Generating Synthetic Multi-label Data Streams

Jesse Read, Bernhard Pfahringer, Geoff Holmes

Department of Computer Science
The University of Waikato
Hamilton, New Zealand
{jmr30,bernhard,geoff}@cs.waikato.ac.nz

Abstract. There are many available methods for generating synthetic data streams. Such methods have been justified by the need to study the efficacy of algorithms on a theoretically infinite stream, and also a lack of real-world data of sufficient size. Although multi-label classification has attracted considerable interest in recent years, most of this work has been carried out in the context of a batch learning environment rather than a data stream. This paper makes an in-depth analysis of multi-label data, and presents a general framework for generating synthetic multi-label data streams.

1 Introduction

A multi-label data stream is a data stream with the same properties as multi-label data. Multi-label learning problems, where an instance is assigned multiple labels from a finite set of labels, have received considerable attention in the machine learning literature, but prior work focusses almost exclusively on a batch learning environment with train-test or cross-validation scenarios. The problem of multi-label data streams has received much less attention.

Many real world practical problems involve data which can be considered as a multi-label data stream. For example news articles, e-mails, RSS-feeds, newsgroups, bookmarking, and medical text classification.

Labels can be considered as subject categories, tags, author names, or even diagnoses (in the case of the medical domain), as long as the set of labels is finite and known at the time of classification. Instances will always arrive in time order. Classification in such an environment involves an emphasis on efficiency and adaptivity:

- **Incremental learning**: examples processed one at a time; must be able to predict as new instances become available
- **Efficiency**: limited amount of time and memory; able to handle large volumes of new instances
- **Adaptivity**: must be able to handle to concept drift

Despite the ubiquitous presence of multi-label data streams in the real world, they can rarely be easily assimilated on a large scale with both labels and timestamps intact and there may issues with sensitive data – for example with e-mail,

personal bookmarking, and medical text corpora. In many cases, in-depth domain knowledge may be necessary to determine and pinpoint changes to the concepts represented by the data.

Hence the reasons to generate synthetic multi-label data streams are to:

- increase the pool of multi-label stream data and thereby also increase the depth of analysis and conclusions which can be drawn in respect to the performance of various algorithms;
- allow a theoretically infinite data stream; and
- help conduct specific analysis of incremental multi-label algorithms.

This paper involves an in depth study of multi-label data, and presents a framework for generating synthetic multi-label data streams in order to facilitate the study and evaluation of multi-label algorithms in this area.

2 Prior Work

The notation to define a multi-label data stream is as follows.

- Let $\mathcal{X} = \mathbb{R}^d$ denote the input space
- Let $X \in \mathcal{X}$ be an *instance*
- Let $L = \{l_1, l_2, \cdots, l_N\}$ denote the finite label set
- Let $l_i \in L$ be a single label
- Let $S = (l_1, l_2, \cdots, l_N) \in \{0, 1\}^N$ be a *label subset* representing $S \subseteq L$ where:

$$S[i] = \begin{cases} 1 & \text{if } l_i \in S \\ 0 & \text{if } l_i \notin S \end{cases}$$

- Let $d = (X, S)$ be a multi-label example consisting of an instance and relevant labels
- Let $D = d_0, d_1, \cdots$ be a theoretically infinite stream of multi-label examples

2.1 Synthetic Multi-label Data

Generating synthetic data streams has been investigated in the past for single-label data. The work in [4] provides the `MOA` framework which contains a variety of methods for the generation and classification of *single-label* data streams. This is expanded by [1] which additionally considers concept drift, as opposed to simply an incremental context. There are also numerous examples of purpose-specific multi-label data being generated.

The authors of [10] generate a multi-label synthetic dataset with three labels and two features. The examples pertaining to certain labels are associated with certain Gaussian distributions. Cai [2] uses a tree structure with random weight vectors generated for each node. Park and Fürnkranz [5] generate data using a number of labels using a set of pairwise constraints. Random permutations are generated which satisfy this set, which are in turn decomposed into binary pairwise preferences.

Kirchenko's [3] synthetic data involves a special hierarchical case where inner nodes represent labels. Synthetic data is generated by building a balanced tree hierarchy and allocating three binary attributes, with 10 training and 5 test instances generated for each label.

Overall, prior work for generating synthetic multi-label serves only to highlight certain characteristics of the algorithms that the authors present. The data usually contains as few as two or three features and labels, relatively few examples, and was never intended for large scale multi-label evaluation. More importantly, none of these data generation techniques are for creating data stream contexts, which is a main focus of this paper.

