Local Surrogates vs Global Surrogates - visual comparison of decision boundaries

Jonas Daugalas and **Emanuel Gerber**

jonas.daugalas@tum.de
emanuel.gerber@tum.de

Abstract

This work explores new methods for comparing local and global surrogates through decision region visualization. We present an implementation for approximating and visualizing decision regions of text classification models. The implementation combines text perturbation, dimensionality reduction and Voronoi diagrams. The proposed method provides more insights about decision boundary landscape complexity compared to basic numerical scores such as accuracy or the R-squared metric.

1 Introduction

Surrogates are a popular method in the domain of explainable AI. They are simple, interpretable models (like Linear Regression models or Decision Trees) that are trained to approximate the predictions of an uninterpretable black box model. From the observations of an explainable surrogate model we can draw conclusions about the black box model.

There are two types of surrogates: local surrogates and global surrogates

1.1 Local Surrogates

Local surrogate models are trained to approximate a black box model prediction around a single data point (Molnar, 2019). The goal of a local surrogate is to identify the reasons behind a particular prediction. An example of a model interpretation method which uses a local surrogate is LIME (Ribeiro et al., 2016). LIME identifies the importance of input features towards the predicted outcome, by applying variations to the input and modeling how the prediction changes depending on the existence of a feature. It can thereby detect those features that determined the outcome of the prediction most dominantly.

1.2 Global Surrogates

Global surrogate models are trained to approximate the predictions of a black box model for all inputs (Molnar, 2019). By interpreting a global surrogate, we can obtain insights about the global decision policies of the black box model.

1.3 Goal

Since both local and global surrogates provide a way for explaining the original black box model, we would like to compare, which one approximates the original model better. For that, we need to compare how well each surrogate represents the original.

However, to the best of our knowledge the only approaches for comparing surrogates are basic accuracy, R^2 or similar measures. For example, the R^2 metric provides a numerical value for how much variance from the black box model is captured by the surrogate model. Unfortunately such single metrics do not provide detailed information about the behaviour of a model or its decision region complexity. Moreover, R^2 gives no indication for how complex models behave in a particular local surrounding.

The goal of this project is to explore a new method for comparing local surrogates and global surrogates by visualizing their decision boundaries. By visually comparing decision landscapes, we hope to get more insights about how different surrogate models approximate the black box model and how complex these models behave locally.

2 Preliminaries

2.1 Exact Boundaries

Visualizing decision regions requires the identification of decision boundaries. In some cases, finding exact decision boundaries of a classifier is trivial. For example in the case of Support Vector Machines, the exact decision boundary is given by the

hyper plane that separates two classes. However, finding decision boundaries of black box machine learning model such as a neural network model is generally not trivial. It can be shown that the problem of finding exact decision boundaries is at least as difficult as the exact robustness certification.

Let's consider a sub-problem of finding the exact decision boundaries: given a range of the input space $I = [\vec{x}_{min}, \vec{x}_{max}]$ and a classifier $f(\vec{x})_\theta$, check if there exists a decision boundary point of model $f(\vec{x})_\theta$ within the range I.

The above formulation leads to the **exact robustness certification** (Günnemann, 2020) problem. The goal of the exact robustness certification is to answer whether a classifier f_θ is adversarial-free around a sample $\vec{x}$ within ϵ radius (measured by a given norm). The robustness certification algorithm yields a positive result if and only if there are no adversarial examples within an ϵ ball around the input sample.

Note that the exact robustness certification with L_∞-norm is equivalent to our decision boundary sub-problem formulation with $I = [\vec{x} - \vec{1}\epsilon, \vec{x} + \vec{1}\epsilon]$. Hence, if we can efficiently solve our decision boundary sub-problem, we also solve the exact robustness certification for the L_∞-norm ϵ ball.

Unfortunately, it is proven that verifying properties (including exact robustness certification) of deep neural networks with ReLU activations is NP-Complete (Katz et al., 2017).

3 Related work

Previous work (Vlassopoulos et al., 2020) has already highlighted the importance of decision boundary approximations for the goal of model explainability. However, there is no current implementation for visualizing decision regions of complex models. Existing visualization methods such as (Raschka, 2018) already provide an implementation for visualizing decision regions for simple models such as decision trees, linear regression, or models with low dimensional inputs. But there is no support for large neural network models with high dimensional inputs, as they are used for many NLP tasks.

