Car Classification by ResNet and IRM

Annette Jing
Department of Statistics
Stanford University
ajing@stanford.edu

Abby Audet
Department of Electrical Engineering
Stanford University
aaudet@stanford.edu

I. Introduction and Motivation

One problem that machine learning has is that the performance of algorithms is usually highly dependent on how well the training dataset represents the population. However, in real life people often cannot obtain nice samples that are able to reflect the idiosyncrasies of each sub-population.

For classification of car images, a natural problem that arises is that certain makes have existed a lot longer than the others, so a naive machine learning algorithm may learn to not classify newer looking models to these makes. This leads to rapid deterioration of pre-trained models that are meant for long-term future use.

Our goal is to implement the recently invented technique "invariant risk minimization" with the aid of a deep residual network transformation to prioritize learning invariant features of car images, and hence hopefully improve the performance of classifiers that can be used to detect fraud in online car sales and automate aspects of car maintenance and repair.

II. Related Work

A. Invariant Risk Minimization (IRM)

Arjovsky, Bottout, Gulrajani, and Lopez-Paz (2020) [1] formulated the problem as follows:
Consider a large set of unseen but related environments $\mathcal{E}_{all}$. We have access to datasets $\{(x_i^e, y_i^e)\}_{i=1}^{n_e}$ collected under training environments $e \in \mathcal{E}_{tr} \subsetneq \mathcal{E}_{all}$. It is assumed that within each environment the data points are independent and identically distributed as (X^e, Y^e), i.e. $(x_i^e, y_i^e) \overset{iid}{\sim} (X^e, Y^e)$. Given some loss function ℓ, define the environmental risk corresponding to environment $e \in \mathcal{E}_{all}$ as

$$R^e(f) := E[\ell(f(X^e), Y^e)].$$

The goal is to find a classifier $\hat{f}$ that minimizes the *out-of-distribution risk*

$$R^{OOD}(f) := \max_{e \in \mathcal{E}_{all}} R^e(f). \tag{1}$$

Intuitively, we might want to pool the data together and utilize a traditional method that falls underneath the *empirical risk minimization (ERM)* paradigm first proposed by Vapnik [9] in 1992. The objective function of such methods can be written generally as

$$\sum_{e \in \mathcal{E}_{tr}} \sum_{i=1}^{n_e} \ell(f(x_i^e), y_i^e).$$

The problem with ERM under this setting is that they prioritize learning correlations that have low variance in large sub-populations over learning correlations that are consistent across different environments. In the case of our example, such algorithms are more likely to pick up signals such as how old-fashioned a car looks rather than features unique to each make.

A second method we may want to try is *robust learning* [2], which minimizes

$$R^{rob}(f) := \max_{e \in \mathcal{E}} R^e(f) - r_e,$$

where r_e are constants that serve as environmental baselines. However, [1] showed that this is actually equivalent to ERM under KKT differentiability.

If we are willing to assume that there exists a "data representation" δ such that $\delta(X^e)$ follows the same distribution across $e \in \mathcal{E}_{all}$, *domain adaptation* methods such as those proposed in [3] and [4] are good choices. While promising, the assumption that there is an "invariant data representation" is fairly strong and hard to verify.

Lastly, we can use the newly proposed invariant risk minimization (IRM) algorithm [1]. Instead of finding a data representation that follows the same distribution across all environments, IRM looks for one that can induce a classifier which is simultaneously optimal for all environments. IRM assumes that there exists a data representation $\delta : \mathcal{X} \to \mathcal{H}$ and a classifier $\hat{g} : \mathcal{H} \to \mathcal{Y}$ such that

$$\hat{g} \in \arg\min_{g: \mathcal{H} \to \mathcal{Y}} R^e(g \circ \delta)$$

for every $e \in \mathcal{E}_{all}$. It then makes sense to find the best set of functions that achieve this in our sample:

$$(\hat{\delta}, \hat{g}) = \arg\min_{\substack{\delta: \mathcal{X} \to \mathcal{H}, \\ g: \mathcal{H} \to \mathcal{Y}}} \sum_{e \in \mathcal{E}_{tr}} R^e(g \circ \delta)$$
$$\text{subject to } \hat{g} \in \arg\min_{g: \mathcal{H} \to \mathcal{Y}} R^e(g \circ \hat{\delta}) \quad \forall e \in \mathcal{E}_{tr}. \tag{2}$$

