figure 2.2 gives an overview of how K affects bias and variance. In general, we will see that as K increases (from 2 to N), bias decreases but variance increases.[7]

The behavior of the bias of K-fold CV is well-documented. [3] shows that for independent training and validation sets, bias depends on the difference in performance of the algorithm trained on N_t versus N observations, where N_t is the number of training observations in a given fold. Since models improve as they are trained on more observations, K-fold CV is negatively biased, but less so as N_t approaches N. Recalling Figure 2.1, we note that choosing values of K that result in training sets that are too small (i.e., to the left of the red line) would likely result in large increases in bias.

The behavior of the variance of K-fold CV is a much more challenging problem in general. Ideally, one would compute the variance of K-fold CV for a given dataset with N observations by drawing many independent datasets of N observations from the same population and simply computing the sample variance of the K-fold CV error estimates on those new datasets. Of course, we typically use CV only when it is difficult to draw new observations from the population, so the variance of the CV estimator needs to be estimated using only the given dataset. Unfortunately, [5] proves that there are no unbiased estimators of the variance of the K-fold CV estimate of the expected test error. Without getting into the details of their proof, this is because the overlaps between the training sets used for K-fold CV produce dependencies between error estimates, so simply using the population variance of the K error estimates underestimates the true variance.

[6] Alternatively, one could think of the error estimates for each of the R resamples r as random variables E_r, so the overall error estimate is $\frac{1}{R}\sum_{r=1}^{R} E_r$. Since $Var(\sum_{r=1}^{R} E_r) = \sum_{r=1}^{R}\sum_{s=1}^{R} Cov(E_r, E_s)$, we see that the variance of the overall estimate increases as the covariance between the errors estimated based on different resamples increases.

[7] The one notable exception is in the cases of very low K (e.g., $K = 2$). In these cases, variance is higher due to the simple fact that there are more ways to assign points into K folds for low values of K. For example, there are $\binom{N}{N/2}$ ways to partition N observations into two folds, which maximizes $\binom{N}{k}$ for all k.

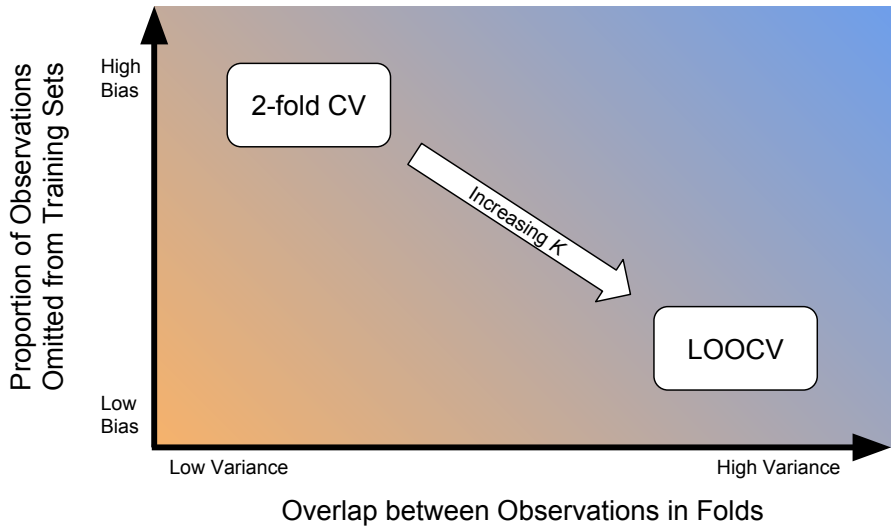

Figure 2.2: Rough intuition for the bias-variance tradeoff in K-fold CV

Regardless, [3] provides some basic intuition for the variance of CV methods. It shows that the variance of a validation-set estimate of model error $\widehat{Err}_{\mathcal{V}}$ is roughly proportional to $\frac{1}{N_v}$ plus the variability of the model trained on N_t observations, where N_v and N_t are the sizes of the validation set and its corresponding training set. The formula shows that as N_v increases, the variance of $\widehat{Err}_{\mathcal{V}}$ decreases. Also, it shows that the variance of $\widehat{Err}_{\mathcal{V}}$ depends on the stability of the given model. [30] points out that most models become more stable as their training sets get larger, so the variance of $\widehat{Err}_{\mathcal{V}}$ should decrease as N_t increases.

While the intuition described in the previous paragraph is useful, it is difficult to apply to K-fold CV, since N_t and N_v are inversely related and depend directly on N and K. Also, model stability varies dramatically across models and sample sizes. Simulation studies are therefore quite useful for better understanding the variance of K-fold CV.

2.5 Comparisons in the Literature

A wide range of simulation studies in the literature have compared K-fold CV to other model-assessment methods. This paper will not dive into the literature, though the interested reader can find brief surveys in [16] and [23].

The most-cited early simulation study is [23], which provides a comprehensive comparison of K-fold CV and the bootstrap using C4.5 decision trees and naive Bayes classifiers. It recommends 10-fold CV in general, as it provides the best balance of low bias and low variance on the datasets used in the paper. Early papers like [23] generally confirm the bias-variance tradeoff of adjusting K in K-fold CV (see, e.g., [7], [13], [42]).

Later studies include comparisons with repeated K-fold CV and implement more modern learning algorithms (e.g., random forests, neural nets). Though a number of these studies exist, we

highlight [22] here.[8] [22] compares both repeated and non-repeated 10-fold CV to the .632+ bootstrap[9]. It uses pruned decision trees, which are good at generalizing to out-of-sample observations, and the discrete adaboost algorithm, which is highly adaptive to training data (i.e., prone to overfitting). In simulations, repeated 10-fold CV produces estimates with similar bias but less variance than single 10-fold CV, so [22] recommends the use of repeated CV. Though the .632+ bootstrap and repeated 10-fold CV both have low bias and similar variance in most cases, the .632+ is badly optimistically biased in some cases, particularly when an adaptive rule is used with large sample sizes of data. [22] thus recommends the use of repeated 10-fold CV in general.