4 Methods

4.1 Black-box models

In this project we used a neural text classification network as our baseline black-box model. Our network receives a single, fixed-size vector as input and returns prediction scores. We transform the natural language input of variable length into this fixed vector format by resolving each word from the input (excluding stopwords) with its GloVe embedding (Pennington et al., 2014). Afterwards we average GloVe embeddings into a single vector (1).

We trained our network on the AG News dataset (agn) and achieved a test accuracy of 89%.

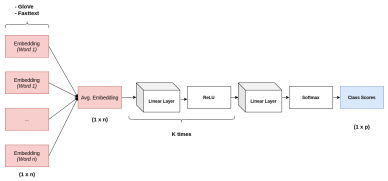


Figure 1: Neural Network Architecture

4.2 Gradient based boundary search

We experimented with gradient based boundary search, which was motivated by techniques from Adversarial Attacks. We attempted to approximate the outline of the decision boundary by finding input values that result in a prediction score with equal confidence between exactly two classes. We used backpropagation on the averaged input embeddings to optimize towards inputs with output scores 0.5 and 0.5 for two different classes. The following code describes the steps for finding local confusion points in a local surrounding of a single sample.

Algorithm 2 Gradient Search Algorithm

1. create random point r in window size ρ around input sample d
2. use backpropagation on random sample r to optimize for "confused" prediction score s (e.g $[0.5, 0.5, 0., 0.]$) over maximum of k training iterations.
3. add $\mathbf{r}$ to local decision boundary if the L2 norm $\|\mathrm{r}-\mathrm{s}\| < \delta$

Besides finding points on the decision boundary, we also optimized towards input points whose predictions are biased towards one of each possible classes.

We visualized the results by applying dimensionality reduction with t-SNE (Van der Maaten and Hinton, 2008) on the inputs and highlighted the inputs corresponding to their class prediction (2). We visualized points on the decision boundary in

red, and class specific inputs in yellow, green, blue, and purple.

In comparison to the adversarial search we also sampled random points in the same window size around the selected input point and visualized the outputs accordingly (2)

We plotted the "decision boundaries" for different original sampled inputs of different classes and with varying locality around them (we tried window sizes in the interval $[10e^{-1}, 10e^{-11}]$). Regardless of the window size and the original data point, we were able to generate input points that were both on the boundaries as well as of different classes. In comparison, using random sampling, we consistently found only members of the the class from the original sample data point.

These findings show that our neural network is susceptible to adversarial attacks. But since random sampling in the same local window around each data point did not find members of different classes, we conclude that gradient search does not help us with our goal of decision region visualization. Instead, it simply highlights model weaknesses and irregularities in the model behaviour.

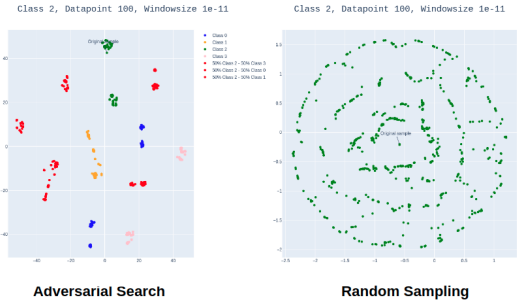


Figure 2: Adversarial Search (left) vs Random Sampling (right)

4.3 Voronoi decision region approximation

Unsuccessful attempts with gradient-based search methods (see 4.2) motivated us to search for alternatives for general decision region approximation.

One approach to approximate a decision boundary between two points is to find a hyperplane that is positioned right in the middle between the two points. In presence of more than two points the mathematics of intersecting hyperplanes may be more involved but the intuition is the same.

Such region partitioning can be achieved by the Voronoi tessellation method (Aurenhammer, 1991). This method partitions space into cells, such that all the points in the input space, for which the distance to a given input point $\vec{x}_i$ is less or equal to any other input point, form a decision region of the point $\vec{x}_i$ class.