Not yet mentioned in the literature is the idea of using clustering to create multi-label data where cluster centers represent labels. This would presume the use of a clustering algorithm which can supply probabilities that an instance belongs to each cluster so that a threshold could be used to influence different degrees of multi-labelling. Any data source could be used if the original time order can be maintained. A related possibility would be to use a time-ordered single-label dataset and to reclassify using this same ranking-and-threshold method.

The advantage of these techniques is to have data with underlying real-world concepts. However, the stream cannot be theoretically infinite unless the source of real world data is, and in such a case the clustering process would then also have to be incremental. Moreover, access to such an extensive and reliable source of real-world data streams is still necessary and domain knowledge is still necessary to analyse concept drift. Finally — this problem would be much more suited to a multi-label *ranking* problem, as opposed to classification. Hence we do not consider this idea further.

The task of generating synthetic multi-label data streams has not yet been thoroughly investigated, and has been mainly specific to certain algorithms or scenarios. In following sections, this paper presents a general framework for multi-label synthetic data generation designed to produce a wide variety of multi-label data in the form of a data stream.

3 Generating Multi-label Data Streams

The main novelty of the framework presented in this paper stems from the use of *problem transformation*, also known as *data transformation*, well known in the multi-label literature [8, 6, 9]. It has shown that it is possible to decompose multi-label data into single-label data. The *reverse transformation is also possible*: single-label data can be transformed into multi-label data. This allows for a generalised framework which can generate multi-label synthetic data independently of the actual data-generation process.

Just as problem transformation classification methods use existing single-label classifiers independently of the transformation method, synthetic multi-label generation can be carried out independently of the data generation method. The `MOA` framework[1] [4] provides state-of-the-art functionality for generating

[1] `http://www.cs.waikato.ac.nz/~abifet/MOA/`

single-label synthetic data streams under a variety of schemes, all of which could be used for creating a multi-label stream. The task of composing a realistic multi-label data stream from single-label data is discussed in depth in this section and later the synthetic data generated is evaluated in comparison to real world data.

Figure 1 outlines the overall process to generate a multi-label example. An initial single-label instance is generated according to label skew, and further single-label examples are generated and are added according to the probabilities that they should occur together. Both the feature spaces and label spaces are combined to form a multi-label example (X, S). All processes, including the combination of feature spaces $(X \oplus X')$ will be described in this section.

```
GenerateML()
 1  ▷ Generate and filter a single-label example according to skew
 2  (X, l) ← filter(Pr(l), SL.generateSL())
 3  ▷ Formation of a multi-label example
 4  (X, S ← {l})
 5  ▷ Adding multiple labels l' where l' ∉ S
 6  while |S| < β
 7        do
 8           ▷ Generate and filter a single-label example
 9           (X', l') ← filter(Pr(l'|S), SL.generateSL())
10           ▷ Combine the feature set, and add the label
11           (X, S) = (X ⊕ X', S = S ∪ l')
12  ▷ Hence new multi-label example: (X, S)
13  return (X, S)
```

Fig. 1: Generating a multi-label example. `SL.generateSL()` represents any single-label data stream generator from (for example the `MOA` framework). $filter(\gamma, D)$ filters instances from a single-label stream D according to γ, and β is a constant to help approximate a certain label cardinality.

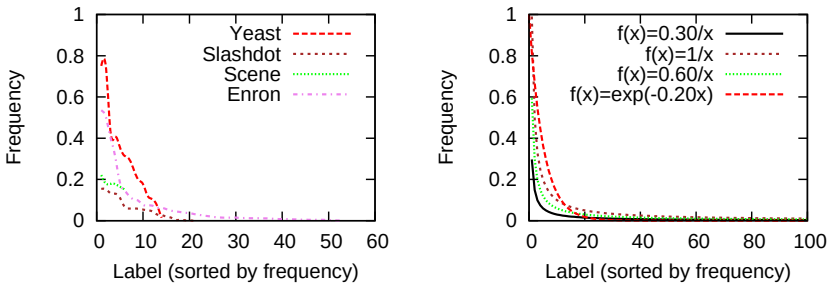
3.1 Label Skew

The phenomenon of *label skew*, where a label or subset of labels are particularly dominant or subordinate in the data, is not unique to multi-label data, but does tend to be particularly prevalent and exaggerated. This is due to the nature of multi-label data: each example can be associated with multiple labels and it is therefore inherently possible for more than one label to dominate the majority of examples (unlike single-label data). Skew in multi-label data is often intuitive, especially to text classification. A label such as `Economy` is likely to be relevant to many examples in a news articles corpus. It is also likely to be found in combination with other labels, for example {`Economy,Politics`}, or {`Economy,New`

`Zealand`}. Therefore `Economy` is likely to be very frequent in the data, while other labels, such as `New Zealand`, only refer to specific subset of news articles therefore occur much more infrequently.

