MARGIN: Uncovering Deep Neural Networks using Graph Signal Analysis

Rushil Anirudh [*1] **Jayaraman J. Thiagarajan** [*1] **Rahul Sridhar** [2] **Peer-Timo Bremer** [1]

Abstract

Interpretability has emerged as a crucial aspect of machine learning, aimed at providing insights into the working of complex neural networks. However, existing solutions vary vastly based on the nature of the interpretability task, with each use case requiring substantial time and effort. This paper introduces MARGIN, a simple yet general approach to address a large set of interpretability tasks ranging from identifying prototypes to explaining image predictions. MARGIN exploits ideas rooted in graph signal analysis to determine influential nodes in a graph, which are defined as those nodes that maximally describe a function defined on the graph. By carefully defining task-specific graphs and functions, we demonstrate that MARGIN outperforms existing approaches in a number of disparate interpretability challenges.

1. Introduction

With widespread adoption of deep learning solutions in science and engineering, obtaining *a posteriori* interpretations of the learned models has emerged as a crucial research direction. This is driven by a community-wide effort to develop a new set of meta-techniques able to provide insights into complex neural network systems, and explain their training or predictions. Despite being identified as a key research direction, there exists no well-accepted definition for interpretability. Instead, in different contexts, it may refer to a variety of tasks ranging from debugging models (Ribeiro et al., 2016), to determining anomalies in the training data (Koh & Liang, 2017). While some recent efforts (Lipton, 2016; Doshi-Velez & Kim, 2017) provide a more formal definition for interpretability as generating *interpretable rules*, these focus on instance-level explanations, i.e. understanding how a network arrived at a particular decision for a single instance.

In practice, interpretability covers a wider range of challenges, such as characterizing data distributions and separating hyperplanes of classifiers, combating noisy labels during training, detecting adversarial attacks, or generating saliency maps for image classification. As discussed below, solutions to all such problems have been proposed each using custom tailored, task-specific approaches. For example, a variety of tools aim to explain which parts of an image are the most responsible for a prediction. However, these cannot be easily repurposed to identify which samples in a dataset were most helpful or harmful to train a classifier.

Instead, we introduce the MARGIN (**M**odel **A**nalysis and **R**easoning using **G**raph-based **In**terpretability) framework, which directly applies to a wide variety of interpretability tasks. MARGIN poses each task as an *hypothesis* test and derives a measure of *influence* that indicates which parts of the data/model maximally support (or contradict) the hypothesis. More specifically, for each task we construct a graph whose nodes represent entities of interest, and define a function on this graph that encodes a hypothesis. For example, if the task is to determine which labels need to be corrected in a dataset with corrupted labels, the domain is the set of samples, while the function can be local label agreement that measures how many neighbors have the same label as the current node. Using graph signal processing (Shuman et al., 2013; Sandryhaila & Moura, 2013) we then identify which samples are most important to describe the label agreement function, which turn out to be those with faulty labels as they introduce significant local variations in the function. Similarly, we can define other graphs and functions to address a number different tasks using the same procedure. This generic formulation, while extremely simple in its implementation, provides a powerful protocol to realize several meta-learning techniques, by allowing the user to incorporate rich semantic information, in a straightforward manner. In a nutshell, the proposed protocol is comprised of the following steps: (i) identifying the domain for interpretability (for e.g. intra-sample vs inter sample), (ii) constructing a neighborhood graph to model the domain, (for e.g. pixel space vs. latent space) (iii) defining an *explanation function* at the nodes of the graph, (iv) performing graph signal analysis to estimate the *influence* structure in the domain, and (v) creating interpretations based on the estimated influence structure. Figure 1 illustrates the steps involved in MARGIN for *a posteriori* interpretability.

Overview: Using different choices for graph construction

[*]Equal contribution [1]Lawrence Livermore National Laboratory, USA. [2]University of California Irvine, USA..

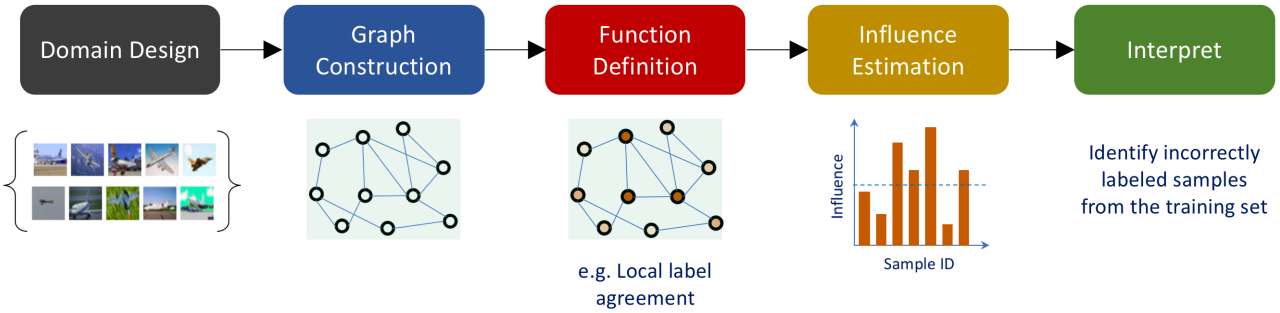


Figure 1. MARGIN - An overview of the proposed protocol for *a posteriori* interpretability tasks. In this illustration, we consider the problem of identifying incorrectly labeled samples from a given dataset. MARGIN identifies the most important samples that need to be corrected so that fixing them will lead to improved predictive models.

