International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 12 | Dec - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

IMAGE SIMILARITY USING LOGISTIC REGRESSION

Ch Mukitha [1], D Sravan Kumar [2], G Aravind [3], Swapna[4]
UG Scholar, Guru Nanak Institutions Technical Campus, Hyderabad[1,2]
Assistant Professor, Guru Nanak Institutions Technical Campus, Hyderabad [3]

ABSTRACT:

Many learning algorithms such as kernel machines, nearest neighbors, clustering, or anomaly detection, are based on distances or similarities. Before similarities are used for training an actual machine learning model, we would like to verify that they are bound to meaningful patterns in the data. In this paper, we propose to make similarities interpretable by augmenting them with an explanation. We develop Logistic Regression & Haar cascade, a scalable and theoretically founded method to systematically decompose the output of an already trained deep similarity model on pairs of input features. Our method can be expressed as a composition of regression explanations, which were shown in previous works to scale to highly nonlinear models. Through an extensive set of experiments, we demonstrate that Logistic Regression robustly explains complex similarity models. Additionally, we apply our method to an open problem in digital humanities: detailed assessment of similarity between historical documents, such as astronomical tables. Here again, Logistic Regression & Haar cascade provides insight and brings verifiability into a highly engineered and problem-specific similarity model.

1 INTRODUCTION

Building meaningful similarity models that incorporate prior knowledge about the data and the task is an important area of machine learning and information retrieval [2]. Good similarity models are needed to find relevant items in databases [3], [5]. Similarities (or kernels) are also the starting point of a large number of machine learning models including discriminative learning [6], [7], unsupervised learning [8], [9], and data embedding/visualization [12], [14]. An important practical question is how to select the similarity model appropriately. Assembling a labeled dataset of similarities for validation can be difficult: The labeler would need to inspect meticulously multiple pairs of data points and come up with exact real-valued similarity scores. As an alternative, selecting a similarity model based on performance on some proxy task can be convenient (e.g., [15],[16]). In both cases, however, the selection procedure is exposed to a potential lack of representatives of the training data (cf. the 'Clever Hans' effect [19]).— In this paper, we aim for a more direct way to assess similarity models, and make use of explainable ML for that purpose. Explainable ML [20], is a subfield of machine learning that focuses on making predictions interpretable for the human. Numerous methods have been proposed in the context of ML classifiers [23], [24 Logistic regression coefficients can be used to determine the relative importance of each input feature in making predictions. The larger the magnitude of the coefficient, the more important the corresponding feature is in predicting the outcome.

.

1.1 OBJECTIVE

In this paper, we bring explainable ML to similarity. We present basic approaches to explain the predictions of a similarity model in terms of input features. The similarity model is considered to be already trained. We first discuss the case of a simple linear model, and then extend the concept to more general nonlinear cases.

1.2 SCOPE OF THE WORK:

To develop a Logistic Regression along with Haarcascade in order to decompose the output of the trained similarity model into pairs of input features.

1.3 EXISTING SYSTEM:

While there has been substantial progress in learning suitable distance metrics, these techniques in general lack transparency and decision reasoning, i.e., explaining why the input set of images is similar or dissimilar. In this work, we solve this key problem by proposing the first method to generate generic visual similarity explanations with gradient-based attention. ¬ We demonstrate that our technique is agnostic to the specific similarity model type, e.g., we show applicability to Siamese, triplet, and quadruplet models. Furthermore, we make our proposed similarity attention a principled part of the learning process, resulting in a new paradigm for learning similarity functions. We demonstrate that our learning mechanism results in more generalizable, as well as explainable, similarity models..

1.3.1 Existing System Disadvantages:

- They fail to encode the position and orientation of objects.
- They fail to encode the position and orientation of objects.
- It tends to be much slower because of operations like maxpool.

1.4 SYSTEM ARCHITECTURE:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity. • Operating System • Platform • Programming Language • Front End : : : : Windows 7/8/10 Spyder3 Python Spyder3

1.4.1 EXPLANATION:

The architecture for the loan prediction model project follows a structured process for data handling and machine learning. It starts with
Data Collection, which involves gathering
the necessary raw data. This data then goes through **Preprocessing**, where it is cleaned and prepared for analysis, including handling missing values and encoding categorical data. Following this, the data is **Split** into training and testing sets. The **Model Building** phase involves training different machine learning models, like Logistic Regression, on the training data. These models are then evaluated using performance metrics to identify the best-performing model. Finally, the selected model is used for **Prediction**, providing insights on the likelihood of loan approval for new applicants. This architecture ensures a systematic and thorough approach to developing a reliable loan prediction model.

