Properties of Object-Level Cross-Validation Schemes for Symmetric Pair-Input Data

Juho Heimonen[1,2], Tapio Salakoski[1,2], and Tapio Pahikkala[1,2]

[1] TUCS - Turku Centre for Computer Science, Turku, Finland
[2] University of Turku, Turku, Finland
firstname.lastname@utu.fi

Abstract. In bioinformatics, many learning tasks involve pair-input data (i.e., inputs representing object pairs) where inputs are not independent. Two cross-validation schemes for symmetric pair-input data are considered. The mean and variance of cross-validation estimate deviations from respective generalization performances are examined in the situation where the learned model is applied to pairs of two previously unseen objects. In experiments with the task of learning protein functional similarities, large positive mean deviations were observed with the *relaxed* scheme due to training–validation dependencies while the *strict* scheme yielded small negative mean deviations and higher variances. The properties of the strict scheme can be explained by the reduction in cross-validation training set sizes when avoiding training–validation dependencies. The results suggest that the strict scheme is preferable in the given setting.

Keywords: cross-validation, pair-input, AUC, K-Nearest Neighbor.

1 Introduction

In supervised learning, the generalization performance is commonly estimated by training a model on one part of the dataset (training set) and evaluating it against another (validation set) to avoid optimistically biased estimates. Cross-validation (CV) is a procedure to estimate the generalization performance by aggregating the results of several such evaluations. [2].

A CV procedure consists of folds, each of which involving training and evaluating a model according to a training–validation split of the dataset. Since an *input* (i.e., a data point) can belong to the training set of one fold and to the validation set of another, CV can be used when the small size of the dataset prevents from obtaining large enough training and validation sets in a single split [6]. The properties of a CV estimator are influenced by the splitting scheme as well as how the performance is measured.

In a general case, CV procedures assume that data are identically distributed and the training set is independent from the validation set [2]. The conventional approach of randomly partitioning data into training and validation sets is not viable when the data contain dependencies [12].

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 384–393, 2014.

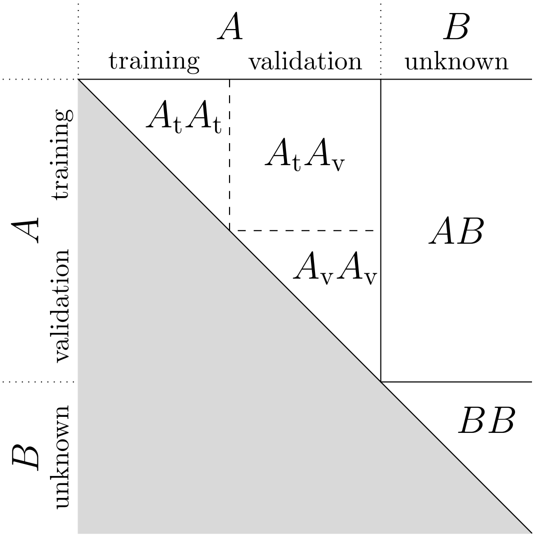

Fig. 1. There are three types of pairs in a symmetric pair-input learning task. The set A contains those objects that are pair members in the dataset on which a model is trained and cross-validated. The set B contains the objects not present in the dataset. The AA, AB, and BB types of pairs differ in the number of members seen in the dataset. The types $A_{\mathrm{t}}A_{\mathrm{t}}$, $A_{\mathrm{t}}A_{\mathrm{v}}$, and $A_{\mathrm{v}}A_{\mathrm{v}}$ are the analogous types within a CV fold with the subscripts referring to the training (t) and validation (v) sets.

This study explores the properties of CV estimators in the case of symmetric pair-input data. Pair-input data consist of inputs that represent pairs of *objects* while symmetry refers to pair members being of a single type with a symmetric relation. Among others, data of this type are encountered in bioinformatics when considering the properties of protein pairs, such as binding [13] or functional similarity. Research in biosciences typically focuses on specific aspects of organisms and knowledge is consequently centered around a subset of proteins. Since the protein pairs of which a particular property is known stem from a limited set of proteins, it is common that a protein is a pair member in several inputs which leads to strong dependencies (see, for example, [13]).