and the explanation function design, we present five case studies to demonstrate the broad applicability of MARGIN for *a posteriori* interpretability. First, in section 5.1 we study a unsupervised problem of identifying samples which well characterize the underlying data distribution, referred to as prototypes and criticisms respectively (Kim et al., 2016). We show that the MARGIN is better at identifying these candidates than state-of-the-art techniques. In section 5.2, we obtain localized image saliency at a pixel level using MARGIN, clearly explaining predictions from a black-box pre-trained model, and show that these strongly agree with techniques that even have access to the entire model. In section 5.3, we identify label corruptions in the training data, and show that MARGIN is able to identify these samples more effectively than recently proposed approaches, while also being able to explain the results intuitively. In section 5.4, we analyze decision surfaces of pre-trained classifiers by determining samples that are the most confusing to the model. Finally, in section 5.5 we extend two recently proposed statistical techniques to detect adversarial examples from harmless examples, and demonstrate that incorporating them inside MARGIN improves their discriminative power significantly.

2. Related Work

We outline recent works that are closely related to the central framework, and themes around MARGIN. Papers pertinent to individual case studies are identified in their respective sections.

Our goal in this paper is to identify a core framework that is capable of being repurposed to several interpretability tasks. This is related to two recent works – Fong et al. (Fong & Vedaldi, 2017) propose to perturb images in a way that they can be repurposed to several other tasks, of which interpretability is one. In (Koh & Liang, 2017), the authors proposed a strategy to select influential samples by extending ideas from robust statistics, which was shown to be applicable to a variety of scenarios. While these approaches are reasonably general, the proposed framework leverages the generality of graph structures, along with the ability to include arbitrary, semantically rich functions defined at each node. To the best of our knowledge, our work is the first to propose such a formulation.

The central idea of MARGIN is to use graph signal processing (GSP), to identify high frequency regions on graph signals. GSP itself is a relatively recent area, where there are two broad classes of approaches – one that builds on spectral graph theory using the graph Laplacian matrix (Shuman et al., 2013), and the other based on algebraic signal processing that builds upon the graph shift operator (Sandryhaila & Moura, 2013). While both are applicable to our framework, we adopt the latter formulation. Our approach relies on defining a measure of influence at each node, which is related to sampling of graph signals. This is an active research area, with several works generalizing ideas of sampling and interpolation to the domain of graphs, such as (Chen et al., 2015; Pesenson, 2008; Gadde et al., 2014). In many of these cases, the signal (or function) is assumed to be known, while one of our contributions is to identify the right function for a given interpretability task. In addition, our hypothesis on analyzing the high frequency content of the function is conceptually similar to (Chen et al., 2017) in being efficient, without requiring the need to solve any sophisticated optimization.

3. A Generic Protocol for Interpretability

In this section, we provide an overview of the different steps of MARGIN and describe the proposed influence estimation technique in the next section.

Domain Design and Graph Construction: The domain definition step is crucial for the generalization of MARGIN across different scenarios. In order to enable instance-level interpretations (e.g. creating saliency maps), a single instance of data, possibly along with its perturbed variants, will form the domain; whereas a more holistic understanding of the model can be obtained (e.g. extracting prototypes/crit-

Table 1. Using MARGIN to solve different commonly encountered interpretability tasks.

Task	Domain	Nodes in $\mathcal{G}$	Function	Explanation Modality
Prototypes/Criticisms	Complete dataset	Samples	MMD (Global, Local)	Sample sub-selection
Explain prediction	Single image	Explanations	Sparsity	Saliency maps
Detect noisy labels	Complete dataset	Samples	Local label agreement	Samples to fix
Characterize attacks	Attacks/Noisy samples	Perturbed samples	MMD (Global)	Attack statistics
Study discrimination	Complete dataset	Samples	Local label agreement	Confusing samples

icisms) by defining the entire dataset as the domain. Regardless of the choice of domain, we propose to model it using neighborhood graphs, as it enables a concise representation of the relationships between the samples.

More specifically, given the set of samples $\{\mathbf{x}_i\}$, we construct a k-nearest neighbor domain graph that captures local geometry of the data samples. The metric for graph construction (that determines neighborhoods/edges) can arise from prior knowledge about the domain or designed based on latent representations from pre-trained models. For example, if we use the latent features from AlexNet (Krizhevsky et al., 2012), the resulting graph respects the distance metric inferred by AlexNet for image classification. Though the difficulty in choosing an appropriate k for designing robust graphs is well known, designing better graphs is beyond the scope of this paper. In our experiments, we find that our results are not very sensitive to the choice of k.