Overall, the literature generally concludes that 10-fold CV (or repeated 10-fold CV) strikes a good balance between bias and variance for model assessment. This guideline will be kept in mind for the simulations in Chapter 5.

[8] See [6], [29], and [39] for some of the other more relevant ones.

[9] A bias-corrected version of the bootstrap for model assessment; see [16] page 252.

Chapter 3

Model Assessment in a Spatial Context

CV methods work well for most model-assessment problems. On spatial data, however, they break down because they depend on assumptions that are often violated in spatial contexts. In this chapter, we elaborate on these issues before surveying modified CV methods for spatial data.

3.1 Challenges with Spatial Data

Spatial data exhibit a few properties that make it difficult to apply standard statistical methods to them. For one, spatial data exhibit *spatial autocorrelation*, where observations close to each other in space have related values. Figure 3.1 shows this phenomenon in a small sample from a forest ecology dataset collected by Albert Y. Kim[1] and David Allen [2]. Note how tree species cluster together, so any attributes related to species, e.g., tree height, canopy width, resource consumption, etc., will also show similar spatial clustering patterns.

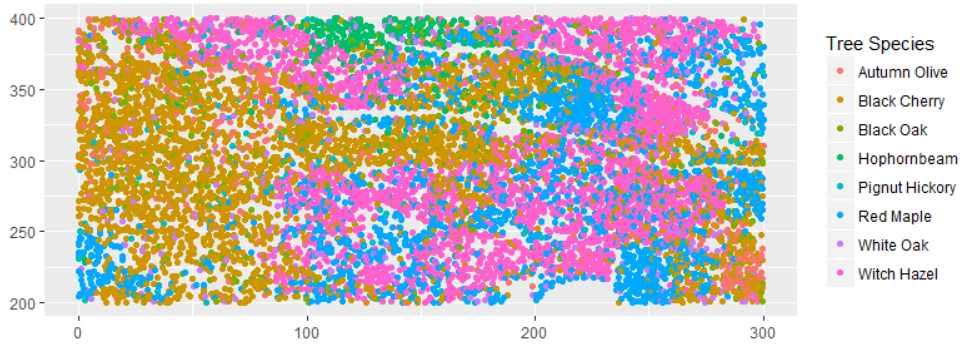

Figure 3.1: Spatial distribution of tree species shows clustering

Spatial autocorrelation often leads to difficulties with statistical methods (e.g., see [25], [26]) because many methods assume that observations are independent and identically distributed (i.i.d.). Spatial autocorrelation, however, means that drawing one observation implies that nearby observations will likely be similar to it, thus making it unreasonable to assume that points are i.i.d. As such, many methods, including modeling and model assessment methods, perform worse when naively applied to spatial data.

[1] Lecturer of Statistics, Amherst College
[2] Assistant Professor in Biology, Middlebury College

Interestingly, CV does not necessarily assume that observations are i.i.d. Instead, it depends on a modified assumption of independence based on sampling and resampling. When scientists gather data, they collect a random *sample* from the population. CV simulates sampling by *resampling*; rather than sampling from the whole population, CV resamples from the given dataset.

CV assumes that the original sample is representative of the population and that its resamples are collected in the same way that a new sample from the population would be collected. If observations are i.i.d., then this assumption is trivially satisfied. Without an i.i.d. assumption, though, the assumption still holds for spatial data so long as the resampled training and validation sets are collected in the same manner as the original sample was: collected in spatial groups.

K-fold CV, however, does not resample spatial groups of data. Instead, it uses random resampling, which produces training and validation sets whose points are distinct but come from overlapping spatial regions. To illustrate this point, Figure 3.2 gives an example of how one fold of 5-fold CV might partition points into a training set (white points) and a validation set (black points).

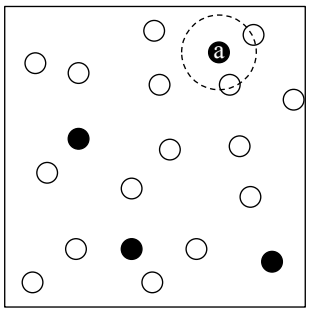

Figure 3.2: Example 5-fold CV fold

Having training-set points interspersed among validation-set points leads to three main challenges for model assessment. First, the presumed independence between the training set and the validation set is broken. Consider the validation-set point a in Figure 3.2. As indicated by the dotted circle, a is very close to two other points in its corresponding training set. Due to spatial autocorrelation, knowing information about the two nearby training-set observations gives information about a.[3] To clarify, suppose that a has a response y_a of the form $y_a = f(\mathbf{x_a}) + \epsilon_a$. The training-set points near a would not only tend to have $\mathbf{x_i}$ similar to $\mathbf{x_a}$, but also tend to have ϵ_i similar to ϵ_a. Thus, if a model overfit the training set, it would perform exceptionally well on a, which would undermine the rationale for using a validation set (instead of training error).

The second challenge is that the spatial autocorrelation structures in the training set are not preserved. The training set in Figure 3.2 contains observations that are more spread out in space, on average, than the original training set. As such, the observations do not exhibit the same degree of spatial autocorrelation as the original training set, which can affect model training.

The final challenge relates to *interpolation* and *extrapolation*. Researchers usually want to assess model extrapolation: how well models can predict observations in spatial regions other than those of the original samples. Such observations will often have different latent-variable values. Interspersing training-set points among validation-set points, however, leads to similarities between the latent-variable values of the training and validation sets, since they are drawn from overlapping spatial regions. Thus, K-fold CV would be better at assessing model interpolation: how well the models can predict observations in the same spatial region as the training set.

[3] [12] notes that spatial observations can exhibit clustering behavior, which would exacerbate this problem.

Challenges in the Literature

The aforementioned theoretical issues with K-fold CV in spatial contexts have led researchers to explore the effects of spatial autocorrelation on model assessment. The field of ecological niche modeling, the modeling of species distributions over time and space, has produced a significant amount of recent research in this area, as ecologists have discovered that failing to use proper validation techniques to evaluate their models leads to unjustified conclusions [34].