Object pairs were categorized in [13] by their composition with respect to a given dataset. Figure 1 illustrates these three types: both members (AA), one member (AB), or no members (BB) belonging to the set A that contains the objects present in the dataset. It was observed that the CV estimator of the generalization performance of a model learned from AA pairs using a conventional scheme is acceptable when considering the performance on AA pairs but optimistically biased when considering the performance on AB or BB pairs [13].

This study examines two CV schemes in the situation where predictions will be made on BB pairs. They differ from conventional splitting schemes in that the splitting is performed on objects, not on inputs, and validation sets are formed based on the selected objects. The *relaxed* scheme, involving models trained on the union of $A_\mathrm{t}A_\mathrm{t}$ and $A_\mathrm{t}A_\mathrm{v}$ (see Fig. 1), is expected to be optimistically biased because validation set inputs are exposed via shared pair members whereas the *strict* scheme, involving models trained on $A_\mathrm{t}A_\mathrm{t}$ and evaluated against $A_\mathrm{v}A_\mathrm{v}$, should not exhibit an optimistically biased behavior because the setup is analogous to learning from AA pairs to predict BB pairs. The strict scheme is expected to be pessimistically biased because the full model is trained on more data than the CV models [1] and have higher variance than the relaxed scheme because its training sets contain less data [11].

Experiments are performed on the prediction of the functional similarity of two proteins from their sequences. While not a typical formulation of the protein function prediction task, which is one of the major tasks in bioinformatics [9], functional similarity serves as an example of a symmetric pair-input problem.

2 Cross-Validation Schemes

Let $\mathcal{O}$ be a set of objects and $\mathcal{Z} \subset \mathcal{X} \times \mathcal{Y}$ a set of instances, where the input space $\mathcal{X} = \mathcal{O}^2$ and the output space $\mathcal{Y} = \{-1, 1\}$. An instance $z = (x, y) \in \mathcal{Z}$ consists of an input $x = (o, o') \in \mathcal{O}^2$ and its associated label y such that $y = 1 \iff x \in \mathcal{R}$, where $\mathcal{R} \subseteq \mathcal{O}^2$ is the symmetric relation of interest. A sequence $Z = ((x_1, y_1), \ldots, (x_n, y_n)) \in \mathcal{Z}^n$, where $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ are the input and label sequences, respectively, is called a training set. The set $\mathcal{O}_Z = \{o : (\exists i)(X_i = (o, o') \vee X_i = (o', o))\}$ is the set of training set objects. An input cannot be associated with both labels. That is, $(x, y) \in \mathcal{Z} \implies (x, -y) \notin \mathcal{Z}$. Also, $(\exists i)(X_i = (o, o')) \implies (\nexists j)(X_j = (o', o))$ because the inputs (o, o') and (o', o) are assumed to have identical representations in addition to their associated labels being identical due to symmetry. Instances having $y = 1$ are called positive instances and those having $y = -1$ negative instances. Let $D_\mathcal{O}$ and $D_\mathcal{Z}$ be probability distributions over $\mathcal{O}$ and $\mathcal{Z}$, respectively.

The outputs of a prediction function $f_Z : \mathcal{X} \to \mathbb{R}$, learned from the training set Z, rank the inputs by how likely their associated $y = 1$. The generalization performance of a prediction function is measured by its conditional expected area under the ROC curve (AUC) [1]

$$\mathrm{A}(f_Z) = \mathrm{E}_{z_+ \sim D_+, z_- \sim D_-}[\mathrm{H}(f_Z(x_+) - f_Z(x_-))] \ , \tag{1}$$

where $z_+ = (x_+, 1)$, $z_- = (x_-, -1)$, and H is the Heaviside step function with $H(0) = \frac{1}{2}$, while D_+ and D_- are the conditional distributions of instances derived from $D_\mathcal{Z}$ given $y = 1$ and $y = -1$, respectively.