Formally, an undirected weighted graph is represented by the triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ denotes the set of nodes, $\mathcal{E}$ denotes the set of edges and $\mathbf{W}$ is an adjacency matrix that specifies the weights on the edges, where $\mathbf{W}_{n,m}$ corresponds to the edge weight between nodes v_n and v_m. Let $\mathcal{N}_n = \{m | \mathbf{W}_{n,m} \neq 0\}$ define the neighborhood of node v_n, i.e. the set of nodes connected to it. The normalized graph Laplacian, $\mathbf{L}$, is then constructed as $\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, where $\mathbf{D}_{nn} = \sum_m \mathbf{W}_{n,m}$ is the degree matrix and $\mathbf{I}$ denotes the identity matrix.

Explanation Function Definition: A key component of MARGIN is to construct an explanation function that measures how well each node in the graph supports the presented hypothesis. Let us illustrate this process with an example – in order to create saliency maps for image classification, one can build a graph where each node corresponds to a potential explanation (i.e. a subset of pixels), while the edges can measure how likely can two explanations produce similar predictions. In such a scenario, one can hypothesize that an *ideal* explanation will be *sparse*, in terms of the number of pixels, since that is more interpretable. Consequently, the size of an explanation can be used as the function. Table 1 shows the domain design, graph construction, and function definition choices made for different use cases. Section 5 will present a more detailed discussion.

Influence Estimation: This is the central analysis step in MARGIN for obtaining influence estimates at the nodes of $\mathcal{G}$, that can reveal which nodes can maximally describe the variations in the chosen explanation function. Implicitly, this step can be viewed as a *soft*-sample selection strategy with respect to the structure induced by the domain graph. We propose to perform this estimation using tools from graph signal analysis. Section 4 describes the proposed algorithm for influence estimation.

From Influence to Interpretation: Depending on the hypothesis chosen for *a posteriori* analysis, this step requires the design of an appropriate strategy for transferring the estimated influences into an interpretable explanation.

4. Proposed Influence Estimation

Given a neighborhood graph $\mathcal{G}$ along with an explanation function $\mathbf{f}$, we propose to employ graph signal analysis to estimate node influence scores. Before we describe the algorithm, we will present a brief overview of the preliminaries.

Definitions: We use the notation and terminology from (Sandryhaila & Moura, 2013) in defining an operator analogous to the *time-shift* or *delay* operator in classical signal processing. During a graph shift operation, the function $\mathbf{f}(n)$ at node v_n is replaced by a weighted linear combination of its neighbors: $\hat{\mathbf{f}} = \mathbf{A}\mathbf{f}$, where $\mathbf{A}$ is the graph shift operator, which is the simplest, non-trivial graph filter. Commonly used choices for $\mathbf{A}$ include the adjacency matrix $\mathbf{W}$, transition matrix $\mathbf{D}^{-1}\mathbf{W}$ and the graph Laplacian $\mathbf{L}$.

The set of eigenvectors of the graph shift operator is referred to as the graph Fourier basis, $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U} \in \mathbb{R}^{N \times N}$, and the Fourier transform of a signal $\mathbf{f} \in \mathbb{R}^N$ is defined as $\mathbf{U}^T\mathbf{f}$. The ordered eigenvalues corresponding to these eigenvectors represent frequencies of the signal, with λ_1 to λ_N representing the smallest to largest frequencies. The notion of frequency on the graph corresponds to the rate of change of the function across nodes in a neighborhood. A higher change corresponds to a high frequency, while a smooth variation corresponds to a low frequency. In this context, the graph filtering using a graph shift operator corresponds to a *low-pass* filter that dispenses high frequency components in the function. Similarly, a simple *high-pass* filter can be easily designed as $\hat{\mathbf{f}_h} = \mathbf{f} - \hat{\mathbf{f}}$.

Algorithm: The overall procedure to obtain influence

Algorithm 1 Influence Estimation

Input: Domain – $\mathbf{X}$, Graph – $G = (\mathcal{V}, \mathcal{E})$ and the explanation function $\mathbf{f}$ defined at the nodes of G

Output: Influence estimate at each node, $I(i), \forall i \in \mathcal{V}$

Construct graph shift operator $\mathbf{A}$ from $\mathbf{X}$ $\hat{\mathbf{f}} = \mathbf{f} - \mathbf{Af}$
foreach $i \in \mathcal{V}$ **do**
  compute $I(i) = ||\hat{\mathbf{f}}(i)||_2^2 \ \forall i \in \mathcal{V}$.
end

scores at the nodes of $\mathcal{G}$ can be found in Algorithm 1. Intuitively, we design a high-pass filter that eliminates the low frequency content and retains the signal energy only at those nodes that characterize the extreme variations of the function. Following the high-pass filtering step, the influence score at a node is estimated as the magnitude of the filtered function value at that node:

$$I(i) = ||\hat{\mathbf{f}_h}(i)||_2^2 \ \forall i \in \mathcal{V}, \tag{1}$$

where $\hat{\mathbf{f}_h}$ corresponds to the high-pass filtered version of $\mathbf{f}$. Interestingly, we find that analyzing the high frequency components of the explanation function often leads to a sparse influence structure, indicating the presence of multiple local optima that corroborate the hypothesis. Conversely, the influence structure obtained from low frequency components is typically dense and hence requires additional processing to qualify regions of disagreement.

