MODEL AGNOSTIC LOCAL INTERPRETABLE METHODS AND AN EMPIRICAL NOTION OF ROBUSTNESS

By

ROHIT SINGH

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

CHENNAI MATHEMATICAL INSTITUTE
Computer Science

JULY 2020

To the Faculty of Chennai Mathematical Institute:

The members of the Committee appointed to examine the thesis of ROHIT SINGH find it satisfactory and recommend that it be accepted.

Madhavan Mukund, Prof., Chair

K. V. Subrahmanyam, Prof.

B Srivathsan, Prof.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisors Prof. Madhavan Mukund and Prof. K. V. Subrahmanyam for the continuous support of my master's thesis and research, for their patience, motivation, enthusiasm, and immense knowledge. The detailed discussions we had and their useful insights helped me in all the time of research and writing of this thesis.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

MODEL AGNOSTIC LOCAL INTERPRETABLE METHODS

AND AN EMPIRICAL NOTION OF

ROBUSTNESS

Abstract

by Rohit Singh, Masters
Chennai Mathematical Institute
July 2020

: Madhavan Mukund

Recently there has been a leap in leveraging the machine learning models in every aspect of life. In most of the cases, it's very hard to comprehend how the model is behaving even for a data scientist. This trend of taking the model as an oracle can be problematic when it is used in high stakes decisions making (where the cost of a wrong decision is high). In this manuscript, we discuss the techniques that can be used to make this oracle more explainable. In the second part, we discuss the robustness of these explanations and provide a metric to quantify the same.

TABLE OF CONTENTS

Chapter One

INTRODUCTION

The trend of using machine learning can be seen in all domains, whether it is corporate, finance, criminal justice, healthcare etc. These models tend to have high accuracy w.r.t the task at hand, but sometimes we need more than mere accuracy as we will see in the following scenarios:

1) Consider the task where we are required to make high stakes decisions using machine learning like in criminal recidivism prediction or the use of an uncertain (but potential life saving) drug on a patient. Now if are using machine learning models for prediction in these tasks we want more than just good accuracy, we want some notion of explanation for the decision made by the model.

If the model is predicting the cancellation of bail to a person we want to make sure that it is not biased towards a particular race (This would be easy to judge if the explanation of model is provided).

If the model is highly confident that the drug will work on a patient we (or the domain expert) would like to know what parameters (patient data) make the model predict this. This again can be achieved if we have models explanation at hand. So, there can be many tasks similar to the one described above where we the goal is a combination of explanation and good accuracy.

2) Machine learning models especially neural networks, though standing on strong mathematical foundation, become too complex to be comprehended by humans when working in high dimensionality data space, simply because of the large number of parameters.
In the face of such obscurity, instead of concluding that there is no way to understand the behaviour of such models, we can use explanation as a tool to understand some aspect of it. Thus, from the research point of view, generating the explanation can itself become a motivational goal.

The first part of this thesis discusses different ways to classify explanation methods, though we will limit ourself to a particular type of explanation methodology called local interpretable methods. The reason for limiting ourself to only this approach will be made clear later.
The second part explores an important aspect of the explanation of models in question, robustness. It is worth noting that as the explanation itself is subjective, the notion of robustness might vary from person to person, thus at the outset, we will define this notion more concretely.

Chapter Two

BACKGROUND

2.1 Interpretability/ Explainability

Quoting from Wikipedia, "An **explanation** is a set of statements usually constructed to describe a set of facts which clarifies the causes, context, and consequences of those facts." and from wiktionary, "an **interpretation** is an act of interpreting or explaining what is obscure."
There is only a subtle difference between these definitions, We won't go into the details and will use these terms interchangeably throughout this thesis.

People have approached explainability from many directions, forming them as a different way of explaining the model. The following ways of grouping interpretability will provide sufficient background to understand this thesis.

2.1.1 Intrinsic vs Post-hoc interpretability

. Intrinsic interpretability is achieved by constructing self-explanatory models which incorporate interpretability directly to their structures. The families in this category include decision trees, rule-based models, linear models, attention models, and so on. In contrast, post-hoc interpretability requires creating a second model to provide explanations for an

existing model.

The main difference between these two approaches lies in the trade-off between model accuracy and explanation fidelity.

Inherently interpretable models can provide an accurate and undistorted explanation but may sacrifice prediction performance to some extent. The post-hoc ones are limited by their approximate nature while keeping the underlying model accuracy intact.

2.1.2 Global vs Local interpretability

. Techniques for interpretability can be further classified as, global and local interpretability. As the name describes, global interpretability explains the global behaviour of the model in terms of its structure and parameters, whereas local interpretability explains the model's reasoning for individual predictions.

In the case of DNNs, one way to achieve global interpretability is by understanding the representations captured by intermediate layer neurons. Note that this will increase transparency.

As for local interpretation, one can provide the importance of each feature that causes the model prediction for a given input, thus help us in uncovering the causal relationship between features and model predictions.

2.1.3 Model agnostic vs Model based interpretability

. Post-hoc interpretable methods can be further classified based on the classes of models they can be applied to. Model agnostic methods don't depend on the model class and apply to every kind of machine learning model. On the other hand, Model-based methods apply to a specific type of model.

2.2 Which methods should we pursue?

In [Rud], Cynthia Rudin describes key issues with Post-hoc interpretability. We will try to show that if we restrict our domain to only local interpretable methods we can avoid those issues.

- **Explainable ML methods provide explanations that are not faithful to what the original model computes.**

  An explainable model that has a 90 agreement with the original model indeed explains the original model most of the time. However, an explanatory model that is correct 90 of the time is wrong 10 of the time. If a tenth of the explanations are incorrect, one cannot trust the explanations, and thus one cannot trust the original black box. If we cannot know for certain whether our explanation is correct, we cannot know whether to trust either the explanation or the original model.

  In case of local interpretable methods

  Indeed, the post-hoc model will have a low-fidelity to the original model, if we want it to be comprehensible (use a limited number of features to explain). But if we limit ourself to only a small sub-region of data space, we can have a post-hoc model with high-fidelity in this local region. Consider for instance that the original model is a highly non-convex plane covering the original data space. Looking at a small sub-region in this plane, we can define it with a simple convex or possibly linear plane. Thus, our domain (restricted post-hoc models i.e., local interpretable methods) can safely overcome this issue.

- **Explanations often do not make sense or do not provide enough detail to understand what the BlackBox is doing.**

  Even if the post-hoc model is correct (have high fidelity to the original model), It is still possible that the explanation leaves out so much information that it can't be trusted.