In each CV fold, a validation set $\mathcal{O}_V$ of objects is picked such that $\mathcal{O}_V \subset \mathcal{O}_Z$. The validation set V of instances is a subsequence of Z such that $(\exists i)(V_i = (x, y)) \iff ((\exists j)(Z_j = (x, y)) \wedge o \in \mathcal{O}_V \wedge o' \in \mathcal{O}_V)$, where $x = (o, o')$, (see $A_\mathrm{v}A_\mathrm{v}$ in Fig. 1) while the training set T of instances is a subsequence of Z. In the relaxed scheme $(\exists i)(T_i = (x, y)) \iff ((\exists j)(Z_j = (x, y)) \wedge (o \notin \mathcal{O}_V \vee o' \notin \mathcal{O}_V))$ (see $A_\mathrm{t}A_\mathrm{t}$ and $A_\mathrm{t}A_\mathrm{v}$ in Fig. 1) while in the strict scheme $(\exists i)(T_i = (x, y)) \iff$

$((\exists j)(Z_j = (x, y)) \wedge o \notin \mathcal{O}_V \wedge o' \notin \mathcal{O}_V)$ (see $A_t A_t$ in Fig. 1). The sequence $C = ((V_1, T_1), \ldots, (V_n, T_n))$ contains the validation and training set pairs of the n folds.

The CV performance $\hat{\mathrm{A}}_{\mathrm{CV}}(Z)$ is an estimator of $\mathrm{A}(f_Z)$ obtained from Z using the learning algorithm that yielded f_Z. The quality of a CV scheme is evaluated using the mean and variance of the deviation $\mathrm{B}(Z) = \hat{\mathrm{A}}_{\mathrm{CV}}(Z) - \mathrm{A}(f_Z)$ which follows the approach taken, for example, in [6] and [1]. The second moment about zero of $\mathrm{B}(Z)$ is also considered.

3 Estimation of AUC

The properties of $\hat{\mathrm{A}}_{\mathrm{CV}}(Z)$ are influenced by how the validation set V and the training set T are selected in each fold but also by how cross-validation AUC is calculated. The choice between the relaxed and strict schemes affects T, which is the focus of this study, while the selection of $\mathcal{O}_V$ affects both V and T.

Two methods to calculate cross-validation AUC are considered: averaging and pooled AUC [5,1]. The former is the mean AUC over folds whereas the latter is calculated from the concatenation of the predictions made in the folds.

In an earlier study, AUC estimators were analyzed in a non-pair-input situation. Non-zero mean deviations were observed for pooled AUC on certain kinds of data which was attributed to predictions from several models being compared although strictly not compatible. Also, estimators involving more comparisons of positive–negative instance pairs were observed to have lower variance than those with fewer comparisons. [1].

Object-leave-two-out CV includes a fold for each of the $\binom{m}{2}$ possible validation sets fulfilling the condition $|\mathcal{O}_V| = 2$, where $m = |\mathcal{O}_Z|$. If $\mathcal{X}$ is restricted to the inputs (o, o') such that $o \neq o'$ (like in this study, see Sect. 4.3), each validation set contains only one instance and all instances are included in exactly one validation set.

In object-n-fold CV, $\mathcal{O}_Z$ is partitioned into n parts of approximately equal sizes with the ith part being $\mathcal{O}_V$ in the ith fold. Consequently, some instances do not belong to any of the validation sets and the number of excluded instances increases as n increases (Fig. 2). To cover all instances, overlapping validation sets can be selected such that two parts form a validation set in each fold which results in $\binom{n}{2}$ folds. In this case, however, the pairs in which the members are from the same part appear in $n - 1$ validation sets while the other pairs appear only once (diagonal blocks vs. non-diagonal blocks in Fig. 2). As n approaches m, overlapping object-n-fold CV approaches object-leave-two-out CV (Fig. 2).