Figure 3 shows an example of a 2-dimensional Voronoi diagram for several input points. Colors represent a class label of the point and the corresponding region.

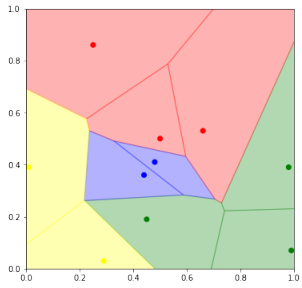


Figure 3: Example 2-dimensional Voronoi diagram.

Decision region approximation of a classifier via Voronoi tessellation method is achieved as described in the following pseudo-code 3:

Algorithm 3 Decision region approximation via Voronoi tessellation

1. sample input points $P = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_n\}$ in the region of interest;
2. classify points P to get predicted labels $\vec{y} = \{y_1, y_2, ..., y_n\}$;
3. partition space with the sample points P into a Voronoi diagram;
4. a Voronoi cell c_i defines a decision region within which the prediction is approximated to be y_i;
5. hyperplane segments between Voronoi cells defining prediction for same class can be ignored;
6. every hyperplane segment between two Voronoi cells c_i and c_j with $y_i \neq y_j$ is part of the decision boundary.

4.4 Decision region approximation in 2D

Using the Voronoi boundary approximation method described in 4.3 we can approximate and visualize local decision landscape around a text sample. For visualization we want a 2-dimensional Voronoi diagram, therefore we also need the inputs to be 2-dimensional real vectors.

One way to have a mapping between text samples and their 2-dimensional numerical representations is to generate those text samples from the

original, such that two text features are perturbed and the extent of each feature perturbation can be expressed as a scalar value. In our case, we chose a text feature to be a specific word of the original sample. By replacing each of the chosen word with another word, we get a perturbed text sample. We measure the similarity between the embedding of the word in the original sample and the embedding of the replaced word to get a numerical value for that particular feature.

Figure 4 shows few perturbed text examples with the described procedure. We use pretrained GloVe embeddings to find similar words for replacement. The cosine distance serves as a numeric value for the distance of the perturbed sample from the original. Each text sample is mapped to a 2D vector $\vec{x} = [x_1, x_2]$ where x_1 is similarity ($1 -$ cosine distance) of the first (yellow) word to the original, and x_2 is similarity of the second (green) word to the original.

Figure 5 illustrates the resulting points plotted in 2D space and the Voronoi diagram forming the decision regions. Note that the original sample corresponds to the point at coordinates $(1, 1)$ because both chosen words are not perturbed and their similarities to the original words are 100%. Other words have lower similarity scores therefore they are lower in both axes.

Figure 4: Text perturbation example by replacing 2 words from the original with similar words by the cosine distance measure between GloVe embeddings of the words.

4.5 Dimensionality reduction for visualization

By limiting inputs to only 2 dimensions, we can visualize the resulting Voronoi diagram directly without any further processing. However, this approach is very constrained. For a more general case we are interested in inputs of higher-dimensional space. We would like to apply perturbations on any number of tokens, or use embeddings as inputs. For these cases we need to be able to visualize more than 2 dimensions. This leaves us with two options:

1. first reduce input dimensions, then apply Voronoi tessellation in 2D;

2. first apply Voronoi tessellation, then reduce result to 2 dimensions.

Both cases lead to problems. Applying Voronoi tessellation in a non-linearly transformed space (first option) does not guarantee to maintain the desirable properties of the Voronoi diagrams in general. In particular, if we have input points in the original space $X = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_n]$, corresponding points in the reduced space $\tilde{X} = [\vec{x}_1^R, \vec{x}_2^R, ..., \vec{x}_n^R]$, and the corresponding Voronoi cells $C = [c_1, c_2, ..., c_n]$, we cannot guarantee that every point $\vec{x}'$ which when transformed would fall within the region of cell c_i would be closest to the input point $\vec{x}_i$ and not any other input point.

In the case of the second option, we don't know how to apply dimensionality reduction to a Voronoi cell in high dimensional space in order to visualize it in 2D.

