RESEARCH NOTE

Understanding covariate shift in model performance [version 1; peer review: 2 approved with reservations]

Georgia McGaughey, W. Patrick Walters, Brian Goldman

Modeling & Informatics, Vertex Pharmaceuticals, Boston, MA, USA

V1 **First published:** 07 Apr 2016, 5(Chem Inf Sci):597
<https://doi.org/10.12688/f1000research.8317.1>
Second version: 17 Jun 2016, 5(Chem Inf Sci):597
<https://doi.org/10.12688/f1000research.8317.2>
Latest published: 17 Oct 2016, 5(Chem Inf Sci):597
<https://doi.org/10.12688/f1000research.8317.3>

Abstract

Three (3) different methods (logistic regression, covariate shift and k-NN) were applied to five (5) internal datasets and one (1) external, publicly available dataset where covariate shift existed. In all cases, k-NN's performance was inferior to either logistic regression or covariate shift. Surprisingly, there was no obvious advantage for using covariate shift to reweight the training data in the examined datasets.

Keywords

covariate shift , model building , ChEMBL , logistic regression , k-NN

This article is included in the **Chemical Information Science** gateway.

Open Peer Review

Approval Status

	1	2
version 3		
(revision)		
17 Oct 2016	view	
version 2		
(revision)		
17 Jun 2016	view	view
version 1		
07 Apr 2016	view	view

1. **Robert Sheridan**, Merck Research Laboratories, Rahway, USA

2. **Martin Vogt** , University of Bonn, Bonn, Germany

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Brian Goldman (brian_goldman@vrtx.com)

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2016 McGaughey G *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

How to cite this article: McGaughey G, Walters WP and Goldman B. **Understanding covariate shift in model performance [version 1; peer review: 2 approved with reservations]** F1000Research 2016, 5(Chem Inf Sci):597 <https://doi.org/10.12688/f1000research.8317.1>

First published: 07 Apr 2016, 5(Chem Inf Sci):597 <https://doi.org/10.12688/f1000research.8317.1>

Introduction

A common prerequisite in supervised learning algorithms is that the training and prediction data arise from the same distribution and are independently and identically distributed (*iid*)¹. Intuitively this is justified, as one should not expect to learn a classifier on one distribution of examples and apply it to accurately predict labels of examples drawn from a different distribution. Covariate shift is a machine learning technique that can be utilized in supervised learning when the training and prediction distributions are known to differ, but the concept being learned remains stationary. A recent book provides an excellent overview of the current state of the art in covariate shift methods².

Covariate shift frequently occurs during the drug discovery process where learning systems are built to predict physicochemical properties of interest. Initially a chemistry team may focus on a particular chemical series, and information from this series is used to train a learning system. As the project progresses, the chemistry team may refocus their efforts on a new, structurally distinct series. The accuracy of prospective computational predictions on the new series may be compromised as these molecules originate from a distribution that is distinct from the molecular set used to train the learning tool.

For example one may wish to build a learning system to predict hERG activity (unwanted cardiovascular toxicity). Initially the computational tool is trained using series A but must now predict on series B. The concept “binding to hERG” is fixed, however the area of interest has transitioned from chemical series A to chemical series B. The feature vectors describing these two sets are likely related but potentially different; and as such, their covariates have shifted. Put more mathematically, the probability of observing a feature vector from the prediction set is different from the

probability of observing a feature vector from the training set. That is, the training and prediction sets are *non-iid*. A well-constructed learning system will recognize that predictions on series B are outside the “domain of applicability” of the model and predict with low confidence. The covariate-shift method attempts to adjust the domain of applicability so that it is more aligned with the prediction set.