4 Experiments

The properties of the relaxed and strict schemes were investigated by conducting experiments on learning protein functional similarities. A protein is a biomolecule composed of amino acid chains folded into a three-dimensional structure that is capable of accomplishing (possibly jointly with other proteins) a particular

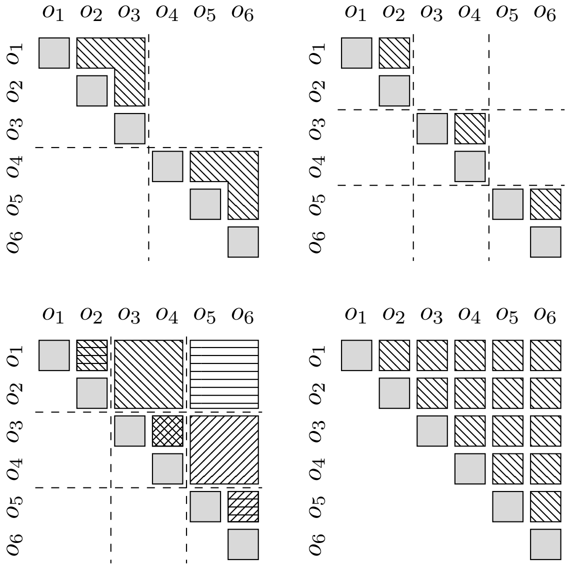
o_1 o_2 o_3 o_4 o_5 o_6
o_1 o_2 o_3 o_4 o_5 o_6
o_1 o_2 o_3 o_4 o_5 o_6
o_1 o_2 o_3 o_4 o_5 o_6
o_1 o_2 o_3 o_4 o_5 o_6
o_1 o_2 o_3 o_4 o_5 o_6
o_1 o_2 o_3 o_4 o_5 o_6
o_1 o_2 o_3 o_4 o_5 o_6

Fig. 2. The validation sets (*patterned areas*) of object-2-fold CV (*upper left*) cover more instances than those of object-3-fold CV (*upper right*). Both overlapping object-3-fold CV (*lower left*) and object-leave-two-out CV (*lower right*) cover all instances but three instances are present in two validation sets, distinguished by pattern types, in the former. *Dashed lines* indicate the boundaries of the parts of $\mathcal{O}_Z = \{o_1, \ldots, o_6\}$ while *grey squares* represent the excluded (o, o) pairs.

task. How the amino acid sequence (and the structure) of a protein defines its function is one of the major topics in bioinformatics.

The task of protein function prediction can be formulated as one of predicting the function of a protein from its sequence [8] while other sources of information may also be utilized as well [4,7]. In this study, instead of directly predicting the function, the functional similarity of two proteins is considered because it fulfills the requirements of symmetric pair-input data.

4.1 Data

Datasets were derived from the Universal Protein Resource[1] (UniProt) [14]. UniProt entries contain amino acid sequences of proteins together with diverse annotations, literature references, and cross-references to other databases. Its UniProtKB/Swiss-Prot section contains manually curated entries while the UniProtKB/TrEMBL section contains unreviewed, computer-annotated entries. Only the former was used in the experiments in order to minimize noise.

[1] `http://www.uniprot.org/`

The functional similarity of two proteins was determined by their Gene Ontology annotations. Gene Ontology[2] (GO) [3] is a comprehensive classification and widely adopted in bioinformatics. It provides hierarchical controlled vocabularies for three complementary domains – molecular function, biological process, and cellular component – referenced by UniProt entries.

Three datasets were created by considering one of the GO domains at the time and the fourth by considering the domains jointly. The information regarding the function was assumed to be complete when an entry had any GO annotation belonging to the given domain(s). All such proteins were included in the dataset while the others were discarded to avoid false negative labels. This produced datasets ranging approximately from 387,000 to 511,000 proteins in size.