Most importantly, ignoring spatial autocorrelation falsely inflates assessments of model accuracy. Using historical data on the distributions of British breeding-bird species, [2] shows that "measures of performance on nonindependent data provided optimistic estimates of models' predictive ability on independent data." Similarly, results of a comprehensive study in [4] confirm that as the independence between training and validation sets increases, assessments of model accuracy decrease, for a variety of assessment and modeling techniques.

The studies cited above focus on validation-set approaches, but ecologists have also studied K-fold CV. [40] shows that using K-fold CV error for assessment leads to overoptimistic error estimates for models that overfit the data. Because CV is highly sensitive to the average distances between training and validation points (see [19]), the proximity of training and validation data in K-fold CV leads to artificially low error estimates for models that can incorporate the information about the validation data contained in the training data. [36] supports this idea. Using data on sea temperatures in the Northern Atlantic, [36] compares K-fold CV estimates of error to extrapolation error, found by assessing model performance on sea temperatures from the Southern Atlantic. They find that K-fold CV significantly underestimates the extrapolation error for neural nets, because neural nets are flexible enough to overfit the spatial autocorrelation structures in the data.

3.2 Methods to Account for Spatial Autocorrelation

There are many ways to modify CV to account for the challenges described in Section 3.1. For spatial data, the resampling in CV needs to mirror the original sampling, so resampled points should be spatially grouped. Figure 3.3 shows an example. On the left, the solid black circles are points that have been randomly sampled (top-left) and resampled (bottom-left). On the right, points are sampled and resampled in spatial clusters. Note that while the sampled/resampled points on the left are uniformly spread across the rectangle, the sampled/resampled points on the right are closer together, which preserves the spatial relationships in the data.

Grouping observations together in specific ways for CV is known as *blocking* [15]. Intuitively, blocking changes the units of resampling from individual observations to user-defined blocks of observations. There are many blocking methods, but this paper will focus on spatial blocking, where blocks are groups of observations that are close to each other in space.

Another modification of CV for spatial data involves the use of *buffer regions*, which are simply points spatially located between the defined training and validation sets that are used in neither set. Using buffer regions accomplishes two goals. First, it reduces the dependencies between points near the edges of training and validation sets [24]. Second, it adds distance between training and validation sets, which increases the amount of extrapolation in a model's predictions on a validation set [15]. In summary, spatial blocking and buffering provide solutions, at least theoretically, to the issues discussed in Section 3.1. The following section will explore how blocking and buffering can be incorporated into CV.

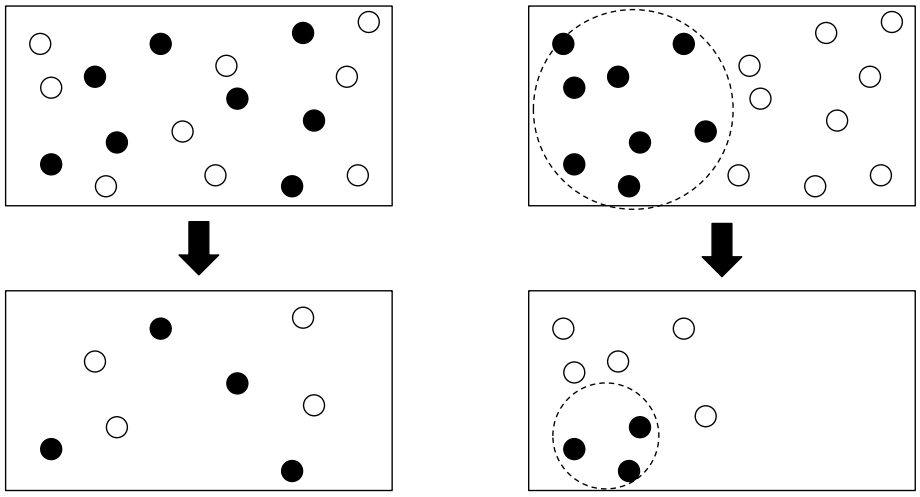

Figure 3.3: Random sampling/resampling vs. spatially grouped sampling/resampling

3.3 Spatial Cross-Validation

Spatial cross-validation involves modifications of CV for spatial data. Instead of defining folds of points at random, folds are defined by spatial boundaries. The most basic case of spatial CV, which we call *grid cross-validation*, defines spatial folds by dividing the data into rectangular regions. The data in each block in the resulting grid is then used as a "fold" for K-fold CV.

One method for incorporating buffer regions into grid CV is described in Algorithm 2, which we call *buffered grid cross-validation* [Kim and Allen, in progress].[4]

Algorithm 2 Buffered Grid Cross-Validation

1: Partition data $(\mathbf{x_i}, y_i)$ into $\mathbf{r} * \mathbf{c}$ folds $F_{r,c}$ of equal spatial dimension ($\mathbf{r}$ rows, $\mathbf{c}$ columns)
2: **for** each fold $F_{r,c}$ **do**
3: `buffer` $\leftarrow \{(\mathbf{x_i}, y_i) \mid (\mathbf{x_i}, y_i) \notin F_{r',c'}, r' = \{r-1, r, r+1\}, c' = \{c-1, c, c+1\}\}$
4: `train` $\leftarrow \{(\mathbf{x_i}, y_i) \mid (\mathbf{x_i}, y_i) \notin$ `buffer`$\}$
5: `validate` $\leftarrow \{(\mathbf{x_i}, y_i) \mid (\mathbf{x_i}, y_i) \in F_{r,c}\}$
6: Train model m_k on `train`
7: Record performance p_k of m_k on `validate`
8: **return** $\frac{1}{\mathbf{r}*\mathbf{c}} \sum_{k=1}^{\mathbf{r}*\mathbf{c}} p_k$

Other Spatial Cross-Validation Methods

Other methods for spatial CV have been proposed in the literature. Some methods block observations in different ways. For example, more oblong rectangular blocks may be useful in many spatial applications, since climate varies more from north-to-south than from east-to-west (see [15]). The `sperrorest` package in R (see [8]) includes spatial blocking based on K-means clustering of the data, as is used to validate landslide models in [17].

