Bulletin of the Seismological Society of America, Vol. 108, No. 4, pp. 2248–2264, August 2018, doi: 10.1785/0120180056

Prioritizing Ground-Motion Validation Metrics Using Semisupervised and Supervised Learning

by Naeem Khoshnevis and Ricardo Taborda

Abstract It has become common practice to validate ground-motion simulations based on a variety of time and frequency metrics scaled to quantify the level of agreement between synthetics and data or other reference solutions. There is, however, no agreement about the importance or weight that it ought to be given to each metric. This leads to their selection often being subjective, either based on intended applications or personal preferences. As a consequence, it is difficult for simulators to identify what modeling improvements are needed, which would be easier if they could focus on a reduced number of metrics. We present an analysis that looks into 11 ground-motion validation metrics using semisupervised and supervised machine learning techniques. These techniques help label and classify goodness-of-fit results with the objective of prioritizing and narrowing the choice of these metrics. In particular, we use a validation dataset of a series of physics-based ground-motion simulations done for the 2008 M_w 5.4 Chino Hills, California, earthquake. We study the relationships that exist between 11 metrics and carry out a process where these metrics are understood as part of a multidimensional space. We use a constrained k-means method and conduct a subspace clustering analysis to address the implicit high-dimensional effects. This allows us to label the data in our dataset into four validation categories (poor, fair, good, and excellent) following previous studies. We then develop a family of decision trees using the C5.0 algorithm, from which we select a few trees that help narrow the number of metrics leading to a validation prediction into the four referenced categories. These decision trees can be understood as rapid predictors of the quality of a simulation, or as data-informed classifiers that can help prioritize validation metrics. Our analysis, although limited to the particular dataset used here, indicates that among the 11 metrics considered, the acceleration response spectra and total energy of velocity are the most dominant ones, followed by the peak ground response in terms of acceleration and velocity.

Introduction

Verification and validation of ground-motion synthetics have received increasing attention in recent years due to advances in deterministic and nondeterministic physics-based earthquake ground-motion simulation, as well as a growing interest in the use of synthetic seismograms for engineering applications. Validation entails the comparison of simulations with observations, whereas verification involves the comparison of simulations with exact or alternative solutions (see Bielak *et al.*, 2010; Taborda and Bielak, 2013, and references therein). Various methods or schemes have been proposed to evaluate, through direct signal-to-signal quantitative comparisons or overall statistical analyses, the similarity between simulation synthetics and recorded data, or with respect to other reference solutions (Anderson, 2004; Kristeková *et al.*, 2006, 2009; Olsen and Mayhew, 2010; Burks and Baker, 2014; Rezaeian *et al.*, 2015). Some of these methods are better suited for verification purposes because they compare the signals at the waveforms level (Kristeková *et al.*, 2006, 2009), whereas others are better suited for validation. Burks and Baker (2014) and Rezaeian *et al.* (2015), for instance, are used for validation in the context of engineering applications, that is, at the level of the dynamic response of buildings. Anderson (2004) and Olsen and Mayhew (2010), on the other hand, are used for validation in the context of the overall characteristics of seismograms when comparing synthetics with data, in both time and frequency. These methods are preferred for validating high-frequency or broadband simulations, where applications may be undefined or the expectation of matching seismograms at the waveforms level is less relevant (unless dealing with low frequencies, $f_{\max} \leq 1$ Hz), and because they offer simple schemes to quantify the goodness-of-fit (GOF) of the simulation results.

Out of the various validation methods just described, Anderson (2004) is perhaps the most widely used today (e.g., Bielak *et al.*, 2010; Chaljub *et al.*, 2010; Guidotti *et al.*, 2011; Maufroy *et al.*, 2015). In essence, this method assesses the similarity of two signals based on the average score of 10 metrics. These metrics measure the cumulative strength of the signals (in terms of Arias intensity and absolute energy), the comparative evolution of the signals in time (through normalized Arias and energy integrals), their peak values (in acceleration, velocity, and displacement), frequency content (in terms of Fourier and response spectra), and their time synchronization (through cross correlation). We, in particular, have consistently relied on a modified version of Anderson's method for the validation of a series of simulations of the 2008 M_w 5.4 Chino Hills earthquake (Taborda and Bielak, 2013, 2014), and for the evaluation of velocity models in southern California through the validation of a large suite of moderate-magnitude earthquake simulations (Taborda *et al.*, 2016).

The strength of the method proposed by Anderson (2004) is that its metrics convey physical meaning to both seismologists and engineers. However, it has been pointed out that some of these metrics are redundant, or that other relevant metrics need to be added. Taborda and Bielak (2013), for instance, included—explicitly—the strong-motion duration (Trifunac and Brady, 1975) as an additional metric, and averaged the Arias-intensity-related and energy-integral-related scores to avoid duplication. In the same spirit, Maufroy *et al.* (2015) reduced the number of metrics, limiting them to only those with comparable units. Unfortunately, none of these alternatives addresses the underlying questions regarding what are the most important parameters that ought to be taken into account when validating ground-motion simulations for a specific application and what level of priority should these parameters be given.

Lack of consensus about how to answer these questions makes the choice of validation methods and the selection of the comparison metrics a subjective one, mostly based on personal preferences and—arguably—on the application intended for the simulation and/or the validation itself. At the same time, with the current multiplicity of metrics, to give to all metrics the same weight in a validation analysis, makes it difficult for simulators to identify the sort of changes needed in their models that could lead to better ground-motion predictions. We think that this situation can be corrected if we can offer data-informed arguments that can help simulators justify focusing on a reduced number of alternative metrics.

To that end, we study the relationships that exist between 11 different metrics—those proposed by Anderson (2004), plus the strong-motion duration—with the objective of offering a prioritized reduced number of metrics that can help predict validation results. We initially treat these metrics as independent variables defining a multidimensional space, and analyze them using machine learning techniques. In particular, we use semisupervised and supervised methods. In machine learning, whether a method is considered unsupervised, semisupervised, or supervised, depends on the level of constraints applied to the process or prior level of classification given to the data. Here, we first use a semisupervised method called constrained k-means to conduct a subspace clustering analysis to label data samples in a given validation dataset (e.g., MacQueen, 1967; Wagstaff *et al.*, 2001). This dataset corresponds to the validation results from a series of simulations done for the 2008 M_w 5.4 Chino Hills, California, earthquake (Taborda and Bielak, 2014). Here, the clustering done by the constrained k-means method allows us to label the data using for validation categories: poor, fair, good, and excellent. These categories were defined by Anderson (2004) and have been adopted—despite some differences—by other GOF methods used in verification and validation (e.g., Kristeková *et al.*, 2009; Olsen and Mayhew, 2010). Upon labeling the data samples in our dataset into these four categories, we use a supervised learning method as implemented in the C5.0 algorithm to obtain a family of decision trees (Quinlan, 1993, 1996). We then select a few of these trees to gain insight about and prioritize the metrics involved in the validation process. The decision trees narrow the number of metrics offering a prediction algorithm into the aforementioned validation categories.

In summary, although specific to the chosen dataset (from the given set of simulations for a single event), our results indicate that among the 11 metrics considered in the analysis, the acceleration response spectra and total energy of velocity are the most dominant ones, followed by the peak ground response in terms of acceleration and velocity. We test our prediction model and offer a discussion about its implications and potential use in future validation efforts.

Validation Metrics

There are various methods and algorithms available to evaluate the similarity between two or more seismograms, both through direct signal-to-signal quantitative comparisons or overall statistical analyses (e.g., Anderson, 2004; Kristeková *et al.*, 2006, 2009; Olsen and Mayhew, 2010; Burks and Baker, 2014; Rezaeian *et al.*, 2015). In earthquake ground-motion simulation, they are used primarily for verification with respect to benchmark or analytical solutions, and for validation with respect to data (i.e., ground-motion records). Here, we focus on the list of metrics proposed by Anderson (2004), with an additional metric for duration, as suggested by others (Olsen and Mayhew, 2010; Maufroy *et al.*, 2015) and as implemented in Taborda and Bielak (2013). These metrics are listed in Table 1.

Following Anderson (2004), when applied to a pair of signals, each one of these metrics yields a GOF score ranging from 0 to 10, where a value of 10 corresponds to two signals having identical characteristics. This scoring scale varies according to the following exponential function:

$$S(p_1, p_2) = 10 \exp\left[-\left(\frac{p_1 - p_2}{\min(p_1, p_2)}\right)^2\right], \qquad (1)$$

Table 1
Validation Metrics

Code	Metric
C1	Arias intensity integral
C2	Energy integral
C3	Arias intensity
C4	Total energy
C5	Peak acceleration
C6	Peak velocity
C7	Peak displacement
C8	Response spectrum
C9	Fourier amplitude spectrum
C10	Cross correlation
C11	Strong phase duration

in which S is the GOF score that results from comparing values p_1 and p_2 from signals 1 and 2, respectively, for each one of the different metrics in Table 1. In the case of C8 and C9, where the values p_1 and p_2 are function of the period (T) and frequency (f), respectively, S is computed for all values of T and f and averaged to produce a single mean score. Anderson also provided guidelines for the process to be applied to the original signals as well as to those resulting from a sequential set of band-pass filters covering the whole frequency range of interest. In his method, this frequency-domain decomposition should have passbands defined following a logarithmic distribution to give more weight to the low frequencies, and the GOF scores of all sub-bands and the broadband be averaged into a single GOF final score. However, in this study, to avoid the additional parameters that would result from this approach (i.e., sub-bands width and filter characteristics, Khoshnevis and Taborda, 2015), we use only the broadband ($f = 0$–4 Hz) results.

One important aspect of Anderson (2004) is that the results of the GOF scores were calibrated by comparing horizontal components of recorded seismograms and other simulations using the first 10 metrics (C1–C10) in Table 1 to find out how the scores were distributed throughout the scoring scale. Anderson concluded that for a typical distribution of scores, the GOF values could be classified into four validation categories: poor (for scores from 0 to 4), fair (4 to 6), good (6 to 8), and excellent (for scores from 8 to 10). Although these categories are arguably subjective, over the years they have been adopted—despite some differences in the ranges—by other GOF methods employed in verification and validation (e.g., Kristeková *et al.*, 2009; Olsen and Mayhew, 2010). As a result, Anderson's method has been used as a reference baseline for various validation studies (e.g., Bielak *et al.*, 2010; Chaljub *et al.*, 2010; Guidotti *et al.*, 2011; Maufroy *et al.*, 2015).

Here, we focus our analysis on the 11 metrics included in Table 1 and investigate the relationships that exist between them to prioritize a reduced number of metrics that can help predict the outcome category one would use to label the results of a given simulation.

