RESEARCH NOTE

REVISED Understanding covariate shift in model performance [version 3; referees: 2 approved]

Georgia McGaughey, W. Patrick Walters, Brian Goldman

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

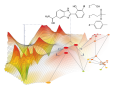
v3 **First published:** 07 Apr 2016, **5**(Chem Inf Sci):597 (doi: 10.12688/f1000research.8317.1)

Second version: 17 Jun 2016, **5**(Chem Inf Sci):597 (doi: 10.12688/f1000research.8317.2)

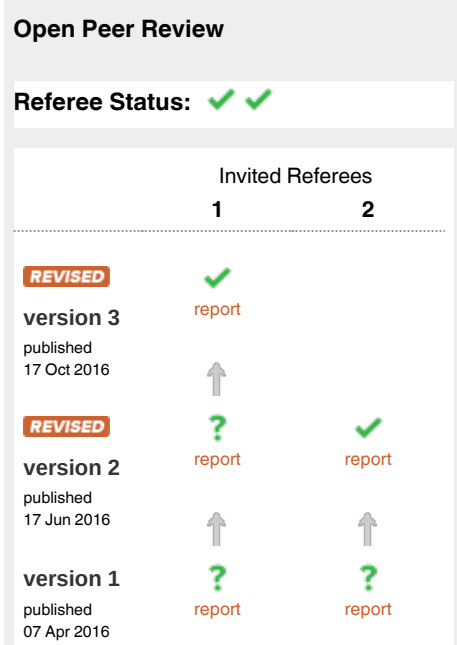
Latest published: 17 Oct 2016, **5**(Chem Inf Sci):597 (doi: 10.12688/f1000research.8317.3)

Abstract

Three (3) different methods (logistic regression, covariate shift and k-NN) were applied to five (5) internal datasets and one (1) external, publically 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.



This article is included in the Chemical information science channel.



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

How to cite this article: McGaughey G, Walters WP and Goldman B. **Understanding covariate shift in model performance [version 3; referees: 2 approved]** *F1000Research* 2016, **5**(Chem Inf Sci):597 (doi: 10.12688/f1000research.8317.3)

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

Competing interests: No competing interests were disclosed.

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

REVISED **Amendments from Version 2**

In response to reviewer feedback, Figure 1 has been replaced with a new version.

See referee reports

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. While standard machine learning classifiers are trained and then used to predict on arbitrary compounds, covariate shifted classifiers must be trained specifically for each prediction dataset. This is because covariate shifted classifiers weight the training distribution to be more similar to the prediction distribution. A recent book provides an excellent overview of the current state of the art in covariate shift methods[2].

Covariate shift frequently occurs during the drug discovery process where learning systems are built to predict physiochemical 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. It is analogous to a nearest neighbor classifier but employs distributions rather than individual examples. Covariate shifted classifiers weight examples from the training set to create a distribution that is more aligned with the prediction set. This weighted data set is then used to train the classifier, resulting in a covariate shifted classifier. As such, covariate shift is applied at the distribution level whereas nearest neighbor methods are applied at the example level. Once a training set has been shifted, it can be used by any machine learning algorithm.

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 shown[3] 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 blue distribution are used for training a classifier to predict on an instance from the green distribution then the red 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.

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 Sugiyama[4]. 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.