1.5 PROPOSED SYSTEM

¬ Image collection for deep learning often involves the use of web scraping techniques to automatically download large numbers of images from online sources such as search engines or social media platforms. These images are often labeled manually or with the help of annotation tools to create a dataset suitable for training deep learning models. ¬ Deep learning models often require large amounts of data to be trained effectively, which can be a challenge when working with images. To address this, techniques such as transfer learning can be used to reuse pre-trained models and fine-tune them on a smaller dataset. Additionally, deep learning models can be trained on large-scale image datasets such as ImageNet or COCO, which provide a broad range of labeled images that can be used to train models for specific tasks.

1.5.1 PROPOSED SYSTEM ADVANTAGES:

- It brings such explain ability and scales to potentially highly complex dataset.
- Improved classification accuracy and efficient use of computational resources.

2 DESCRIPTION

2.1 GENERAL:

Logistic regression is a supervised learning algorithm used for binary classification problems, while Haar cascade is an object detection algorithm used for detecting objects in images or videos. We apply logistic regression to classify images and then use Haar cascade to detect specific objects in those images. We first train a logistic regression model on a dataset of labeled images. The logistic regression model will then predict the label of new images. After that, we can apply Haar cascade to detect specific objects in the images based on the label predicted by the logistic regression model. Once the logistic regression model is trained, we can use it to predict the label of new images. Then, we can apply Haar cascade to detect a person in the image predicted by the logistic regression model. Using logistic regression and Haar cascade in combination can improve the accuracy of object detection in images. By using the logistic regression model to predict the label of the images, we can reduce the search space for the Haar cascade algorithm, making it more efficient and accurate. Combining logistic regression and Haar cascade can be useful for specific tasks such as object detection in images. Logistic regression can be used to predict the label of the images, and then Haar cascade can be used to detect specific objects based on the predicted label. This combination of methods can improve the accuracy of object detection and make the process more efficient. Altogether, the method we propose brings transparency into a key ingredient of machine learning: similarity. Our contribution paves the way for the systematic design and validation of similarity-based ML models in an efficient, fully informed, and human-interpretable manner..

2.2 METHODOLOGIES

2.2.1 MODULES NAME:

- Dataset
- Importing the necessary libraries
- Retrieving the images
- Splitting the dataset
- Building the model
- Apply the model and plot the graphs for accuracy and loss
- Accuracy on test set
- Saving the Trained Model

1) Dataset: In the first module, we developed the system to get the input dataset for the training and testing purpose. We have taken the dataset for deep similarity in images. The image similarity dataset consists of 274 images of 5 different persons.

2) Importing the necessary libraries: We will be using Python language for this. First, we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, cv2(OpenCV) for image processing tasks and other libraries such as pandas, numpy, matplotlib and tensorflow.

3) Retrieving the images: We will retrieve the images and their labels. Then resize the images to (64,64) as all images should have same size for recognition. Then convert the images into numpy array.

4) Splitting the dataset: Split the dataset into train and test. 80% train data and 20% test data. A. Convolutional Neural Networks The objectives behind the first module of the course 4 are: • To understand the convolution operation • To understand the pooling operation • Remembering the vocabulary used in convolutional neural networks (padding, stride, filter, etc.) • Building a convolutional neural network for multi-class classification in images B. Computer Vision Some of the computer vision problems which we will be solving in this article are: 1. Image classification 2. Object detection 3. Neural style transfer 14 One major problem with computer vision problems is that the input data can get really big. Suppose an image is of the size 68 X 68 X 3. The input feature dimension then becomes 12,288. This will be even bigger if we have larger images (say, of size 720 X 720 X 3). Now, if we pass such a big input to a neural network, the number of parameters will swell up to a HUGE number (depending on the number of hidden layers and hidden units). This will result in more computational and memory requirements – not something most of us can deal with.

5) Building the model: For building the model we will use sequential model from keras library. Then we will use FaceNet model consists of a deep convolutional neural network (CNN) that takes an input image of a face and outputs a vector of features that represent the face. The CNN is trained on a large dataset of faces to learn how to extract facial features that are discriminative and invariant to variations in lighting, pose, and facial expression. The resulting feature vectors have the property that faces from the same person are mapped to similar vectors, while faces from different people are mapped to dissimilar vectors. This makes it possible to perform face recognition by comparing the feature vectors of two faces and computing the distance between them.