[4] See Figure 3.4 for a visual.

Other methods incorporate buffer regions in different ways. *Spatial leave-one-out cross-validation* (SLOO), proposed in [24],[5] is a natural spatial variation of leave-one-out CV. Rather than excluding only one observation from the training set in each fold, SLOO excludes all observations within a small buffer region of the validation observation to preserve independence. [31] proposes a variation of SLOO called *spatial K-fold cross-validation* (SKCV). SKCV is K-fold CV, except for each fold, observations within a small buffer radius of the validation set observations are excluded from the training set. To help visualize these methods, Figure 3.4 illustrates one fold each of buffered grid CV, SLOO, and SKCV (respectively). Black points are points in the validation set, gray points are points in the buffer, and white points are points in the training set for the given fold.

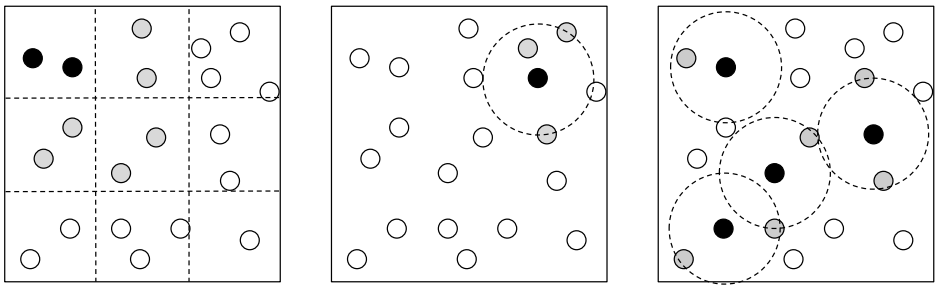

Figure 3.4: One fold of buffered grid CV, SLOO, and SKCV

The three algorithms in Figure 3.4 differ in the sizes of their training sets, validation sets, and buffer regions, which theoretically affects their bias and variance. Buffer regions are particularly interesting with respect to bias and variance. Their primary role is to increase independence between training and validation sets, which should reduce the optimistic bias of nonspatial error estimates. Buffers also reduce overlaps between training sets by removing points from each training set, which may decrease estimate variance (see Section 2.4). At the same time, buffer regions remove data that would have otherwise been used for training, which may cause pessimistic bias. Also, the described buffering methods all remove random numbers of points, which can increase estimate variance. Since buffer regions may theoretically affect bias and variance in many ways, simulation-based results will be important to better understand their effects.

Regardless, we can make some general conclusions about the spatial CV methods we have introduced. For one, we note that SLOO is similar to LOOCV. Since it excludes the fewest buffer points and has the largest possible training set in each of its folds, it should have the lowest bias, but relatively high variance (and computational cost). The bias and variance of buffered grid CV can be tuned by $\mathbf{r}$ and $\mathbf{c}$, with high values leading to similar performance as SLOO, and low values leading to more bias but less variance. SKCV is similar to buffered grid CV. For low values of K and highly clustered points, SKCV may exclude most points from its training sets in each fold, and thus be highly biased. On the other hand, letting $K = N$ leads to SLOO.

Currently, we are unfortunately not aware of any literature that empirically compares these methods. The authors seem to propose them because they are the most natural extensions to CV for the spatial setting, but there is much room for further study.

[5] An earlier paper, [37], proposes SLOO as well, though it does not go into the same level of detail as [24].

Chapter 4

Cross-Validation for Model Selection

Thus far, we have studied CV for model assessment in both nonspatial and spatial contexts. While model assessment is useful on its own, it is also frequently used to compare models. This chapter surveys some theoretical and simulation-based results for CV model selection.

4.1 Why Cross-Validation for Model Selection?

Researchers have developed a huge variety of model selection methods for different modeling frameworks and selection goals. For example, Mallow's C_p [27], Aikaike's Information Criterion (AIC), [1], and the Bayesian Information Criterion (BIC) [33] are all analytic model selection criteria that see frequent use in a wide range of application.

This thesis, however, focuses on CV, a nonparametric model assessment method, for two primary reasons. First, it is not obvious how the theoretical justifications for the methods described above extend to spatial contexts where independence assumptions break down. For example, [3] notes that Mallow's C_p assumes that the variance of the response does not depend on predictor values, which seems particularly untenable in spatial data where both predictor and response values tend to cluster. Second, extending those methods to different models is more difficult than similar extensions of CV, which needs no modification in most cases. In general, CV is much easier to understand than C_p, AIC, or BIC, which can make it more appealing to use, particularly for non-statisticians. Though nonparametric methods like CV can involve much more computation, modern computers have made them relatively easy to run on most datasets.

4.2 Model Selection for Identification

In model selection, the goal is to choose the "best" model from a family of potential models. In general, the "best" model for a given problem can be defined in two ways (see [3]):

1. The model expected to have the least error on new data (selection for *estimation*)
2. The simplest model that can effectively describe the data (selection for *identification*)

In selection for estimation, the goal is to choose the model that will have the least error on new data, given a training set $\mathcal{T}$. Doing so involves estimating the conditional test error $Err_{\mathcal{T}}$ of the candidate models and selecting the one with the least $Err_{\mathcal{T}}$. Selection for identification, on the other hand, aims to choose the model that best describes underlying patterns in the population [22]. Such a model would produce low-error estimates not only if trained on the given $\mathcal{T}$, but also if trained on a new $\mathcal{T}'$ drawn from the same population. Selection-for-estimation methods thus

select the models with the least Err. Since CV is a better estimator of Err (see Section 2.3), this chapter focuses on selection for identification.[1]

4.3 Bias and Variance in Model Selection

The model assessment methods of Chapters 2 and 3 prioritized low bias, since their purpose was to estimate the actual values of out-of-sample error. In model selection, though, the particular error estimates are less important; instead, the ability to confidently compare estimates becomes vital. Low variance is therefore more important than low bias to some extent. Even if the error estimates for all of the models in consideration are highly biased, CV can still select the best model so long as it produces low-variance estimates and its outputs are all biased in the same way.

