Astronomy and Computing 12 (2015) 67–72

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

Nearest neighbor density ratio estimation for large-scale applications in astronomy

J. Kremer [a,*], F. Gieseke [b], K. Steenstrup Pedersen [a,c], C. Igel [a,c]

[a] Department of Computer Science, University of Copenhagen, Sigurdsgade 41, 2200 Copenhagen, Denmark
[b] Institute for Computing and Information Sciences, Radboud University Nijmegen, Toernooiveld 212, 6525 EC Nijmegen, Netherlands
[c] Space Science Center, University of Copenhagen, 2100 Copenhagen, Denmark

ARTICLE INFO

Article history:
Received 27 March 2015
Accepted 17 June 2015
Available online 29 June 2015

Keywords:
Methods: data analysis
Methods: statistical
Galaxies: distances and redshifts
Sample selection bias
Nearest neighbors
Large-scale learning

ABSTRACT

In astronomical applications of machine learning, the distribution of objects used for building a model is often different from the distribution of the objects the model is later applied to. This is known as sample selection bias, which is a major challenge for statistical inference as one can no longer assume that the labeled training data are representative. To address this issue, one can re-weight the labeled training patterns to match the distribution of unlabeled data that are available already in the training phase. There are many examples in practice where this strategy yielded good results, but estimating the weights reliably from a finite sample is challenging. We consider an efficient nearest neighbor density ratio estimator that can exploit large samples to increase the accuracy of the weight estimates. To solve the problem of choosing the right neighborhood size, we propose to use cross-validation on a model selection criterion that is unbiased under covariate shift. The resulting algorithm is our method of choice for density ratio estimation when the feature space dimensionality is small and sample sizes are large. The approach is simple and, because of the model selection, robust. We empirically find that it is on a par with established kernel-based methods on relatively small regression benchmark datasets. However, when applied to large-scale photometric redshift estimation, our approach outperforms the state-of-the-art.

1. Introduction

In many machine learning applications labeled (training) and unlabeled (test) data do not follow the same distribution. One reason can be that the labeled patterns have not been sampled randomly. In astronomy such a sample selection bias arises because objects that are expected to show more interesting properties are preferred when it comes to costly high-quality spectroscopic follow-up observations; other objects whose scientific value may not be that obvious (e.g., seemingly star-like objects) may be overlooked (Mortlock et al., 2011). One way to address this bias is to weight the labeled training sample according to the ratio between the two probability distributions (Huang et al., 2007). As this true ratio is usually not available, one has to estimate it from a finite sample. The crucial point is to control the variance of the estimator. Empirically, it seems promising to reduce the variance of the estimator by accepting a slightly higher bias (Sugiyama et al., 2008). This gives rise to ratio estimators that, in practice, perform better than the naïve approach of estimating the two densities separately.

In this work, we improve a simple nearest neighbor density ratio estimator (Lima et al., 2008) by combining it with a principled way of performing model selection (Sugiyama and Müller, 2005). The approach compares well to established kernel-based estimators on a variety of standard, small-sized regression datasets. Furthermore, by selecting proper hyperparameters and by taking huge amounts of patterns into account, we experimentally show that the estimator yields better results compared to the state-of-the-art on a large-scale astronomical dataset.

Let each data point be represented by a feature vector $\mathbf{x}$ from a domain $\mathcal{X}$ with a corresponding label $\mathbf{y}$ from a domain $\mathcal{Y}$. We consider scenarios in which the learner has access to some labeled (source) data S sampled from $p_s(\mathbf{x}, \mathbf{y})$ and a large sample of unlabeled (target) data T sampled from $p_t(\mathbf{x}, \mathbf{y})$. While $p_s(\mathbf{x}, \mathbf{y})$ and $p_t(\mathbf{x}, \mathbf{y})$ may not coincide, we assume that $p_s(\mathbf{y}|\mathbf{x}) = p_t(\mathbf{y}|\mathbf{x})$ for all $\mathbf{x}$ and that the support of p_t is a subset of the support of p_s. This is usually referred to as *covariate shift*, a particular type of *sample selection bias*. In this case the probability density ratio

* Corresponding author.
E-mail address: jan.kremer@di.ku.dk (J. Kremer).

http://dx.doi.org/10.1016/j.ascom.2015.06.005
2213-1337/

between target and source distribution at a given point reduces to $\beta(\mathbf{x}) = \frac{p_t(\mathbf{x})}{p_s(\mathbf{x})}$.

Different strategies have been proposed to address covariate shift, such as finding a common feature space or re-weighting the source patterns. The latter is conceptually simple, and there are several approaches to estimate appropriate weights via density ratio estimation (Huang et al., 2007; Lima et al., 2008; Sugiyama and Müller, 2005; Bickel et al., 2007; Cortes et al., 2008; Loog, 2012; Quionero-Candela et al., 2009; Izbicki et al., 2014; Kanamori et al., 2009). These methods are, for example, based on reducing the problem to probabilistic classification between the target and source dataset (Bickel et al., 2007), on using kernel-based methods to match means in an induced Hilbert space (Huang et al., 2007), or on using nearest neighbor queries to estimate the mismatch between the densities by counting patterns in local regions (Lima et al., 2008; Loog, 2012). It is crucial to control the variance of such an estimator via regularization. Depending on the algorithm at hand, the regularization can take the form of, for example, a kernel bandwidth (Huang et al., 2007), the rank of a low-rank kernel matrix approximation (Izbicki et al., 2014), or a weight norm (Kanamori et al., 2009). The involved parameters are often set by heuristics such as the median of pairwise distances for the kernel bandwidth (Schölkopf and Smola, 2002). As an alternative, Sugiyama and Müller (2005) suggest a model selection criterion that is unbiased under covariate shift. In the following, we employ this criterion for selecting the neighborhood size of the nearest neighbor estimator via cross-validation. Then, we empirically show that the resulting algorithm can outperform the computationally more expensive state-of-the-art kernel-based estimator due to its ability to consider larger samples in less time.