Study Dataset

We select the simulation results and validation analysis done by Taborda and Bielak (2014) as our validation dataset. In that study, the authors carried out deterministic simulations for the 2008 M_w 5.4 Chino Hills, California, earthquake using a finite-element approach. The simulations, done for a kinematic finite-fault model of the earthquake, were computed for a maximum frequency $f_{\max} = 4$ Hz and a minimum shear-wave velocity $V_{S_{\min}} = 200$ m/s. In total, Taborda and Bielak (2014) did three simulations, each for a different velocity model (CVM-S4, CVM-H, CVM-H+GTL; see Small *et al.*, 2017). The modeling domain covered an area of 180 km × 135 km that included all the major sedimentary basins and other relevant geologic structures in the greater Los Angeles region. Their validation analysis consisted of comparisons with data recorded during the event at 336 ground-motion monitoring stations, for the three components of motion (east–west, north–south, and up–down). The simulation domain and the stations used in that study are shown in Figure 1, and a sample of the validation results obtained by the authors using Anderson's approach is shown in Figure 2. This figure, in particular, shows the spatial distribution of GOF scores (interpolated from the values at each station) for the comparison of the broadband synthetics and data, for the three velocity models used by Taborda and Bielak (2014). In each case, the results are the average of the GOF results for the three components of motion and the scores for the 11 metrics from Table 1. A subsequent study by Taborda *et al.* (2016) using multiple events found that the model CVM-S (v.4.26.M01; see Small *et al.*, 2017) to be the model that most consistently led to better simulation results. Here, however, we use the GOF scores obtained by Taborda and Bielak (2014) independently of the velocity models and/or the components of motion.

We refer to the set of 11 scores associated with a pair of signals from the work done by Taborda and Bielak (2014) as one of the data samples that compose our dataset. Although at times we will make distinctions between the velocity models and the components of motion for visualization purposes, the clustering analysis to be described in the Data Analysis Method section was done using all the data samples in the validation dataset. The motivation behind this choice was that the dataset, as a collection of GOF values, was independent of the simulations and serves here as a generic set of data samples for the purpose of identifying the correlations that exist between the different metrics in Table 1. As such, given the simulations for each velocity model (3), the components of motion (3), and the number of stations used in the validation (336), gave us a large enough dataset, with 3024 data samples. Figure 3 illustrates the idea of a lack of dependence on the simulations by comparing the statistical distribution of the GOF scores of the simulations organized by velocity models and components. It is clear that although there are some differences between them, these are negligible. In other words, the statistical distribution of the data for each metric is about the

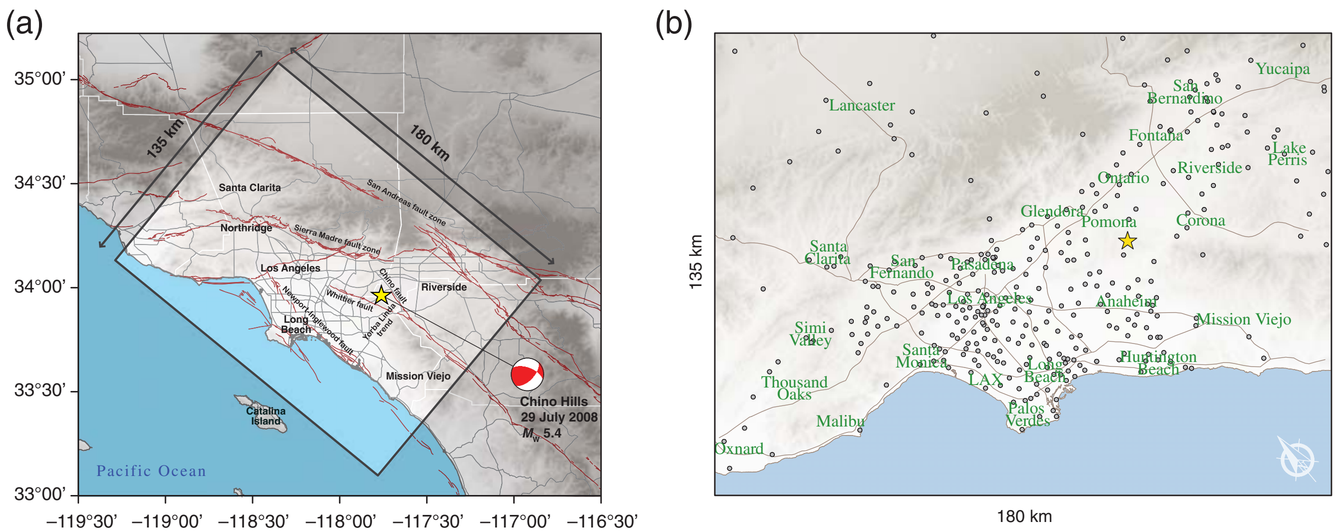

Figure 1. (a) Region of interest used by Taborda and Bielak (2014) for the simulations of the 2008 M_w 5.4 Chino Hills earthquake, including the epicenter, focal mechanism, and major quaternary faults. In the background, the main roads and topography are shown for visual reference. (b) Distribution of the 336 stations (gray dots) used by Taborda and Bielak (2014) for the validation analysis of their simulations within the modeling domain shown in (a). Roads, city names, and the hillshade topography are also shown here in the background for reference. The color version of this figure is available only in the electronic edition.

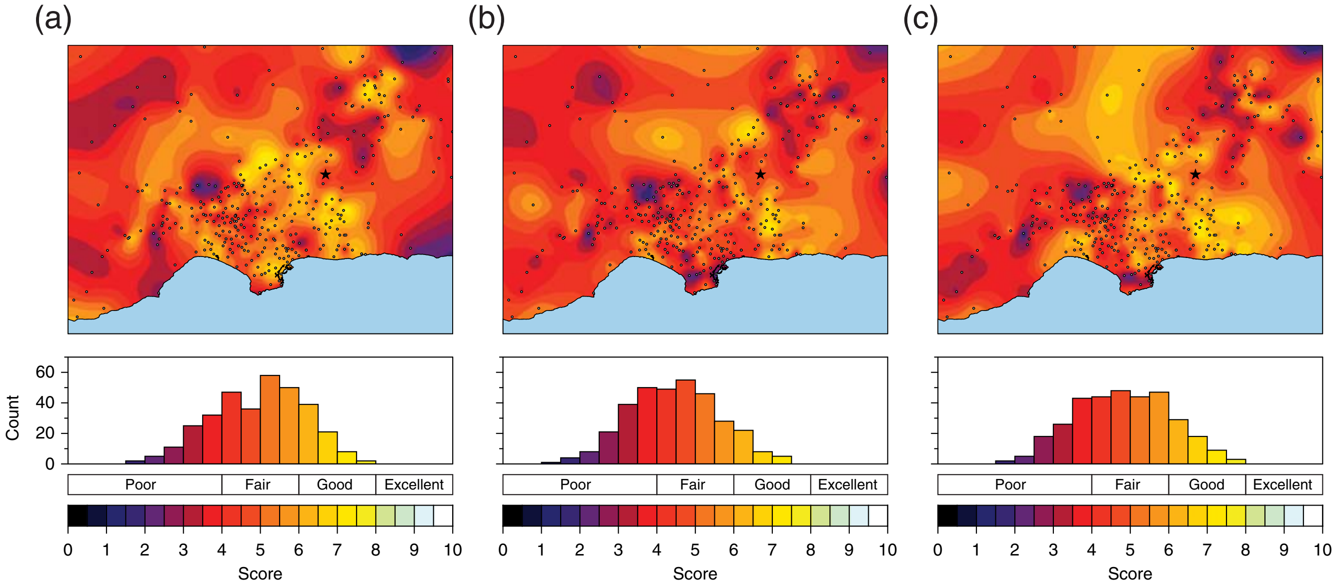

Figure 2. Validation results obtained by Taborda and Bielak (2014) in the form of goodness-of-fit (GOF) values across the region of interest obtained from comparisons between data recorded at ground-motion monitoring stations and simulations for three different velocity models: (a) CVM-S, (b) CVM-H, and (c) CVM-H+GTL. The distribution of the results at the bottom shows the count of stations on the GOF scale defined by Anderson (2004) and the ranges of the validation categories: poor, fair, good, and excellent. The color version of this figure is available only in the electronic edition.

same independently of model or component. Consequently, and to use a dataset as large as possible, we thought it acceptable to combine all the data samples into a single dataset.

Data Analysis Method

We are interested in proposing an algorithm with a reduced and prioritized number of validation metrics based on previously acquired collection of validation data samples (i.e., our dataset). A common method to do this is to identify rules with disjunctive characteristics, in the form of decision trees, that lead to outcomes representative of the overall validation analysis. In our case, we define such outcome in terms of four validation categories or classes representative of the quality of the validation, namely, poor (P), fair (F), good (G), and excellent (E). The development of such decision trees requires a proper classification of the data,

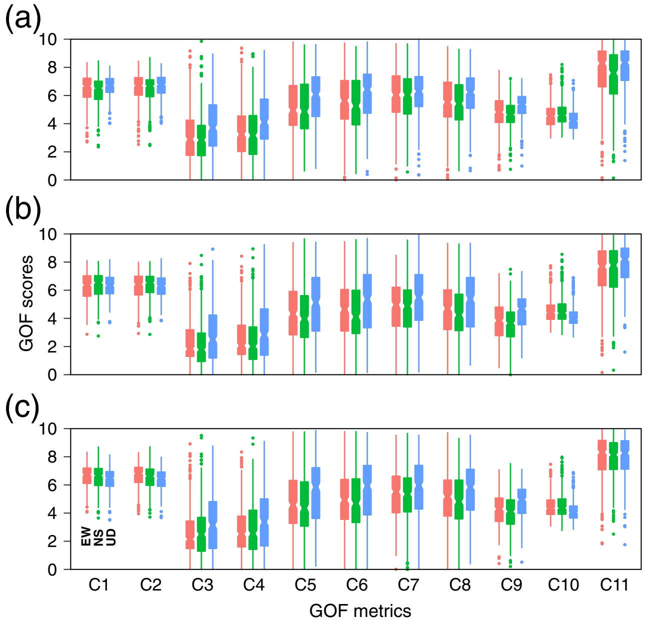

Figure 3. Statistical distribution of the GOF dataset obtained from Taborda and Bielak (2014) shown in the form of box plots for each metric (C1–C11, see Table 1), and components of motion (north–south [NS], east–west [EW], and up–down [UD]), for the simulations done with velocity models: (a) CVM-S, (b) CVM-H, and (c) CVM-H+GTL. In each case, the boxes represent the interquartile range (IQR = Q3 − Q1), the medians are indicated by a notch in the boxes, and the vertical lines show the range of the data, with outliers (data less than Q1 − 1.5IQR and greater than Q3 + 1.5IQR) shown as scattered dots. The color version of this figure is available only in the electronic edition.

which can be done through a clustering process. This implies applying the labels P, F, G, and E to the data samples in our dataset. The Clustering section explains the data processing analysis we put in place to label the data samples in our dataset and subsequently obtain three alternative decision trees.