Covariate shift methods typically reweight instances in the training data so that the distribution of training instances is more closely aligned with the distribution of instances in the prediction set. This is accomplished by providing more weighting during model building to an instance in the training set that are similar to an instance in the prediction set. It has been shown³ that the appropriate importance weighting factor $w(x)$ for each instance “ x ” in the training set is:

$$w(x) = \frac{p_p(x)}{p_t(x)} \quad (1)$$

where $p_t(x)$ is the probability of seeing instance x in the training set and $p_p(x)$ is the probability of seeing x in the prediction set. It is important to note that only the feature vector values (not their labels) are used in reweighting. The importance weighting scheme is intuitively understandable. If the probability of seeing a particular instance from the training set in the prediction is very small, then this instance should carry little weight during the training process and consequently have little effect on the decision function.

Figure 1 plots two Gaussian distributions and $w(x)$. If instances from the red distribution are used for training a classifier to predict on an instance from the green distribution then the blue curve gives the importance of each instance. Note the increased importance for instances from the training distribution overlapping with high-density regions of the prediction distribution.

Figure 1. Train, prediction and importance.

Methods

For our experiments, we use a logistic regression classifier where each training instance is weighed by its importance $w(x)$. For the calculation of $w(x)$ we use the Kullback-Leibler Importance Estimation Procedure (KLIEP) method developed by Sugiyama⁴. The KLIEP method is based on the Kullback-Leibler divergence theorem and attempts to find weights to minimize the divergence from $p_{train}(x)$ to $p_{predict}(x)$. Briefly, the importance is modeled as a linear function:

$$\hat{w}(x) = \sum_{i=1}^b \alpha_i * \varphi_i(x) \quad (2)$$

The α_i are the weights to be learned and φ_i the basis functions. The importance weight from Equation 1 can be rearranged and used to estimate the probability of observing a feature vector in the predictive set.

$$\hat{p}_p(x) = w(x)p_t(x) \quad (3)$$

The KL divergence from $p_p(x)$ to its estimate $\hat{p}_p(x)$ can then be expressed as:

$$KL[p_p(x) || \hat{p}_p(x)] = \int p_p(x) \log \left(\frac{p_p(x)}{p_t(x)\hat{w}(x)} \right) dx$$

After algebraic manipulation, removing terms independent of $\hat{w}(x)$ and adding constraints to ensure proper normalization, a final objective function to be maximized can be derived as (see 4 for details):

$$\begin{aligned} & \text{maximize} \left[\sum_{j=1}^{n_p} \log \left(\sum_{i=1}^b \alpha_i \varphi_i(x_j) \right) \right] \\ & \text{subject to: } \sum_{j=1}^{n_t} \sum_{i=1}^b \alpha_i \varphi_i(x_j) = 1 \\ & \text{and } \alpha_1, \alpha_2, \dots, \alpha_b \geq 0 \end{aligned}$$

The resulting problem is convex and can be solved using standard optimization techniques. The result is an expression for $w(x)$ that allows calculating weights for a training instance x . These weights can then be incorporated when training a classifier to obtain a co-variate shifted version of the classifier.

Toy example

To demonstrate the use of covariate shift methods, we repeated a simple toy experiment as detailed in 3. Figure 2 graphically displays the results we obtained.

The red training points are drawn from two (2) two-dimensional Gaussian distributions representing a class 1 and a class 2. The green prediction points are drawn from a slightly rotated version of the training distributions. The red line plots the classifier obtained when training on only the training points; the green line plots the classifier trained on both the training and prediction points (the optimal classifier in this case). The blue line plots the classifier trained on the training data that was weighted by the importance factor as estimated by the KLIEP method. Note how the blue line is shifted towards the optimal classifier, demonstrating the effect of the KLIEP algorithm and covariate shift.

Experiments

Dataset 1. The BACE IC50 data derived from the ChEMBL database

<http://dx.doi.org/10.5256/f1000research.8317.d117882>

Units are in nM.

Using the Python programming language, we implemented the KLIEP method combined with logistic regression and applied it to five different in-house ADME (absorption, distribution, metabolism and excretion) datasets. We compare KLIEP+Logistic Regression (KL+LR) to Logistic Regression (LR) and a k-NN classifier ($k=5$)⁵.