Of course, the assumption that the estimates are "biased in the same way" for all of the models in consideration is quite strong and difficult to check. It thus may be preferable to only use CV to select between similar kinds of models (e.g., the same supervised learner with different variable inputs), since the overall behaviors of similar, non-overfit models should not change too drastically between appropriate training sets. In any case, while low-bias estimates of model error are nice for model selection, it is more important to have low-variance estimates so that the error estimates for different models can be compared to each other. In theory, then, K-fold CV with $K \ll N$ may produce the best model selection results.

4.4 Cross-Validation for Selection in the Literature

Theoretical Results

The theoretical criterion for selection-for-identification methods is *consistency*. Intuitively, a model selection method is consistent if the probability that the selection method chooses the "true" model approaches one as the training set gets arbitrarily large.

A small body of research has studied the consistency properties of K-fold CV under different conditions. The most famous result is Shao's finding that for linear models, LOOCV is asymptotically inconsistent, while leave-d-out CV[2] is consistent so long as $\frac{d}{N} \to 1$ as $N \to \infty$ [35]. [42] produces similar findings for leave-d-out CV in linear regression, finding that the asymptotic probability of selecting the correct model increases as a function of $\frac{d}{N}$. [41] studies consistency in the classification setting, and finds that K-fold CV is consistent so long as $\frac{N_v}{{N_t}^2} \to \infty$ as $N \to \infty$, where N_v is the number of validation-set points, and N_t is the number of training-set points.

These papers show that for asymptotic consistency, the number of observations in the validation sets must be much larger than what is typically used for model assessment. Some bias in terms of smaller training sets is traded off for larger validation sets to better account for the variation in the model assessments [43]. In general, leaving more observations out for validation improves model selection consistency.

Simulation-Based Results

The theoretical results described above generally hold true in simulation studies as well. [7] compares the model-selection performances of LOOCV and 10-fold CV, finding that 10-fold CV

[1] For more on selection for estimation, see [3] for an excellent (if somewhat technical) survey.

[2] CV with $\binom{d}{N}$ folds, where each of the $\binom{d}{N}$ possible sets of d observations is used as a validation set in one fold.

outperforms LOOCV in terms of selecting models of the right dimension, though both methods significantly underestimate the true errors of the final model they produce. [42] compares 2-fold, 5-fold, 10-fold, and LOO CV on linear models on small samples ($N = 20$). Interestingly, it finds that 2-fold CV performs worse than the other methods; asymptotic results do not entirely hold in the small-sample case.[3] [35] compares LOOCV with other CV methods (not covered in this paper), and finds that all methods have negligible probabilities of selecting underparameterized models. LOOCV also performs more poorly than the other methods that leave out more observations.

Notably, the literature for both the theoretical and simulation-based results discussed here is somewhat dated, and the simulation-based results are run with sample sizes far smaller than would be expected in modern applications. Unfortunately, more recent literature on the theoretical and empirical performances of CV is sparse.

Before concluding this section, one interesting recent result on the use of CV for model selection should be highlighted. In [21], Jung draws on the ideas from Shao's papers (e.g., [35]) to propose *multiple predicting cross-validation* (MPCV) as alternative to K-fold CV for model selection. Like K-fold CV, MPCV first randomly partitions the given data into K folds. For each iteration, though, rather than using $K - 1$ folds for training and one fold for validation, MPCV uses one fold for training and the other $K - 1$ for validation. The $K - 1$ error estimates for each observation are averaged in the final error estimate. Jung proves the consistency of MPCV (for linear models), though he notes that it should only be used when one fold contains enough data to provide a "reasonable fit" for the given model. He also shows how MPCV outperforms K-fold CV at model selection for a variety of both simulated and real datasets, since it has much less of a tendency to select overfit models. We explore Jung's idea of "flipping" training and validation sets to achieve better model selection properties in Chapter 5.

4.5 Model Selection in a Spatial Context

The spatial CV methods discussed in Chapter 3 can be applied to model selection. In theory, the bias-variance considerations discussed in Section 4.3 still apply: while low bias is desirable, the best methods should be those with the lowest variance. The bias-variance discussions at the end of Section 3.3 therefore help us hypothesize that buffered grid CV with a coarse grid or SKCV with low K should outperform SLOO at model selection.

As was the case for nonspatial, resampling-based methods for model selection, though, there is sparsity in the literature that specifically tackles the problem of model selection using resampling methods on spatial data.[4] Interestingly, a highly cited 2004 review of model selection methods for ecology does not mention any nonparametric resampling methods [20]. It seems that while spatial resampling for model assessment is growing in popularity (see e.g., the papers in Section 3.1), many researchers still prefer to use model-selection methods that are more theoretically supported (albeit under assumptions that are not necessarily satisfied by the data) and less computationally intensive. In general, there appears to be significant opportunity to study how the spatial model assessment methods described in Chapter 3 perform on the problem of model selection with spatial data.

[3] The relatively large variance of 2-fold CV (see Footnote 7 in Chapter 2) may be a factor here as well.

[4] There is, however, a small body of research on CV model selection for dependent time-series data, which is similar to one-dimensional spatial data, in some sense. For reference, the reader can look to [9], [10], [11], [18], [28], and [32]).

Chapter 5

Simulations

Chapters 3 and 4 gave heuristic justifications for why spatial blocking and buffering should improve CV model assessment and selection on spatial data. As noted then, few studies have compared such techniques to other methods. This chapter takes some small steps toward addressing this gap in the literature. We compare spatial versus nonspatial resampling techniques on simulated spatial data, concluding with some preliminary results.