4.2 Features and Labels

Each protein sequence was represented by a vector containing the frequencies of amino acids as well as the frequencies of bigrams of adjacent amino acids categorized into four classes according to [10]. A protein pair was represented by the sum of the two protein feature vectors. This low-dimensional representation is more suitable for the K-Nearest Neighbor classifiers used in the experiments (see Sect. 4.3) than high-dimensional representations.

A protein pair was labeled positive if its members had any GO annotation in common. The hierarchy of GO classes was not taken into account.

4.3 Experiment Details

The set $\mathcal{Z}$ was defined as the set of instances covering all protein pairs (o, o') such that $o \neq o'$ to avoid trivially positive instances skewing performance scores. Both $D_{\mathcal{O}}$ and $D_{\mathcal{Z}}$ were chosen to be uniform distributions. Since its exact value is impractical to calculate, the conditional expected AUC was estimated from a random sample S with the Wilcoxon–Mann–Whitney statistic [5]. For each dataset, the sequence S was drawn without replacement from $\mathcal{Z}$ such that $|S| = 10^4$. Let $\mathcal{O}_S = \{o : (\exists i)(S_i = (x, y) \wedge (x = (o, o') \vee x = (o', o)))\}$.

The relaxed and strict schemes were evaluated with all possible combinations of the four datasets, two validation set selection methods (object-ten-fold or object-leave-two-out), and two AUC calculation methods (averaging or pooled). Note that averaging AUC cannot be calculated in the object-leave-two-out case because each validation set contains only one instance.

The sampling distribution of deviations was obtained from one thousand independent repeats. In each repeat, a sequence $O = (o_1, \ldots, o_n)$ of 100 proteins was conditionally drawn without replacement from $\mathcal{O}$ given that $o_i \notin \mathcal{O}_S$. The training set Z was formed by including the inputs (o, o') fulfilling the condition $(\exists i)(O_i = o) \wedge (\exists j)(O_j = o')$.

In all experiments, K-Nearest Neighbor classifiers were trained with inverse distance weighing. The parameter K was varied from $K = 10$ to $K = 100$ in steps of ten to analyze its effect.

[2] http://www.geneontology.org/

Table 1. The observed mean deviations for $K = 50$. *10x* and *LTO* refer to object-ten-fold and object-leave-two-out CV while A and P refer to averaging and pooled AUC, respectively.

	Relaxed			Strict		
Dataset	A-10x	P-10x	LTO	A-10x	P-10x	LTO
Union	0.1799	0.1870	0.1919	−0.0113	−0.0209	−0.0187
Molecular function	0.1509	0.1614	0.1642	−0.0167	−0.0282	−0.0285
Biological process	0.0673	0.0909	0.0907	−0.0508	−0.0402	−0.0410
Cellular component	0.1745	0.1797	0.1842	−0.0220	−0.0290	−0.0246

Table 2. The observed variances of deviations for $K = 50$. *10x* and *LTO* refer to object-ten-fold and object-leave-two-out CV while A and P refer to averaging and pooled AUC, respectively.

	Relaxed			Strict		
Dataset	A-10x	P-10x	LTO	A-10x	P-10x	LTO
Union	0.0020	0.0017	0.0009	0.0028	0.0028	0.0015
Molecular function	0.0032	0.0025	0.0012	0.0051	0.0046	0.0025
Biological process	0.0111	0.0086	0.0027	0.0160	0.0147	0.0064
Cellular component	0.0016	0.0015	0.0009	0.0027	0.0028	0.0016

5 Results and Discussion

The relaxed and strict schemes resulted in positive and negative mean deviations, respectively, and the experiments with the relaxed scheme yielded lower variances of deviations than their counterparts with the strict scheme. Increases in K resulted in decreases in the means in both schemes, though the effect was minor in the strict scheme, while the variances increased in the strict scheme and decreased in the relaxed scheme. The peak generalization performance was reached in the given range of K with the *Union* and *Cellular component* datasets.