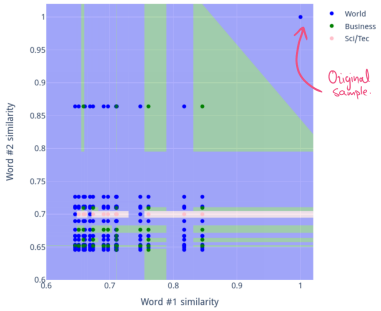


Figure 5: Decision region visualization in 2 dimensions.

4.6 Decision region visualization

Even though Voronoi tessellation in the dimensionality-reduced space may not preserve some desirable properties of decision region approximation (see 4.5), the result may still be useful for model comparisons. This is the decision region approximation and visualization method that we used in our experiments for surrogate comparisons.

At the core of the method is the Voronoi decision region approximation procedure as described in the algorithm 3. Two important details to be taken care of are: text to numerical vector mapping, and high-dimensional input handling.

Depending on the scenario, mapping text inputs to numerical vectors is done in one of two ways. When we want to visualize an arbitrary set of text samples, for example a subset of the training set, each text sample is mapped to the average of the GloVe embeddings of each word in the text sample. This way each sample corresponds to a vector in $\mathbb{R}^{100}$. In another scenario, we want to inspect a local surrounding of a single text sample. In this scenario we employ a text perturbation technique similar as described in subsection 4.4 figure 4 but not limited to 2 word perturbations. In contrast to the 2-dimensional perturbation variant, now we allow every word of the text sample to be replaced or removed. First, for each perturbation we generate a random binary mask of length M, where M is the number of words in the original text sample. The mask indicates which words of the text sample will be replaced or removed, and which words will stay unchanged. The words marked by the mask are either completely removed with some probability, or replaced with a similar (or not similar) word. Finally, the numerical representation of the perturbed text sample is a vector of length M, where the i-th element is: 1 if the i-th word was not perturbed; 0 if the i-th word was removed; and $1-\text{cosine\_similarity}(w_i^O, w_i^P)$ if the i-th word was replaced, where w_i^O is the embedding of the i-th word in the original text sample and w_i^P is the embedding of the i-th word in the perturbed text sample.

The high-dimensional inputs are handled by simply reducing their dimensions before passing them to the Voronoi tessellation. We are using t-SNE dimensionality reduction method to transform M-dimensional input vectors to 2-dimensional.

Figure 6 illustrates the decision region approximation and visualization method for the scenario of local decision landscape around a given text sample.

4.7 Surrogates

For the comparison experiments we used linear Support Vector Machines (SVM) and Logistic Regression as both local and global surrogates.

Text inputs were preprocessed into concatenations of one-hot-encoded vectors. Each such vector represents a word in the vocabulary. The vocabulary size V was kept as a hyperparameter to obtain models of varying complexity. The more complex surrogates were trained to use a larger vocabulary

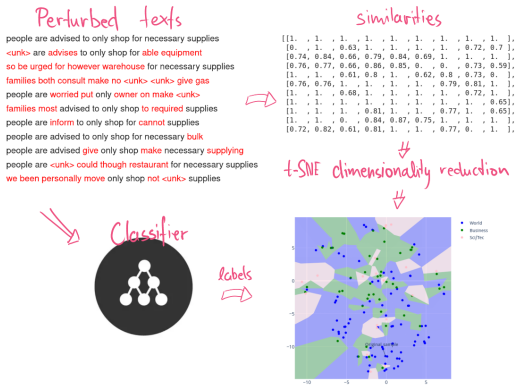


Figure 6: Decision region approximation and visualization method.
Top left: the original text sample and the generated perturbations.
Bottom left: original text sample together with perturbed texts are classified with the classifier to acquire predicted labels.
Top right: each text sample corresponds to a similarity vector.
Bottom right: final visualization by computing the Voronoi diagram on the dimensionality-reduced similarity vectors and coloring regions with the predicted class labels.

size (more features) for training and predictions, while simpler surrogates were limited to a smaller vocabulary (less features).

Both linear SVM and logistic regression models are interpretable. They have a coefficient for each input feature (a word from the vocabulary in our case) of every class. A coefficient specifies the importance of the corresponding word for the given class prediction.