5. Case Studies

5.1. Case Study I - Prototypes and Criticisms

A commonly encountered problem in interpretability is to identify samples that are prototypical of a dataset, and those that are statistically different from the prototypes (called criticisms). Together, they can provide a holistic understanding about the underlying data distribution. Even in cases where we do not have access to the label information, we seek a hypothesis that can pick samples which are representatives of their local neighborhood, while emphasizing statistically anomalous samples. One such function was recently utilized in (Kim et al., 2016) to define prototypes and criticisms, and it was based on Maximum Mean Discrepancy (MMD).

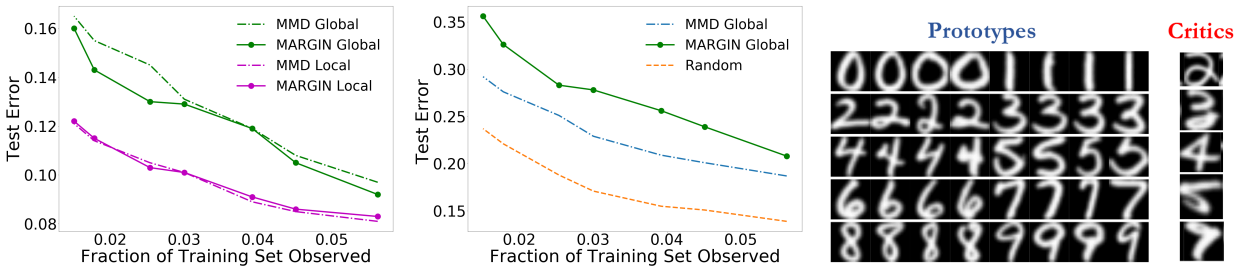
Formulation: Following the general protocol in Figure 1, the domain is defined as the complete dataset, along with labels if available. Since this analysis does not rely on pre-trained models, we construct the neighborhood graph based on conventional metrics, e.g. Euclidean distance. Inspired by (Kim et al., 2016), we define the following explanation function: For each sample $\mathbf{x}_i$, we remove the chosen sample and all its connected neighbors from the graph to construct the set $\bar{\mathcal{X}} = \{\mathbf{x}_j, j \notin (i \cup \mathcal{N}_i)\}$, and estimate the function at the i^{th} node as $f(i) = \text{MMD}(\bar{\mathcal{X}}, \bar{\mathcal{X}} \cup \mathbf{x}_i)$. In cases of labeled datasets, the kernel density estimates for the MMD computation are obtained using only samples belonging to the same class. We refer to these two cases as *global* (unlabeled case) and *local* (labeled case) respectively. The hypothesis is that the regions of criticisms will tend to produce highly varying MMD scores, thereby producing high frequency content, and hence will be associated with high MARGIN scores. Conversely, we find that the samples with low MARGIN scores correspond to prototypes since they lie in regions of strong agreement of MMD scores. More specifically, we consider all samples with low MARGIN scores (within a threshold) as prototypes, and rank them by their actual function values. In contrast to the greedy inference approach in (Kim et al., 2016) that estimates prototypes and criticisms separately, they are inferred jointly in our case.

Experiment Setup and Results: We evaluate the effectiveness of the chosen samples through predictive modeling experiments. We use the USPS handwritten digits data for this experiment, which consists of 9,298 images belonging to 10 classes. We use a standard train/test split for this dataset, with 7,291 training samples and the rest for testing. For fair comparisons with (Kim et al., 2016), we use a simple 1-nearest neighbor classifier. As described earlier, we consider both unsupervised (*global*) and supervised (*local*) variants of our explanation function for sample selection.

We expect the prototypical samples to be the most helpful in predictive modeling, i.e., good generalization. In Figure 2(a), we observe that the prototypes from MARGIN perform competitively in comparison to the baseline technique. More importantly, MARGIN is particularly superior in the global case, with no access to label information. On the other hand, criticisms are expected to be the least helpful for generalization, since they often comprise boundary cases, outliers and under-sampled regions in space. Hence, we evaluate the test error using the criticisms as training data. Interestingly, as shown in Figure 2(b), the criticisms from MARGIN achieve significantly higher test errors in comparison to samples identified using *MMD-critic* based optimization in (Kim et al., 2016). Furthermore, examples of the selected prototypes and criticisms from MARGIN are included in Figure 2(c).

5.2. Case Study II - Explanations for Image Classification

Generating explanations for predictions is crucial to debugging black-box models and eventually building trust. Given a model, such as a deep neural network, that is designed to classify an image into one of r classes, a plausible *explanation* for a test prediction is to quantify the importance of different image regions to the overall prediction, i.e. produce a saliency map. We posit that perturbing the salient regions should result in maximal changes to the prediction.



(a) Training with prototypes. (b) Training with criticisms. (c) Selected Samples

Figure 2. Using MARGIN to sample prototypes and criticisms. In this experiment, we study the generalization behavior of models trained solely using prototypes or criticisms.

In addition, we expect *sparse* explanations to be more interpretable. In this section, we describe how MARGIN can be applied to achieve both these objectives.

Formulation: Since we are interested in producing explanations for instance-level predictions using MARGIN, the domain corresponds to a possible set of explanations for an image. Note that, the space of explanations can be combinatorially large, and hence we adopt the following greedy approach to construct the domain. We run the SLIC algorithm (Achanta et al., 2012) with varying number of superpixels, say $\{50, 100, 150, 200, 250, 300\}$, and define the domain as the union of superpixels from all the independent runs. In our setup, each of these superpixels is a plausible explanation and they become the nodes of $\mathcal{G}$.