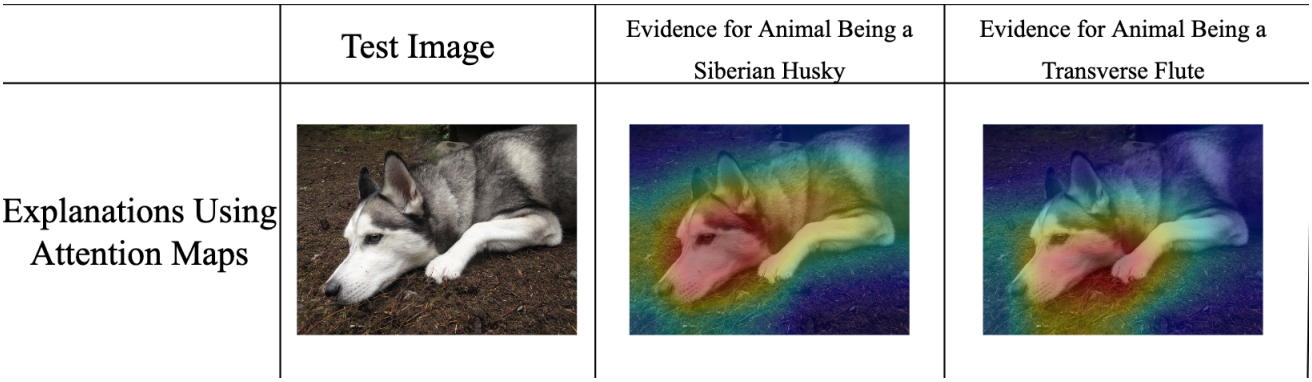


Figure 2.1 Saliency [SVZ14] does not explain anything except where the network is looking. We have no idea why this image is labeled as either a dog or a musical instrument when considering only saliency. The explanations look essentially the same for both classes. Figure credit: Chaofan Chen and [Che]

For example, consider figure 2.1 where the saliency map [SVZ14] is used to explain the behaviour of a model. Here the explanation is just a heat map of where the model is looking but the more valuable information of what it is doing with that part isn't given. Thus, the evidence (explanation) of two different data points could be the same which makes the explanation unreliable.

In case of local interpretable methods

One of the qualities that an explanation should possess is it should be comprehensible, which means it should be small enough to be digested by the audience. In principle, the importance of all features for a prediction can be given but often discarded due to the above requirement. Thus, an explanation should be enough to give end-user the understanding of model's prediction by using a limited amount of information, which is precisely what our domain methods do.

As for the saliency example, it is one way to explain the models' behaviour. Surely, it isn't the best but that doesn't mean that post-hoc models can't incorporate the required details.

2.3 Local Interpretable Methods

It is useful to perceive local interpretable methods as the problem of assigning an *attribution* value, sometimes also called "relevance" or "contribution", to each input feature of a network. As an example, consider a DNN to classify the input $x \in \mathbb{R}^N$ over $C = [c_1, c_2, ..., c_c]$ classes, let the output produced by network is $O = [p_1, p_2, ..., p_c]$ (where $p_i \in [0, 1]$ representing the probability of class i for input x and $\sum_{i=1}^{c} p_i = 1$).
Given a specific target c, the goal of an attribution method is to determine the contribution $R^c = [R_1^c, R_2^c, ..., R_N^c] \in \mathbb{R}^N$ of each input feature x_i to the output O.
As we already know these methods come in two flavours, Model agnostic and Model based, as described below :

2.3.1 Gradient based

. These are backpropagation based methods that compute the attributions for all input features in a single forward and backward pass through the network. Sometimes several of these steps are necessary, but the number does not depend on the number of input features. Following are some gradient based methods:

- **Saliency maps** ([SVZ14]) constructs attributions by taking the absolute value of the partial derivative of the target output O_c with respect to the input features x_i.

- **Gradient * Input** ([Shr+16]) computes the attribution by taking the (signed) partial derivatives of the output with respect to the input and multiplying them with the input itself.

- **Integrated Gradients** ([STY17]), similarly to Gradient * Input, computes the average gradient while the input varies along a linear path from a baseline $\bar{x}$ to x. The baseline is defined by the user and often chosen to be zero.

- **Layer-wise Relevance Propagation (LRP)** ([Bac+15]) is computed with a backward pass on the network.
- **DeepLIFT** ([SGK17]) proceeds in a backward fashion, similarly to LRP.

The formula for attribution function and examples are shown in Figure 2.3. The difference in the explanation lies in the way they define the attribution value of each pixel in the image.