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

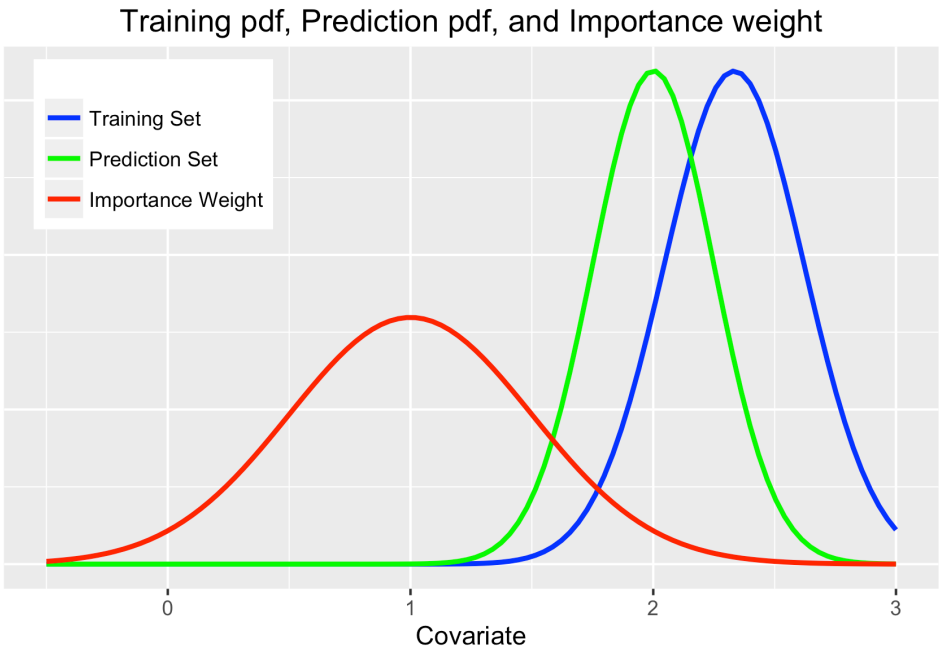


Figure 1. Train, prediction and importance.

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):

$$\underset{\{\alpha_l\}_{l=1}^{b}}{maximize}\left[\sum_{j=1}^{n_p}\log\left(\sum_{l=1}^{b}\alpha_l\varphi_l(x_j)\right)\right]$$

$$subject\ to: \sum_{j=1}^{n_t}\sum_{l=1}^{b}\alpha_l\varphi_l(x_j) = 1$$

$$and\ \alpha_1,\alpha_2,\ldots,\alpha_b \geq 0$$

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 covariate 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 beta secretase IC_{50} 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 for determining weights for use in covariate shift[5]. In principle, covariate shift is applicable to any classifier that allows weighting of input instances (e.g. support vector machines and random forest). For this study we wanted to isolate the effects of covariate shift and therefore selected a classifier without adjustable parameters and used logistic regression (LR). Logistic regression is a classification technique analogous to linear regression and is applicable when the dependent variable is categorical[6]. We combined logistic regression with KLIEP and applied it to five different in-house ADME (absorption, distribution, metabolism and excretion) assays and one external dataset (beta secretase). The cutoff values for determining the binary categories for the compounds in each dataset are listed in Table 1. Due to inherent noise in the assays we discard data where the assay values are between the positive and negative cutoffs listed in the Table 1. We compare KLIEP+Logistic Regression (KL+LR) to Logistic Regression and a k-NN (using Tanimoto similarity) classifier (k=5).

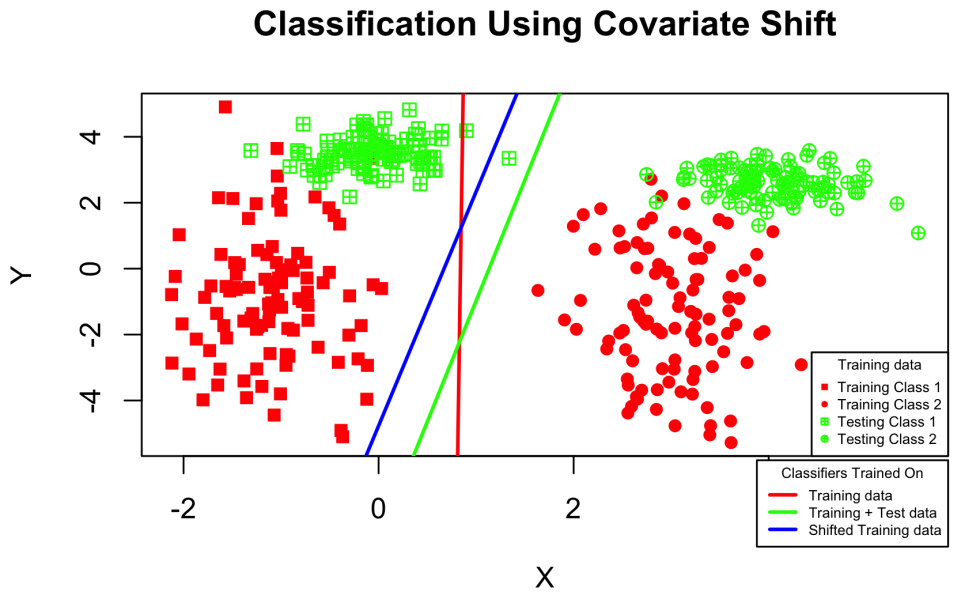


Figure 2. Classification using covariate shift.

Table 1. Proprietary Assays Utilized for Covariate Shift Analysis.

Data Set	Positive Cutoff	Negative Cutoff
hERG	IC50 <10uM	IC50 > 15uM
Human Liver Microsome (HLM)	stable > 60% remain	unstable < 30% remain
Rat Liver Microsome (RLM)	stable > 60% remain	unstable < 30% remain
Solubility (water)	insoluble < 10uM	soluble > 200uM
Solubility (DMSO)	insoluble < 10uM	soluble > 50uM

Legend: The cutoff values for determining the binary categories (actives or inactives) for the compounds in each dataset are listed.