Assuming that a test image $\mathcal{I}$ is assigned the class j with softmax probability $p(j)$, for each of the explanations i, we mask those pixels in the image and use the pre-trained model to obtain the softmax probability $\hat{p}_i(j)$ and measure its saliency as $|p(j) - \hat{p}_i(j)|$. Using these estimates, we obtain pixel-level saliency, S, as a weighted combination of their saliency from different superpixels (inversely weighted by the superpixel size). This dense saliency is similar to previous approaches such as (Zeiler & Fergus, 2014; Zhou et al., 2014).

Note that, this saliency estimation process did not impose the sparsity requirement. Hence, we use MARGIN to obtain influence scores based on their sparseness. To this end, we construct neighborhoods for explanations based on their impact on the predictions, i.e. edges are computed based on their $|p(j) - \hat{p}_i(j)|$ values. The explanation function at each node is defined as the ratio of the size of the superpixel corresponding to that node and the size of the largest superpixel in the graph. Intuitively, MARGIN finds the sparsest explanation for different level sets of the saliency function, $|p(j) - \hat{p}_i(j)|$. Subsequently, we compute pixel-level influence scores, I, as a weighted combination of their influences from different superpixels. The overall saliency map is obtained as $S_{final} = S \odot I$, where $\odot$ refers to the Hadamard product.

Experiment Setup and Results: Using images from the ImageNet database (Russakovsky et al., 2015), and the AlexNet (Krizhevsky et al., 2012) model, we demonstrate that MARGIN can effectively produce explanations for the classification. Figure 3 illustrates the process of obtaining the final saliency map for an image from the *Tabby Cat* class. Interestingly, we see that the mouth and whiskers are highlighted as the most salient regions for its prediction. Figure 4 shows the saliency maps from MARGIN for several other cases. For comparison, we show results from Grad-CAM (Selvaraju et al., 2017), which is a white-box approach that accesses the gradients in the network. We find that, using only a black-box approach, MARGIN produces explanations that strongly corroborate with Grad-CAM and in some cases produces more interpretable explanations. For example, in the case of an *Ice Cream* image, MARGIN identifies the ice cream, and the spoon, as salient regions, while Grad-CAM highlights only the ice cream and quite a few background regions as salient. Similarly, in the case of a *fountain* image, MARGIN highlights the fountain, and the sky, while Grad-CAM highlights the background (trees) slightly more than the fountain itself, which is not readily interpretible.

5.3. Case Study III - Detecting Incorrectly Labeled Samples

An increasingly important problem in real-world applications is concerned with the quality of labels in supervisory tasks. Since the presence of noisy labels can impact model learning, recent approaches attempt to compensate by perturbing the labels of samples that are determined to be high-risk of being corrupted, or when possible have annotators check the labels of those high-risk samples. In this section, we propose to employ MARGIN to recover incorrectly labeled samples. In particular, we consider a binary classification task, where we assume $\beta\%$ of the labels are randomly flipped in each class. In order to identify samples which were incorrectly labeled, we select samples with the highest MARGIN score, followed by simulating a human user correcting the labels for the top K samples. Ideally, we would

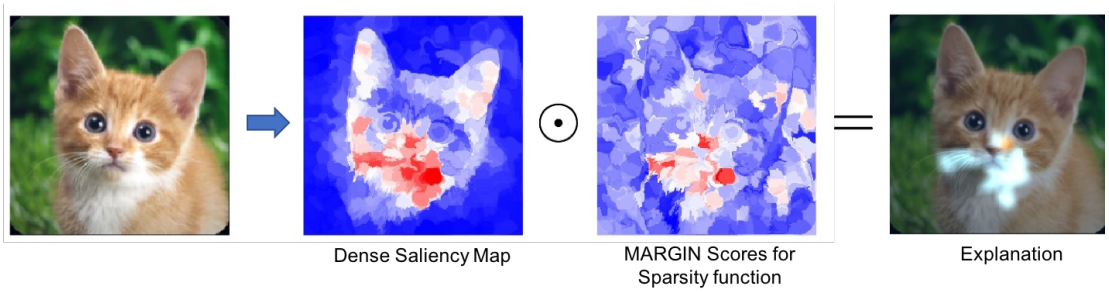


Figure 3. We show the entire process of constructing the saliency map for one particular image (Tabby Cat) from ImageNet. From left to right: original image, (dense) saliency map S, sparsity map I, and finally the explanation from MARGIN, S_{final}.