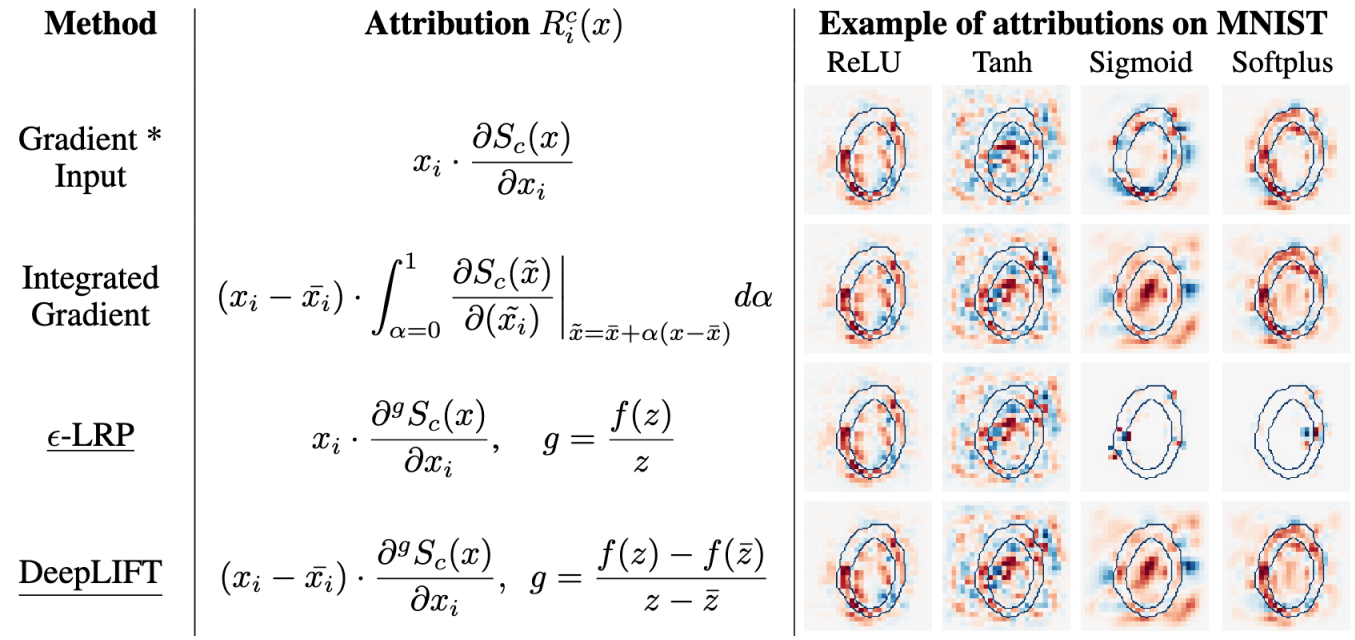


Figure 2.2 Mathematical formulation of gradient-based attribution methods [Anc+18]

2.3.2 Perturbation based

. Perturbation-based methods directly compute the attribution of an input feature (or set of features) by removing, masking or altering them, and running a forward pass on the new input, measuring the difference with the original output. The method of altering the features to generate the perturbation set is what separates these methods form each other. Some of these methods are described below :

- **Occlusion-1** ([ZF14]) replaces one feature x_i at a time with a zero baseline and measures the effect of this perturbation on the target output, i.e., $O_c(x) - O_c(x_{x_i=0})$ where we use $x_{[x_i=v]}$ to indicate a sample $x \in \mathbb{R}^N$ whose i-th component has been replaced with v. For the choice of baseline refer to the paper.

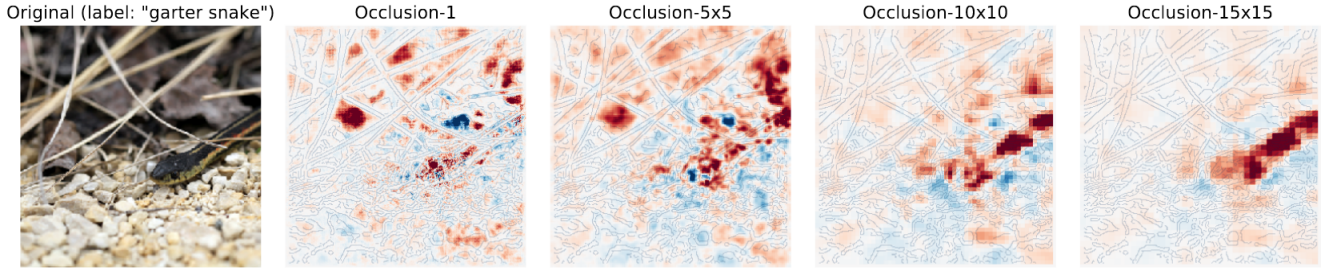


Figure 2.3 Attributions generated by occluding portions of the input image with squared grey patches of different sizes. Notice how the size of the patches influence the result, with focus on the main subject only when using bigger patches. [Anc+18]

- **LIME** ([RSG16]) perturb the input data and check what happens to the predictions. It generates a new dataset consisting of perturbed samples and the corresponding predictions of the black-box model. On this new dataset LIME then trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest.

  Mathematically, local surrogate models with interpretability constraint can be expressed as follows:

$$\text{explanation}(x) = \underset{g \in G}{\operatorname{argmin}}\, L(f, g, \pi_x) + \Omega(g)$$

  The explanation model for instance x is the model g (e.g. linear regression model) that minimizes loss L (e.g. mean squared error), which measures how close the explanation is to the prediction of the original model f (e.g. an xgboost model), while the model complexity $\Omega(g)$ is kept low (e.g. prefer fewer features). G is the family of possible explanations, for example all possible linear regression models. The proximity measure π_x defines how large the neighborhood around instance x is that we consider for the

explanation. A detailed example of LIME is shown in figure 2.4.

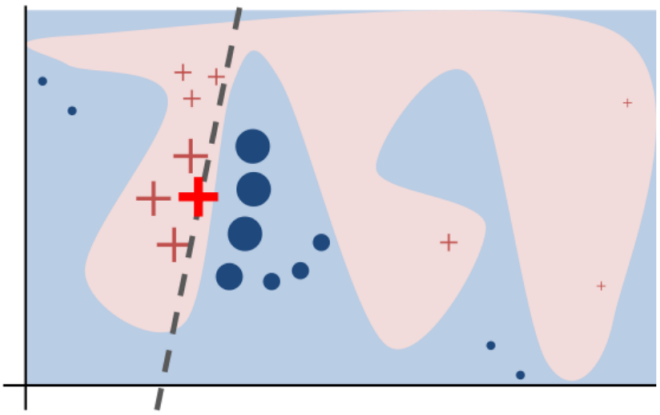


Figure 2.4 Toy example to present intuition for LIME.The black-box model's complex decision function f(unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented hereby size). The dashed line is the learned explanation that is locally (but not globally) faithful. [RSG16]

In practice, LIME only optimizes the loss part. The user has to determine the complexity, e.g. by selecting the maximum number of features that the linear regression model may use.

The recipe for training local surrogate models:

- Select your instance of interest for which you want to have an explanation of the black box prediction.
- Perturb your dataset and get the black box predictions for these new points.
- Weight the new samples according to their proximity to the instance of interest.
- Train a weighted, interpretable model on the dataset with the variations.
- Explain the prediction by interpreting the local model.

Figure 2.5, shows an example where lime is used to explain the prediction of two classifiers on the task of document classification over two classes.