Illustrating typical observations, Tables 1 and 2 show the means and variances, respectively, of the observed deviations of CV estimates from respective (estimated) generalization performances for $K = 50$. The means of the observed generalization performances for $K = 50$ are 0.5857, 0.6524, 0.7423, and 0.6351, in the order of the datasets in the tables.

The absolute values of the deviation means are generally approximately an order of magnitude lower and the deviation variances higher but of the same order of magnitude in the experiments with the strict scheme than in their counterparts with the relaxed scheme. The *Biological process* dataset differs from the others by having notably lower absolute values in the relaxed setting, higher absolute values in the strict setting, and higher variances in both settings.

The above observations are reflected in the second moments about zero being approximately an order of magnitude lower in the experiments with the strict

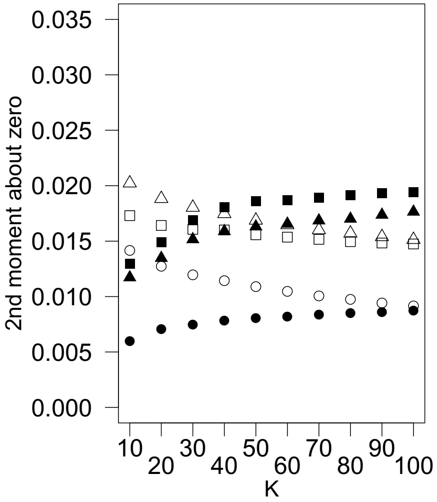

Fig. 3. The second moments about zero of the relaxed and strict schemes become equal at approximately $K = 30$, $K = 60$, or $K = 100$ in the *Biological process* dataset (*left*) whereas they are well-separated in the *Molecular function* dataset (*right*). Triangle, square, and circle refer to *P-10x*, *A-10x*, and *LTO* (see Tables 1 and 2) whereas hollow and solid symbols denote the relaxed and strict schemes, respectively.

scheme than in their counterparts with the relaxed scheme in all except the *Biological process* dataset. The changes in mean and in variance as K increases both contribute toward the decreasing and increasing trends in the second moments seen with the relaxed and strict schemes, respectively. However, as illustrated in Figure 3, the point after which the relaxed scheme yields lower second moments depends on the dataset and the CV details.

The absolute values of the deviation means are higher in the experiments with pooled AUC than in their counterparts with averaging AUC in all but one experiment pair. This is not surprising given that pooling can have either a positive or negative effect on deviations [1]. An increase in the number of positive–negative instance comparisons (*A-10x* < *P-10x* < *LTO*, see Table 2) generally has a decreasing effect on variance, as expected, although *A-10x* and *P-10x* are in the opposite order in two experiment pairs for high K values.

The observed deviation means suggest that the positive effect of training–validation dependencies generally dominates over the negative effect of the reduced size of training sets and, consequently, that the strict scheme is preferable to the relaxed scheme in the setting where the learned model will be applied to pairs of two previously unseen objects. However, given the limited number of experiments in this study, it remains unanswered to what extend these observations can be generalized to other datasets and/or learning algorithms. Particularly, the results obtained with the *Biological process* dataset raises the question whether the unexpectedly small differences in the absolute values of the mean deviations between the two schemes are due to the properties of the dataset or due to the schemes generally yielding more similar absolute values as generalization performance increases. In the latter case, the strict scheme would not necessarily be preferable at high performance levels although it would still have the advantage of yielding conservative estimates.

5.1 Future Directions

The results of this study illustrate the potential of the strict scheme. In future experiments, the scheme will be applied to a variety of learning algorithms and datasets to get a better understanding of its behavior. With preliminary results from another dataset suggesting otherwise, it is of particular interest to investigate whether higher absolute values of deviations should be expected at higher levels of generalization performance as is hinted by the *Biological process* dataset. Different approaches to select validation sets (see, for example, [2]) will also be examined in order to discover their properties when operating on objects instead of on instances. Last, the analysis of the strict scheme will be expanded to the experimental setup outlined in [13] where some pairs of objects are not included in the dataset due to incomplete knowledge of objects.