Although label skew is naturally exaggerated in multi-label data by the process of adding multiple labels to single-label data, for the purpose of introducing and controlling concept drift (addressed below), finer grained control over this skew is necessary. Exponential or asymptotic distributions can be used to determine frequencies over a random ordering of the class labels. For example, $f(j) = \frac{\alpha}{j}$, where $f(j)$ represents the frequency of the jth label for some constant α.

New single-label examples can be filtered according to this distribution and the label skew of a data stream D can easily be manipulated by changing α or $f(j)$. When a dataset's label skew is ordered and plotted, a visual representation is obtained. Figure 2 displays the label skews of some real multi-label datasets and functions which approximate them.



(a) Real-world datasets skew (b) Function approximations

Fig. 2: Label skew for various datasets (a) and some function approximations (b). Labels (x axis) sorted by frequency (y axis).

3.2 Label Distribution

The most fundamental difference between multi-label data and single-label data is that instances may be associated with multiple labels, as opposed to a single class label. A multi-label dataset has an average number of labels assigned per example. The average number of labels per example is the *label cardinality*:

$$LC(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} |S_i|$$

Label distribution refers to the overall *composition of the label cardinality*: the frequency of label set sizes in the dataset. This can also be represented as a function $g(i) = n$ where n percent of instances have i labels assigned to them.

A general distinction can be made between two types of label distribution:

Type A : Most examples contain a single label. This is typical of many text and media classification scenarios where most examples fit naturally under a single label scheme, but multi-labelling has been introduced to resolve classification ambiguities. This is the most common type of multi-label data.

Type B : Most examples contain more than one label. The label set is usually very domain-dependent and chosen specifically to represent a multi-label scheme.

Examples of *Type A* include news articles, and media such as images and video. Most images, for example, may fit naturally into a single-label scheme and may have labels such as `Mountains`, `Forest`, or `Sea`. Multiple labels are used to resolve occasional ambiguities such as when `Mountains` and `Forest` are both relevant to one particular image. A good real-world example of this is the *Scene* dataset[2].

Examples of *Type B* include biological datasets where genes are expected to have multiple functions and text datasets like the *Enron* dataset. *Enron* originates from an e-mail corpus[3] and this version[4] of the dataset contains categories which almost take the form of a checklist and were obviously conceived with consideration for multi-label representation. A small subset of *Enron*'s label set is shown in Figure 3 to illustrate a *Type B* labelling scheme.

Label	Note
Attachment(s)	The e-mail contains attachment
Forwarded	The e-mail was forwarded
Legal Advice	The e-mail contains legal advice
Humor	Written with a tone of humour
Admiration	Written with a tone of admiration
...	...

Fig. 3: An example (from *Enron*) of *Type B* label distribution

The label distribution of both types approximates a Poisson distribution (Equation 1). Values of k and λ depend on the data type. *Type A*'s distribution can be approximated $0, POISS(k = \{0, 1, \cdots, |L|\}, \lambda \approx 0.25)$. *Type B*'s distribution can be approximated by $0, POISS(k = \{1, \cdots, |L|\}, \lambda = LC(D))$ (in the latter case $k = 1$ initially, and $LC(D)$ is as defined above).

$$POISS(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{1}$$

[2] *Scene* can be obtained from `http://mlkd.csd.auth.gr/multilabel.html#Datasets`

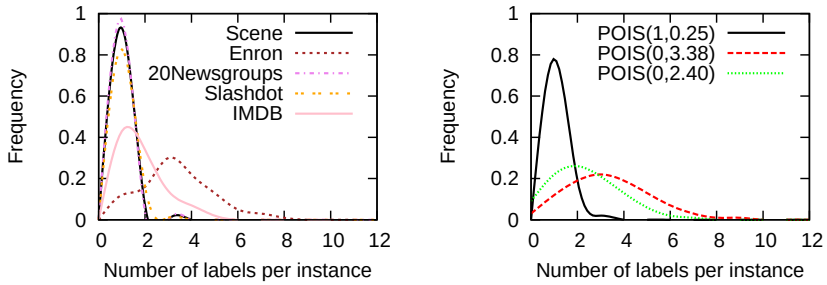
[3] `http://www-2.cs.cmu/~enron/`

[4] Using the labelling scheme `http://bailando.sims.berkeley.edu/enron_email.html`, obtainable from `http://www.cs.waikato.ac.nz/~jmr30/#datasets`

Figure 4 shows the label cardinality distributions of real multi-label datasets alongside the Poisson functions which approximate them.