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, OpenEye[7] 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[8,9] such that others could use their own fingerprints and independently support or refute our claims. We chose the beta secretase IC_{50} 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. Using OpenEye path fingerprints and K-Means clustering we clustered the dataset into two clusters, A and B. Under cross-validation, a logistic regression classifier was able to separate the two clusters with a high level of accuracy (90%) indicating that the clustered dataset would be appropriate for application of the covariate shift algorithm. Ten random subsets of molecules from cluster A were used to train a

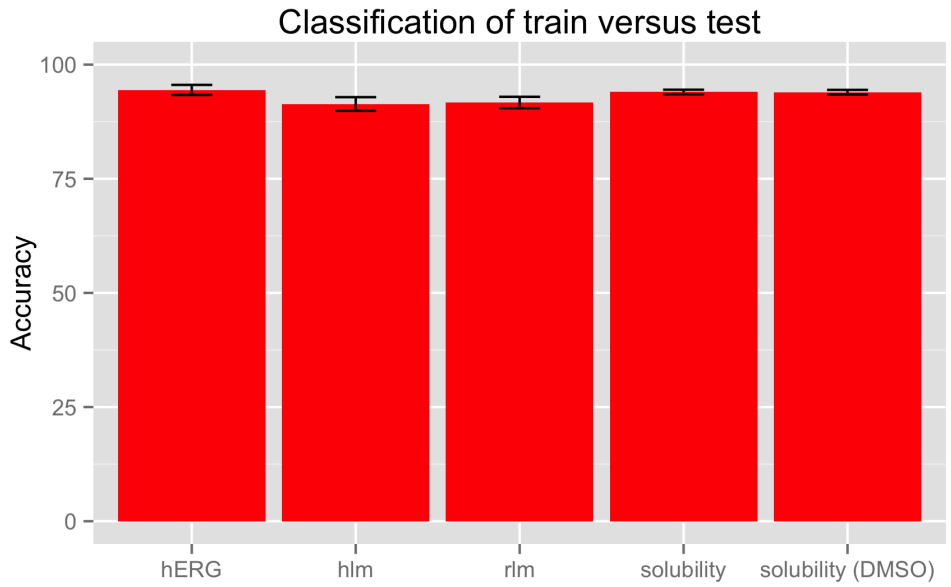


Figure 3. Classification of train versus test.

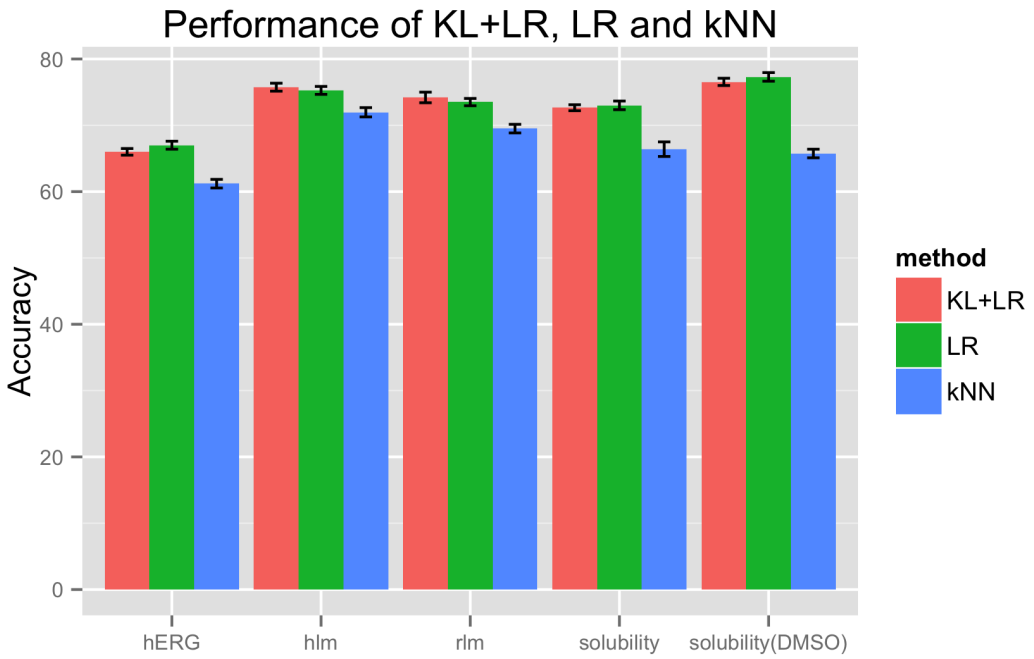


Figure 4. Performance of KL+LR, LR and k-NN.

logistic regression classifier using covariate shift which was then used to predict on molecules from cluster B. The performance of the shifted classifier was compared to an unshifted classifier trained and tested on the same clustered datasets and random splits. The process was repeated by training on molecules from cluster B and predicting on molecules from cluster A. Analogous to the internal datasets, as measured by overall classifier accuracy, there was no statistical advantage for application of covariate shift (Shifted Accuracy: 82.95% +/- 1.6%; Unshifted Accuracy 82.73% +/- 1.2%).