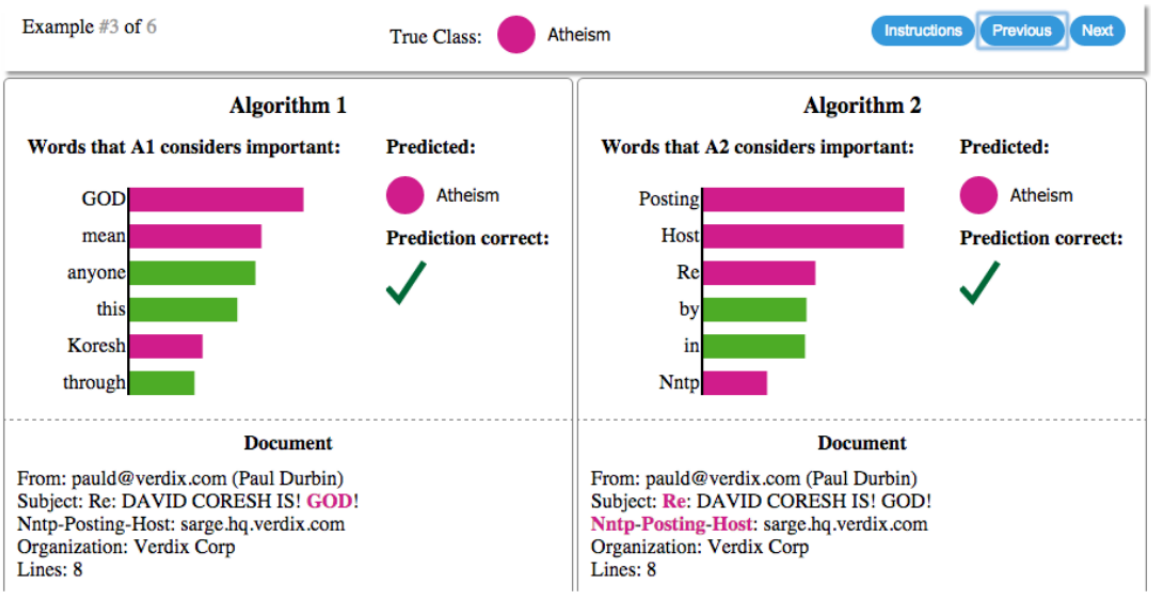


Figure 2.5 Explaining individual predictions of competing classifiers trying to determine if a document is about "Christianity" or "Atheism". The bar chart represents the importance given to the most relevant words, also highlighted in the text. Color indicates which class the word contributes to (green for"Christianity", magenta for "Atheism"). [RSG16]

NOTE : LIME can also provide global explanation by explaining a set of individual instances (SUBMODULAR PICK).

- **Anchor** ([RSG18]) An anchor explanation is a rule that sufficiently "anchors" the prediction locally – such that changes to the rest of the feature values of the instance do not matter. In other words, for instances on which the anchor holds, the prediction is (almost)always the same. For example, the anchors in Figure 2.6 state that the presence of the words "not bad" virtually guarantee a prediction of positive sentiment (and "not good" of negative sentiment).

  Formally, given an instance x to be explained, a rule or an anchor A is to be found, such that it applies to x, while the same class as that of x gets predicted for a fraction of at least τ of x's neighbors where the same A is applicable. A rule's precision results from evaluating neighbors or perturbations (following $D_x(z|A)$) using the provided machine

learning model (denoted by the indicator function $1_{f(x)=f(z)}$).

Or to put it mathematically, we need to find A for instance x s.t

$$\mathbb{E}_{D(z|A}[1_{f(x)=f(z)}] \geq \tau, A(x) = 1$$

The above equation can be framed as an optimisation problem.

Let $prec(A) = \mathbb{E}_{D(z|A}[1_{f(x)=f(z)}]$ and $cov(A) = \mathbb{E}_{D(z)}[A(z)]$.

For an arbitrary D and black-box model f, it is intractable to compute this precision directly. Instead, we introduce a probabilistic definition: anchors satisfy the precision constraint with high probability

$$P(prec(A) \geq \tau) \geq 1 - \delta$$

then the search for an anchor is the following combinatorial optimization problem,

$$\max_{A \ s.t \ P(prec(A) \geq \tau) \geq 1-\delta} cov(A)$$

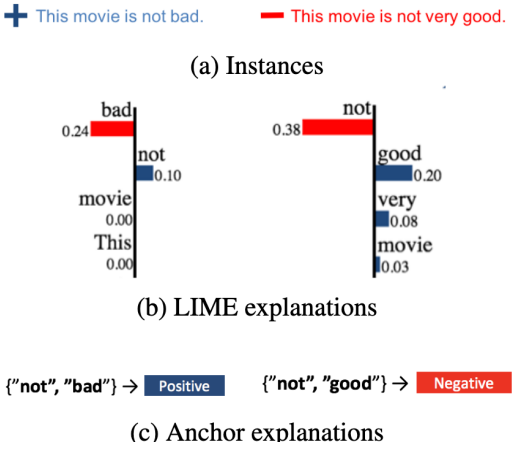


Figure 2.6 Sentiment predictions, LSTM [RSG18]

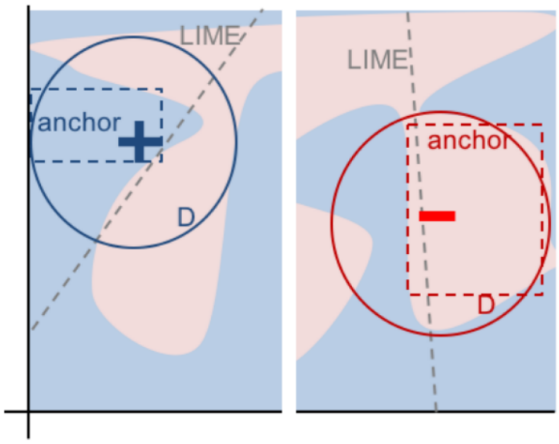


Figure 2.7 LIME vs. Anchors – A Toy Visualization. [RSG18]

Chapter Three

ROBUSTNESS

3.1 Motivation

One crucial property that local interpretable model should pose is robustness. There are many ways to define robustness. The intuition is that similar inputs should not produce substantially different explanations. The reason why this is indeed a crucial property can be reasoned via the following arguments:

- In order for an explanation to be valid around a point, it should be constant in the close vicinity or should change marginally.
- If we seek an explanation that can be applied in a predictive sense around the point of interest as described above, then robustness of the simplified model implies that it can be approximately used in lieu of the true complex model, at least in a small neighborhood.

3.2 Related Work

Recent work to develop a notion of robustness appears in ([AJ18]). Here the authors' notion of robustness captures variations of a prediction's "explanation" with respect to changes in the input leading to that prediction.

The authors use neighbourhood-based local Lipschitz continuity defined as:

Definition $f : \chi \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is **locally Lipschitz** if for every x_0 there exist $\delta \geq 0$ and $L \in \mathbb{R}$ such that $\|x - x_0\| \leq \delta$ implies $\|f(x) - f(x_0)\| \leq L\|x - x_0\|$.