In practice, label cardinality (LC) is never greater than about 5.0. If $LC(D) \gg 5.0$, the problem is usually better treated as a *hierarchical* problem, or *keyword* problem where keywords are *not* assigned based on a predetermined categorical structure intended to facilitate browsing, but rather searching, linking or lookup structures; likewise where $|L| \gg 100$.

Earlier, in Figure 1, a constant β is used to control the assignment of labels so as to adhere to one of the two distributions. This constant is closely linked to the desired label cardinality: i.e. $LC \approx \beta$.



(a) Real-world datasets (b) Function approximations

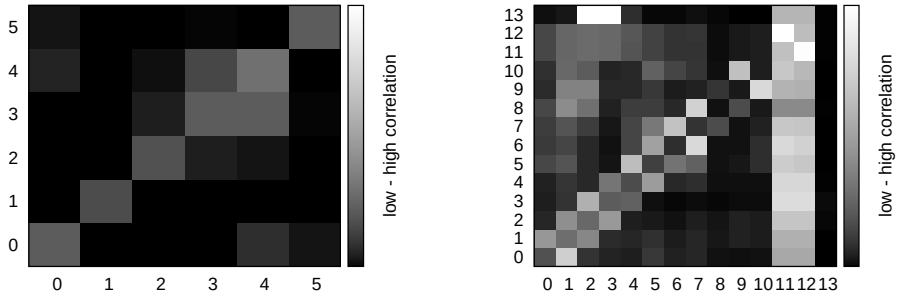
Fig. 4: Label distributions of real-world datasets (a) and approximations (b). The number of labels per example (x axis) against frequency (y axis).

3.3 Label Relationships

There is wide consensus in the literature about the existence of label interdependencies in multi-label data [6, 9, 7, 10]. The underlying relationships between labels in the data is reflective of the problem domain. The degree of label dependency varies, but any real world data in which labels are completely independent of each other is not an interesting multi-label problem, but rather $|L|$ separate binary filtering problems. This implies that labels cannot be selected independently or randomly to create a synthetically generated multi-label example.

Multi-label relationships usually emerge from a problem domain. These relationships can be viewed as a $|L| \times |L|$ *probability matrix* m where $m[k][j] = Pr(l_k|l_j)$. Figure 5 shows matrix representations for the *Scene* (a) and *Yeast* (b) datasets which represent *Type A* and *Type B* data, respectively. The label frequencies are displayed in the matrix diagonal, i.e. $m[k][k] = Pr(l_k)$.

The correlations are related to label skew (covered in Section 3.1). That is to say $Pr(l_j|l_k)$ is high if $Pr(l_j)$ is high, and correspondingly low when $Pr(l_j)$ is low. This is most noticeable in *Yeast* for labels l_{11} and l_{12} where these labels



(a) *Scene* dataset (Type A data). (b) *Yeast* dataset (Type B data).

Fig. 5: Label relationship matrices displayed as heatmaps. The matrix diagonal represents $P(l_i)$ for each $l_i \in L$.

are associated with high frequency and many correlations. Label l_8 and l_{13} show the converse. Other shade differences represent domain dependent factors which can be represented in synthetic data by randomisation. In this particular *Type A* dataset (*Scene*), there is only weak skew and the domain-dependent label correlations stand out clearer.

To generate an artificial matrix m to simulate domain-dependent label correlations, $\epsilon\%$ of rows under column $m[j]$ are filled with normally-distributed random numbers where $\mu = Pr(l_j)$ and $\sigma = 1.0$ (other cells are left as ≈ 0.0). ϵ is related to label cardinality and should be set low for *Type A* data (low label cardinality), and high for *Type B* data (higher label cardinality).