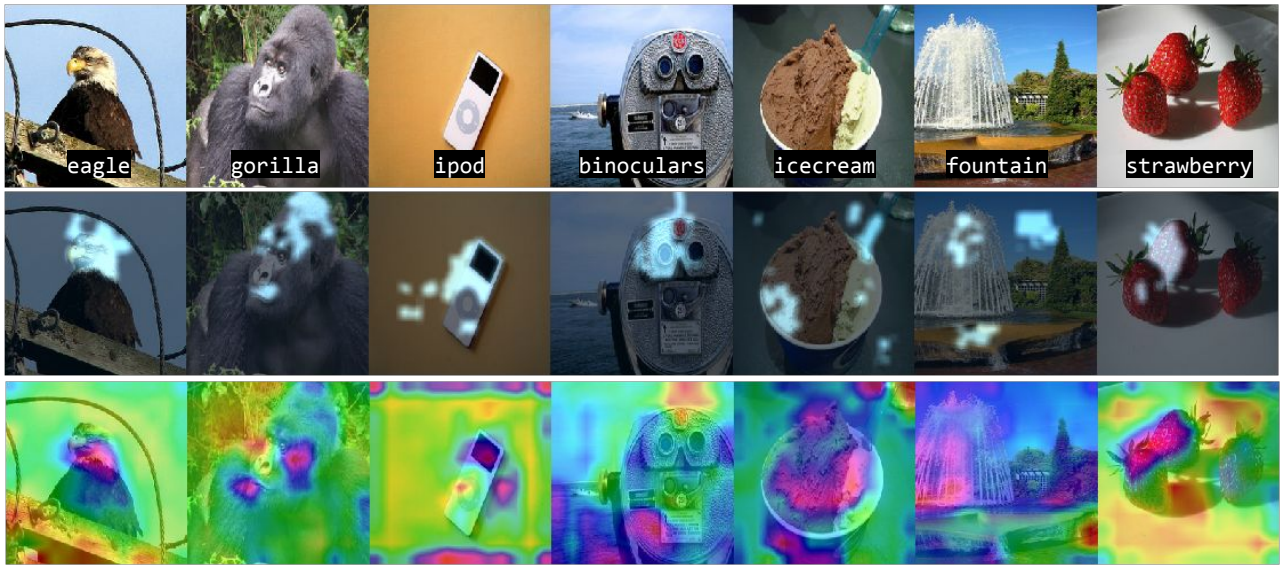


Figure 4. Our approach identifies the most salient regions in different classes for image classification using AlexNet. From top to bottom: original image, MARGIN's explanation overlaid on the image, and Grad-CAM's explanation. Note our approach yields highly specific, and sparse explanations from different regions in the image for a given class.

like K, the number of samples checked by the user, to be as small as possible.

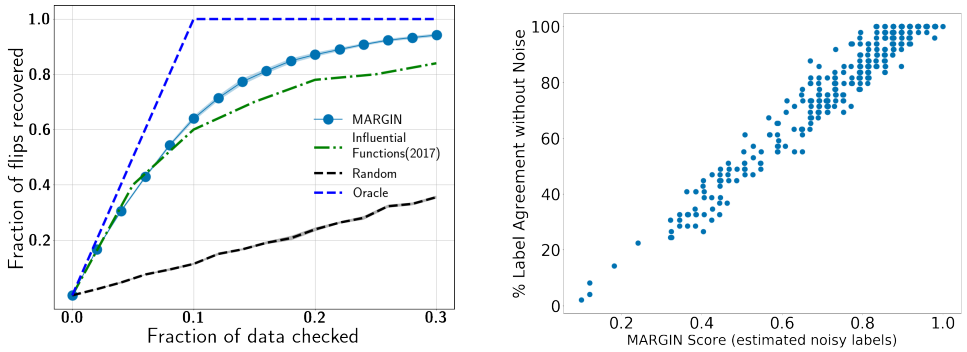
Formulation: Similar to Case Study I, the entire dataset is used to define the domain and a user-defined metric is used to construct the graph. Since we expect the flips to be random, we hypothesize that they will occur in regions where the labels of corrupted samples are different from their neighbors. Instead of directly using the label at each node as the explanation function, we believe a more smoothly varying function will allow us to extract regions of high frequency changes more robustly. As a result, we propose to measure the level of *distrust* at a given node, by measuring how many of its neighbors disagree with its label:

$$\mathbf{f}(i) = 1 - \frac{\sum_{j \in \mathcal{N}_i} L(j,i)}{|\mathcal{N}_i|}, \tag{2}$$

where $L(j,i)$ is 1 only if nodes j and i share the same label; $|.|$ denotes the cardinality of a set.

Experiment Setup and Results: We perform our experiments on the Enron Spam Classification dataset (Metsis et al., 2006), containing 4138 training examples, with an imbalanced class split of around 70:30 (non-spam:spam). Following standard practice, we randomly corrupt the labels of 10% of the samples. For the Enron Spam dataset, we extracted bag-of-words features of 500 dimensions corresponding to the most frequently occurring words. These features are then used to construct a k-NN graph with the number of neighbors k fixed at 20, and we report average results from 10 repetitions of the experiment. We compare our approach with three baselines: *(i) Influence Functions:* We obtain the most influential samples using Influence Functions (Koh & Liang, 2017). *(ii) Random Sampling (iii) Oracle:* The best case scenario, where the number of labels corrected is equal to the number of samples observed. Following (Koh & Liang, 2017), we vary the percentage of influential samples chosen, and compute the *recall* measure, which corresponds to the fraction of label flips recovered in the chosen subset of samples.

As seen in Figure 5(a), we see that our method is nearly 10



(a) Detecting label flips in the Enron dataset (Metsis et al., 2006).