5.1 Simulation Studies

Our simulations aim to show that for spatially structured data, spatial resampling methods indeed outperform nonspatial methods. To do so, we adopt simplified versions of a simulation in [15] (Box 1), running two simulations: one for model assessment, and another for model selection.

Both simulation procedures begin as follows. First, we generate 100 datasets, each containing 500 points placed uniformly at random in a 100×100 grid. For each point, we randomly generate spatially correlated[1] variables `X1`, `X2`, and `X3` and spatially correlated noise `e_spatial`. Finally, we generate the response `val` as $f(\texttt{X1}, \texttt{X2}, \texttt{X3}) + \texttt{e\_spatial} + \texttt{e}$, where `e` is nonspatial random noise.

For each dataset $\mathcal{T}$, the true conditional error $Err_{\mathcal{T}}$ error of the given model is computed by fitting the model to $\mathcal{T}$ and computing its average root-mean-square error over all 99 other datasets. Intuitively, we treat the other 99 datasets as a large draw from the same population as $\mathcal{T}$, so performance on that large sample should be a good estimate of performance on the overall population. Expected Err is computed by taking the average of all 100 $Err_{\mathcal{T}}$ values.

Model Assessment Simulation

The model assessment simulation compares the true errors of linear models to the error estimates produced by training error, 16-fold[2] CV error, LOOCV error, 4×4 buffered grid CV error, and SLOO CV error (with buffer size 15[3]). The data-generating function used is:

$$f(x, y, z) = 2\sin(\pi x) + x + 2y + 4 * \mathbb{1}_{z>0}(z),$$

[1] Generated by converting the pairwise distance matrix into a covariance matrix for a multivariate normal. We use the distance decay function $f(x) = \sigma^2 * (1 - \frac{x}{x+10})$ to scale distances into appropriate covariance matrix values.

[2] We choose to use 16-fold CV rather than 10-fold CV (as suggested in Section 2.5) because its validation sets are more similar in size to those of 4×4 buffered grid CV, which is better for comparing the two methods.

[3] We choose a buffer size of 15 because it generally produces buffers similar in size to those of 4×4 buffered grid CV, which is better for comparing the two methods.

where $\mathbb{1}_{z>0}(z)$ is the indicator function that is 1 if $z > 0$ and 0 otherwise.[4]

For each simulated dataset, we compute the aforementioned five estimates of the out-of-sample error of the linear model m: `val` $\sim$ `X1` $+$ `X2` $+$ `X3`. Table 5.1 shows the average values of the five error estimates over the 100 simulated datasets, along with the average true error Err. Training error is the most optimistic assessment method, as expected. Also, we generally notice

Assessment Method	Average Error Estimate
Training Error	1.606
LOOCV	1.621
16-fold CV	1.621
SLOO CV	1.672
Buffered Grid CV	1.710
True Error	1.775

Table 5.1: Average error estimates from five assessment methods, and true error

that the nonspatial CV methods are more optimistic than the spatial CV methods.

To better visualize how these error estimates compare to true error, we examine a density plot of the differences between the 100 error estimates and $Err_{\mathcal{T}}$ values, for each method (Figure 5.1). The vertical line at 0 represents an unbiased estimate of true error. We note that all five

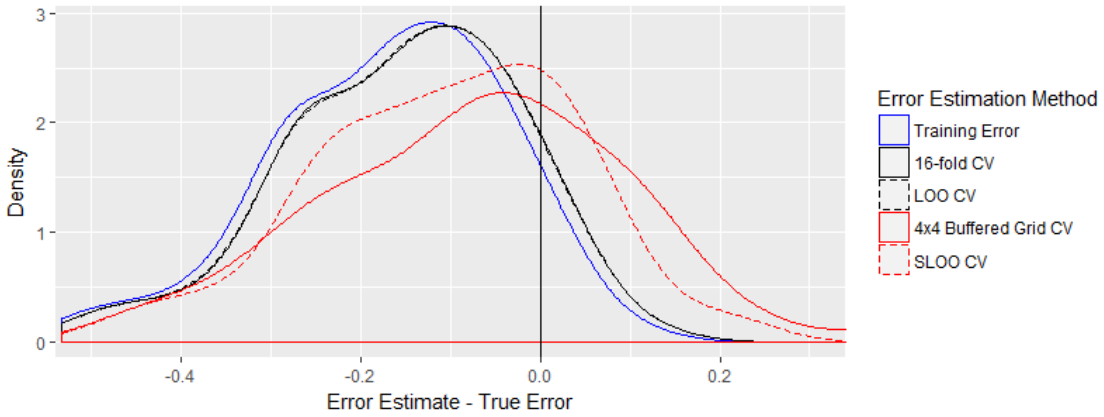

Figure 5.1: Differences between error assessments and true error

assessment methods are overoptimistic: the peaks of their distributions lie to the left of 0. Training error, in blue, is the most overoptimistic. Importantly, however, we note that the spatial error estimates in red tend to be less optimistic than the nonspatial estimates in black. In other words, spatial resampling methods tend to produce error estimates closer to the true error.

Table 5.1 and Figure 5.1 suggest that, at least for the simple simulation presented here, spatial resampling methods indeed outperform nonspatial ones at model assessment. We confirm not only that spatial methods are less overoptimistic on average, but also that their estimates actually tend to be closer to true error than the estimates made by nonspatial methods.

[4] Admittedly, this function is chosen in a relatively arbitrary manner. The goal was to make the signal roughly linear, but complex enough that the learning curve for a linear model will not achieve a minimum with too few training observations (which would eliminate the pessimistic bias that K-fold CV suffers for low values of K).