While this is a challenging bi-leveled optimization problem, Arjovsky et al. [1] showed that if we consider linear classifiers and non-linear data representations, (2) can be reduced to

$$\hat{\delta} = \arg\min_{\delta: \mathcal{X} \to \mathcal{Y}} \sum_{e \in \mathcal{E}_{tr}} R^e(\delta) + \lambda \left\| \nabla_g R^e(g \circ \delta) \Big|_{g=1} \right\|_2^2, \tag{3}$$

where $\lambda \in [0, \infty)$ is a regularization parameter that balances the ERM term and the penalty term for violations of

invariance. Here, the classifier g is set to 1 and the data representation δ has become the de facto predictor.

To close this section off, we note that the authors of [1] have derived a condition under which the classifier defined by (3) will also minimize the out-of-distribution risk (1). While we will not delve into the mathematical details of the condition, we note that it is a restriction on how much the training environments can be co-linear, which makes intuitive sense as the algorithm needs a sufficiently diversified $\mathcal{E}_{tr}$ to be able to tell which underlying features are truly invariant.

B. Deep Residual Network (ResNet)

One problem with using IRM defined by (3) for image classification is the assumption that the optimal classifier is linear, which is highly unlikely to be true. To mitigate this problem, we perform feature selection using a pre-trained deep residual network [6] that returns a 1000-dimensional feature vector before running IRM.

III. Datasets and Evaluation Metrics

A. Dataset

We use the Car Connection Picture Dataset, which contains 323 classes consisting of 64,467 images, covering models manufactured between 1990 and 2020 [5].

B. Classification categories

Since our goal is out-of-distribution prediction, we split the dataset into two parts: pictures of cars manufactured before 2010 and those manufactured after 2010 (including 2010). The task is to predict the makes of pictures of cars manufactured before 2010 using models trained on pictures of cars manufactured after 2010.

For the implementation of IRM, we further split the training dataset into three "environments" defined by which of $2010 \sim 2013$, $2014 \sim 2017$, and $2018 \sim 2020$ the car's year of manufacturing falls in. Hence, we have in total 4 data splits.

There are in total 12 makes that are present in each split: Chrysler, Jeep, GMC, Aston Martin, Honda, Chevrolet, Ford, Land Rover, Mitsubishi, Acura, BMW, and Lincoln. We only keep pictures of cars belonging to makes that are present in each split. The size of the splits are given below:

Testing Dataset	**Training & Validation Dataset**		
2000 ∼ 2009	**2010 ∼ 2013**	**2014 ∼ 2017**	**2018 ∼ 2020**
1463	7859	9522	5894

C. Image Pre-processing

We resize all images to be 256 pixels on their shortest size and then crop the middle section to get 256×256 square images.

D. Loss Function & Evaluation Metrics

As the task it image classification, we use the cross-entroy loss function implemented by torch.nn.CrossEntropyLoss.

The methods are evaluated based on their overall accuracy across all categories.

IV. Methods

A. Baselines

For baselines, we want to use methods we have learned in class that follow the ERM principle.

a) Naive Bayes: To apply naive Bayes to image classification, we treat the pixels in an image as words in a message. Since each pixel can only take integer value between 0 and 255, our dictionary is $\{0, ..., 255\}$. Additionally, the output would be categorical instead of binary because there are more than just two types of car makes and models.

b) ResNet Logistic Regression: As described in subsection II-B, we transform the pixels of each picture by a pre-trained ResNet to get a 1000-dimensional feature vector, which is then used as an input in a regular multinomial logistic regression.

B. ResNet Invariant Risk Minimization

As before, the pixels are transformed by a pre-trained ResNet into feature vectors.