Figure 2. Classification using covariate shift.

For each dataset the molecules were sorted by compound registration date. The first 75% of the data comprised the master training set while the remainder formed the master prediction set. Temporal ordering of the data represents the evolving coverage of chemical space by drug discovery projects and consequently captures the natural “shifting” of the covariates. Classifier performance statistics are generated by performing twenty different runs, each on a random 80% of the master files. Performance statistics for each classification task are then obtained by averaging the results of the twenty individual folds. In all cases, OpenEye⁶ path fingerprints are used as feature vectors. We experimented with different fingerprints provided by OpenEye (MACCS 166 bit structural keys and circular fingerprints) and found that they had no significant effect on the outcome.

To ensure the data was amenable to covariate shift we generated classifiers separating “training” from “prediction” data. Figure 3 shows performance of LR on this separation task. For each dataset we are able to compute highly accurate classifiers. This indicates that the training and prediction data are drawn from different distributions and hence are appropriate for covariate shift methods. This is a necessary condition for covariate shift but does not imply model improvement over unweighted data.

Figure 4 compares the performance of KL+LR, LR and k-NN on the five (5) datasets. One can see from the graph that KL+LR failed to provide any statistical improvement over standard LR.

We extended the study to include an external dataset provided by ChEMBL^{7,8} such that others could use their own fingerprints and

independently support or refute our claims. We chose the beta secretase IC₅₀ data as it is a well established biochemical screen, highly accurate and contains > 7000 data points crossing multiple orders of magnitude, which are publically available.

Analogous to the internal datasets, we were able to demonstrate that the data could be separated and thereby appropriate for application of the covariate shift algorithm. Unfortunately, the outcome was the same: that is, as measured by overall classifier accuracy, there was no statistical advantage for reweighting the training set.

A possible explanation for the failure of the covariate shift method to provide a boost in predictive performance could be that the calculated importance weights are all similar. This would cause each training example to exert the same influence on the decision function and thus the importance weighting would have no effect. This was not the case. Figure 5 plots the cumulative distribution function of the importance weight for the training set compound. The plot demonstrates that weights are distributed across a range of classifier performance.

Conclusions

We have applied the KLIEP method to five (5) internal data sets and one (1) external data set where covariate shift was evident. Although KL+LR was an advantage over k-NN, there is no statistical advantage of reweighting the training dataset. We are surprised with this outcome and are currently exploring other datasets where application of covariate shift may improve the predictions.

Figure 3. Classification of train versus test.

Figure 4. Performance of KL+LR, LR and k-NN.

Figure 5. Cumulative distribution function.

Data availability

F1000Research: Dataset 1. The BACE IC50 data derived from the ChEMBL database, [10.5256/f1000research.8317.d117882](https://doi.org/10.5256/f1000research.8317.d117882)⁹

the beta-secretase data set and contributed to the experimental design. PW provided oversight.

Competing interests

No competing interests were disclosed.

Author contributions

BG conceived the study. BG designed the experiments and carried out the research. GM wrote the manuscript and provided

Grant information

The author(s) declared that no grants were involved in supporting this work.