As opposed to the (global) Lipschitz criterion, here both δ and L depend on the anchor point x_0. Armed with this definition they quantify the robustness of model f in term of constant L.

Naturally, this quantity is rarely known a-priori, and thus has to be estimated. A straightforward way to do this involves solving, for every point x_i of interest, an optimization problem:

$$\hat{L}(x_i) = \underset{x_j \in N_\epsilon(x_i)}{\operatorname{argmax}} \frac{\|f(x_i) - f(x_j)\|_2}{\|x_i - x_j\|_2}$$

where $N_\epsilon(x_i)$ is a ball of radius ϵ centered at x_i.

Experimental Results

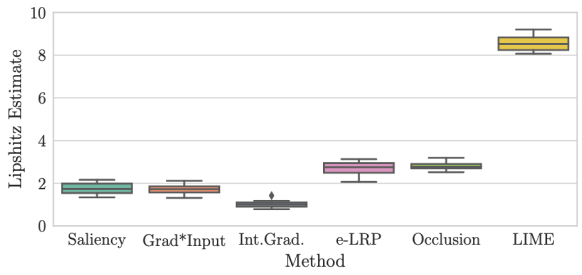


Figure 3.1 Local Lipschitz estimates computed on 100 test points on MNIST explanations. [AJ18]

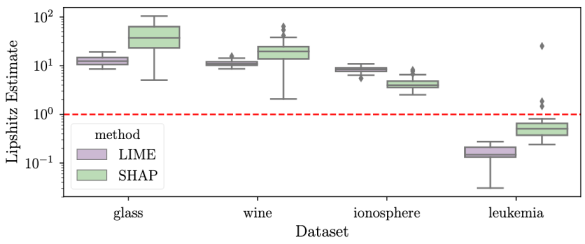


Figure 3.2 Local Lipschitz estimates computed on 100test points on various UCI classification datasets. [AJ18]

Figure 3.3 shows the robustness score for a sample example from MNIST dataset for different local interpretable methods. It is evident from the figure that for perturbation based method (especially LIME) the explanation is more volatile to noise as compared to gradient-based methods and thus have higher value of δ from the definition of local lipschitz continuity which suggest that it is less robust.

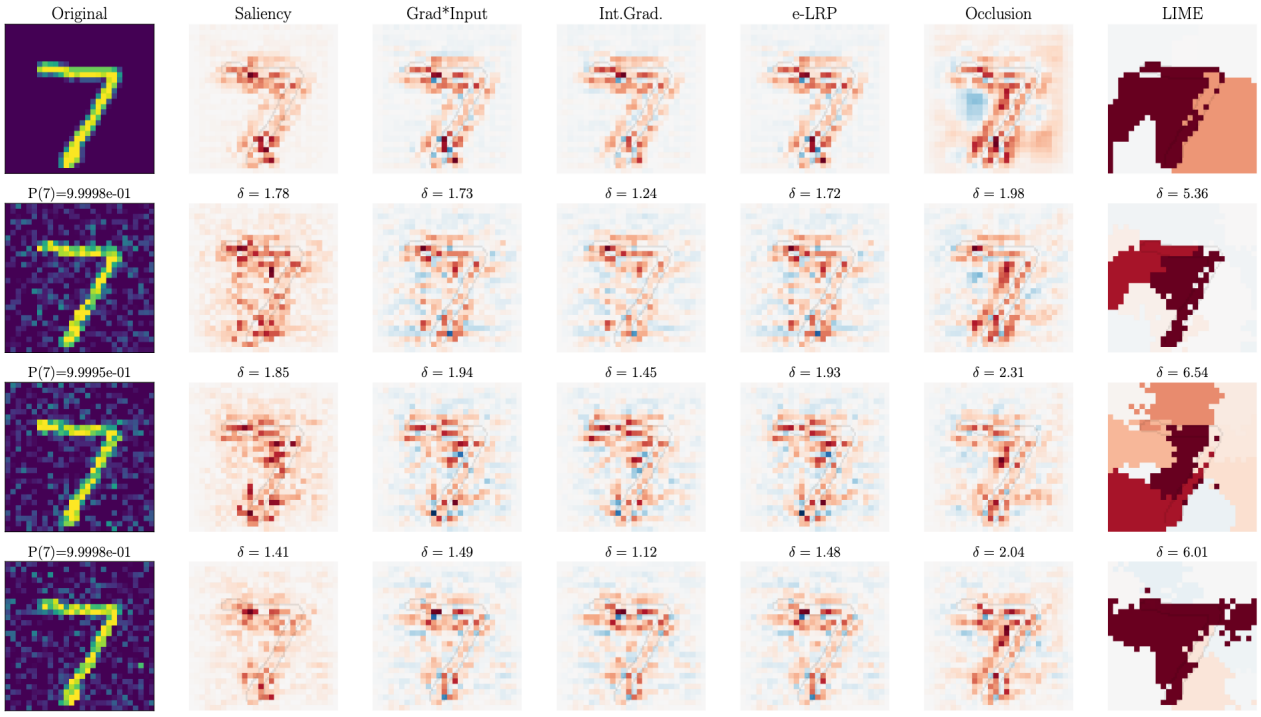


Figure 3.3 Explanations of a CNN model prediction's on a example MNIST digit (top row) and three versions with Gaussian noise added to it. The perturbed input digits are labeled with the probability assigned to the predicted class by the classifier. Here δ is the ratio $\|f(x) - f(x^{'})\|_2 / \|x - x^{'}\|_2$ for the perturbed $x^{'}$, which are not adversarially chosen as in the above definition. [AJ18]

3.3 Our Formulation

The explanation of local interpretable methods we have discussed so far provides the features (set of pixels in case of image data and set of words in case of text data) that are important in the vicinity of the addressed data point. We will name these features as the anchor [1] and the rest of the features as anchor complement.

We make two observation about the anchor and its complement as :

1. Intuitively, if the anchor indeed depicts important features in the local neighbourhood

[1] This anchor is just an explanation of a specified method and shouldn't be confused with anchor in [RSG18]

(meaning the explanation is robust) then the prediction of the original model should be more volatile to change in the anchor as compared to changes in anchor complement.

2. Typically a complex machine learning model (DNNs in most cases) is just a highly non-convex plane over input data space. If we demand the explanation to be robust we expect that a small change in anchor should bring a small change in original models prediction. Likewise, a small change in anchor complement should bring a relatively bigger change in original models prediction.