We implemented an algorithm that finds the practical IRM estimator defined in (3), where, given a mini-batch $\{(X_j^2, Y_j^e)\}_{j=1}^{2b}$, the L^2 penalty term is estimated by

$$\sum_{i=1}^{b} \nabla_w \ell(w \cdot \Phi(X_{2i}^e), Y_{2i}^e))|_{w=1} \cdot \nabla_w \ell(w \cdot \Phi(X_{2i-1}^e), Y_{2i-1}^e))|_{w=1}.$$

This is an unbiased estimate for $\|\nabla_w R^e(w \cdot \Phi)|_{w=1}\|_2^2$ in the penalty term of IRM as long as the mini-batch is scrambled beforehand. The regularization parameter λ is chosen by cross-validation.

C. Multilayer Fine-tuned ResNet

The two aforementioned methods involving ResNet only "fine tunes" the last layer of the network, whereas there is an existing fine-tuning algorithm in the fast.ai library that performs fine-tuning in multiple layers of the deep neural network. This method serves as a benchmark of how one of the most groundbreaking image classification algorithms performs on this dataset.

V. Experimental Results

For the confusion matrices in this section, the vertical axis presents the true categories, and the horizontal axis presents the predicted categories.

A. Baseline: ResNet Logistic Regression

Since the naive Bayes methods performs badly even in-sample, we only include ResNet logistic regression as our baseline.

The ResNet logistic regression has an overall accuracy of 21.3%.

B. ResNet Invariant Risk Minimization

ResNet Invariant Risk Minimization has an overall accuracy of 23.1%, which is marginally better than ResNet Logistic Regression's performance.

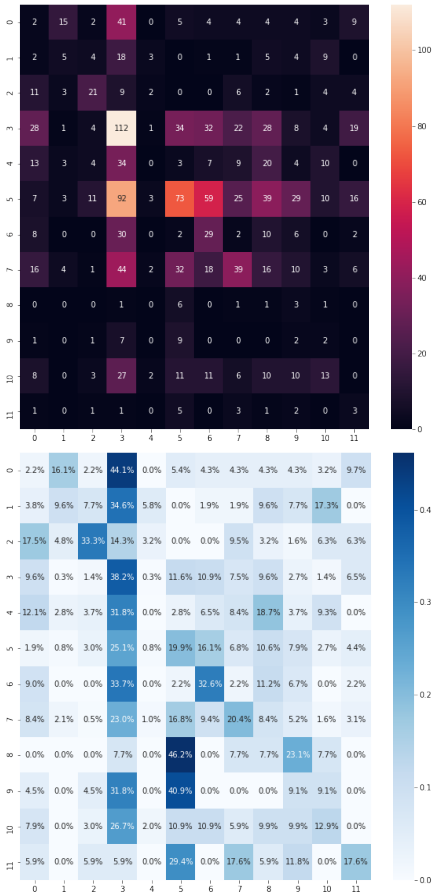
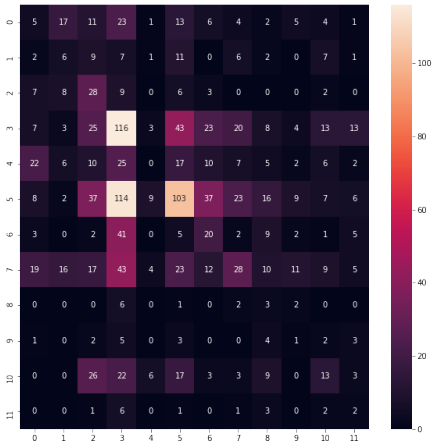


Fig. 1. Confusion matrices for ResNet logistic regression.



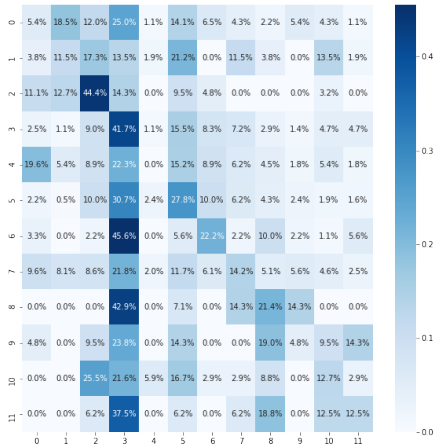


Fig. 2. Confusion matrices for ResNet IRM.