This article is structured as follows: in Section 2 we briefly discuss two state-of-the-art kernel-based estimators that serve as a baseline in our experimental evaluation. In Section 3 we present a nearest neighbor-based density ratio estimator and show how it can be extended to perform automatic model selection. In Section 4 we evaluate the proposed nearest neighbor density ratio estimator with integrated model selection in comparison to other methods on a medium-sized regression benchmark and on a large-scale astronomical dataset for photometric redshift estimation. In Section 5 we conclude and give possible directions for future work.

2. Kernel-based density ratio estimation

In density ratio estimation, kernel-based estimators are considered the state-of-the-art (Sugiyama et al., 2010). Among these, kernel mean matching (KMM) (Huang et al., 2007) and the spectral series estimator (Izbicki et al., 2014) have shown to perform particularly well.

Given some input space $\mathcal{X}$, a kernel is a positive semi-definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for which $\forall x, z \in \mathcal{X} : k(x, z) = \langle \Phi(x), \Phi(z) \rangle_{\mathcal{H}}$, where $\Phi : \mathcal{X} \to \mathcal{H}$ maps elements of the input space to a kernel-induced Hilbert space $\mathcal{H}$ (Aronszajn, 1950). Kernel mean matching aims at matching the means of two distributions in $\mathcal{H}$ by solving the problem

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \beta_i \Phi(\mathbf{x}_i^{(s)}) - \frac{1}{N_t} \sum_{i=1}^{N_t} \Phi(\mathbf{x}_i^{(t)}) \right\|_{\mathcal{H}}^2 \tag{1}$$

$$\text{subject to } \beta_i \in [0, B] \quad \text{and} \quad \left| \sum_{i=1}^{N_s} \beta_i - N_s \right| \leq N_s \epsilon, \tag{2}$$

where N_s is the number of source domain patterns and N_t is the number of target domain patterns. The parameter B restricts the maximum possible weight and ϵ bounds the deviation of the mean weight from 1. Cortes et al. (2008) show that the solution to Eq. (1) converges with high probability to the true density ratio if the kernel induced by $\Phi(\mathbf{x})$ is universal (Steinwart and Christmann, 2008). The kernel function, which implicitly defines Φ and $\mathcal{H}$, is typically chosen from a parameterized family of functions, and the kernel parameters are parameters of KMM-based approaches.

The spectral series estimator (Izbicki et al., 2014), although motivated differently, minimizes an unconstrained version of Eq. (1) for computing training weights. Instead of bounding the weights via B and their mean via ϵ, the solution is regularized by the rank J of a low-rank approximation of the kernel Gram matrix between training points—which results when expanding Eq. (1). Unlike KMM, the spectral series estimator can compute weights not only for the source sample, but also for arbitrary patterns. This allows for selecting the kernel parameters and J via cross-validation, as we shall see later.

Negative theoretical results in the analysis of weighting methods (Ben-David et al., 2010; Ben-David and Urner, 2012) suggest that sample sizes have to be prohibitively large to guarantee reliable weights. However, empirically it has been found that re-weighting often does improve results. Our method is motivated by typical tasks in astronomy, where we deal with large labeled samples and huge unlabeled samples in feature spaces of relatively low dimensionality (e.g., up to $\mathbb{R}^{10}$). For such rather benign scenarios, we aim at estimating weights with high accuracy by taking into account hundreds of thousands of labeled and unlabeled patterns. However, both KMM as well as the spectral series estimator involve $|S| \times |T|$ kernel matrices in their general form. Thus, they are not directly applicable to scenarios with hundreds of thousands of patterns. Special cases might be addressed in a more efficient way. Still, the general cases with non-linear kernel functions involve the computation of such kernel matrices and, depending on the method, quadratic programming, matrix inversion, or eigenvalue decomposition, which exhibit at least a quadratic running time (Bern and Eppstein, 2001; Golub and Van Loan, 1989; Kojima et al., 1989). Therefore, we are considering nearest neighbor-based density ratio estimation, which can be implemented more efficiently.

For the matrix decompositions in the spectral series estimator we used an efficient $\mathcal{O}(n^2)$-algorithm (Dhillon, 1998). Both, decomposition as well as the nearest neighbor search, could be sped up by using approximation schemes (e.g., see Arya et al., 1994; Halko et al., 2011), but we decided not to introduce such approximations with corresponding hyperparameters in our study.

3. Nearest neighbor density ratio estimation revisited

We consider the algorithm proposed by Lima et al. (2008) to estimate appropriate ratios via nearest neighbor queries, see Algorithm 1. The efficiency of the approach is ensured via the use of k–d trees. For the sake of completeness, we briefly sketch how these spatial data structures can be used to speed up nearest neighbor search before outlining the details of the density ratio estimator.

3.1. Nearest neighbor search in low dimensions

A classical *k –d tree* (Bentley, 1975) is a binary tree constructed from a d-dimensional point set $S \subset \mathbb{R}^d$. The inner nodes correspond to hyperplanes splitting the data in $\mathbb{R}^d$ and the leaf nodes define a partitioning of S. The tree can be built recursively in $\mathcal{O}(|S| \log |S|)$ time. Starting from the root node numbered by 0 and $S_0 = S$, each inner node v with children u and w partitions the data S_v into two almost equal-sized subsets S_u and S_w. If S_v contains only a single point (or a predefined number of points), v becomes a leaf node. At tree level j, the datasets are split according to the median in dimension $j \bmod d + 1$.