These two points will be the basis for our notion of robustness for local interpretable methods. Now, we will try to put these points in a more formal way. It is fair to point out at the outset that our notion is solely based on the experimentation and thus in a sense it is empirical.

Let the original model be f. Take N different data points from input space.
Let the anchor (or one can say the explanation) of the ith point (N^i) be (A^i_{LE}) and the anchor complement be $(A^i_{LE})'$.
Suppose the average $L1$ *norm* over the N examples is L^1_{avg}.
We will try to generate noise on (A^i_{LE}) and $(A^i_{Le})'$ for different percentages of L^1_{avg}, in increasing order (*bins*).
Let L and U represent the lowest and highest bin values respectively.
For example, $bins$ = [0.02, 0.05, 0.1, 0.2]

NOTE: The reason to choose $L1$ *norm* over other L *norms* is given in the Appendix A. Moreover one can check what is anchor, $l1$ *norm* and how to generate noise on (A^i_{LE}) and $(A^i_{LE})'$ for different data types.

Formalisation of point 1

Let N_i be the ith example. Let p be the smallest bin such that adding $p\%$ noise to (A^i_{LE}) generate a noisy data point (A^i_{MCN}) with $f(N^i) \neq f(A^i_{MCN})$.

Likewise, Let q be the smallest bin such that adding $q\%$ noise to $(A^i_{LE})'$ generate a noisy data point $(A^i_{MCN})'$. with $f(N^i) \neq f((A^i_{MCN})')$.

Now, the robustness (as per point 1) of this explanation can be calculated via following formula:

$$R^i_1 = \frac{1}{U-L}(q-p)$$

There are some extreme cases that we need to take into account, these are:

- p **and** q **exceeds** U In this case we will put the R^i_1 to 0, as we haven't found any adversarial example (in our limited noisy attempt) that could change the original prediction but this adversarial example might be present for some different configuration of noise.
- p **exceeds** U In this case we put the R^i_1 to 1.
- q **exceeds** U In this case we put the R^i_1 to -1.

Formalisation of point 2

Before delving into the mathematical formalisation, let's revisit the intution behind it. What it talks about is of divergence, the divergence of the misclassified noisy anchor and anchor complement from original prediction.

The robustness (as per point 2) of an explanation is then the squeezed difference (squeezed in [0, 1]) of anchor complement divergence score and anchor divergence score.

NOTE: Refer to the Appendix B to check the way we calculate divergence score.

3.4 Experiments/Results[2]

We studied both perturbations as well as gradient-based methods to calculate and compare the robustness score over two different datasets: Imagenet and MNIST.

We used authors implementation of LIME [3], alibi library [4] for Anchor and DeepExplain [5] for the rest of the methods. The sample results for perturbation and gradient-based methods on a sample input from Imagenet are shown in table 3.1 and 3.2 respectively.

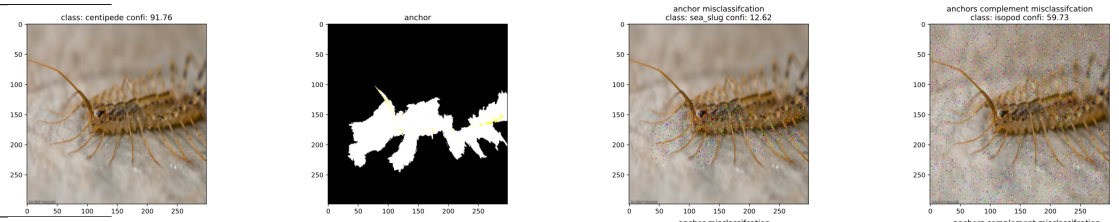
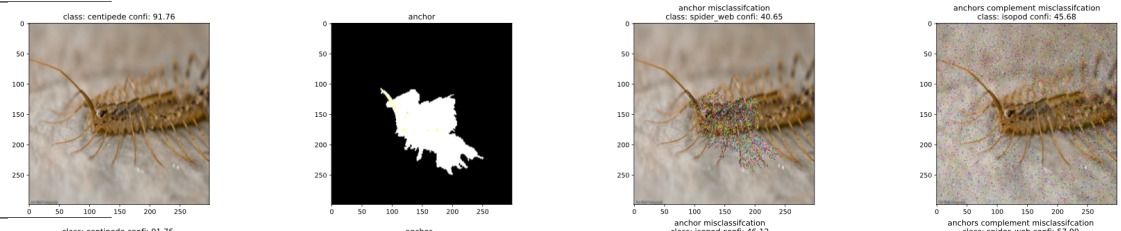
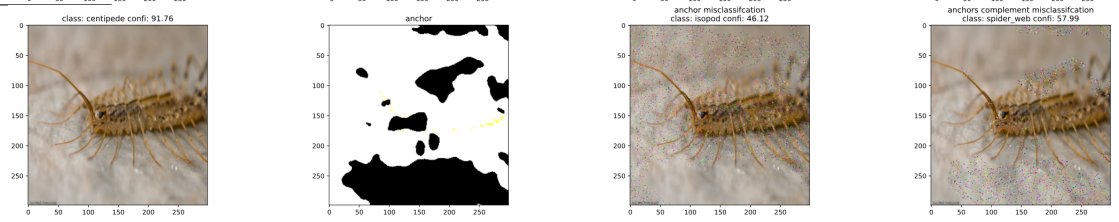
Method	Result
LIME	
Anchor	
Occlusion	

Table 3.1 Result of perturbation based methods

The anchors of these methods look very different due to the nature of the explanation they provide. Perturbation based methods give segments of the image as an explanation while on the other hand, gradient-based methods give the pixel set that achieves selective

[2] https://github.com/rohit9650/robustness

[3] https://github.com/marcotcr/lime [RSG16]

[4] https://github.com/SeldonIO/alibi

[5] https://github.com/marcoancona/DeepExplain

gradient threshold.