Model Selection Simulation

The model selection simulation uses "flipped" 16-fold CV and "flipped" buffered grid CV[5] in addition to the five assessment methods from the assessment simulation. The data-generating function is:

$$f(x, y, z) = 2\sin(\pi x) + x + 2y.$$

For each dataset we compute the seven error estimates for four models:

1. m_1: `val` $\sim$ `X1` (underspecified model 1)
2. m_2: `val` $\sim$ `X2` (underspecified model 2)
3. m_3: `val` $\sim$ `X1` + `X2` (true model)
4. m_4: `val` $\sim$ `X1` + `X2` + `X3` (overspecified model)

Each model-assessment method "selects" the model m_i that it estimates to have the least error.

For a baseline reference, we first show results for the model selection simulation performed without spatial autocorrelation, i.e. with `X1`, `X2`, `X2`, and the noise generated as independent random Normals. Table 5.2 shows the number of times each model was selected by each model-assessment method. Ideally, we want our selection methods to choose the true model m_3, in bold. We note that the corresponding spatial and nonspatial methods perform nearly the same,

Selection Method	Selected m_1	Selected m_2	**Selected $\mathbf{m_3}$**	Selected m_4
Training Error	0	0	**0**	100
16-Fold CV	0	0	**71**	29
Flipped 16-Fold CV	0	0	**75**	25
LOOCV	0	0	**85**	15
4×4 Buffered Grid CV	0	0	**73**	27
Flipped 4×4 Buffered Grid CV	0	0	**77**	23
SLOO CV	0	0	**85**	15

Table 5.2: Models selected by seven selection methods using nonspatial data

as expected for nonspatial data. Interestingly, the LOO CV methods perform the best at model selection, picking the correct model 85% of the time. Finally, the "flipped" CV methods appear to slightly outperform their standard counterparts, though not significantly.

Table 5.3 shows the results for the model selection simulation performed with spatial autocorrelation, as described above. We notice a few differences from Table 5.2. First, selection on spatial data appears to be a much more difficult task; the success rates of most of the CV selection methods drops from 70-80% to 40-60%. Importantly, however, we see that the spatial methods significantly outperform their nonspatial counterparts on spatial data. The nonspatial methods tend to select the overspecified model m_4. We hypothesize that this tendency is related to their lack of buffer regions. m_4 contains `X3`, a spatial variable with no defined relation to the spatial variable `val`. According to [26], though, even if `X3` has a spatial pattern independent of `val`, it will be more correlated with `val` than a nonspatial variable `X4` would be. Because the nonspatial methods do not use buffer regions, their training sets contain points near their validation sets. Thus, they are more likely to capture the spurious relationships between `X3` and `growth`, which would lead them to select m_4.

The spatial methods, unlike the nonspatial ones, select the correct model m_3 over 50% of the

[5] "Flipped" methods apply Jung's idea (see Section 4.4), where training sets are instead used for validation and v.v.

Selection Method	Selected m_1	Selected m_2	**Selected $\mathbf{m_3}$**	Selected m_4
Training Error	0	0	**0**	100
16-Fold CV	0	2	**38**	62
Flipped 16-Fold CV	0	1	**44**	56
LOOCV	0	13	**47**	53
4×4 Buffered Grid CV	0	6	**68**	26
Flipped 4×4 Buffered Grid CV	1	12	**48**	39
SLOO CV	0	6	**77**	23

Table 5.3: Models selected by six selection methods using spatial data

time.[6] Similar to the nonspatial methods, though, they still have a tendency to select overspecified models. Also, in both Tables 5.2 and 5.3, LOOCV and SLOO CV appear to have the best performance for nonspatial and spatial methods, respectively.

Overall, the selection simulations show that the spatial CV methods discussed in this paper are better at model selection than the nonspatial CV methods. The simulations also show that "flipped" CV methods may provide slightly strongly selection performance, though such a conclusion requires further study.

Finally, the selection simulations show that LOO CV methods seem to have the best selection performance, which generally differs from the conclusions in the literature discussed in Chapter 4 which stated that leaving out more observations in each CV fold led to better selection. As noted in Section 4.4, though, the cited studies are relatively dated. In particular, they use sample sizes of less than 100, far smaller than the 500 observations used in the selection simulations here or the larger datasets that we generally work with in the modern world. Also, the cited studies generally compared LOOCV to CV methods other than K-fold CV (e.g., leave-d-out CV). We tentatively conclude that K-fold CV with $K \ll N$ is not a particularly good method for model selection, even though it is the most popular CV method for model assessment.

[6] With the slight exception of flipped buffered grid CV, which selects the underspecified model m_2 with surprising frequency. Note, though, that the signal for `X1` is relatively weak and noisy compared to the signal for `X2`; as such, selecting the underspecified model m_2 is wrong but not wholly unreasonable.

Chapter 6

Conclusion

Spatial data present unique challenges for CV model assessment and selection. In this paper, we explored these challenges and studied potential solutions using both theory and simulations.

Most importantly, our simulations showed that the spatial CV methods introduced in Chapter 3 indeed outperformed nonspatial CV methods at both assessment and selection on spatial data. Spatial CV methods tended to be less overoptimistic about model performance and were also better at selecting the true model instead of the overspecified model.

The simulations also showed that LOO CV methods seemed to have the best selection properties. This result generally contradicts the literature, which suggests that using larger validation sets in resampling methods should improve model selection. We note, however, that the literature is relatively dated, using small sample sizes and focusing less on K-fold CV in particular. Further research will be needed before more conclusive results can be reached.

In any case, the work in this paper leaves much room for further exploration. For one, a lot can be done by studying the effects of simply adjusting the parameters of our simulations. Seeing how spatial and nonspatial CV methods perform for different data-generating functions, supervised learning methods, signal-to-noise ratios, or levels of spatial autocorrelation, e.g., can help us develop a better understanding of how these methods might perform on real data. The background and simulation framework in this paper[1] can hopefully provide a solid jumping-off point for future work in this area.

Another interesting avenue of study might involve extending different CV methods to spatial contexts. This paper only studied K-fold CV, but there are a huge range of CV methods in the literature (e.g., see [3]). Some of these methods have been shown to have better model assessment and/or selection properties than K-fold CV (e.g., see [35] for an example for i.i.d. data, or [32] for a particularly relevant paper using time-series data). Extensions of these methods to spatial data could yield powerful new assessment and selection methods.