Clustering

The first step toward obtaining a decision tree is to label the data according to their attributes. The inherent attribute of our data are the GOF values, but because of the multiplicity of metrics and lack of clarity about their relationships, we need to label the data according to the validation categories. We do this by means of a clustering process.

Clustering is an unsupervised data-mining process used to group data in a multidimensional space based on their attributes (Fayyad, 1996). According to Jain *et al.* (1999), clustering can be classified in two categories: hierarchical and partitional. Technical details aside, the basic difference is that hierarchical algorithms create nested partitions, whereas partitional algorithms produce singular partitions.

There is no single clustering process that can be applied to every dataset (Hartigan, 1985; Jain and Dubes, 1988; Dy and Brodley, 2004). Consequently, one needs to make a choice. We use a partitional, distance-based method known as constrained k-means. The standard k-means method is an unsupervised process for partitioning an n-dimensional population into k clusters with a minimum within-cluster attributes variance (e.g., MacQueen, 1967). The constrained k-means method, on the other hand, is a semisupervised approach that extends the standard method by allowing the use of background information in the form of clustering restrictions—thus the upgrade to a semisupervised method.

Given a k number of clusters, where each cluster is identified by its center, the standard process starts by computing the distances of all other data points to the center of the clusters, and grouping them based on their proximity to the clusters' centers. Once this is done, the center of each cluster is updated based on the average attributes of its data points, and the process is repeated until the clusters become stable.

This process is sensitive to the initial selection of the number of clusters and their centers. To mitigate this, constrained k-means introduces two types of constraints: must-link and cannot-link (Wagstaff *et al.*, 2001). The must-link constraint specifies instances in which two data points must be linked, that is, be in the same cluster. The cannot-link constraint specifies instances in which data cannot be in the same cluster. This prevents the process from converging into a local minimum and defines constrained k-means as a semisupervised method. Figure 4 illustrates the differences between the standard and constrained k-means methods for a single clustering iteration on a small 2D dataset.

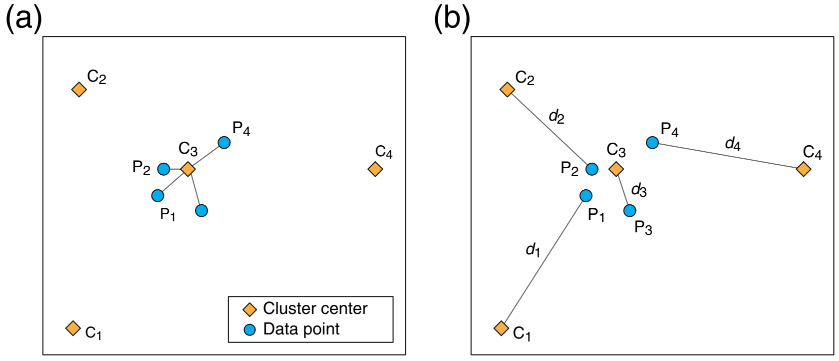

Figure 4. Representation of the (a) ordinary, and (b) constrained k-means approaches for four data points (P) and four cluster centers (C) in a 2D space, where, for the case of the constrained k-means approach, all the data points are set to be cannot-link points. The color version of this figure is available only in the electronic edition.

In our implementation, we limit the clustering to four validation categories: poor (P), fair (F), good (G), and excellent (E). The cluster centers are randomly selected at the start, but we apply constraints by adding a subset of four artificial data samples into our dataset such that they have cannot-link conditions. These artificial data samples are associated with the different validation categories and are such that they have GOF scores equal to 3, 5, 7, and 9, across all metrics.

The concepts of cluster centers and distances, as exemplified in Figure 4, imply the existence of a 2D space. Here, the dimensions are defined by the GOF metrics. We are therefore dealing with an 11D space. In clustering, the term “dimensions” is equivalent to the concept of the data features. Within this context, the GOF metrics are the features defining a multidimensional space. In such multidimensional space defined by the 11 features, the distances are obtained using the Euclidean expression:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{n} (x_{i,l} - x_{j,l})^2}, \qquad (2)$$

corresponding to the distance d between the data point x_i and the cluster center x_j in the n-dimensional domain, in which $x_{i,l}$ is the lth feature of the data point x_i, and $x_{j,l}$ is the lth feature of the cluster center x_j. It is, however, unpractical to expect the patterns defining the clusters to be observable across all features. Such high-dimensional issues are well known (see, for instance, Dy and Brodley, 2004; Parsons *et al.*, 2004), and can be tackled using subspaces. Therefore, instead of analyzing all 2^{11} possible subspaces, we focus only on subspaces with two, three, and four features. In total, we analyze 550 subspaces, 55 subspaces with two features, 165 with three features, and 330 with four features.

Unfortunately, not all the subspaces will have clearly distinguishable clusters (i.e., some will not satisfy the cannot-link constraints even after a large number of iterations). Such subspaces are discarded, and all others are used to label the data. In an ideal case, each subset of data samples corresponding to the comparisons—between synthetic and recorded signals at a particular location (or station) in a simulation—done using the 11 metrics will be labeled 550 times, and the final label is taken as the mode. For example, if after all the subspaces are accounted for, a station has labels $\{F, F, F, P, F, G, E, F, F\}$, then such a station will be given a final label F. Once all the samples in the dataset have been properly labeled, we proceed with the decision-tree analysis.

Decision Trees

Building decision trees is a supervised learning process used to approximate a target function as a sequence of disjunctive conditions designed to measure the effectiveness of a set of attributes to classify data and predict outcomes. The theory behind decision trees is well established (e.g., Quinlan, 1986, 1993; Mitchell, 1997). Decision trees are, however, nonunique. They depend on the dataset, the user parameters, and the algorithm employed to build them (Murthy, 1998). Therefore, before describing our choice of the decision-tree algorithm and parameters, we revise three aspects about the dataset: data classes, attributes equivalence, and dataset balance.

First, by data classes we refer to whether the classes of data are properly distinguishable or not, and to whether the data attributes contribute to those classes or not. In our case, data classes are handled by means of the clustering process. As we described before, at the end of the clustering process, all the data, samples in our dataset are labeled into one out of the four validation categories. In decision trees, however, we refer to these categories as classes. We are therefore interested in working with a labeled dataset or a properly classified dataset using the validation classes P, F, G, and E, which is something that has been guaranteed by the prior clustering process.

Second, by attributes equivalence, we refer to whether the data attributes are comparable with each other on equivalent terms or not. There are cases, either because of the method or because of the data, that the data attributes need to be standardized. In artificial neural networks, for instance, the data need to be normalized on a $[-1, 1]$ or $[0,1]$ scale. Normalization is also necessary when data attributes come in different unit scales or value ranges (e.g., Wu *et al.*, 2010). In our case, the concept of attributes in the context of the decision-tree analysis is the same as the concept of features in clustering or GOF metrics in validation. Because all our metrics are dimensionless and defined within the same numerical scale (varying from 0 to 10), normalization is not necessary and the process of building a decision tree is not susceptible to the attributes scales.

Last, there is the issue of balance. A balanced dataset is one that has about the same number of samples per class. Algorithms used for building decision trees tend to perform better with balanced datasets (e.g., Weiss and Provost, 2003; Branco *et al.*, 2016). Imbalanced datasets, on the other hand, are those with a significant disparity in the number of samples in each class. Imbalanced datasets can be improved by a resampling process. There are two basic resampling approaches: undersampling and oversampling (Branco *et al.*, 2016). Undersampling discards data from the subsets with larger number of samples. Oversampling replicates data until reaching appropriate number of samples. Both processes are done by randomly picking data to either be discarded or replicated. In our case, as we will see later, we apply oversampling to one of our dataset classes.

In general, oversampling increases the possibility of overfitting, which occurs when the training data lead the algorithm to produce a decision tree that predicts outcomes too close to or exactly the same as the training data. This is not desirable because such a tree would lead to inaccurate predictions for other unseen data. Overfitting can be avoided by applying heavy pruning methods. (Pruning, as the word implies, is the process of cutting branches off a tree. Heavy pruning methods limit the complexity of the tree by constraining the number of branches and/or depth of a tree.)

Upon completing the clustering and resampling processes, the next step is to subdivide the dataset in two, a training dataset (with 70% of the samples, picked randomly), and a testing dataset (with the remaining 30%). We first use the training dataset to build a large suite of potential decision trees, and then use the testing dataset to evaluate and pick the best possible decision tree(s). There are several algorithms for building decision trees (e.g., ID3, C4.5, C5.0, CART; see Breiman *et al.*, 1984; Quinlan, 1986, 1993, 1996). Here, we use the C5.0 algorithm as implemented by Kuhn *et al.* (2017). This algorithm is the latest update to the original ID3 and subsequent C4.5 algorithms (Quinlan, 1993, 1996). C5.0 is superior to its predecessors because it reduces the limitations for handling numerical attributes and missing data, and it has additional features such as the development of boosted models. Discussing the differences between these algorithms is out of the scope of the article.

In general, given a dataset, the process of building the tree consists of recursively identifying the attributes in the training data that are most likely to predict an outcome. The process is recursive because new branches and decision nodes are created based on the remaining data at every branch and level, until the tree reaches a certain depth or when other conditions are met. The effectiveness of an attribute A in classifying any subset S from the training data is measured through the information gain G as

$$G(S,A) = E(S) - \sum_{v \in A[a]} \frac{|S_v|}{|S|} E(S_v), \tag{3}$$

in which $A[a]$ is the discrete set of all possible values of attribute A, S_v is the subset of S for which attribute A has value v, and $E(\cdot)$ is the entropy function given by

$$E(S) = \sum_{i=1}^{c} (-p_i \log_2(p_i)), \tag{4}$$

in which p_i is the proportion of S belonging to class i, and c are the different values that a given target attribute can take.

Entropy measures the homogeneity of a set of data. The higher the entropy, the more even the distribution of the data across classes. Conversely, an extremely low value of entropy would mean most of the data fall within a particular class. With this in mind, the information gain G of an attribute A for the dataset S, or $G(S,A)$ in equation (3), is the expected reduction in entropy caused by partitioning the dataset according to the attribute A (Mitchell, 1997), which we consider to be discrete. For the selection of a threshold for a continuous attribute $A(a)$, please refer to Quinlan (1996).