Global surrogates were trained on the AG_NEWS dataset, taking black-box model predictions as labels. We used $C = 0.1$ for the inverse of the regularization strength. Several models were trained by varying vocabulary size hyperparameter $V \in [400, 35000]$. The training dataset for the local surrogates was generated for each input sample of interest by the pertubation strategy as described in subsection 4.6 and labeled by the black-box model. We used $C = 10$ for the inverse of the regularization strength. Several models were trained by varying the vocabulary size hyperparameter $V \in [60, 480]$. Figures 15, 16, 17, 18 contain accuracy and R^2 scores for global SVM, local SVM, global logistic regression, and local logistic regression surrogates respectively.

4.8 Decision region approximation for global surrogates

We visually compared our global surrogate model (linear SVM) with the black box model by highlighting the differences of predictions. In figure

7 we plot the decision regions (areas in the background) from the black box model using Voronoi tesselation. We use UMAP on the input points (averaged GloVe embeddings) to visualize model predictions in 2D and indicate the predicted class label by the point color. We highlight predicted data points in red, where the global surrogate prediction deviates from the black box model prediction. This visualization shows, that the prediction of the global surrogate especially deviates from the black box model near the approximated boundaries.

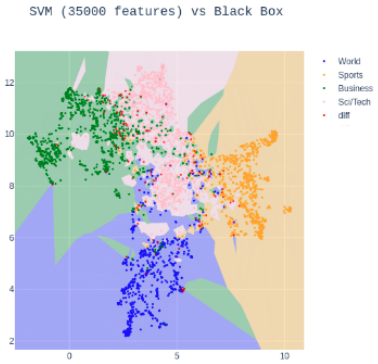


Figure 7: Global surrogate model vs black box model

Further comparisons with global surrogates of varying complexity also indicate, that global surrogate models approximate the black box model better with increasing model complexity of the global surrogate (see 13).

4.9 Surrogate decision region comparisons

We used the decision region approximation and visualization method (as described in subsection 4.6) to compare few surrogates.

Figure 8 shows the comparison of local and global SVM surrogates in a local surrounding of one input sample. In this case the original text sample is "In times of world cup pandemic, people are advised to only shop for necessary supplies". We generated perturbations of the sample and used them for producing all of the visualization plots in the figure. In figure 14 we apply a binary coloring scheme, where we distinguish the color of matching predictions against differing predictions between the surrogate models and the black box-model.

The experiment reveals a number of useful aspects of the visualization method. First, the global surrogate plot looks very different from the black box model plot, even though it has high accuracy and R^2 scores on the AG_NEWS validation dataset. Second, we can visually observe proportions of different classes. Third, for the local surrogates, we get a feeling of how the complexity of the models manifests in the complexity of the decision regions.

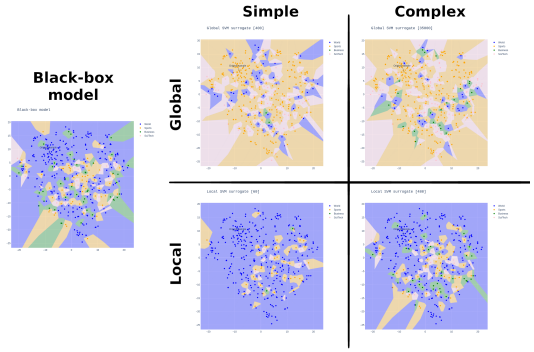


Figure 8: Global and local SVM surrogate comparison.
Left: decision region approximation of the original **black-box** model.
Right top row: decision region approximation of two **global** SVM surrogates (simple on the left, complex on the right).
Right bottom row: decision region approximation of two **local** SVM surrogates (simple on the left, complex on the right).

5 Conclusion

We introduced a new method for comparing local and global surrogates by visualizing decision boundaries.

Initially, we tried to find these decision boundaries using adversarial search, which showed that adversarial search exploits the weaknesses in the model and is not suited for outlining decision landscapes.

We came up with a method for approximating and visualizing decision boundaries combining Voronoi tessellation, text perturbation, and dimensionality reduction. We visually compared the decision region plots of different surrogates with different model complexities. Our implementation for decision boundary visualization is applicable to general text classification models and provides additional insights about the decision landscape complexity compared to a single-digit metric such as R^2. Furthermore, the surrogate comparisons confirm the intuition that our local surrogates are better at locally approximating the black-box model compared to the global surrogates.