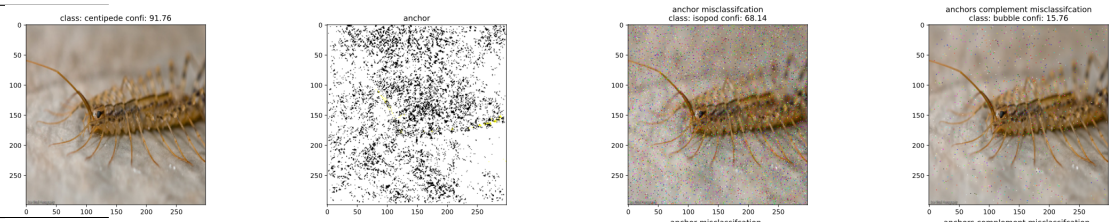
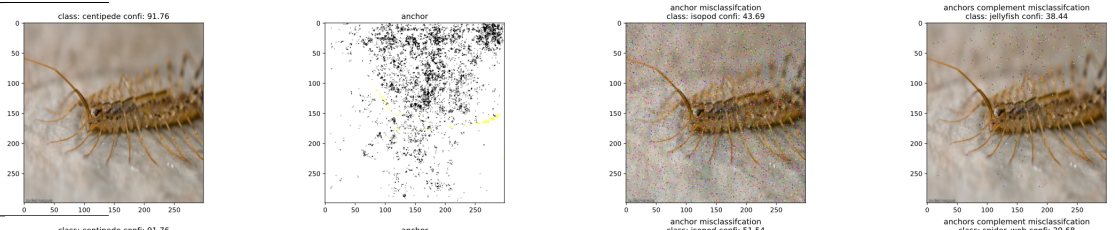
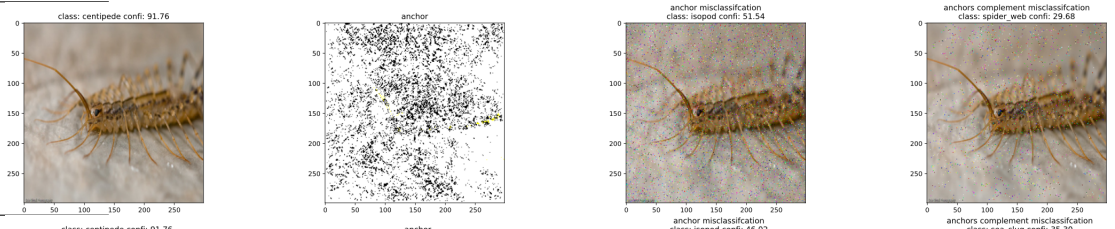
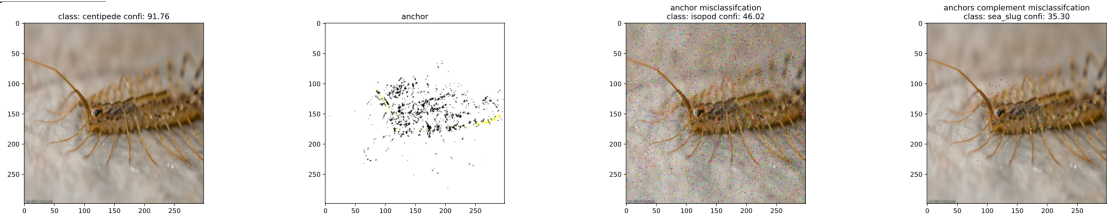
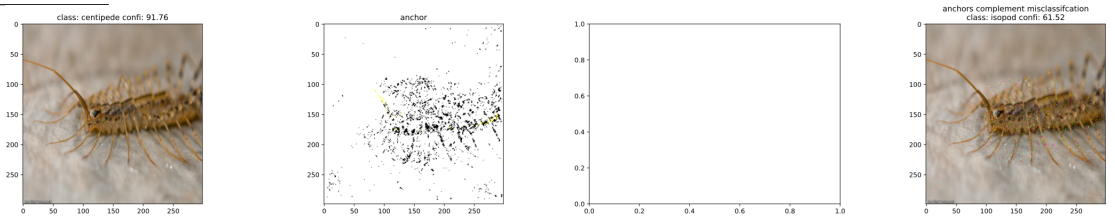
Method	Result
$\in$ **LRP**	
Saliency	
Grad*input	
Intragrad	
DeepLift	

Table 3.2 Result of gradient based methods

It is worth noting that the entries in the last two columns of result tables can be empty. The reason for this behaviour is that we didn't find any misclassification when noise was added to anchor or anchor complement (depending on which one is empty). Also, both the entries can not be simultaneously empty as in that case we would have discarded this sample from contributing to the robustness score for the specified method (see the observation in our formulation section).

Finally, the overall result of robustness score for these methods for datasets: Imagenet (over 100 data points) and MNIST (over 1000 data points) is shown in table 3.3.

	Robustness Results			
	Imagenet (over 100 data points)		*MNIST (over 1000 data points)*	
Method Name	**# of data points**	**robustness**	**# of data points**	**robustness**
LIME	93	0.56	184	0.66
Anchor	79	0.18	200	0.29
Occlusion	93	-0.23	199	0.71
∈LRP	91	-0.32	197	0.68
Saliency	95	-0.32	333	0.83
Grad*Input	92	-0.30	191	0.72
IntraGrad	95	-0.33	184	0.69
DeepLIFT	92	-0.39	203	0.73

Table 3.3 Robustness score of local interpretable methods over imagenet and mnist data

3.5 Conclusion

The final table 3.3 suggest that perturbation based methods are more robust on Imagenet dataset but are less robust as compared to gradient-based methods on MNIST dataset. The reason behind this observation is the need for tweaking a large number of parameters for these methods in order to work well for a given dataset. On the other hand, gradient-based methods are restricted to a particular type of ML model that is Neural Networks and can't be applied to other models. So, we suggest that the perturbation based methods should be the first choice to set a benchmark on the explanation of the original model.

REFERENCES

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks:Visualising image classification models and saliency maps". In: *ICLR* (2014).

[ZF14] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision* (2014), pp. 818–833.

[Bac+15] Sebastian Bach et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wiserelevance propagation." In: *PloS one, 10(7):e0130140* (2015).

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" In: *ACM* (2016), pp. 1135–1144. DOI: 10.1145/2939672.2939778. URL: http://arxiv.org/abs/1602.04938.

[Shr+16] Avanti Shrikumar et al. "Not just a blackbox: Learning important features through propagating activation differences". In: *arXiv preprint arXiv:1605.01713* (2016).

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences". In: *Proceedings of the 34th International Conference on Machine Learning* 70 (2017), pp. 3145–3153. DOI: http://proceedings.mlr.press/v70/shrikumar17a.html.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *Proceedings of the 34th International Conference on Machine*

Learning 70 (2017), pp. 3319–3328. DOI: http://proceedings.mlr.press/v70/sundararajan17a.html.