In the particular case of the C5.0 algorithm, in addition to the general steps just described, the result of the process of building a decision tree depends on other logical and numerical parameters. We set the options to (a) perform feature selection or winnowing, (b) evaluate possible advanced splits of data, (c) use a confidence factor (CF), and (d) use a threshold for the number of samples that go in the tree leaves. The feature selection option is used by the algorithm to choose the most important attribute over others; the option of evaluating advanced splits prevents the use of a hard threshold during the classification by considering different probabilities in assigning data to different classes; the CF is used to control the level of pruning; and the threshold for the number of samples in a leaf is set to a minimum number $S_{\min}$ (also known as minCases) to limit the level of complexity of the tree. Additional details about these options are available in Quinlan (1993) and Kuhn *et al.* (2017). For this study, to obtain a variety of tree options, we build trees for varying values of CF and $S_{\min}$.

Each tree resulting from this process has different qualities. We are interested in selecting a tree that is highly effective, but with a low level of complexity. In other words, we seek a tree with a good ratio of accuracy for predicting the classification outcome of the data, while using a reduced number of attributes (GOF metrics) in only a few number of steps (as given by a small number of decision nodes or a tree with shallow depth). The number of metrics, the number of nodes, and the depth of a tree are directly seen from the topology of the tree, and are often proportional (shallow trees tend to use less nodes and thus less metrics). In general, smaller trees are preferable because they are easy to understand and often more accurate predictors (Quinlan, 1996).

The effectiveness of the tree, on the other hand, needs to be measured. To that end, we use the factor F_β proposed by van Rijsbergen (1979) as

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}, \tag{5}$$

in which P and R are the precision and recall factors, respectively, and β is a weighting factor between the two. P and R are defined as

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{and} \quad R = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{6}$$

respectively. Here, TP, FP, and FN are the number of samples in the testing dataset considered to be true positives, false positives, and false negatives, respectively.

These values are typically ordered in a confusion matrix. For the simplest case of two categories, a confusion matrix takes the form

$$\begin{array}{lll} & & \text{Prediction} \\ & & \begin{array}{cc}\text{Positive} & \text{Negative}\end{array} \\ \text{Actual Value} & \begin{array}{l}\text{Positive}\\ \text{Negative}\end{array} & \begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN}\end{bmatrix} \end{array},$$

in which TN is the number of true negative samples. Confusion matrices can be larger depending on the number of classes. In our case, the confusion matrices are size 4 × 4, to compare the actual number of data samples classified as P, F, G, and E with the number of samples predicted by the decision trees for each one of the same validation classes.

We compute F_β in equation (5) using $\beta = 1$ to give equal weight to P and R (McCarthy and Lehnert, 1995). This is done for all the trees obtained using all possible combinations of CF and $S_{\min}$ values. Then, we select trees with high levels of effectiveness as measured by F_1, commensurate to such trees having low complexities, that is, reduced number

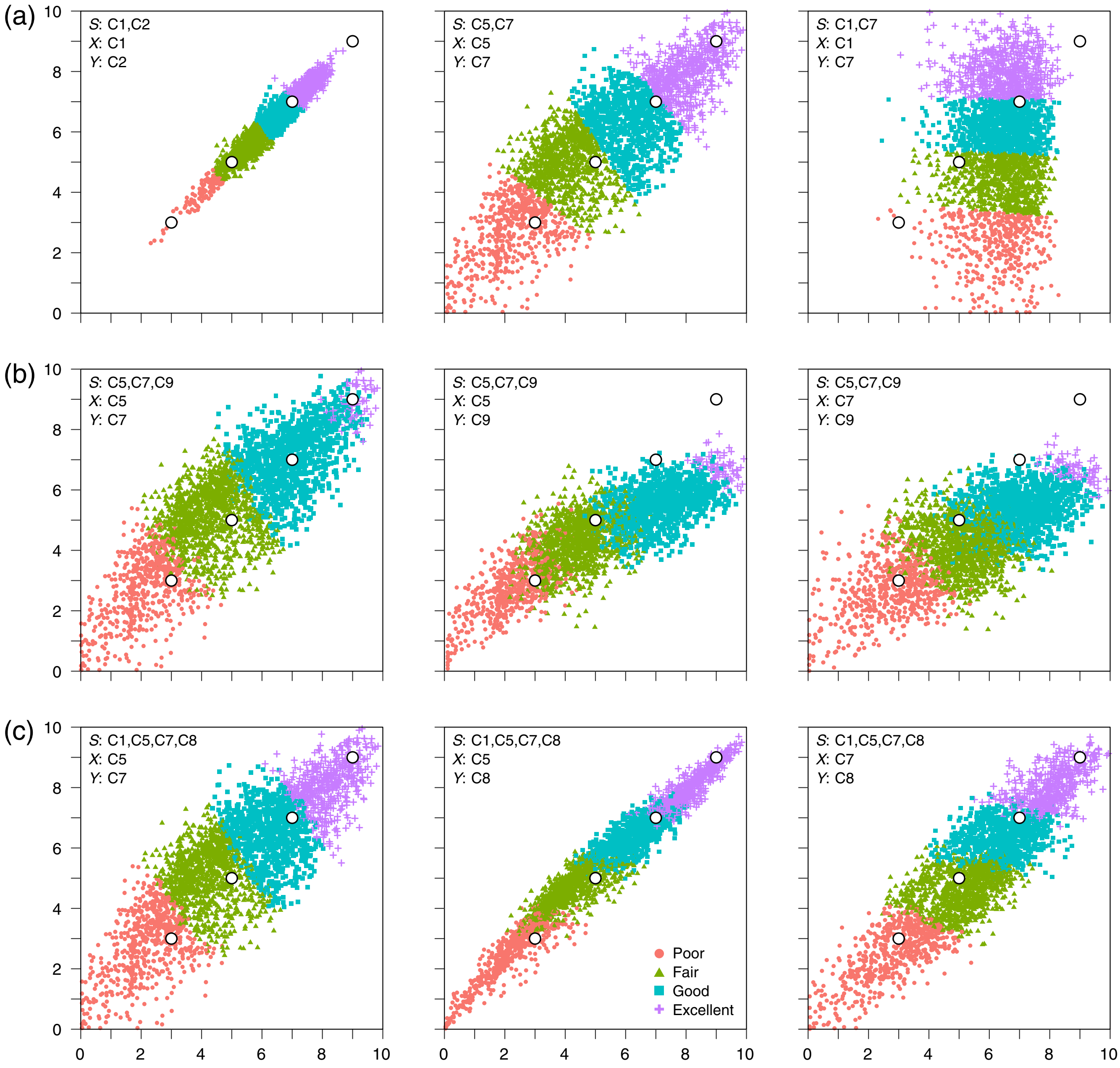

Figure 5. A sample of the results from the multidimensional clustering analysis showing the clusters from a 2D perspective into the relationships between different GOF metrics. In each case, the labels near the upper left corner indicate the features considered in the subspace analysis, and the features associated with the horizontal (x) and vertical (y) axes in the plot. (a) Two-, (b) three-, and (c) four-feature subspace analyses. In each case, the poor, fair, good, and excellent clusters are indicated with circle, triangle, square, and cross symbols, respectively. Empty circles indicate the location of the artificial cannot-link stations we introduced as background knowledge to the clustering process. The color version of this figure is available only in the electronic edition.

of metrics and shallow depths, which is precisely the goal set at the start. F_1 is obtained based on the testing dataset as opposed to the training dataset.

Results

As explained in the methodology, the first step is to carry out a clustering process on the dataset to properly label the data samples according to the classes to be used in the decision-tree algorithm. Figure 5 presents a sample of the results obtained after applying the constrained k-means method, including subspacing. Recall that there are a total of 550 subspaces. Figure 5 shows three examples for each one of the subspaces considered, that is, three for each one of the two-, three-, and four-feature subspace analyses. However, because it is not practical or possible to present three- or higher-dimensional plots, we display the results from a 2D perspective into the subspaces by picking two metrics at a

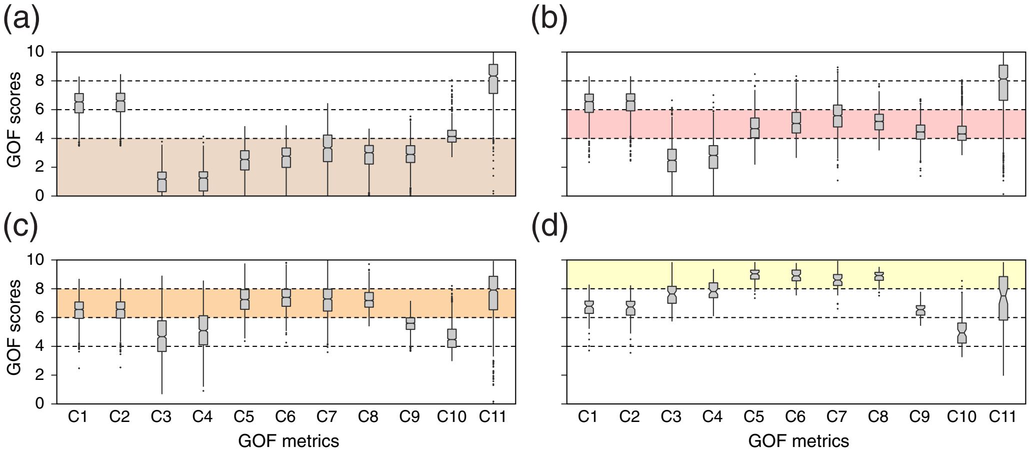

Figure 6. Statistical distribution of the dataset after the clustering analysis, as partitioned into the four validation categories: (a) poor, (b) fair, (c) good, and (d) excellent. The distributions are shown in the form of box plots for each metric (C1–C11). In each case, the medians are indicated by a notch in the boxes of the central quartiles, and the vertical lines represent the interquartile range, with outliers shown as scattered dots. Dashed lines and background shadows represent the boundaries of the poor, fair, good, and excellent categories as defined by Anderson (2004). The color version of this figure is available only in the electronic edition.

time. (The only cases in which this is a direct view into the clustering results are the two-feature analysis cases.)