(b) Examining the incorrectly labeled samples with their influence score.

Figure 5. Detecting incorrectly labeled samples using MARGIN.

percentage points better than the state-of-the-art Influence Functions, achieving a recall of nearly 0.95 by observing just 30% of the samples. In Figure 5(b), we study how MARGIN scores the incorrectly labeled samples. On the y-axis, we show the percentage of the neighbors that agree with the original label (if there was no corruption) – this is a proxy measure to identify which samples lie closer to the classification boundary vs the ones that are farther away. The x-axis shows the MARGIN score, and we see a clear trend, which indicates a strong preference for samples that lie farther away from the classification boundary. In other words, this corresponds strongly to correcting the least number of samples which can lead to the most gain in validation performance when using a trained model.

5.4. Case Study IV - Interpreting Decision Boundaries

While studying black-box models, it is crucial to obtain a holistic understanding of their strengths, and more importantly, their weaknesses. Conventionally, this has been carried out by characterizing the decision surfaces of the resulting classifiers. In this experiment, we demonstrate how MARGIN can be utilized to identify samples that are the most confusing to a model.

Formulation: In order to adopt MARGIN for analyzing a specific model, we construct the graph using latent representations inferred from the model. Since decision surface characterization is similar to Case Study III, we use the local label agreement measure in (2) as the explanation function.

Experiment Setup and Results: We perform an experiment on 2-class datasets extracted from ImageNet and MNIST. More specifically, in the case of ImageNet, we perform decision surface characterization on the classes *Tabby Cat* and *Great Dane*. We used the features from a pre-trained AlexNet's penultimate layer to construct the graph. For the MNIST dataset, we considered data samples from digits '0' and '6', and we used the latent space produced using a convolutional neural network for the analysis. A selected subset of samples characterizing the decision surfaces of both datasets are shown in Figure 6. For ImageNet, it is clear that the model gets confused whenever the animal's face is not visible, or if it is in a contorted position, or occluded. Similarly, in the MNIST dataset, the examples shown depict atypical ways in which the digits '0' and '6' can be written.

5.5. Case Study V - Characterizing Statistics of Adversarial Examples

In this application, we examine the problem of quantifying the statistical properties of adversarial examples using MARGIN. Adversarial samples (Biggio et al., 2013; Szegedy et al., 2013) refer to examples that have been specially crafted, such that a particular trained model is 'tricked' into misclassifying them. This is done typically by perturbing a sample, sometimes in ways imperceptible to humans, while maximizing misclassification rates. In order to better understand the behaviour of such adversarial examples, there have been studies in the past to show that adversarial examples are statistically different from normal test examples. For example, an MMD score between distributions is proposed in (Grosse et al., 2017), and a kernel density estimator (KDE) in (Feinman et al., 2017). However, these measures are global, and provide little insight into individual samples. We propose to use MARGIN to develop these statistical measures at a sample level, and study how individual adversarial samples differ from regular samples.

Formulation: As in other case studies, MARGIN constructs a graph, where each node corresponds to an example that is either adversarial or harmless, and the edges are constructed using neighbors in the latent space of the model, against which the adversarial examples have been designed. We consider two kinds of functions in this experiment: i) **MMD Global:** Similar to 5.1, we use the MMD score between the whole set, and the set without a particular sample and its neighbors. This provides a way to capture statistically rarer samples in the dataset; ii) **KDE:** We also use the

(a) Most confusing samples for AlexNet pre-trained on ImageNet for the *Tabby Cat* and *Great Dane* classes

(b) Most confusing samples for a CNN trained on MNIST (for the 0/6 classes)

Figure 6. Using MARGIN to sample near decision boundaries.

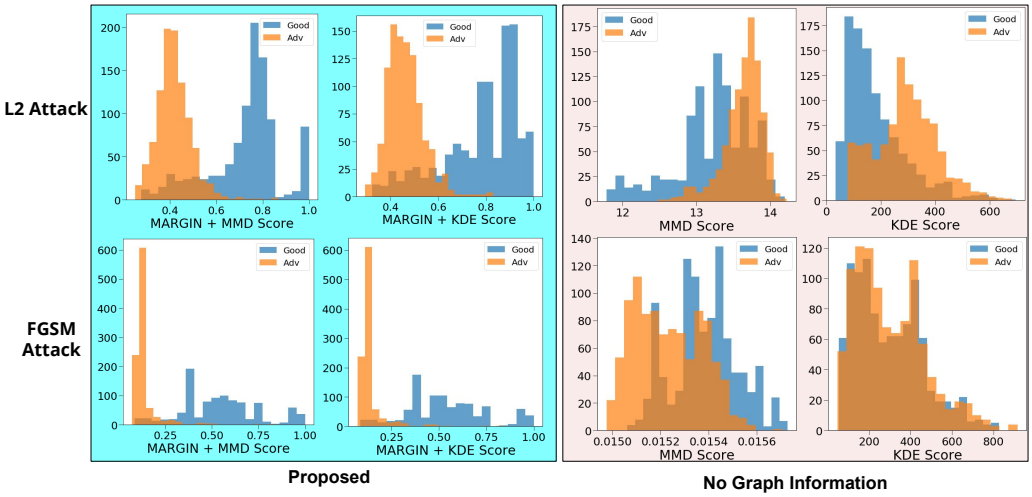


Figure 7. A comparison of statistical scores to identify adversarial samples with and without incorporating graph structure. We see that including the structure results in a much better separation between adversarial and harmless examples. In addition, regions of overlap can easily be explained.