[AJ18] David Alvarez-Melis and Tommi S. Jaakkola. "On the Robustness of Interpretability Methods". In: *cs.LG* (2018). DOI: https://arxiv.org/pdf/1806.08049.pdf.

[Anc+18] Marco Ancona et al. "TOWARDS BETTER UNDERSTANDING OF GRADIENT-BASED ATTRIBUTION METHODS FOR DEEP NEURAL NETWORKS". In: *ICLR* (2018). DOI: https://openreview.net/pdf?id=Sy21R9JAW.

[RSG18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High-Precision Model-Agnostic Explanations". In: *Association for the Advancement of ArtificialIntelligence (www.aaai.org)* (2018). URL: https://homes.cs.washington.edu/~marcotcr/aaai18.pdf.

[Mol19] Christoph Molnar. *Interpretable Machine Learning A Guide for Making Black Box Models Explainable.* Christoph Molnar, 2019.

[DLH20] Mengnan Du, Ninghao Liu, and Xia Hu. "Techniques for Interpretable Machine Learning". In: *ACM* 63.1 (2020), pp. 68–77. DOI: https://cacm.acm.org/magazines/2020/1/241703-techniques-for-interpretable-machine-learning/fulltext.

[Che] Checkermallow. *Canis lupus winstonii (Siberian Husky)*. URL: https://www.flickr.com/photos/132792051@N06/28302196071/in/photolist-K7Y9RM-utZTV9-QWJmHo-QAEdSE-QAE3pL-TvjNJu-%20tziyrj-EWFwEx-DWb7T4-DTRAWu-CYLBpP-DMUVn2-dUbgLG-ccuabw-57nNvJ-UpDv4D-eNyCQP-q8aWpJ-86gced-QLBwiG-QP7k6v-aNxiRc-rmTdLW-oeTM8i-d1rkCG-ueSwz4-%20dYKwJx-7PxAPF-KFUqKN-TkarEj-7X5FZ2-7WS6Z2-7X5Gwa-7X5GkT-7Z8w5s-s4St8A-%20qsa12b-7X8Vqs-7X8VLy-7X5Gm6-7X5Gjp-PTy69W-7X8VQ3-7X8VEy-7X5GqD-iaMjUN-7X8VgE-odbiWy-TkacgQ-7X5Gk4/.

[Rud] Cynthia Rudin. *Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and UseInterpretable Models Instead*. URL: https://arxiv.org/pdf/1811.10154.pdf.

[Sin] Rohit Singh. *Robustness experiment code for master's thesis CMI*. URL: https://github.com/rohit9650/robustness.

APPENDIX

Appendix A

Reason for choosing $L1$ norm

We need L_{avg} of some L *norm* so that we can add noise to either (A^i_{LE}) or $(A^i_{LE})'$ for different % of L_{avg} and get misclassified anchor (A^i_{MCN}) or misclassified anchor complement $(A^i_{MCN})'$ respectively. So, one would want that misclassified examples distribute nicely over different % of L_{avg}.

From our empirical study of $L1$ *norm* and $L2$ *norm* over 50 examples from different categories of imagenet dataset, we get the following result (we have result for 2 classes only):

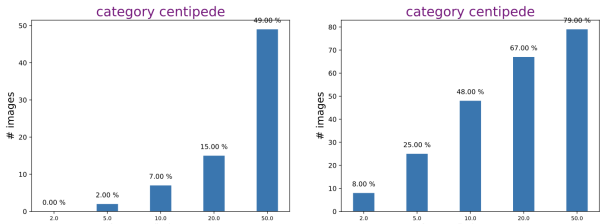


Figure A.1 L^2_{avg} vs L^1_{avg}

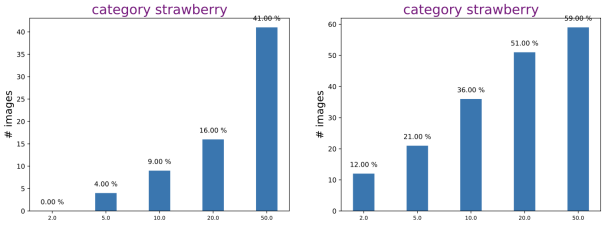


Figure A.2 L^2_{avg} vs L^1_{avg}

This experiment suggests that L^1_{avg} would be a good choice for our work.

Image data type

- **Anchor and Anchor complement** The local interpretable methods provide explanation of image data type as a collection of pixels. We call this set of pixels as Anchor and the remaining of pixels as anchor complement.
- **Calculating $L1$ norm** We used the usual way of calculating $L1$ *norm*.
- **Generating noise to (A^i_{LE}) and $(A^i_{LE})'$** We used the Gaussian noise with standard mean and variance to put noise on Anchor or Anchor complement for different fraction

and epsilons values, and see what bin value they fall in i.e., p and q.

Text data type

- **Anchor and Anchor complement** Local interpretable methods provide explanation of Text data type (say when the task is of classification) as a collection of words. We call this set of words as Anchor and the remaining of words in the sentence or document as anchor complement.

- **Calculating $L1$ norm** We can use word vectors of all the words present in document and take the $L1$ *norm* of their average vector.

- **Generating noise to (A^i_{LE}) and $(A^i_{LE})'$** We can tweak the word vectors of words present in the Anchor or Anchor complement, and see to what bin value the $L1$ *norm* of document average vector fall in i.e., p and q.

Appendix B

Divergence score

labels is a list of $(class, probability)$ over all classes. Now, if *true_labels* represent the original prediction and *labels*1, *labels*2 represent the misclassified anchor and anchor complement prediction respectively, then one way to calculate divergence is using the code given below.

```
# sort original label w.r.t prediction. highest first
def sortSecond(val):
  return val[1]
true_labels.sort(key=sortSecond, reverse=True)

labels1_dict = {}
labels2_dict = {}

for i in range(len(labels1)):
  labels1_dict[labels1[i][0]] = labels1[i][1]
  labels2_dict[labels2[i][0]] = labels2[i][1]

i = 1
divergence1 = 0
divergence2 = 0
while True:
  if i >= len(true_labels) or true_labels[i-1][1] == 0:
    break
  divergence1 += i * abs(true_labels[i-1][1] - labels1_dict[true_labels[i-1][0]])
  divergence2 += i * abs(true_labels[i-1][1] - labels2_dict[true_labels[i-1][0]])
  i += 1

divergence_diff = divergence2 - divergence1

robustness = divergence_diff / (len(labels1) / 2)

while robustness > 1:
  robustness /= 10
```

Figure B.1 divergence score implementation