Some aspects of Figure 5 are worth discussing. First of all, one must keep in mind that these are not typical correlation plots. How well defined the clusters are from each other or with respect to the metrics are important aspects to consider. Although in most plots all four clusters are clearly distinguishable, there are others in which that is not the case. In the combinations in the three-feature analysis, for instance, the excellent cluster has a limited presence. This is due to the influence of the lower C9 values (see Fig. 3) on the correlations with C5 and C7. This is not necessarily a bad thing, as it helps understand the different relationships between the various GOF metrics. In the best of cases, the combinations C1–C2 and C5–C8 in the two- and four-feature analysis, for instance, exhibit an almost perfect proportionality between the metrics. That means either one of the metrics in those combinations is redundant. Relationships with redundant features are those where knowledge about one of the features provides a direct view into the other. On the other hand, the combination C1–C7 in the two-feature analysis presents an example of an irrelevant feature. We say C1 is irrelevant because it provides no insight about the outcome of C7 or that of the clusters. Identifying subspaces with redundant and irrelevant features is important because, on the one hand, the former help reduce the number of necessary features, whereas on the other hand, the latter can essentially be discarded because of their weak contribution to the decision-making process (Dy and Brodley, 2004).

Figure 6 shows the statistical distribution (box plots) of the samples once the dataset is partitioned into the four GOF validation classes. This is similar to Figure 3, but after the clustering process is completed. Separately, we prepared similar plots to look at the influence of the velocity models and components of motions on the results of the clustering process, and as observed before, they had no significant differences with the aggregate of all the samples in the dataset. Figure 6 is particularly important because it highlights the outcome of the clustering process and provides a preview into the decision-tree analysis results. Metrics C5–C8 consistently fall within Anderson's poor, fair, good, and excellent classifications, and that metrics C3 and C4 also show a progressive variation in sync with these categories. This means that these metrics are likely the best predictors for the outcome of the validation process, as we will see upon performing the decision-tree analysis. On the other hand, metrics C1 and C2, and C9–C11 are almost invariant, therefore less or not decisive in the validation process. C11, in particular, shows a broader (larger boxes) distribution, which indicates that it is less effective in predicting the final validation class.

In total, the clustering process results in 816, 1253, 879, and 76 data samples for the poor, fair, good, and excellent classes, respectively. These groups are shown in Figure 7. As it can be seen in this figure, the number of samples in the excellent class is significantly less than those in the poor, fair, and good classes. Therefore, before moving on with the decision-tree analysis, it is necessary to resample the subset of the excellent class, for which we use the oversampling approach described in the Data Analysis Method section. Oversampling is nothing else but a replication of data samples. This process is done randomly, that is, through a random selection of data samples from the original set to be duplicated until one increases the number of total samples in the oversampled set to a desired target number. In the case of the excellent class, we applied oversampling until reaching a total of 760 samples, as indicated with the dashed line in Figure 7. Because the original size was 76 samples, that means we applied an oversampling ratio of 10×. According to Weiss and Provost (2003), an oversample rate of 10× is

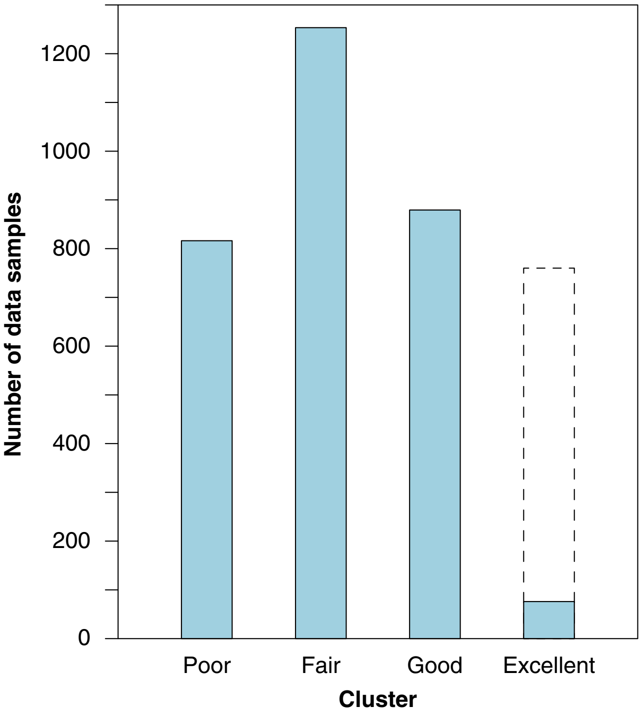

Figure 7. Number of data samples in each class (poor, fair, good, and excellent) after conducting a multidimensional constrained k-means clustering process using subspace analysis for two, three, and four features (GOF metrics). The dashed-line bar indicates the number of samples in the excellent class after oversampling. The color version of this figure is available only in the electronic edition.

considered to be acceptable in this type of data processing. (Arguably, we could have undersampled the fair class as well but we deemed that unnecessary. Besides, we used a heavy pruning process to prevent overfitting.) Once this process was completed, we went on with the decision-tree analysis.

In total, we generated 20,000 trees using the C5.0 algorithm for all possible combinations of the parameters CF and $S_{\min}$, in which CF was chosen to vary between 0 and 1 at intervals of size 0.01, and $S_{\min}$ was chosen to vary between 1 and 200 at unitary intervals. Despite our choice for small intervals, the algorithm often reached recurrent tree structures for different CF and $S_{\min}$ values. Therefore, in reality, from the 20,000 combinations for which we ran the algorithm, only 66 unique trees were found.

For each one of these unique trees, we computed the effectiveness factor F_1 from equation (5) and extracted the total number of nodes in the trees and their depth. Figure 8 shows the results of F_1 for all the trees and its distribution in terms of the number of decision attributes or GOF metrics used in the trees as a function of the number of nodes and the depth of each tree. Recall that we are interested in finding a sequence of decisions (represented by disjunctive decision nodes in a tree) that can lead to good GOF predictions (i.e., high values of F_1) using a reduced number of attributes (GOF metrics). In general, all the trees obtained with the C5.0 algorithm are good in terms of the effectiveness factor (F_1 values close to 1). Then, our choice comes down to using a reduced number of metrics. Having several trees with 2, 3, and 4 metrics (as opposed to 11), the following factor in the decision is choosing trees with algorithms using a small number of steps to reach the prediction. This is given by a combination between the depth of the tree and the number of nodes in the tree (i.e., trees with low complexity).

Based on these criteria, we selected three candidate trees: T1, T2, and T3. These trees are shown in Figure 9. T1 is the simplest of the three, and T2 and T3 share some of their topology on the right side. T3 is the most complex of them. More complex trees tend to be deeper, have more nodes, and employ more attributes, all of which depends—in part—on the pruning process. More complex trees have

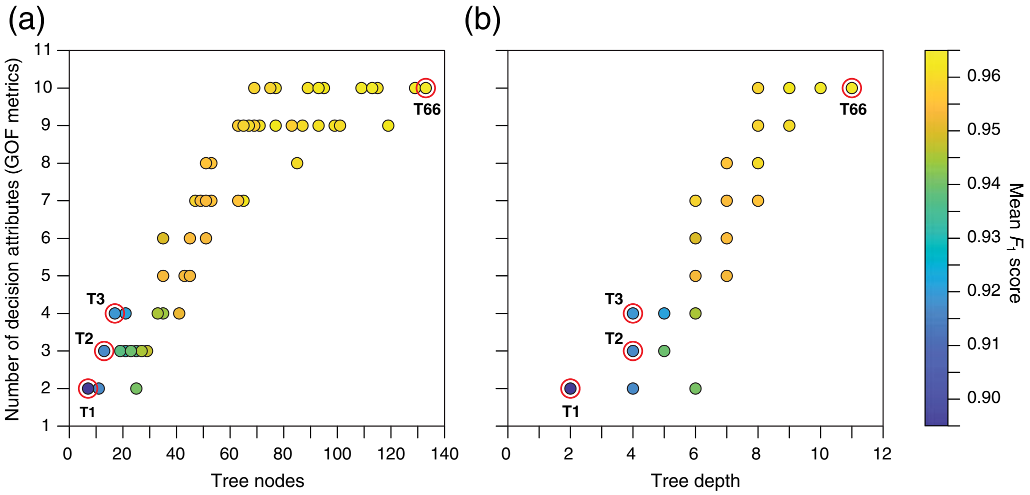

Figure 8. Accuracy of tree predictions in terms of the factor (F_1) from equation (5) as indicated by dots distributed with respect to the number of attributes (GOF metrics) as a function of: (a) the number of nodes and (b) the depth of the trees. The rings around some of the dots indicate select trees used as reference in the Results and Testing sections, Figures 9–11, and Tables 2–7. The color version of this figure is available only in the electronic edition.