3.4 The Feature Space

A complete framework must be able to transform generated single-label examples into multi-label examples, and to do so must consider the *feature space*, and more importantly, the relationship between feature attributes and labels. Text data is both intuitive to examine, and also representative of the majority of multi-label data streams. Tables 1 and 2 show the most frequent words for labels occurring *exclusively* of each other, together *in combination*, and also the *global* most frequent words, for comparison. Figures 6a and 6b show the Gaussian distributions for specific examples taken from the tables. Slashdot[5] contains summaries of news articles and *20 Newsgroups*[6] contains newsgroup posts.

Referring to these tables and figures, two feature-label effects can be seen which contain information that can benefit a multi-label algorithm:

A *feature-label effect* is where a feature identifies a certain label. An intuitive example is in the *Slashdot* dataset where 'linux' pertains strongly to the label

[5] `http://slashdot.com`
[6] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

`Linux`, while 'mobile' pertains to `Mobile`, and both words are relevant where these labels are found in combination.

A *feature-combination effect* is where a feature identifies a combination of labels. Often some words may occur frequently only when two labels are found in combination. This is the case in the *20 Newsgroup* dataset for the word 'arms'. This feature is relevant to `politics.guns` but tends to occur even more frequently when the newsgroup post is also posted to `misc.religion`.

There are also various *random effects*. Words like 'anonymous' are generic and do not provide information regarding the presence of labels or combinations of labels. They may occur less frequently in a label combination simply because with an average paragraph length of n words, over several labels, there are fewer words between labels and words which are more strongly relevant (i.e. resulting from the *feature-label effect* and *feature-combination effect*) take preference.

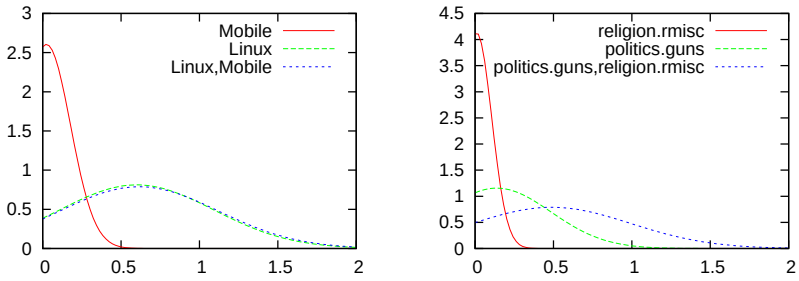
A surprisingly uncommon and irrelevant effect is the average occurrence of two features in a combination: $P(x|\{A,B\}) \approx (P(x|A) + P(x|B))/2$ for feature x and labels `A`,`B`. This effect can also be considered random because it does not tend to indicate the presence of either a specific label or combinations of labels.

Table 1: *Slashdot*. Most frequent words for labels `Linux` and `Mobile`

Global	`Linux`	`Mobile`	{`Linux`,`Mobile`}
anonymous	linux	mobile	linux
reader	ubuntu	iphone	open
game	source	anonymous	windows
story	open	reader	phone
reports	released	phone	netbook
world	anonymous	android	source
years	kernel	apple	mobile
released	software	phones	free

Table 2: *20 Newsgroups*. Most frequent words for labels `politics.guns` and `religion.misc`

Global	`politics.guns`	`religion.misc`	{`politics.guns`,`religion.misc`}
don	people	don	jews
1	don	people	arms
2	gun	christian	bear
people	time	god	don
time	government	years	koresh
good	fbi	good	fbi
make	guns	time	people
3	waco	make	news



(a) *Slashdot* for 'linux' (b) *20 Newsgroups* for 'arms'

Fig. 6: Word frequencies for certain labels individually and in combination.

In implementation, parameters can be used to influence the proportions of the two effects and a mapping is used to carry this out where each feature attribute is mapped to either a single label or a label pair or neither while the remaining proportion implies random effects. Each attribute either implies the presence of a particular label (the *feature-label effect*), implies the presence or absence of a particular combination the (*feature-combination effect*), or does not imply anything (a *random effect*). The process is outlined in Figure 7.