References

AG's corpus of news articles. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html. [Online; accessed 02-February-2021].

Franz Aurenhammer. 1991. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405.

Stephan Günnemann. 2020. Machine Learning for Graphs and Sequential Data. Lecture (IN2323) at Technische Universität München.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. 2017. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer.

Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

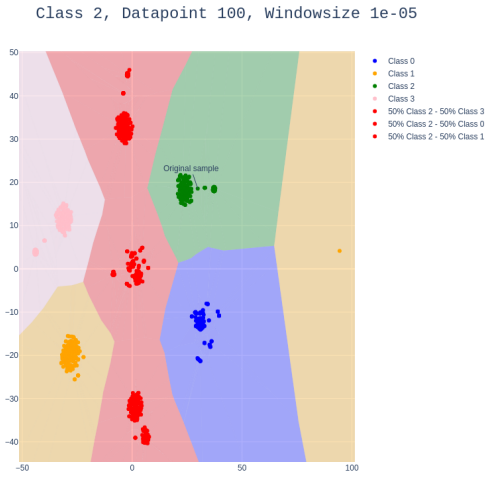
Sebastian Raschka. 2018. Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*, 3(24).

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier.

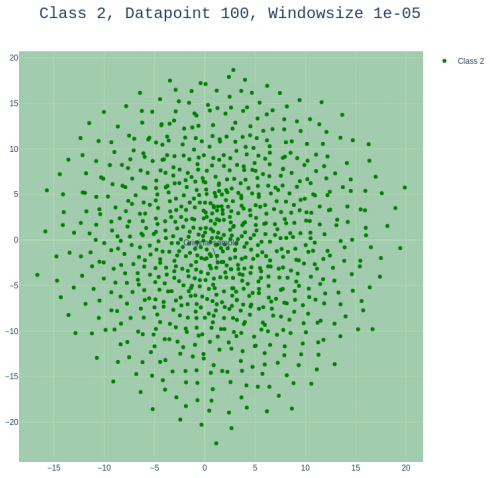
Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Georgios Vlassopoulos, Tim van Erven, Henry Brighton, and Vlado Menkovski. 2020. Explaining predictions by approximating the local decision boundary. *arXiv preprint arXiv:2006.07985*.