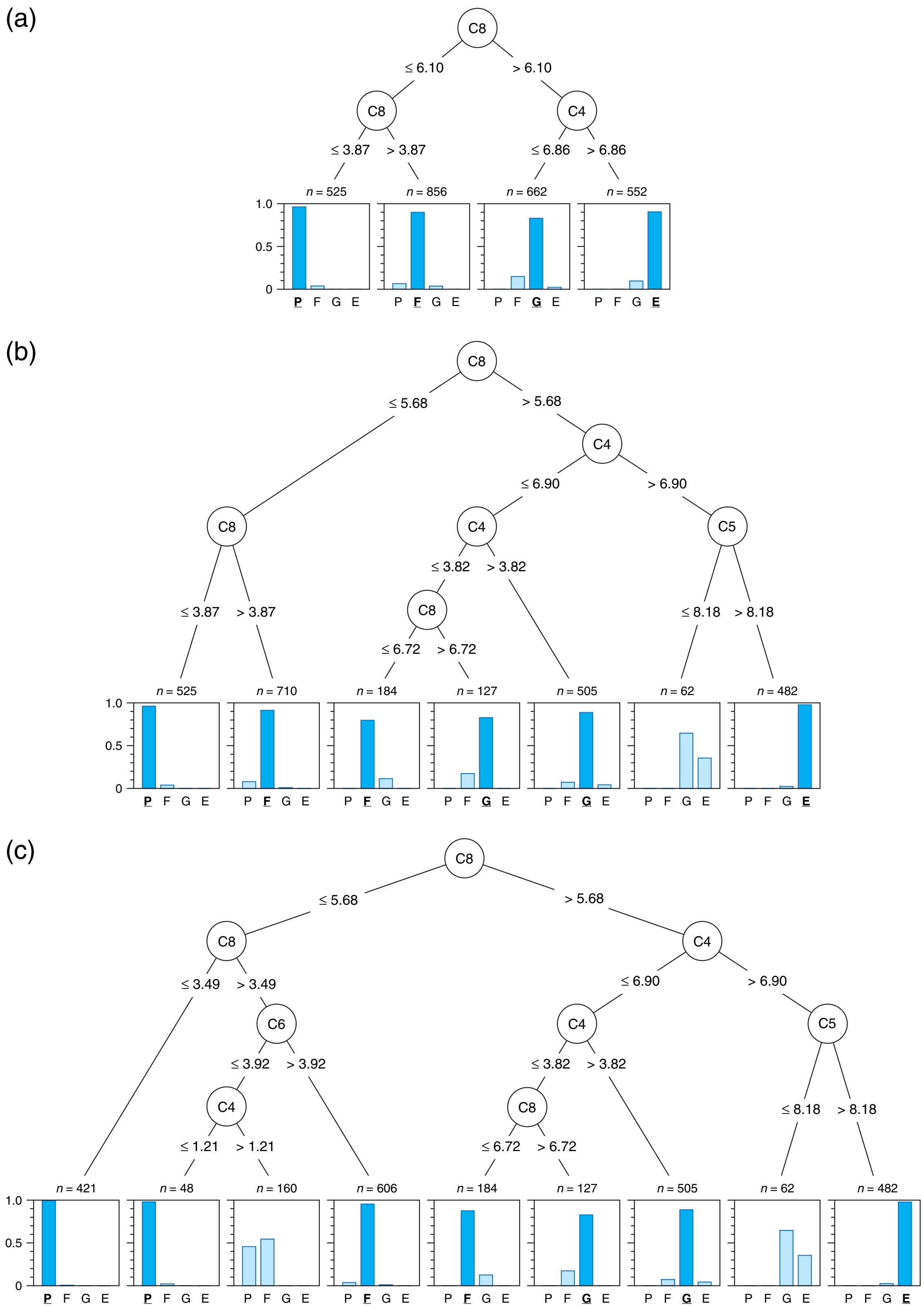

Figure 9. Selected trees (a) T1, (b) T2, and (c) T3. In each case, the decision nodes of the trees contain the code corresponding to the metric (see Table 1) used and the branches beneath each node show the limit value to be used to select the next level in the tree. At the bottom, normalized histograms are shown to indicate the distribution of the samples at each leaf node according to the validation classes with codes P, F, G, and E for poor, fair, good, and excellent, respectively. At the top of each histogram is the total count of samples at the corresponding leaf node. Histograms highlight the dominant validation class. The color version of this figure is available only in the electronic edition.

Table 2
Confusion Matrix Results for Decision Tree T1

		Prediction			
		P	F	G	E
Actual	P	242	11	0	0
	F	13	313	16	0
	G	0	42	207	15
	E	0	0	23	231

Table 3
Confusion Matrix Results for Decision Tree T3

		Prediction			
		P	F	G	E
Actual	P	248	14	0	0
	F	7	343	18	0
	G	0	9	221	33
	E	0	0	7	213

Table 4
T1 Precision, Recall, and F_1 Values per Class

Class	P	R	F_1
P	0.96	0.95	0.95
F	0.92	0.86	0.88
G	0.78	0.84	0.81
E	0.91	0.94	0.92
Mean			0.90

Table 5
T3 Precision, Recall, and F_1 Values per Class

Class	P	R	F_1
P	0.95	0.97	0.95
F	0.93	0.93	0.93
G	0.84	0.89	0.86
E	0.96	0.86	0.91
Mean			0.91

higher F_1 values, but they may not necessarily be more practical. In total, T1, T2, and T3 use 2, 3, and 4 attributes, respectively. All coincide in the use of the total energy (C4) and response spectra (C8) as key metrics. T2 adds the peak acceleration (C5), and T3 adds the peak velocity (C6) to the previous metrics.

At the bottom of each tree in Figure 9 are the histograms of the data that land on each leaf node. (The count of samples here is done based on the training dataset.) The decision tree assigns the final validation category based on the dominant class in each leaf. As such, the samples in the second leaf node from the right in T3 are categorized as good despite a significant portion of them being excellent. This is a natural trade-off embedded in the use of decision trees.

The actual effectiveness (i.e., the average F_1 values shown in Fig. 8), however, is measured based on the testing dataset. Recall that F_1 depends on the number of true predictions and false predictions measured by the confusion matrix and the precision and recall factors. Tables 2 and 3 show the confusion matrices for T1 and T3, respectively; and Tables 4 and 5 show the corresponding results for P, R, and F_1. As it can be seen in Tables 2 and 3, both trees lead to strong diagonal confusion matrices, meaning that the classification into poor, fair, good, and excellent is well defined. The differences between both matrices are actually minor and small with respect to their diagonal values, and the F_1 values in Tables 4 and 5 are all near to or above 0.9. The values obtained for tree T2 are consistent with these observations.

Another aspect of interest is the level of participation of each metric in the analysis of data, as the GOF values are run through the nodes of the decision trees (i.e., as the tree is traversed). The results of such level of participation are listed in Table 6 for trees T1, T2, T3, and T66. (The tree T66 is the most complex of them all as inferred from the number of metrics involved and the number of nodes and levels seen in Fig. 8.) The percentages in this table represent the amount of data that is seen by a decision node associated with any particular metric. The nodes are not unique, and metrics are often used in different nodes in a given tree. Therefore, these percentages accumulate for each metric as the tree is traversed. A percentage of 100 means that a given metric has the opportunity to see all the data samples through one or multiple nodes, a low percentage means only a few data samples are evaluated by that metric, and a null percentage means the metric plays no role in the decision tree (it is not present in any node).

Table 6
Data Analysis Participation (in Percent) for Select Trees

Code	Metric	T1	T2	T3	T66
C1	Arias integral	—	—	—	1.5
C2	Energy integral	—	—	—	0.5
C3	Arias intensity	—	—	—	9.6
C4	Total energy	46.8	52.4	60.2	100.0
C5	Peak acceleration	—	21.5	21.5	72.2
C6	Peak velocity	—	—	37.0	56.3
C7	Peak displacement	—	—	—	19.4
C8	Response spectrum	100.0	100.0	100.0	100.0
C9	Fourier spectrum	—	—	—	23.8
C10	Cross correlation	—	—	—	22.2
C11	Strong phase duration	—	—	—	—

The results in Table 6 highlight the fact that, for all trees, the metrics for total energy (C4) and response spectra (C8) are consistently the most relevant ones in the decision algorithms. They are followed by the peak acceleration (C5) and peak velocity (C6). On the other end, the strong phase duration (C11) plays no role whatsoever in any of the trees, and the

Table 7
Data Analysis Participation (in Percent) for Select Trees after Removing Certain Metrics

		Without C8				Without C5 and C8			
Code	Metric	T1	T2	T3	T66	T1	T2	T3	T66
C1	Arias integral	—	—	—	—	—	—	—	—
C2	Energy integral	—	—	—	2.3	—	—	—	—
C3	Arias intensity	—	—	—	53.2	—	60.4	61.2	99.1
C4	Total energy	44.3	81.8	81.8	83.2	47.44	48.7	80.5	54.6
C5	Peak acceleration	100.0	100.0	100.0	100.0	—	—	—	—
C6	Peak velocity	58.6	63.5	62.8	82.4	100.0	100.0	100.0	100.0
C7	Peak displacement	—	—	—	32.4	—	—	—	43.2
C8	Response spectrum	—	—	—	—	—	—	—	—
C9	Fourier spectrum	—	—	—	48.6	—	—	—	72.2
C10	Cross correlation	—	—	—	24.9	—	—	—	20.5
C11	Strong phase duration	—	—	—	—	—	—	—	8.1

Arias integral (C1), energy integral (C2), and Arias intensity (C3) have only small to marginal participations. The remaining metrics—peak displacement (C7), Fourier spectrum (C9), and cross correlation (C10)—have a somewhat significant participation, but only in the more complex trees.

An additional aspect worth highlighting here is the fact that the clustering and decision-tree processes address the problem of correlation between the metrics in a natural way. For instance, one would expect the Arias intensity (C3) and peak acceleration (C5), or the Arias intensity (C3) and total energy (C4) to be related. Considering them all without distinction could lead to double weighing their influence, and assuming a 1-to-1 relationship may overstate their level of correlation. Instead, the process used here determines which of them has a better chance to predict the outcome in the presence of the other and preserves that metric as part of the decision tree.

One can put this to test by removing one or more metrics and see how the participation of the metrics rearranges. Table 7 shows these participations for trees with similar topologies (1) when removing the response spectrum (C8), and (2) when removing both the peak acceleration (C5) and the response spectrum (C8). In the first case, the peak acceleration (C5) takes the place of the most determinant metric, followed by the total energy (C4) and peak velocity (C6), depending on the tree. In the second case, the peak velocity (C6) becomes the most dominant, followed by the Arias intensity (C3) and the total energy (C4), which vary in their level of participation depending on the tree. These additional alternatives emphasize the role of the total energy (C4) and the peak velocity (C6), and show that other metrics such as the peak acceleration (C5), peak displacement (C7), and Fourier spectrum (C9) are also relevant. They also reaffirm the low or null levels of contribution of the metrics associated with the shape of the Arias integral (C1), the energy integral (C2), and the cross correlation (C10) and strong phase duration (C11). All this is consistent with what we observed in Figure 6.

That being said, a low participation does not necessarily mean that a given metric is not relevant at all. Arguably, the strong phase duration is highly regarded as an important parameter for strong-motion records in engineering. What the results we present here mean is that, in the context of this particular set of 11 metrics, to classify whether a simulation result is poor, fair, good, or excellent, other metrics such as the total energy (C4) are much more influential in the final result than the strong phase duration (C11). This, of course, is done under the assumption that one seeks to narrow the selection of metrics without loss of insight about the outcome of the validation process when compared with the outcome obtained with the whole suite of the 11 metrics used here.

In the end, the final selection of a preferred tree comes down to reducing complexity in the analysis without compromising the interpretation of results. We favor tree T1 because it is based only on three decision steps, and two GOF metrics: the total energy (C4) and response spectra (C8).