References

1. Zadrozny B: **ICML 04 Proceedings of the twenty-first international conference on Machine learning**. ACM New York, 2014.
2. Quiñero-Candela J, Sugiyama M, Schwaighofer A, *et al.*: **Dataset Shift In Machine Learning**. MIT Press, Cambridge, Massachusetts, 2009.
[Reference Source](#)
3. Shimodaira H: **Improving predictive inference under covariate shift by weighting the log-likelihood function**. *J Stat Plan Inference*. 2000; **90**(2): 227–244.
[Publisher Full Text](#)
4. Sugiyama M, Suzuki T, Nakajima S, *et al.*: **Direct importance estimation for covariate shift adaptation**. *Ann Inst Stat Math*. 2008; **60**(4): 699–746.
[Publisher Full Text](#)
5. **A Matlab implementation of the KLIEP algorithm is freely available:**
<http://www.ms.k.u-tokyo.ac.jp/software.html#KLIEP>
[Reference Source](#)
6. **OpenEye Scientific Software**. (version 2014.Feb), 9 Bisbee Ct, Suite D, Santa Fe NM, USA.
[Reference Source](#)
7. Gaulton A, Bellis LJ, Bento AP, *et al.*: **ChEMBL: a large-scale bioactivity database for drug discovery**. *Nucleic Acids Res*. 2012; **40**(Database Issue): D1100–1107.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
8. Bento AP, Gaulton A, Hersey A, *et al.*: **The ChEMBL bioactivity database: an update**. *Nucleic Acids Res*. 2014; **42**(Database Issue): D1083–1090.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
9. McGaughey G, Walters WP, Goldman B: **Dataset 1 in: Understanding covariate shift in model performance**. *F1000Research*. 2016.
[Data Source](#)

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 25 April 2016

<https://doi.org/10.5256/f1000research.8943.r13393>

© 2016 Vogt M. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Martin Vogt

Department of Life Science Informatics, Bonn-Aachen International Center for Information Technology (B-IT), LIMES (Life & Medical Sciences Institute) Program Unit Chemical Biology and Medicinal Chemistry, University of Bonn, Bonn, Germany

The study investigates the influence of accounting for covariate shift in classification performance using logistic regression models. Overall, this short paper is very well and clearly written, however the method section should be expanded (see below). Although no increase in performance could be established by accounting for covariate shift, it provides an excellent basis for further investigations.

Suggestions/Corrections:

The method section should be expanded:

1. I assume all models were trained as binary classifiers. This is potentially confusing as the chosen ADME properties in the experimental data could also have been modelled using regression models. This should be stated clearly and explained how labels (good/bad) are assigned to the training instances for the different ADME properties (and how labels are assigned to the ChEMBL data given the potencies).
2. Which basis functions (kernels?) were used in equation (2)?
3. What distance measure was used for k-NN (e.g., Soergel/Tanimoto, Hamming)?
4. In Figure 3 (and 4), given the imbalance in data size between training and test set, consider reporting the balanced accuracy. E.g. a trivial classifier classifying each compound as "training" compound would have an accuracy of 75% based on the imbalance of the data set, which needs to be taken account when interpreting Figure 3.
5. The authors provide a data set for download although they do not explicitly report the results for that data set. The results should be reported.

Typos:

- In the formula for KL on page 3 the two vertical bars should have the same size.
- In Figure 1, the labels for the red and blue line are mixed up.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 08 April 2016

<https://doi.org/10.5256/f1000research.8943.r13266>

© 2016 Sheridan R. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Robert Sheridan

Cheminformatics Department, Merck Research Laboratories, Rahway, NJ, USA

This is potentially an important negative result in QSAR, however I think some revision is necessary because some aspects are unclear.

The title "Understanding covariate shift..." is a little weak. One could say "Failure of covariant shift to improve model performance..."

It needs to be explicitly pointed out in the introduction that in most QSAR one builds a model then is able to predict arbitrary compounds. On the other hand, to use covariant shift, one must know which molecules one is predicting before one can generate the model. One can regard "lazy learning" as an extreme version of covariant shift: neighbors of the test set molecules are given weights of 1.0 and all other molecules are given weights of 0.

I need a little more explanation in words of how the weighting is done for training set compounds. Since we are using substructure descriptors here, I am finding it hard to visualize. For example, are we just using distance to the nearest test set example, or are we looking at overlap of the training set descriptors with the distribution of test set descriptors?

Practically no explanation is given as to what QSAR methods are being used. I know what K-NN is and I presume LR is linear regression. Why weren't popular methods like random forest, SVM, or PLS tried?

The color key in Figure 1 does not seem to match what is in the text. In any case, perhaps a better way of looking at would be the [enclosed figure](#).

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research