KDE of each sample, as proposed in (Feinman et al., 2017), where we measure the discrepancy of each sample against the training samples from its predicted class. While these measures on their own may not be very illustrative, they are useful functions to determine influences within MARGIN.

Experiment Setup and Results: We perform experiments on 2000 randomly sampled test images from the MNIST dataset (LeCun, 1998), of which we adversarially perturb 1000 images. We measure MARGIN scores using both MMD Global, and KDE, against two popular attacks – the Fast Gradient Sign Method (FGSM) attack (Goodfellow et al., 2014), and the L2-attack (Carlini & Wagner, 2017b). We use the same setup as in (Carlini & Wagner, 2017a), including the network architecture for MNIST. The resulting MARGIN score determined using algorithm 1 is more discriminative, as seen in Figure 7. As noted in (Carlini & Wagner, 2017a), the MMD and KDE measures were not very effective against stronger attacks such as the L2-attack. This is reflected to a much lower degree even in our approach, where there is a small overlap in the distributions. We also find that the overlapping regions correspond to samples from the training set that are extremely rare to begin with (like criticisms from section 5.1).

6. Conclusions

We proposed a generic framework called MARGIN that is able to provide explanations to popular interpretability tasks in machine learning. These range from identifying prototypical samples in a dataset that might be most helpful for training, to explaining salient regions in an image for classification. In this regard, MARGIN exploits ideas rooted in graph signal processing to identify the most influential nodes in a graph, which are nodes that maximally affect the graph function. While the framework is extremely simple, it is highly general in that it allows a practitioner to include rich semantic information easily in three crucial ways – defining the domain (intra-sample vs inter-sample), edges (pre-defined/native/model latent space), and finally a function defined at each node. The graph based analysis easily scales to very sparse graphs with tens of thousands of

nodes, and opens up several opportunities to study problems in interpretable machine learning.

Acknowledgement

This work was performed under the auspices of the U.S. Dept. of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

Achanta, Radhakrishna, Shaji, Appu, Smith, Kevin, Lucchi, Aurelien, Pascal, Fua, and Susstrunk, Sabine. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:2274–2282, 2012.

Biggio, Battista, Corona, Igino, Maiorca, Davide, Nelson, Blaine, Šrndić, Nedim, Laskov, Pavel, Giacinto, Giorgio, and Roli, Fabio. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Carlini, Nicholas and Wagner, David. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017a.

Carlini, Nicholas and Wagner, David. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017b.

Chen, Siheng, Varma, Rohan, Sandryhaila, Aliaksei, and Kovačević, Jelena. Discrete signal processing on graphs: Sampling theory. *IEEE transactions on signal processing*, 63(24):6510–6523, 2015.

Chen, Siheng, Tian, Dong, Feng, Chen, Vetro, Anthony, and Kovačević, Jelena. Fast resampling of 3d point clouds via graphs. *arXiv preprint arXiv:1702.06397*, 2017.

Doshi-Velez, Finale and Kim, Been. A roadmap for a rigorous science of interpretability. *arXiv preprint arXiv:1702.08608*, 2017.

Feinman, Reuben, Curtin, Ryan R, Shintre, Saurabh, and Gardner, Andrew B. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Fong, Ruth and Vedaldi, Andrea. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*, 2017.

Gadde, Akshay, Anis, Aamir, and Ortega, Antonio. Active semi-supervised learning using sampling theory for graph signals. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 492–501. ACM, 2014.

Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Grosse, Kathrin, Manoharan, Praveen, Papernot, Nicolas, Backes, Michael, and McDaniel, Patrick. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Kim, Been, Khanna, Rajiv, and Koyejo, Oluwasanmi O. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems(NIPS).*, pp. 2280–2288, 2016.

Koh, Pang Wei and Liang, Percy. Understanding black-box predictions via influence functions. *International Conference on Machine Learning (ICML).*, 2017.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Yann. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Lipton, Zachary C. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

Metsis, Vangelis, Androutsopoulos, Ion, and Paliouras, Georgios. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pp. 28–69, 2006.

Pesenson, Isaac. Sampling in paley-wiener spaces on combinatorial graphs. *Transactions of the American Mathematical Society*, 360(10):5603–5627, 2008.

Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Sandryhaila, Aliaksei and Moura, José MF. Discrete signal processing on graphs. *IEEE transactions on signal processing*, 61(7):1644–1656, 2013.

Selvaraju, Ramprasaath, Cogswell, Michael, Das, Abhishek, Vedantam, Ramakrishna, Parikh, Devi, and Batra, Dhruv. Grad-cam: Visual explanations from deep networks via gradient-based localization. *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Shuman, David I, Narang, Sunil K, Frossard, Pascal, Ortega, Antonio, and Vandergheynst, Pierre. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Zhou, Bolei, Khosla, Aditya, Lapedriza, Agata, Oliva, Aude, and Torralba, Antonio. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.