The two schemes considered in this study are expected to fail to reliably estimate the generalization performance of a learned model when predictions will be made on inputs where an object seen in the dataset is paired with a previously unseen object (AB pairs in Fig. 1). Adapting the strict scheme to this setting likely requires only minor modifications.

6 Conclusions

Two CV schemes for symmetric pair-input data were considered. They differ from conventional CV schemes by acknowledging the fact that inputs represent pairs of objects. They first make training–validation splits on objects and then use the selected objects to form training and validation sets. The strict scheme avoids dependencies between the training and validation sets that would arise from shared pair members by discarding offending instances from the training sets. Consequently, its folds are analogous to learning a model from a dataset and making predictions on pairs that are composed of objects not encountered in the dataset. The relaxed scheme utilizes all instances in each fold and is hence similar to conventional CV schemes that assume independent instances.

The properties of the relaxed and strict schemes were examined in the task of learning functional similarities of proteins. Four datasets were derived from UniProt database and evaluated using various combinations of AUC calculation method and validation set selection method. Positive mean deviations were observed for the relaxed scheme while negative mean deviations were observed for the strict scheme. The strict scheme yielded lower absolute values of deviation means but higher deviation variances than the relaxed scheme. These observations can be explained by dependencies between training and validation sets, relative training set sizes, and the properties of the AUC calculation methods used in the experiments.

The results suggest that the generalization performance of a model is better estimated by the strict scheme than the relaxed scheme in the situation where predictions will be made on pairs of previously unseen objects. Such pairs may be encountered in significant numbers, for example, when predicting protein–protein binding [13]. However, further experiments are needed to get a better understanding of the properties of the strict scheme.

Acknowledgements. This study was supported by the Academy of Finland. We would like to thank the anonymous reviewers for their helpful comments.

References

1. Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., Salakoski, T.: An experimental comparison of cross-validation techniques for estimating the area under the roc curve. Computational Statistics and Data Analysis 55, 1828–1844 (2011)
2. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Statistics Surveys 4, 40–79 (2010)
3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. 25, 25–29 (2000)
4. Bork, P., Dandekar, T., Diaz-Lazcoz, Y., Eisenhaber, F., Huynen, M., Yuan, Y.: Predicting function: from genes to genomes and back. J. Mol. Biol. 283, 707–725 (1998)
5. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159 (1997)
6. Braga-Neto, U.M., Dougherty, E.R.: Is cross-validation valid for small-sample microarray classification? Bioinformatics 20, 374–380 (2004)
7. Eisenberg, D., Marcotte, E.M., Xenarios, I., Yeates, T.O.: Protein function in the post-genomic era. Nature 405, 823–826 (2000)
8. Han, L., Cui, J., Lin, H., Ji, Z., Cao, Z., Li, Y., Chen, Y.: Recent progresses in the application of machine learning approach for predicting protein functional class independent of sequence similarity. Proteomics 6, 4023–4037 (2006)
9. Lee, D., Redfern, O., Orengo, C.: Predicting protein function from sequence and structure. Nat. Rev. Mol. Cell Biol. 8, 995–1005 (2007)
10. Mei, S., Fei, W.: Amino acid classification based spectrum kernel fusion for protein subnuclear localization. BMC Bioinformatics 11(suppl. 1), S17 (2010)
11. Nadeau, C., Bengio, Y.: Inference for the generalization error. Machine Learning 52, 239–281 (2003)
12. Pahikkala, T., Suominen, H., Boberg, J.: Efficient cross-validation for kernelized least-squares regression with sparse basis expansions. Machine Learning 87, 381–407 (2012)
13. Park, Y., Marcotte, E.M.: Flaws in evaluation schemes for pair-input computational predictions. Nat. Methods 9, 1134–1136 (2012)
14. The UniProt Consortium: Activities at the Universal Protein Resource (UniProt). Nucleic Acids Res. 42, D191–D198 (2014)