Testing

We now test tree T1 on the original simulation results from Taborda and Bielak (2014), for different velocity models and components of motion. In this case, we no longer aggregate all data from the simulation but now look at individual simulations done with different velocity models, and the three components of motion separately, as it would normally be done during any ground-motion simulation validation. The results obtained with the T1 validation algorithm are shown in Figure 10. We should note that they are not supposed to be the same as those obtained by Taborda and Bielak (2014) because T1 no longer gives the same weight to all metrics but relies only on the two metrics C4 and C8. A drawback of the T1 algorithm is that because its outcomes are GOF validation classes (poor, fair, good, and excellent) as opposed to GOF validation scores (with values from 0 to 10), once one obtains the results for the validation process for different components of motion (as shown in the rows of Fig. 10), there is no natural way of taking averages as one would do for scores in a numerical scale (i.e., as typically done when using Anderson, 2004). Therefore, to combine results from the three components into a single validation classification for each station, we define the following rules:

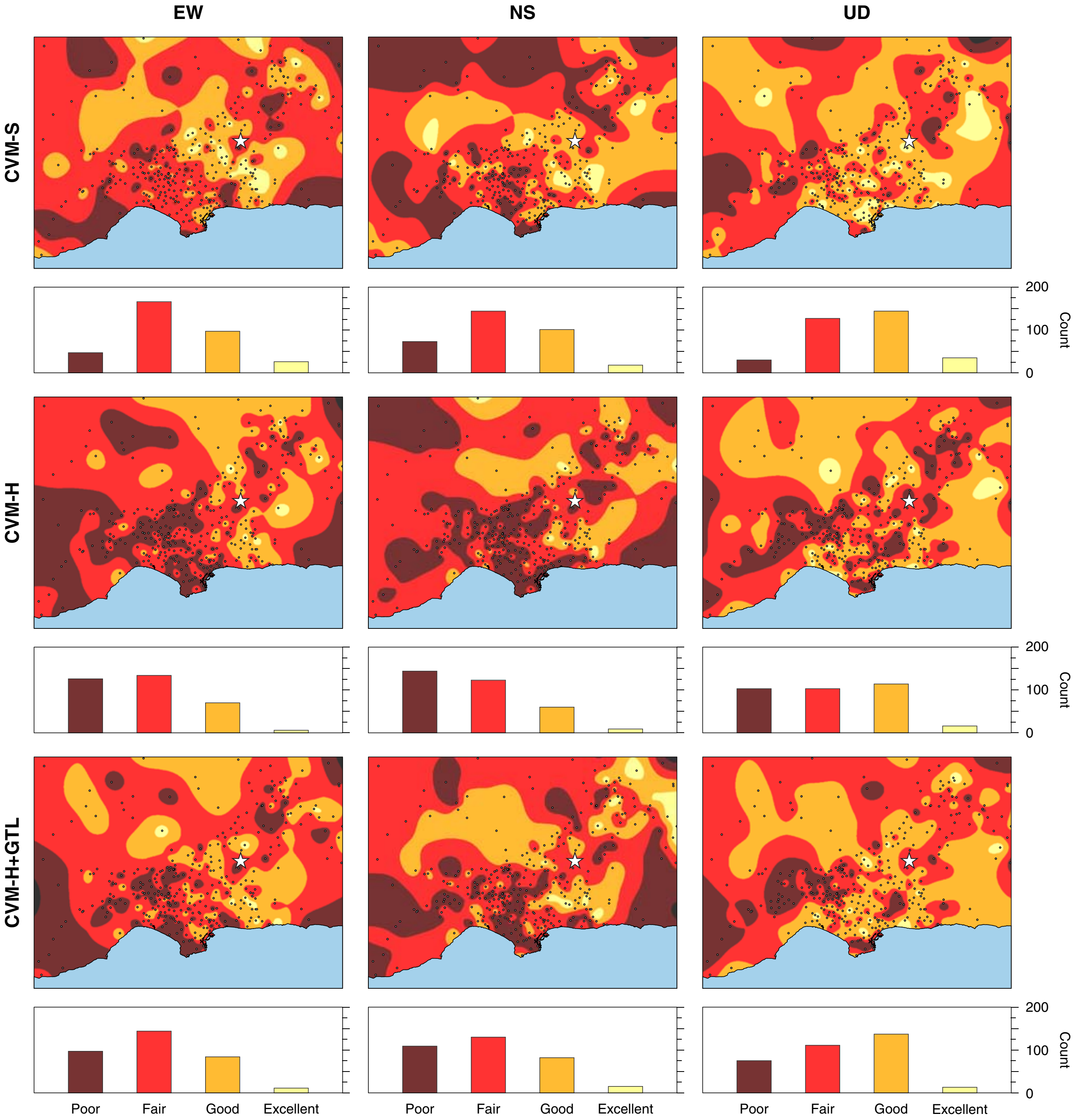

Figure 10. Results obtained using the validation algorithm of the selected decision tree T1 on the 2008 Chino Hills earthquake simulation results reported by Taborda and Bielak (2014). Here, the validation process is carried out separately for each component of motion (EW, NS, and UD), for three different simulations corresponding to simulations done using the southern California velocity models: CVM-S, CVM-H, and CVM-H+GTL. Contours maps are drawn for illustration purposes only based on the spatial distribution of the GOF validation class assigned to each station. The bottom histograms show the count of stations in each validation class. Stations are indicated with dots, and the epicenter with a star. The color version of this figure is available only in the electronic edition.

- If the three components share the same validation class, then that class is assigned as the combination result.
- If two components share the same validation class, but one differs, then the class shared by the former two is assigned as the combination result, thus favoring the majority.
- If all three components have different validation classes, then the final combination result is set as the lower class of the three, penalizing the lack of conclusive results.

We call the result of applying these rules a tree T1 combination. Applying them to the validation results shown in

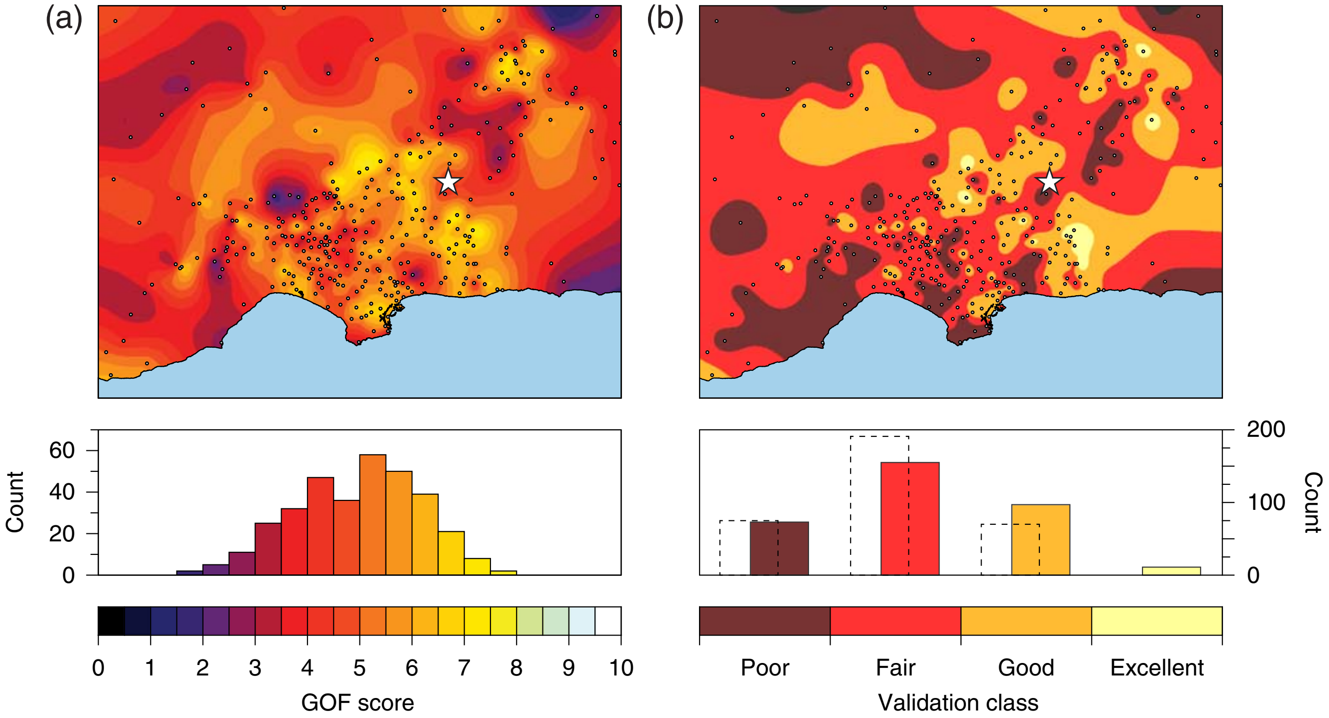

Figure 11. Comparison of GOF validation scores obtained using (a) an 11-metric Anderson (2004)-type GOF scoring, and (b) the T1 combination GOF validation classification. In both cases, the top plots show maps with a distribution of the scores or classes outcome at each station, where the contours are drawn for illustration purposes only; while the bottom plots show histograms with the count of stations for each GOF score interval or validation class. On the right side, the results of both methods are compared by superimposing the results of the counts from using Anderson's method (dashed-line empty bars) next to those obtained using the T1 algorithm (filled bars). The color version of this figure is available only in the electronic edition.

Figure 10 for the case of the simulation done using the model CVM-S is shown in Figure 11, which compares the outcome with the result of the average scores obtained using the traditional Anderson (2004)-type GOF method. This figure exemplifies the differences or similarities between the numeric GOF validation-scores scale and the proposed T1 GOF validation classes.

In our view, the T1 results are simpler to assimilate and equally informative. Looking at both plots in Figure 11, it is possible to argue that they lead to similar conclusions in respect to the overall validation of the simulation, and in respect to some particular areas and specific locations. Nonetheless, it must be said that we do not expect to see a 1-to-1 relationship, as evidenced by the differences in the histograms when compared in equal terms (see dashed line counts on the right side corresponding to the count of the original results from the left side histogram drawn next to the counts of the results from the T1 algorithm). Finding the best possible match is the scope of a future work, whereas here our focus was on identifying the relevance of the different metrics. This is important because the original approach proposed by Anderson (2004) favors a uniform weighing of metrics, and the results shown here indicate this may not be the preferred strategy.

Conclusions

We present the results of a machine learning analysis using semisupervised and supervised methods on a large dataset comparing synthetics and data based on multiple GOF metrics used in ground-motion validation with the goal of prioritizing and reducing the number of metrics, and develop an application independent decision algorithm. As a result of the data processing analysis, we propose a simplified algorithm based on a decision tree which uses only two metrics as opposed to the initial 11 available in the dataset. In particular, the proposed algorithm uses three (disjunctive decision) steps based on the values obtained for the metrics of the total energy and acceleration response spectra. We also propose rules to allow for the new class-based validation criteria to combine results obtained from different components of motion, and demonstrate that the results obtained with the proposed algorithm using two metrics are comparable to those obtained with the score-based validation results used in other recent validation studies. One could implement similar rules to combine results from a frequency-band analysis, using, for example, the mode or majority validation class.

We recognize, however, that the proposed decision-tree algorithm may not be a definitive one because of a potential bias on the fact that the dataset used here, although large enough from a statistical point of view, came from simulations done for a particular earthquake, in a particular region, and using a particular set of metrics. In a future follow-up study, it would be ideal to refine the tree adding other simulation datasets (i.e., from different earthquakes, regions, models, and using additional metrics) to arrive to a sufficiently sound and robust decision tree. The procedural steps laid out here, nonetheless, remain valid, and there is no reason why not to use it also in other contexts utilizing other metrics (e.g., structural responses metrics such as drift ratio) provided that they are properly normalized. In summary, we can say that for the case of the metrics used here, we showed that the procedure and background information used for clustering and decision making is stable, and it is likely that—despite the limitations just described—the metrics of energy and response spectra (along with peak acceleration and velocity as suggested by the additional trees) will prevail as those among the most decisive ones.