3.5 Concept Drift

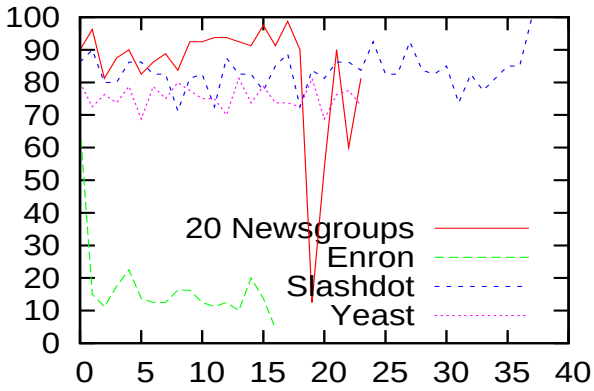
It is known that real-world data streams inevitably begin to show changes to the concepts they represent [1]. This known as *concept drift*. If the concept drift is particularly abrupt, it may be called *concept shift*.

In addition to concept drift in the feature space, as found in single-label data streams, multi-label data also involves concept drift to label cardinality, label skew, label distribution, label-label relationships and feature attribute-label relationships. All of these have been discussed above. Some multi-label concept drift may also involve a change to the label set (L) itself. Figures 8a and 8b show the effect under two measures of shift: "label set coverage" refers to the percentage of instances where label sets overlap the label sets of the initial instances. Label combinations are recorded for the first 100 examples $d_0, \cdots, d_{99}$ and then the percentage of reused combinations is plotted for each of the following blocks of 100 examples: $d_{100}, \cdots, d_{199}, d_{200}, \cdots, d_{299}, \cdots$. This is a form of measuring concept drift in the *label space*. "Accuracy" refers to classification performance under Naive Bayes with a threshold to create multi-label sets (refer to the *ranking and threshold* method in Section 4.1). This is a way to measure concept drift in both the *instance space*, and *feature space*. *Yeast* is a randomised batch dataset, as opposed to a stream, and is displayed for purposes of comparison. In both cases, there is indication of concept drift when the plot is unstable. *20 Newsgroups* shows a very abrupt change in the

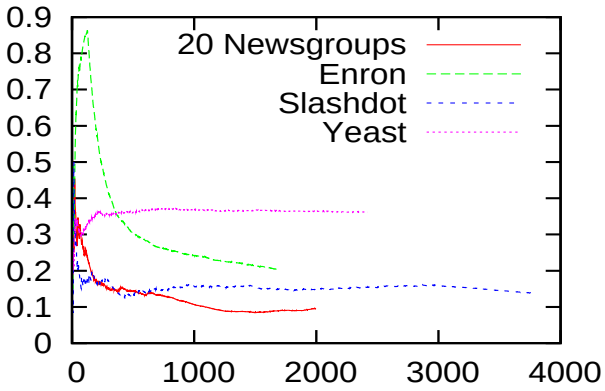
```
CreateML()
▷ Begin with ML example of an empty instance X and relevant labels S
(X = (x_1, x_2, ··· , x_N) ∈ 0^N, S)
▷ Generate SL examples to use, one for each label in S
(W_1, l_1), (W_2, l_2), ··· , (W_|S|, l_|S|) : l_i ∈ S
▷ Generate two binary examples; one positive, one negative
(V_1, 0), (V_2, 1)
▷ A mapping of feature attributes to labels or label pairs
ζ
▷ For each feature attribute in the feature space X
for a ← 1 ... |X|
    do
        ▷ if the attribute maps to a single label
        if |ζ[a]| ≤ 1
            then
                ▷ and if that label is relevant to this example
                if ζ[a] = l_i : l_i ∈ S
                    then
                        ▷ Use value from relevant SL example (W_i, l_i)
                        X[a] = W_i[a]
                    else
                        ▷ Use average from all SL examples (random effect)
                        X[a] ← AVG(W_1[a], W_2[a], ··· , W_|S|[a])
                ▷ otherwise, if the attribute maps to a label pair
            else if |ζ[a]| = 2
                    then
                        ▷ and if that label pair is relevant to this example
                        if ζ[a] ⊆ S
                            then
                                ▷ Use value from positive binary example
                                X[a] = V_2[a]
                            else ▷ Use value from negative binary example
                                X[a] = V_1[a]
        ▷ A ML example with completed feature space and label set
        (X, S)
```

Fig. 7: Creating a multi-label example from several single-label and binary examples into a multi-label example. The process can be governed by the mapping ζ to either influence more of either effect and the empty set $\emptyset$ can be used to create a random effect.

label space, while *Enron* shows pronounced drift early on. *Slashdot* varies only slightly more than the batch dataset *Yeast*.



(a) Label set coverage over time



(b) Average accuracy over time

Fig. 8: Label set coverage and accuracy measured over time on real-world data sets.