A Appendix

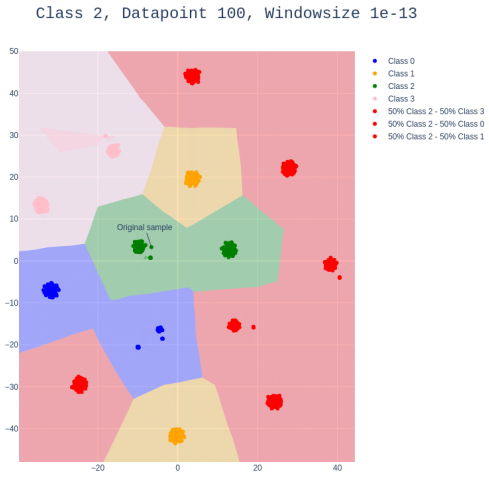


(a) Gradient Search

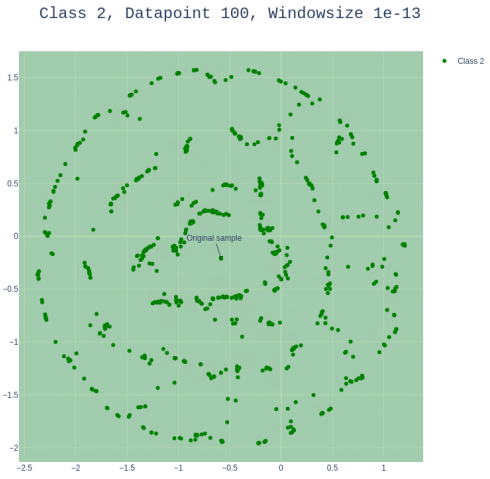


(b) Random Sampling

Figure 9: Gradient Search vs Random Sampling

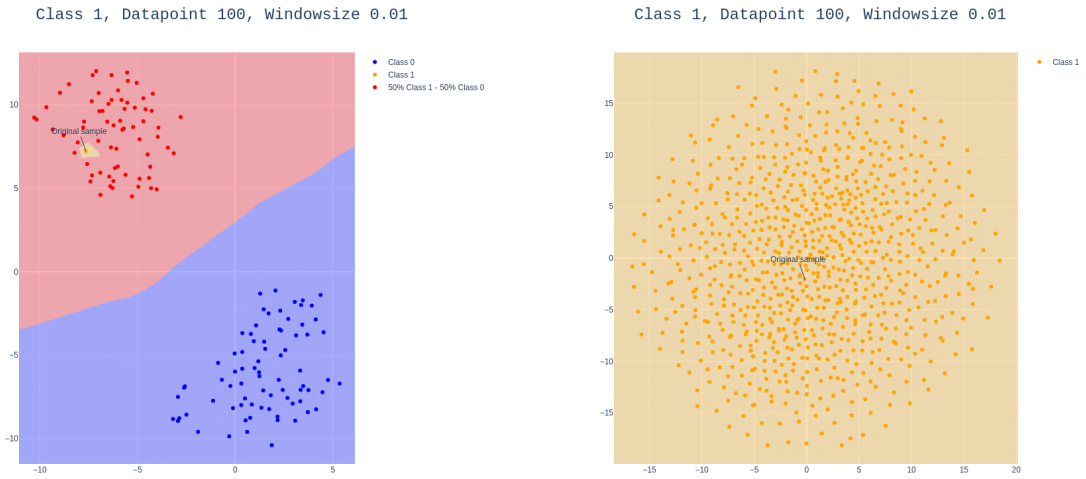


(a) Gradient Search



(b) Random Sampling

Figure 10: Gradient Search vs Random Sampling



(a) Gradient Search (b) Random Sampling

Figure 11: Gradient Search vs Random Sampling

(a) Gradient Search (b) Random Sampling

Figure 12: Gradient Search vs Random Sampling

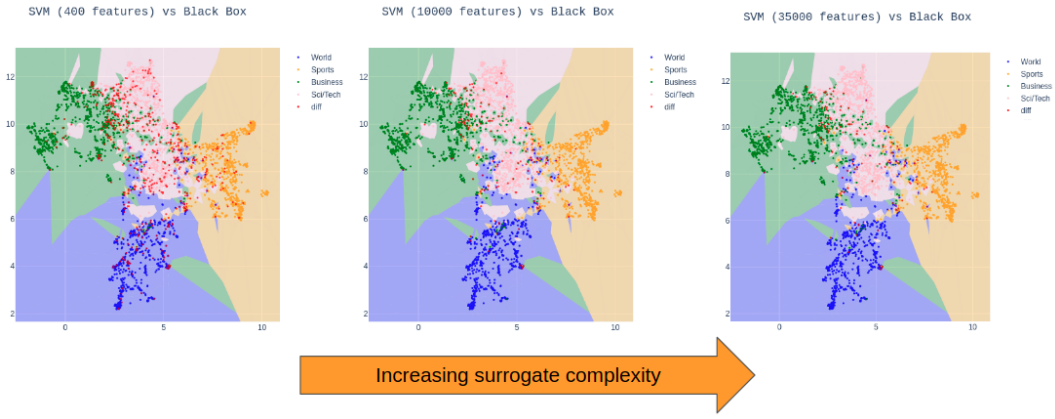


Figure 13: Global surrogate vs black box model for surrogate models of increasing complexity