As statistical modeling grows in popularity, it becomes increasingly important to better understand model assessment and selection. In many situations, such as the spatial contexts explored in this thesis, data do not follow naive assumptions of independence. Failing to account for dependencies can lead to a variety of consequences.

This thesis lays a foundation for understanding and exploring assessment and selection for spatially dependent data, though there is clearly much more work to be done. Dependent data are ubiquitous, and it is often difficult to come up with methods to deal with them in model assessment and selection. A better understanding of how to deal with spatial dependencies in these situations may be a key step in the right direction.

[1] Much of the publicly available code used for this project has been put on https://github.com/jche/spatial-cv.

Bibliography

[1] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

[2] Miguel B Araújo, Richard G Pearson, Wilfried Thuiller, and Markus Erhard. Validation of species–climate impact models under climate change. *Global Change Biology*, 11(9):1504–1513, 2005.

[3] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

[4] Volker Bahn and Brian J McGill. Testing the predictive performance of distribution models. *Oikos*, 122(3):321–331, 2013.

[5] Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of K-fold cross-validation. *Journal of Machine Learning Research*, 5(Sep):1089–1105, 2004.

[6] Simone Borra and Agostino Di Ciaccio. Measuring the prediction error. a comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis*, 54(12):2976–2989, 2010.

[7] Leo Breiman and Philip Spector. Submodel selection and evaluation in regression. the x-random case. *International Statistical Review*, pages 291–319, 1992.

[8] Alexander Brenning. Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The R package sperrorest. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 5372–5375. IEEE, 2012.

[9] Prabir Burman, Edmond Chow, and Deborah Nolan. A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358, 1994.

[10] Patrick S Carmack, William R Schucany, Jeffrey S Spence, Richard F Gunst, Qihua Lin, and Robert W Haley. Far casting cross-validation. *Journal of Computational and Graphical Statistics*, 18(4):879–893, 2009.

[11] C-K Chu and James Stephen Marron. Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics*, pages 1906–1918, 1991.

[12] Christopher Daly. Guidelines for assessing the suitability of spatial climate data sets. *International Journal of Climatology*, 26(6):707–721, 2006.

[13] Bradley Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.

[14] Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference*, volume 5. Cambridge University Press, 2016.

[15] Roberts et al. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 2017.

[16] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, 2001.

[17] JN Goetz, Alexander Brenning, H Petschko, and P Leopold. Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling. *Computers & Geosciences*, 81:1–11, 2015.

[18] Jeffrey D Hart and Seongbaek Yi. One-sided cross-validation. *Journal of the American Statistical Association*, 93(442):620–631, 1998.

[19] Robert J Hijmans. Cross-validation of species distribution models: removing spatial sorting bias and calibration with a null model. *Ecology*, 93(3):679–688, 2012.

[20] Jerald B Johnson and Kristian S Omland. Model selection in ecology and evolution. *Trends in Ecology & Evolution*, 19(2):101–108, 2004.

[21] Yoonsuh Jung. Multiple predicting K-fold cross-validation for model selection. *Journal of Nonparametric Statistics*, pages 1–19, 2017.

[22] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745, 2009.

[23] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145. Stanford, CA, 1995.

[24] Kévin Le Rest, David Pinaud, Pascal Monestiez, Joël Chadoeuf, and Vincent Bretagnolle. Spatial leave-one-out cross-validation for variable selection in the presence of spatial autocorrelation. *Global Ecology and Biogeography*, 23(7):811–820, 2014.

[25] Pierre Legendre. Spatial autocorrelation: trouble or new paradigm? *Ecology*, 74(6):1659–1673, 1993.

[26] Jack J Lennon. Red-shifts and red herrings in geographical ecology. *Ecography*, 23(1):101–113, 2000.

[27] Colin L Mallows. Some comments on C_p. *Technometrics*, 15(4):661–675, 1973.

[28] James Stephen Marron. Partitioned cross-validation. *Econometric Reviews*, 6(2):271–283, 1987.

[29] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[30] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. In *Advances in Neural Information Processing Systems*, pages 307–313, 2000.

[31] Jonne Pohjankukka, Tapio Pahikkala, Paavo Nevalainen, and Jukka Heikkonen. Estimating the prediction performance of spatial models via spatial K-fold cross validation. *International Journal of Geographical Information Science*, 31(10):2001–2019, 2017.

[32] Jeff Racine. Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics*, 99(1):39–61, 2000.

[33] Gideon Schwarz et al. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[34] PAGE Segurado, Miguel B Araujo, and WE Kunin. Consequences of spatial autocorrelation for niche-based models. *Journal of Applied Ecology*, 43(3):433–444, 2006.

[35] Jun Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88(422):486–494, 1993.

[36] RJ Telford and HJB Birks. The secret assumption of transfer functions: problems with spatial autocorrelation in evaluating model performance. *Quaternary Science Reviews*, 24(20-21):2173–2179, 2005.

[37] RJ Telford and HJB Birks. Evaluation of transfer functions in spatially structured environments. *Quaternary Science Reviews*, 28(13-14):1309–1316, 2009.

[38] Waldo R Tobler. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46(1):234–240, 1970.

[39] Sonja Wehberg and Martin Schumacher. A comparison of nonparametric error rate estimation methods in classification problems. *Biometrical Journal*, 46(1):35–47, 2004.

[40] Seth J Wenger and Julian D Olden. Assessing transferability of ecological models: an underappreciated aspect of statistical validation. *Methods in Ecology and Evolution*, 3(2):260–267, 2012.

[41] Yuhong Yang. Comparing learning methods for classification. *Statistica Sinica*, pages 635–657, 2006.

[42] Ping Zhang. Model selection via multifold cross validation. *The Annals of Statistics*, pages 299–313, 1993.

[43] Yongli Zhang and Yuhong Yang. Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, 187(1):95–112, 2015.