

# Classifier Adaptation at Prediction Time

Xuewen Yang

June 22 2018

## Abstract

*Classifiers for object categorization are usually evaluated by their accuracy on a set of i.i.d. test examples. This provides us with an estimate of the expected error when applying the classifiers to a single new image. In real application, however, classifiers are rarely only used for a single image and then discarded. Instead, they are applied sequentially to many images, and these are typically not i.i.d. samples from a fixed data distribution, but they carry dependencies and their class distribution varies over time.*

*In this work, the author argues that the phenomenon of correlated data at prediction time is not a nuisance, but a blessing in disguise. They describe a probabilistic method for adapting classifiers at prediction time without having to re-train them. They also introduce a framework for creating realistically distributed image sequences, which offers a way to benchmark classifier adaptation methods. Experiments on the ILSVRC2010 and ILSVRC2012 datasets show that adapting object classification systems at prediction time can significantly reduce their error rate, even with no additional human feedback.*

## 1. Classifier Adaptation at Prediction Time

This paper formally introduces the problem setting and describes a simple yet effective method to adapt pre-trained classifiers to a new distribution.

### 1.1. Notation

The author works in a multi-class classification setting, where the goal is to assign outputs. In contrast to the classical domain adaptation setting, they assume that a pre-trained multiclass classifier is available, which is probabilistic, i.e.

They are interested in improving the accuracy at prediction time by adapting the predicted class scores, not in changing the training procedure. This setting has the advantage that it is applicable to classifiers of arbitrary parametric form, including e.g., convolutional networks [1], random forests [2] or one-versus-rest support vector machines [3]

with Platt scaling. It is sufficient that the functions  $f_y$  are available in executable form, and we also do not need access to the original training set. This is important for real world application, where the training set could be too large to keep around for prediction time, or might be commercially valuable and not available to customers.

### 1.2. Classifier adaptation

In this section the author introduces their first contribution, a technique for on the fly classifier adaptation. First, they assume that the distribution at prediction time is fixed but differs from the distribution at training time by a change in class proportions. The objects themselves, however, do not change their visual appearance, i.e. If  $\Pi$  were known they could immediately derive an optimal adaptation rule in closed form, see [6] for a detailed derivation.

### 1.3. On the fly estimation of the class proportions

In practice, the class proportions,  $\Pi$ , are unknown, so we cannot simply compute the class-prior adaptation and predict optimally. Instead, at any time  $t = 1, 2, \dots$  we form an estimate of the class proportions,  $\Pi^t$ , using the available data, and define the correspondingly adapted classifier.

## 2. A Benchmark for Classifier Adaptation

In existing image categorization benchmarks the training and test sets have independent samples with identical data distributions. Since this i.i.d. condition is typically not fulfilled when one just collects images, e.g. from the Internet, it is enforced artificially. For example, for the ImageNet challenges, training and test sets are created by randomly selecting a prescribed number of images for each class from a larger annotated corpus [5].

### 2.1. Generating Realistic Image Sequences

The main idea for re-introducing dependencies into the test data is to change the sampling process by which the test sets are created. Instead of sampling a fixed number of images independently for each class, the author creates sequences of dependent samples using a two-stage latent variable construction: 1) we create sequences of class labels, us-

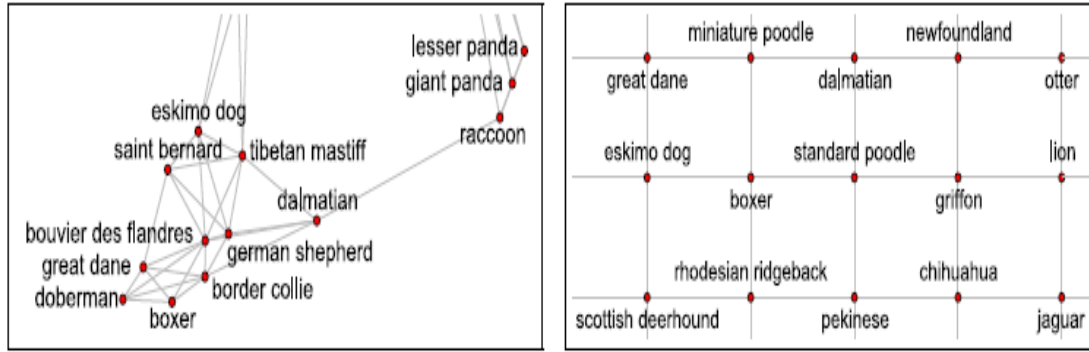


Figure 1. Excerpts of the MDS (left) and KS (right) graphs for the ILSVRC2010 classes with WordNet distance (not all edges are drawn).

ing one of three techniques described below, 2) for each class label in the sequence, we sample one example image of the respective class out of the image corpus.

The first method, MDS, relies on (metric) multidimensional scaling. It assigns a 2D coordinate tuple to each class such that all pairwise distances are preserved as well as possible. The result is a highly heterogeneous class graph in which groups of similar classes form dense clusters, and large gaps occur between group of classes that are pairwise dissimilar, see Figure 1(left).

The KS method is based on kernelized sorting [4]. It arranges the classes on a 2D grid based on their pairwise distance to each other. Neighboring classes in the grid are usually semantically similar, but all neighbors have the same distances, so there are no clusters or gaps, see Figure 1(right).

## References

- [1] Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2009. 1
- [2] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 2012. 1
- [3] C. H. Lampert. Kernel methods in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 2009. 1
- [4] N. Quadrianto, A. J. Smola, L. Song, and T. Tuytelaars. Kernelized sorting. *IEEE TPAMI*, 2010. 2
- [5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1
- [6] M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 2002. 1