C. Multilayer Fine-tuned ResNet

Multilayer Fine-tuned ResNet has an overall accuracy of 33.7% - a lot better than the previous two methods, but still has room for improvement.

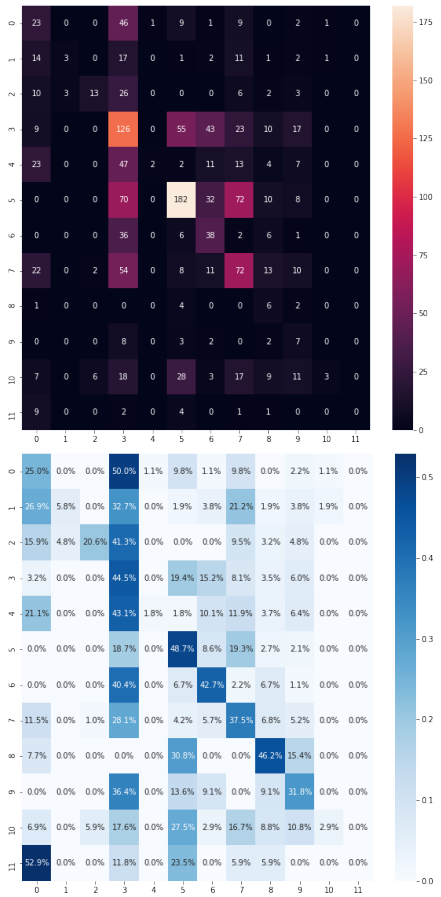


Fig. 3. Confusion matrices for multilayer fine-tuned ResNet.

VI. Conclusion

In our experiments, IRM performs just marginally better than our baseline logistic regression. This can be due to several reasons:

- The practical IRM objective function defined by (3) is only equivalent to the general objective function (2) if we assume the optimal classifier to be linear. Even with ResNet, this assumption can still be too restrictive for image classification. A recent paper by Rosenfeld, Ravikumar, and Risteski [7] demonstrates that IRM can fail catastrophically unless the test data are sufficiently similar to the training distribution under non-linearity.
- Our IRM implementation does 1000 passes across the training dataset, which may not be enough for the loss function to converge.

The multilayer fine-tuned ResNet does much better than ResNet logistic regression and ResNet IRM, but it performance is still lackluster. This may be due to the lack of data augmentation and preprocessing.

Code

The code can be found at: https://github.com/AnnetteJing/CS229-Final-Project-Car-Classification

Contribution

Annette: Literature review, experiment design, implementation of ResNet logistic regression, ResNet IRM, and multilayer fine-tuned ResNet, report.
Abby: Data exploration, implementation of naive Bayes, poster.

References

[1] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. "Invariant risk minimization," arXiv:1907.02893, 2020.
[2] J. A. Bagnell. "Robust supervised learning," AAAI, 2005.
[3] J. Blitzer, R. McDonald, and F. Pereira. "Domain adaptation with structural correspondence learning," Conference on Empirical Methods in Natural Language Processing, P. 120-128, 2006.
[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky. "Domain adversarial training of neural networks," JMLR, 2016.
[5] N. Gervais. The Car Connection Picture Dataset. [Online], 2020. Available: https://github.com/nicolas-gervais/predicting-car-price-from-scraped-data/tree/master/picture-scraper.
[6] K. He, X. Zhang, S. Ren, J. Sun. "Deep Residual Learning for Image Recognition," CVPR, 2016.
[7] E. Rosenfeld, A. Risteski. "The Risks of Invariant Risk Minimization," arxiv:2010.05761, 2020.
[8] F. Tafazzoli, K. Nishiyama, and H. Frigui. "A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition," CVPR Workshops, 2017.
[9] V. Vapnik. "Principles of risk minimization for learning theory," NIPS, 1992.