6) Apply the model and plot the graphs for accuracy and loss: We will compile the model and apply it using fit function. Then we will plot the graphs for accuracy and loss

7) Accuracy on test set: We got an accuracy of 100% on test set. 8) Saving the Trained Model: Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle . Make sure you have pickle installed in your environment. Next, let's import the module and dump the model into.pkl file

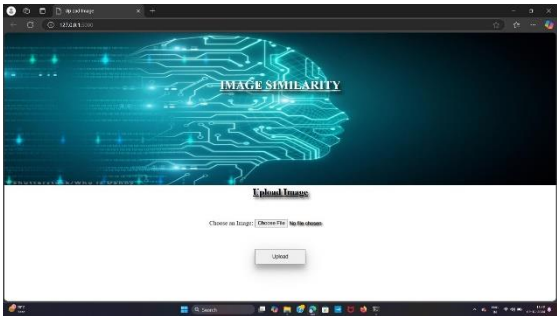
8) Saving the Trained Model: Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle . Make sure you have pickle installed in your environment. Next, let's import the module and dump the model into.pkl file

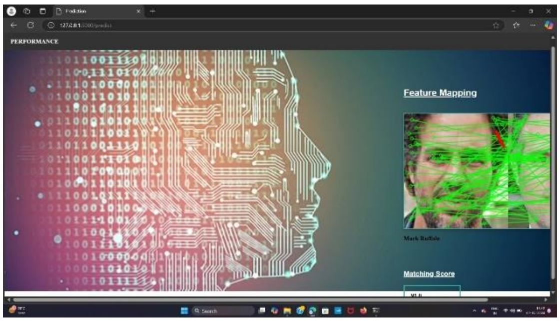
2.3 TECHNIQUE USED OR ALGORITHM USED

- CNN attention

We aim to learn a distance metric to measure the similarity between two images x1 and x2. Our key innovation includes the design of a flexible technique to produce similarity model explanations by means of CNN attention, which we show can be used to enforce trainable constraints during model training. This leads to a model equipped with similarity explanation capability as well as improved model generalizability. With our proposed mechanism to compute similarity attention, one can generate attention maps, to explain why the similarity model predicted that the data sample satisfies the similarity criterion.

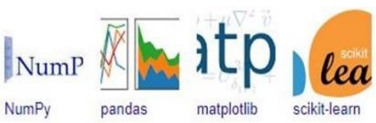
3 RESULT:





Libraries used in python:

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as data frames.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.



4 FUTURE ENHANCEMENT

Future work will extend the presented techniques from binary towards n-ary similarity structures, especially aiming at incorporating the different levels of reliability of the input features.

5 CONCLUSION

We have contributed a theoretically well-founded method to explain similarity in terms of pairs of input features. Our method called Logistic Regression can be expressed as a composition of regression computations. It therefore inherits its robustness and broad applicability, but extends it to the novel scenario of similarity explanation. The usefulness of Haar cascade was showcased on the task of understanding similarities as implemented by the Logistic regression, where it could predict transfer learning capabilities and highlight clear cases of ‘Clever Hans’ [19] predictions. Furthermore, for a practically relevant problem in the digital humanities, Haar cascade was able to demonstrate with very limited data the superiority of a task-specific similarity model.

6 REFERENCES

[1] A. Zien, G. R€atsch, S. Mika, B. Sch€olkopf, T. Lengauer, and K.-R. M€uller, “Engineering support vector machine kernels that recognize translation initiation sites,” Bioinformatics, vol. 16, no. 9, pp. 799–807, 2000.

[2] C. D. Manning, P. Raghavan, and H. Sch€utze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008. [3] A. Nierman and H. V. Jagadish, “Evaluating structural similarity in XML documents,” in Proc. Int. Workshop Web Databases, 2002

. [4] E. Pampalk, A. Flexer, and G. Widmer, “Improvements of audio-based music similarity and genre classification,” in Proc. 6th Int. Conf. Music Inf. Retrieval, 2005.

[5] P. Willett, J. M. Barnard, and G. M. Downs, “Chemical similarity searching,” J.Chem.Inf.Comput. Sci., vol. 38, no. 6, pp. 983–996, 1998.
[6] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006
. [7] B. Sch€olkopf and A. J. Smola, Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA, USA: MIT Press, 2002
[8] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in Proc. 5th Berkeley Symp. Math. Statist. Probability, 1967.
[9] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” ACM Comput. Surv., vol. 31, no. 3, pp. 264–323, 1999.
[10] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell.,vol.22,no.8,pp.888–905,Aug.2000.
[11] B. Sch€olkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” Neural Comput., vol. 13, no. 7, pp. 1443–1471, 2001.
[12] B. Sch€olkopf, A. J. Smola, and K.-R. M€uller, “Nonlinear component analysis as a kernel eigenvalue problem,” Neural Comput.,vol.10, no. 5, pp. 1299–1319, 1998.
[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Proc. 26th Int. Conf. Neural Inf. Process. Syst., 2013.
[14] L. van der Maaten and G. E. Hinton, “Visualizing non-metric similarities in multiple maps,” Mach. Learn., vol. 87, no. 1, pp. 33–55, 2012.
[15] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the SMO algorithm,” in Proc. 21st Int. Conf. Mach. Learn., 2004. 53
[16] S. Sonnenburg, G. R€atsch, C. Sch€afer, and B. Sch€olkopf, “Large scale multiple kernel learning,” J. Mach. Learn. Res., vol. 7, pp. 1531–1565, 2006.
[17] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” J. Mach. Learn. Res., vol. 10, pp. 207–244, 2009.
[18] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012.
[19] S. Lapuschkin, S. W€aldchen, A. Binder, G. Montavon, W. Samek, and K.-R. M€uller, “Unmasking Clever Hans predictors and assessing what machines really learn,” Nat. Commun., 2019.
[20] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. M€uller, Eds.,Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, vol. 11700. Berlin, Germany: Springer, 2019.
[21] Z. C. Lipton, “The mythos of model interpretability,” Commun. ACM, vol. 61, no. 10, pp. 36–43, 2018.
[22] G. Montavon, W. Samek, and K.-R. M€uller, “Methods for interpreting and understanding deep neural networks,” Digit. Signal Process., vol. 73, pp. 1–15, 2018.
[23] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. M€uller, “How to explain individual classification decisions,” J. Mach. Learn. Res., vol. 11, pp. 1803–1831, 2010.
[24] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. M€uller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” PLoS One, vol. 10, no. 7, Jul. 2015.
[25] M. T. Ribeiro, S. Singh, and C. Guestrin, ““Why should I trust you?”: Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016.
[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep

networks via gradient-based localization," in Proc. IEEE Int. Conf. Comput. Vis., 2017.
[27] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. M€uller, "Explaining nonlinear classification decisions with deep Taylor decomposition," Pattern Recognit., vol. 65, pp. 211–222, 2017.
[28] K. Simonyan and A. Zisserman, "Very deep convolutional net-works for large-scale image recognition," in Proc. Int. Conf. Learn. Representations, 2015. [29] M. Valleriani et al., "The emergence of epistemic communities in the sphaera corpus: Mechanisms of knowledge evolution," J. Historical Netw. Res., vol. 3, pp. 50–91, 2019. [30] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science, vol. 290, no. 5500, pp. 2323–2326, 2000.
[31] R. R. Coifman and S. Lafon, "Diffusion maps," Appl. Comput. Harmon. Anal., vol. 21, no. 1, pp. 5–30, Jul. 2006.
[32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in Proc. Eur. Conf. 54 Comput. Vis., 2014.
[33] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in Proc. Int. Conf. Learn. Representations, 2017. [34] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017.
[35] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in Proc. Int. Conf. Learn. Representations, 2014.
[36] D. Smilkov, N. Thorat, B. Kim, F. B. Viegas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," CoRR, 2017. [37] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in Proc. 34th Int. Conf. Mach. Learn., 2017.
[38] J. Kauffmann, K.-R. M€uller, and G. Montavon, "Towards explaining anomalies: A deep Taylor decomposition of one-class models," Pat-tern Recognit., vol. 101, 2020.
[39] B. Micenkova, R. T. Ng, X. Dang, and I. Assent, "Explaining outliers by subspace separability," in Proc. IEEE 13th Int. Conf. Data Mining, 2013.
[40] J. Kauffmann, M. Esders, G. Montavon, W. Samek, and K.-R. M€uller, "From clustering to cluster explanations via neural networks," CoRR, 2019.
[41] K. Xu et al., "Show, attend and tell: Neural image caption generation with visual attention," in Proc. 32nd Int. Conf. Mach. Learn., 2015. [42] M. Zheng, S. Karanam, T. Chen, R. J. Radke, and Z. Wu, "Learning similarity attention," CoRR, 2019.
[43] M. Tsang, D. Cheng, and Y. Liu, "Detecting statistical interactions from neural network weights," in Proc. Int. Conf. Learn. Representations, 2018
[44] T. Cui, P. Marttinen, and S. Kaski, "Recovering pairwise interactions using neural networks," CoRR, 2019.
[45] S. Leupold, "Second-order Taylor decomposition for explaining spatial transformation of images," master's thesis, Elect. Eng. Comput. Sci., Technische Universita€t Berlin, Berlin, Germany, 2017.
[46] M. Simon, E. Rodner, T. Darrell, and J. Denzler, "The whole is more than its parts? From explicit to implicit pose normalization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 3, pp. 749–763, Mar. 2020.
[47] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2015.
[48] J. D. Janizek, P. Sturmfels, and S. Lee, "Explaining explanations: Axiomatic feature interactions for deep networks," CoRR, 2020. 55

[49] C. Watkins, “Dynamic alignment kernels,” in Advances in Large Mar-gin Classifiers. Cambridge, MA, USA: MIT Press, 1999.

[50] K. Tsuda, M. Kawanabe, G. R€atsch, S. Sonnenburg, and K.-R. M€uller, “A new discriminative kernel from probabilistic models,” Neural Comput., vol. 14, no. 10, pp. 2397–2414, 2002.