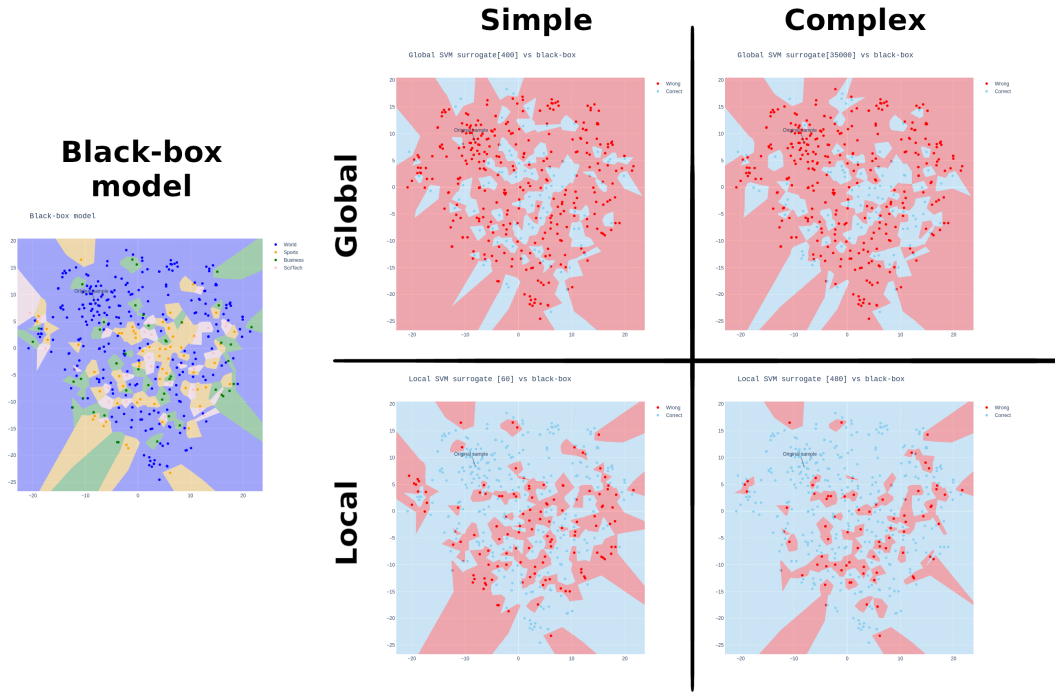


Figure 14: Global and local SVM surrogate comparison.
Left: decision region approximation of the original **black-box** model.
Right top row: decision region approximation of two **global** SVM surrogates (simple on the left, complex on the right).
Right bottom row: decision region approximation of two **local** SVM surrogates (simple on the left, complex on the right).
For the black-box model colors correspond to class labels. For the surrogate models red color indicates surrogate predictions different from black-box model, and blue color indicates predictions matching the black-box model.

Surrogate max features	Train acc	Val acc	Train R^2 score	Val R^2 score
400	0.836344	0.828042	0.581992	0.563991
2000	0.927677	0.914833	0.814351	0.776176
10000	0.964615	0.936125	0.916956	0.837201
35000	0.973844	0.940417	0.941095	0.846505

Figure 15: Accuracy and R^2 scores for global SVM surrogates of varying vocabulary size (max features column) on AG_NEWS datasets.

Surrogate max features	Train acc	Val acc	Train R^2 score	Val R^2 score
60	0.66975	0.672	-0.172182	-0.266986
120	0.71075	0.696	0.008753	-0.226727
240	0.83350	0.754	0.460689	0.036143
480	0.91500	0.826	0.703191	0.209022

Figure 16: Accuracy and R^2 scores for local SVM surrogates of varying vocabulary size (max features column) on perturbations around example text "In times of world cup pandemic, people are advised to only shop for necessary supplies".

Surrogate max features	Train acc	Val acc	Train R^2 score	Val R^2 score
400	0.834865	0.833917	0.572765	0.580339
2000	0.924156	0.915542	0.803204	0.786826
10000	0.953292	0.936042	0.884547	0.841450
35000	0.958937	0.936542	0.899887	0.843413

Figure 17: Accuracy and R^2 scores for global logistic regression surrogates of varying vocabulary size (max features column) on AG_NEWS datasets.

Surrogate max features	Train acc	Val acc	Train R^2 score	Val R^2 score
60	0.67650	0.684	-0.113768	-0.022989
120	0.71750	0.708	0.031316	-0.009040
240	0.82225	0.762	0.431662	0.125809
480	0.91675	0.820	0.697286	0.311807

Figure 18: Accuracy and R^2 scores for local logistic regression surrogates of varying vocabulary size (max features column) on perturbations around example text "In times of world cup pandemic, people are advised to only shop for necessary supplies".