Recent work by [1] on single-label data streams models concept drift with a sigmoid function. The sigmoid function in Equation 2 represents concept drift for instances $d_0 \cdots d$. This function is also suitable for creating concept drift in multi-label data, where sigmoid functions are applied to all aspects of multi-label data: the label skew, distribution, and relationship matrix.

$$sig(d) = \frac{1}{(\Delta x + e^{-s(d-d_0)})} \tag{2}$$

A value x may represent any value of the original concept, and x' the same value in the new concept. To generate a new concept, x' is chosen randomly from a Gaussian distribution where $\mu = x$ and $\sigma = v$ where if $x' < 0 || x' > 1$ then $x' = -x'$, and where v is supplied as a global parameter to control the *extent* of concept drift. Hence the change for a value x is $\Delta x = (x' - x)$.

The variable s controls the *abruptness* of the drift. Large values of s create rapid concept drift while smaller values create a more gradual concept drift. The value of s is directly related to the length of change $(d_0 - d)$ via a constant e.g. $(d_0 - d) = \frac{s}{8}$.

Figure 9 displays sigmoid functions given different values of s and v. Note that in practice, the functions would be centered around $x' - ((x' - x)/2)$ and not 0.5.

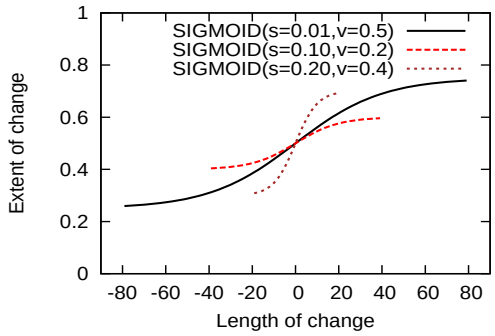


Fig. 9: Sigmoid functions for different changes under different values of s and Δx (v).

4 Results and Discussion

Table 3 shows the range of parameters for generating multi-label data streams under the `MOA`-based framework. An approximation of label cardinality $(-z)$ and a mapping of feature-label relationships $(-a, -b)$ is sufficient to influence all multi-label dimensions of a dataset.

4.1 Resulting Datasets

In Table 4, statistics are displayed of three real-world multi-label datasets and two synthetic multi-label datasets generated using the framework introduced in this paper. *Synth6* has been designed to approximate *Scene* and *Synth8* is intended to represent a new *Type B* dataset.

Table 5 displays the performance of various standard multi-label base methods on the same datasets as in Table 4. The *majority combination* simply selects

Table 3: Possible parameters for synthetic data generation.

parameter	type	description	symbol
$-g$	class	single-label generator	`SL`
$-i$	int	number of instances	$\|D\|$
$-c$	int	number of labels	$\|L\|$
$-u$	int	number of attributes	$\|X\|$
$-r$	int	random seed option	
$-z$	float	desired label cardinality	β, ϵ
$-a$	float	proportion *Label-Effect* mappings	ζ
$-b$	float	proportion *Combination-Effect* mappings	ζ
$-v$	int	average extent of change	Δ
$-x$	int	length/range of change	$d - d_0$
$-p$	int	beginning point of change	d_0

Table 4: Statistics relating to real-world and synthetic datasets.

Method	*Scene*	*Synth6*	*Yeast*	*Enron*	*Synth8*
$\|L\|$	6	6	14	53	23
$\|X\|$	300	300	100	1000	500
Type	A	A	B	B	B
Attributes	num.	num.	num	sparse	num.
Label Cardinality	1.07	1.06	4.24	3.38	2.73
Percent Unique	0.006	0.012	0.082	0.442	0.313

the most popular label combination for each test example. The other methods are well known problem transformation methods, all reviewed in [8]. The *label powerset* method treats each multi-label set as a single label, the *binary relevance* method treats each label as a separate binary problem, and the *ranking and threshold* method ranks the relevance of labels to each test example and selects a subset of the highest ranked labels using a threshold to be the multi-label classification set.

Problem transformation methods require a base single-label classifier to carry out classifications. We use Naive Bayes as the base classifier, which allows incremental classification, even for the label powerset method which requires labels to be added dynamically. The table compares the accuracy of these methods. Accuracy is determined as in [8], but in this case the accuracy is measured for each new example in the stream in a *test then train* scenario.