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.

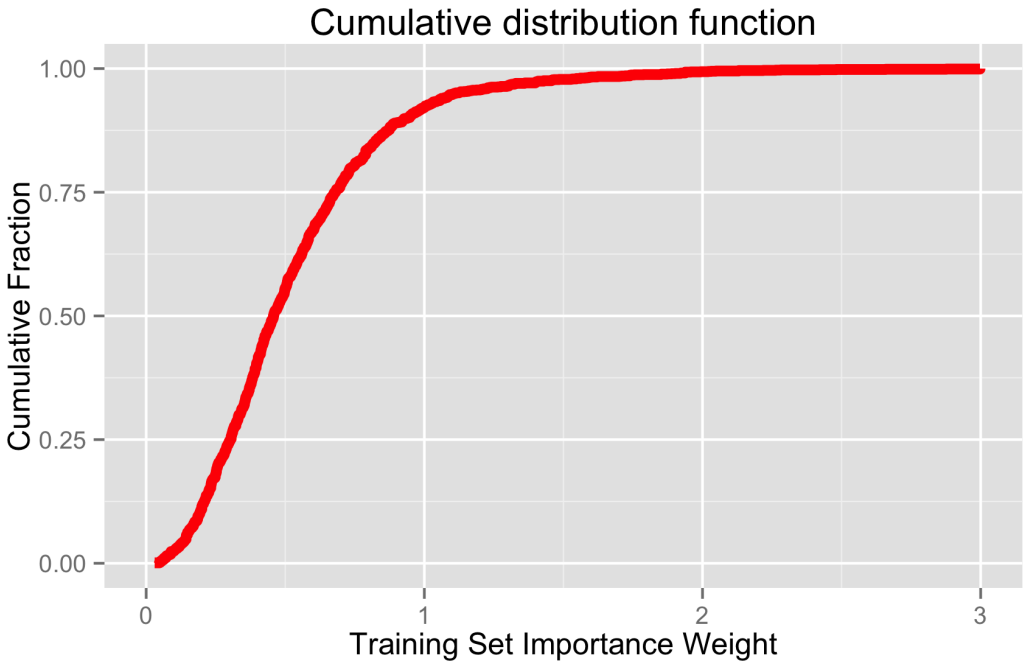


Figure 5. Cumulative distribution function.

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.

Data availability

F1000Research: Dataset 1. The beta secretase IC_{50} data derived from the ChEMBL database, 10.5256/f1000research.8317.d117882[10]

Author contributions

BG conceived the study. BG designed the experiments and carried out the research. GM wrote the manuscript and provided the beta-secretase data set and contributed to the experimental design. PW provided oversight.

Competing interests

No competing interests were disclosed.

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. Quiñonero-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.**
Reference Source
6. Hastie T, Tibshirani R, Friedman J: **The Elements of Statistical Learning Theory.** New York, NY: Springer, 2009.
Publisher Full Text
7. **OpenEye Scientific Software.** (version 2014.Feb), 9 Bisbee Ct, Suite D, Santa Fe NM, USA.
Reference Source
8. 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
9. 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
10. McGaughey G, Walters WP, Goldman B: **Dataset 1 in: Understanding covariate shift in model performance.** *F1000Research.* 2016.
Data Source

Open Peer Review

Current Referee Status: ✔ ✔

Version 3

Referee Report 18 October 2016

doi:10.5256/f1000research.10228.r17055

✔ **Robert Sheridan**
Cheminformatics Department, Merck Research Laboratories, Rahway, NJ, USA

The only issue I am still having trouble with is equation 1.

w(x)=Pp(x)/Pt(x)

where the P's represent probabilities for molecule x. Given that the training and test sets are distinct points (molecules) in a chemical descriptor space and the training and test sets do not overlap, how are the probabilities for molecule x calculated from the other molecules?