Identifying the total energy and response spectra metrics as decisive metrics in the comparisons between synthetics and observations is a key contribution to validation of ground-motion simulations. This contributes to clarifying a standing question in the area of validation, and it provides an indication to simulators about where to focus in the search for improvements in their models.

Data and Resources

Calculations, data processing, and some initial figures were done using R, the language and environment for statistical computing and graphics (https://www.r-project.org, last accessed May 2018), and the C5.0 package (https://CRAN.R-project.org/package=C50, last accessed May 2018). Additional calculations and figures were done using MATLAB (http://www.mathworks.com, last accessed May 2018). Map figures were prepared using the Generic Mapping Tools (GMT; http://gmt.soest.hawaii.edu/, last accessed May 2018). Additional editing of figures was done using Adobe Illustrator (http://www.adobe.com/Illustrator, last accessed May 2018). The validation dataset used here was readily available to the authors and can be provided without restrictions upon request.

Acknowledgments

This work was supported by National Science Foundation (NSF) awards: "SI2-SSI: Community Software for Extreme-Scale Computing in Earthquake System Science" (ACI-1450451), and "Improving Earthquake Forecasting and Seismic Hazard Analysis Through Extreme-Scale Simulations" (OAC-1713792); and Southern California Earthquake Center (SCEC) awards: "An Examination of the Correlations Between Different Goodness-of-Fit Metrics Based on a Large Dataset of Ground Motion Simulation Validations" (16-067), and "Application of Machine Learning in Deterministic Ground Motion Simulation" (17-239). SCEC is funded by NSF Cooperative Agreement EAR-1600087 and U.S. Geological Survey Cooperative Agreement G17AC00047. The SCEC contribution number for this article is 8015. The authors thank two anonymous reviewers and Associate Editor Thomas Brocher for their detailed and constructive comments, which were very helpful in improving the original article.

References

Anderson, J. G. (2004). Quantitative measure of the goodness-of-fit of synthetic seismograms, *Proc. 13th World Conf. on Earthquake Eng.*, Vancouver, Canada, Int. Assoc. Earthquake Eng. Paper 243.

Bielak, J., R. W. Graves, K. B. Olsen, R. Taborda, L. Ramírez-Guzmán, S. M. Day, G. P. Ely, D. Roten, T. H. Jordan, *et al.* (2010). The ShakeOut earthquake scenario: Verification of three simulation sets, *Geophys. J. Int.* **180,** no. 1, 375–404, doi: 10.1111/j.1365-246X.2009.04417.x.

Branco, P., L. Torgo, and R. P. Ribeiro (2016). A survey of predictive modeling on imbalanced domains, *ACM Comput. Surv.* **49,** no. 2, 31, doi: 10.1145/2907070.

Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*, Chapman & Hall/CRC, Boca Raton, Florida.

Burks, L. S., and J. W. Baker (2014). Validation of ground-motion simulations through simple proxies for the response of engineered systems, *Bull. Seismol. Soc. Am.* **104,** no. 4, 1930–1946, doi: 10.1785/0120130276.

Chaljub, E., P. Moczo, S. Tsuno, P.-Y. Bard, J. Kristek, M. Kaser, M. Stupazzini, and M. Kristekova (2010). Quantitative comparison of four numerical predictions of 3D ground motion in the Grenoble Valley, France, *Bull. Seismol. Soc. Am.* **100,** no. 4, 1427–1455, doi: 10.1785/0120090052.

Dy, J. G., and C. E. Brodley (2004). Feature selection for unsupervised learning, *J. Mach. Learn. Res.* **5,** 845–889.

Fayyad, U. M. (1996). Data mining and knowledge discovery: Making sense out of data, *IEEE Expert* **11,** no. 5, 20–25, doi: 10.1109/64.539013.

Guidotti, R., M. Stupazzini, C. Smerzini, R. Paolucci, and P. Ramieri (2011). Numerical study on the role of basin geometry and kinematic seismic source in 3D ground motion simulation of the 22 February 2011 M_w 6.2 Christchurch earthquake, *Seismol. Res. Lett.* **82,** no. 6, 767–782, doi: 10.1785/gssrl.82.6.767.

Hartigan, J. A. (1985). Statistical theory in clustering, *J. Classification* **2,** no. 1, 63–76, doi: 10.1007/BF01908064.

Jain, A. K., and R. C. Dubes (1988). *Algorithms for Clustering Data*, Prentice-Hall, Inc, Englewood Cliffs, New Jersey.

Jain, A. K., M. N. Murty, and P. J. Flynn (1999). Data clustering: A review, *ACM Comput. Surv.* **31,** no. 3, 264–323, doi: 10.1145/331499.331504.

Khoshnevis, N., and R. Taborda (2015). Sensitivity of ground motion simulation validation criteria to filtering, *Proc. 12th Int. Conf. Appl. Stat. Probab. Civil Eng. (ICASP)*, Vancouver, Canada, 12–15 July, doi: 10.14288/1.0076157.

Kristeková, M., J. Kristek, and P. Moczo (2009). Time-frequency misfit and goodness-of-fit criteria for quantitative comparison of time signals, *Geophys. J. Int.* **178,** no. 2, 813–825, doi: 10.1111/j.1365-246X.2009.04177.x.

Kristeková, M., J. Kristek, P. Moczo, and S. M. Day (2006). Misfit criteria for quantitative comparison of seismograms, *Bull. Seismol. Soc. Am.* **96,** no. 5, 1836–1850, doi: 10.1785/0120060012.

Kuhn, M., S. Weston, M. Culp, N. Coulter, and R. Quinlan (2017). User's manual for the package 'C50', available at https://cran.r-project.org/web/packages/C50/C50.pdf (last accessed May 2018).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations, *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. I, Oakland, California, 281–297.

Maufroy, E., E. Chaljub, F. Hollender, J. Kristek, P. Moczo, P. Klin, E. Priolo, A. Iwaki, T. Iwata, V. Etienne, *et al.* (2015). Earthquake ground motion in the Mygdonian basin, Greece: The E2VP verification and validation of 3D numerical simulation up to 4 Hz, *Bull. Seismol. Soc. Am.* **105,** no. 3, 1398–1418, doi: 10.1785/0120140228.

McCarthy, J. F., and W. G. Lehnert (1995). Using decision trees for coreference resolution, *Proc. 14th Int. Joint Conf. Artif. Intell.*, Montreal, Quebec, 20–25 August, available at https://arxiv.org/abs/cmp-lg/9505043 (last accessed May 2018).

Mitchell, T. (1997). *Machine Learning*, McGraw-Hill, New York, New York.

Murthy, S. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey, *Data Min. Knowl. Discov.* **2,** no. 4, 345–389, doi: 10.1023/A:1009744630224.

Olsen, K. B., and J. E. Mayhew (2010). Goodness-of-fit criteria for broadband synthetic seismograms, with application to the 2008 M_{w} 5.4 Chino Hills, California, earthquake, *Seismol. Res. Lett.* **81,** no. 5, 715–723, doi: 10.1785/gssrl.81.5.715.

Parsons, L., E. Haque, and H. Liu (2004). Subspace clustering for high dimensional data: A review, *ACM SIGKDD Explor. Newsl.* **6,** no. 1, 90–105, doi: 10.1145/1007730.1007731.

Quinlan, J. R. (1986). Induction of decision trees, *Mach. Learn.* **1,** no. 1, 81–106, doi: 10.1023/A:1022643204877.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Burlington, Massachusetts.

Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5, *J. Artif. Intell. Res.* **4,** 77–90, doi: 10.1613/jair.279.

Rezaeian, S., P. Zhong, S. Hartzell, and F. Zareian (2015). Validation of simulated earthquake ground motions based on evolution of intensity and frequency content, *Bull. Seismol. Soc. Am.* **105,** no. 6, 3036–3049, doi: 10.1785/0120140210.

Small, P., D. Gill, P. J. Maechling, R. Taborda, S. Callaghan, T. H. Jordan, K. B. Olsen, G. Ely, and C. A. Goulet (2017). The unified community velocity model software framework, *Seismol. Res. Lett.* **88,** no. 6, 1539–1552, doi: 10.1785/0220170082.

Taborda, R., and J. Bielak (2013). Ground-motion simulation and validation of the 2008 Chino Hills, California, earthquake, *Bull. Seismol. Soc. Am.* **103,** no. 1, 131–156, doi: 10.1785/0120110325.

Taborda, R., and J. Bielak (2014). Ground-motion simulation and validation of the 2008 Chino Hills, California, earthquake using different velocity models, *Bull. Seismol. Soc. Am.* **104,** no. 4, 1876–1898, doi: 10.1785/0120130266.

Taborda, R., S. Azizzadeh-Roodpish, N. Khoshnevis, and K. Cheng (2016). Evaluation of the southern California seismic velocity models through simulation of recorded events, *Geophys. J. Int.* **205,** no. 3, 1342–1364, doi: 10.1093/gji/ggw085.

Trifunac, M. D., and A. G. Brady (1975). A study on the duration of strong earthquake ground motion, *Bull. Seismol. Soc. Am.* **65,** no. 3, 581–626.

van Rijsbergen, C. J. (1979). *Information Retrieval*, Butterworth-Heinemann, London, United Kingdom.

Wagstaff, K., C. Cardie, S. Rogers, and S. Schroedl (2001). Constrained κ-means clustering with background knowledge, *Proc. of the Eighteenth International Conf. on Machine Learning*, 577–584.

Weiss, G. M., and F. Provost (2003). Learning when training data are costly: The effect of class distribution on tree induction, *J. Artif. Intell. Res.* **19,** 315–354, doi: 10.1613/jair.1199.

Wu, C. L., K. Chau, and C. Fan (2010). Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques, *J. Hydrol.* **389,** no. 1, 146–167, doi: 10.1016/j.jhydrol.2010.05.040.

Center for Earthquake Research and Information
The University of Memphis
3890 Central Avenue
Memphis, Tennessee 38152
nkhshnvs@memphis.edu
(N.K.)

Department of Civil Engineering,
and Center for Earthquake Research and Information
The University of Memphis
3890 Central Avenue
Memphis, Tennessee 38152
ricardo.taborda@memphis.edu
(R.T.)

Manuscript received 11 February 2018;
Published Online 26 June 2018