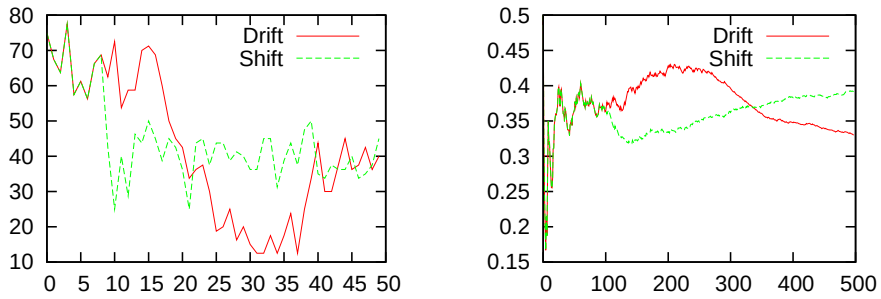
The variety in the results is to be expected due to the different dimensions of each dataset and each method but, importantly, accuracy is higher than the default method. This means that the method for combining label and feature spaces is creating multi-label relevant information simulative of real-world data.

Finally synthetic concept drift is considered. Figure 10 plots the *label set coverage* 10a and *average accuracy* 10b over time. Label set coverage varies over the range of the *drift*, before stabilising afterwards. In terms of *shift*, the coverage drops sharply and stabilises. Accuracy decreases suddenly at the beginning of

Table 5: The average accuracy of various methods on real and synthetic datasets.

Method	*Scene*	*Synth6*	*Yeast*	*Enron*	*Synth8*
Majority Combination	18.31	18.50	39.05	17.11	20.17
Label Powerset	60.60	34.50	46.63	42.47	37.25
Binary Relevance	46.22	27.50	42.28	18.73	31.42
Ranking and Threshold	65.60	30.25	34.71	23.31	26.37

the *shift*, but is able to gradually recover, whereas it declines more slowly over the longer period of the *drift* and is more negatively effected in the long run. This is comparable to the analysis of real-world data earlier in Figure 8.



(a) Label set coverage over time

(b) Average accuracy over time

Fig. 10: Label set coverage and accuracy measured over time on a synthetic data set.

5 Conclusions and Future Work

This paper conducted an in-depth analysis of multi-label data and how the concepts relating to such data change over time in a data stream context. This lead to a framework for generating synthetic multi-label data streams. This framework is based on the concept of problem transformation – it creates a multi-label data stream from a single-label data generator independently of the actual data generation process. It is possible to generate a wide variety of multi-label data by configuring a number of parameters. These parameters allow the manipulation of the multi-label aspects of the data as well as the introduction of concept drift.

Analysis indicates that the data is closely representative of real-world data and therefore able to serve for the analysis and evaluation of incremental multi-label algorithms.

Future work will involve conducting large-scale evaluations of multi-label algorithms using the synthetic multi-label data streams which the framework

is capable of generating. This will aid investigations into the multi-label data stream context.

References

1. Albert Bifet and Ricard Gavaldà. Adaptive parameter-free learning from evolving data streams. Technical report, LSI R09-9 Departament de Llenguatges i Sistemes Informàtics Universitat Politècnica de Catalunya, 2009.
2. Lijuan Cai. *Multilabel Classification over Category Taxonomies*. PhD thesis, Department of Computer Science, Brown University, May 2008.
3. Svetlana Kiritchenko. *Hierarchical Text Categorization and its Application to Bioinformatics*. PhD thesis, Queen's University, Kingston, Canada, 2005.
4. Richard Kirkby. *Improving Hoeffding Trees*. PhD thesis, Department of Computer Science, University of Waikato, 2007.
5. Sang-Hyeun Park and Johannes Fürnkranz. Multi-label classification with label constraints. Technical report, Knowledge Engineering Group, TU Darmstadt, 2008.
6. Jesse Read, Bernhard Pfahringer, and Geoff Holmes. Multi-label classification using ensembles of pruned sets. *ICDM '08: IEEE International Conference on Data Mining*, 0:995–1000, 2008.
7. Liang Sun, Shuiwang Ji, and Jieping Ye. Hypergraph spectral learning for multi-label classification. In *KDD '08: 14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 668–676, New York, NY, USA, 2008. ACM.
8. Grigorios Tsoumakas and Ioannis Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2007.
9. Grigorios Tsoumakas and Ioannis P. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML '07: 18th European Conference on Machine Learning*, 2007.
10. Rong Yan, Jelena Tesic, and John R. Smith. Model-shared subspace boosting for multi-label classification. In *KDD '07: 13th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 834–843, New York, NY, USA, 2007. ACM.