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Version 2

Referee Report 21 June 2016

doi:10.5256/f1000research.9428.r14444

✔ **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 revised paper has been significantly improved and addresses a number of issues raised in the original reviews. However, a few issues remain that should be addressed.

1. The description of Figure 1 is inconsistent with the figure legend. It seems that the red and blue labels in the legend of Figure 1 need to be swapped to make the figure consistent with the description in the text: The red and green curves look like pdfs whose integrals are very similar and close to 1 while the blue

curve has a much larger area inconsistent with a pdf.

2. For the ChEMBL data set the inactivity/activity cutoffs used should be mentioned.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Referee Report 20 June 2016

doi:10.5256/f1000research.9428.r14472

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

Having read the revised paper it seems better, but I'm still somewhat puzzled about a few things.

1. I am getting the feeling that the K-NN method is meant as a baseline control method since by definition K-NN looks at only the training set compounds close to the test set compounds, so there is an implicit selection of training set compounds, and this should have a similar effect as covariant shift. This is not explicitly said in the paper.

2. The authors do not try sophisticated but more "standard" classification methods like random forest or SVM, and don't say why not.

3. Both myself and the other reviewer seem confused by Figure 1. The red line is supposed to be the importance weight. However, it implies that the highest weights are given in a region of descriptor space far away from both training and test sets.

I have read this submission. I 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.

Competing Interests: No competing interests were disclosed.

Author Response 21 Jun 2016

Georgia McGaughey, Vertex Pharmaceuticals, USA

I'm responding directly to the questions you posed, in that order.

1. Agreed that we could have added such a sentence to the paper.

2. We didn't examine more standard classification methods as we were specifically studying whether there was a benefit in using covariate shift. We were not interested in exploring RF or SVM as that was out of scope for this paper.

3. The three curves in Figure 1 were all drawn as per a Gaussian distribution and we thought it

would look odd to have the red line go straight to zero when, for example, x=1. The purpose of the importance weight being close to 0 (on the y axis) when x=2, for instance, is because there's much overlap between the training and the prediction set. When, for instance, there isn't much overlap (ie, x=1.5), the importance weight goes up. I can see why one might be confused when x=1; that was merely to show that when there's minimal overlap between prediction and training, that the importance weight is quite high.

Competing Interests: No competing interests were disclosed.No competing interests were disclosed.

Referee Response 02 Aug 2016

Martin Vogt, University of Bonn, Germany

Concerning Figure 1:
According to equation (1) w(x) = p_p(x)/p_t(x) there is a certain symmetry between the importance weight w(x) and the training pdf p_t(x).

This means that in Figure 1 the red curve can either show the importance weight and the blue curve the training pdf or vice versa without compromising the accuracy of the figure. What speaks against interpreting the blue curve as a pdf is that it clearly has a much larger area than the green curve representing the prediction pdf, which should be 1 in both cases. From visual inspection the red curve has an area much closer to 1 than the blue curve and it should thus be interpreted as training pdf while the blue curve represents the importance weight. Even though the figure is only used for illustration purposes it could be improved by either relabelling or by rescaling the red and blue curves accordingly.

Competing Interests: No competing interests were disclosed.

Author Response 03 Aug 2016

Georgia McGaughey, Vertex Pharmaceuticals, USA

The figure is really meant to be an illustration of the importance weight - and not mathematically accurate with respect to pdfs. In actuality, the green and blue curve look like pdfs but they really are not. The green and blue curves are as such not normalized using the same scales - so the area under each curve does not sum to 1. We could make the blue curve a shifted version of the green curve then recompute the importance weight. What we were hoping to illustrate is that if an example from the training (blue) was pulled at x = 1 it would be very important for training because it is rare and the testing (green) set has non-zero support at x = 1.

Additionally, another perhaps confusing aspect about the figure is that the y-axis represents two values on different scales: 1) the importance weight and 2) the probability (relative) of seeing a training/test example. We will redraw the figure and upload a new version.

Competing Interests: No competing interests.

Version 1

Referee Report 25 April 2016

doi:10.5256/f1000research.8943.r13393

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.

I have read this submission. I 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.

Competing Interests: No competing interests were disclosed.

Referee Report 08 April 2016

doi:10.5256/f1000research.8943.r13266

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.

I have read this submission. I 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.

Competing Interests: No competing interests were disclosed.

Discuss this Article

Version 2

Author Response 17 Aug 2016

Georgia McGaughey, Vertex Pharmaceuticals, USA

New version of Figure 1 has been submitted. The AUCs of the prediction and training sets are now the same and the label for the y axis has been removed to avoid confusion (but a legend clearly defines what each line refers to). The red line (the importance weight) has remained the same as before.

Competing Interests: No competing interests.