

# A Normalization Method for Contextual Data: Experience from a Large-Scale Application

Sylvain Létourneau<sup>1</sup>, Stan Matwin<sup>2</sup>, and Fazel Famili<sup>1</sup>

<sup>1</sup> Integrated Reasoning Group,  
National Research Council of Canada, Ottawa  
{sletour, fazel}@ai.iit.nrc.ca

<sup>2</sup> School of Information Technology and Engineering,  
University of Ottawa, Canada  
stan@site.uottawa.ca

**Abstract.** This paper describes a pre-processing technique to normalize contextually-dependent data before applying Machine Learning algorithms. Unlike many previous methods, our approach to normalization does not assume that the learning task is a classification task. We propose a data pre-processing algorithm which modifies the relevant attributes so that the effects of the contextual attributes on the relevant attributes are cancelled. These effects are modeled using a novel approach, based on the analysis of variance of the contextual attributes. The method is applied on a massive data repository in the area of aircraft maintenance.

**Keywords:** Learning in contextual domains, attribute normalization, data-mining.

## 1 Introduction

In this paper, we address the problem of learning models from contextual data, i.e. data acquired from a system operating in a dynamic environment. We assume that the contextual data contains information on both the environment and the system under study. A wide variety of real world industrial applications generate such contextual data. The information on the environment is typically represented by a limited set of *contextual attributes* such as humidity, temperature, and pressure while the domain specific information is described by a set of *relevant attributes* such as start-time and exhaust gas temperature. What we need is a learning approach that will learn appropriate models while taking into account the effects of the contextual attributes on the relevant attributes.

The pre-processing domain independent method proposed here is a form of feature selection/extraction approach in which we normalize the relevant attributes with respect to the contextual attributes. In this manner their influence will be taken into account by a one-time transformation of the values of performance attributes, which will remove the dependency. The output of our pre-processing method is a new set of relevant attributes that are available for

learning. The models obtained from the new normalized attributes will be more general than the ones obtained without normalization, since they are independent from the variations in the contextual attributes.

The paper describes our approach for handling contextual data as well as results obtained from experiments with a large volume of commercial aircraft engine data. The data consists of automatically acquired sensor measurements from the auxiliary power units (APU) of 34 Airbus A320 over the last two years along with all repairs done on these aircraft engines. The end goal of the project is to discover patterns from these data that can be used to predict component failures.

One of the advantages of our application was the existence of manufacturer’s normalization formulas which come from statistical experiments with real engines operating in a controlled environment. For the purpose of evaluation, we have cast the prediction of an engine component failure as a classification task. We have ran a standard inductive learner on this task using non-normalized data, normalized data with manufacturer’s formulas, and normalized data obtained with our approach. We have compared the results with respect to the standard accuracy. We have also compared the results of learning from different datasets with respect to the numbers of false positives, as this type of errors is critically important in our application.

The problems related to context-sensitive applications are fairly new to the machine learning community. Turney & Halasz [4] describe a project in which contextual normalization is applied to the diagnosis of engine faults. The normalization in that work is performed by a transformation based on the average and standard deviations of the contextual parameters.

Related to dynamic environments, Taylor & Nakhaeizadeh [3] describe in general the state of the art in learning in dynamically changing domains. They list a number of directions for further research, leading to the the development of robust, scalable systems for dynamic environments. Our work belongs to two of the three fundamental subproblems raised by them: drift/change detection, adaptation/classifier update, characterization and/or explanation of change.

The paper is organized as follows. Section 2 describes the normalization approach. Section 3 describes the data used for validation as well as the results. Section 4 consists of discussion and conclusion. This is a report on the work in progress. A longer version of this paper [2] is available.

## 2 The normalization approach

The normalization approach proposed here is a pre-processing technique that transforms the contextual sensitive data in order to cancel the effects of the contextual attributes. As usual a fixed set of attributes is used to represent the data. The attributes are divided in two categories: the performance attributes  $p_1, \dots, p_m$  and the contextual attributes  $c_1, \dots, c_k$ . Contextual attributes can be numeric or symbolic while performance attributes have to be numeric. The

output of the normalization process is a set of normalized performance attributes  $p'_1, \dots, p'_m$  that are no longer affected by the variations in  $c_1, \dots, c_k$ .

The normalization approach is composed of two steps: the *contextual analysis* and the *normalization*. In the contextual analysis, we model the effects of each contextual attribute on the performance attributes. During the normalization step, we use the output of the contextual analysis to cancel the effects of the contextual attributes. These two steps may need to be repeated more than once to completely cancel the effects of contexts. Figure 1 presents the global algorithm of the proposed approach. The following sections describe the contextual analysis and the normalization processes.

**Algorithm RemoveContextEffects**

*Input:* A dataset in which each of the  $N$  instances is described by a set of attributes  $\{p_1, \dots, p_m, c_1, \dots, c_k\}$ , where  $p_i$  and  $c_i$  are the performance and contextual attributes, respectively.

*Output:* A normalized set of performance attributes  $\{p'_1, \dots, p'_m\}$

```

{ $CM_1, \dots, CM_k$ } = ContextualAnalysis( $\{p_1, \dots, p_m, c_1, \dots, c_k\}$ );
While  $\exists$  a cluster name  $\neq 1$  in  $\{CM_1, \dots, CM_k\}$  do begin
     $\{p'_1, \dots, p'_m\} =$  Normalize( $\{p_1, \dots, p_m, c_1, \dots, c_k\}, \{CM_1, \dots, CM_k\}$ );
     $\{CM_1, \dots, CM_k\} =$  ContextualAnalysis( $\{p'_1, \dots, p'_m, c_1, \dots, c_k\}$ );
end;
return the new set of performance values  $\{p'_1, \dots, p'_m\}$ ;

```

**Fig. 1.** Remove Contextual Effects Algorithm

## 2.1 Contextual Analysis

The aim of the contextual analysis is to model the effects of the contextual attributes on the performance attributes. We perform this step using the analysis of variance framework. The contextual attributes are analyzed independently. The procedure is as follows. We first partition each  $c_i$  into a set of intervals. A unique label is assigned to each interval. Secondly, we map  $c_i$  to  $c'_i$  by replacing each value by its corresponding interval label. Finally, we cluster this set of intervals for each of the  $m$  performance attributes. Figure 2 summarizes this procedure. As output, the contextual analysis returns a set of  $k$  contextual matrices  $CM_1, \dots, CM_k$ , where each  $CM_i$  describes the clustering of the performance attributes  $p_1, \dots, p_m$  according to  $c_i$ .

**Partitioning a contextual attribute into a set of intervals** The first step is to partition each  $c_i$  into interval cells that are appropriate for ANOVA. The performance attributes are not used during the partitioning step. The number of intervals created may differ from one contextual attribute to another. A slightly different approach is used for numerical and symbolic attributes. When  $c_i$  is symbolic, an interval is defined for each observed outcome (i.e. each interval corresponds to a value). When  $c_i$  is numeric, the intervals are built in such a way that i) the intervals are approximately of equal size, and ii) each interval has a minimum of 50 elements. We introduce these conditions due to their impact on

**Algorithm ContextualAnalysis**

*Input:* A dataset as described in Algorithm RemoveContextEffects.

*Output:* A set of  $k$  matrices  $CM_1, \dots, CM_k$  describing the contextual effects of each  $c_i$ .

For  $i:=1$  to  $k$  do begin

    Partition  $c_i$  in a set of  $r$  intervals labeled  $1, \dots, r$ ;

    Record definition of intervals in first two columns of  $CM_i$ ;

    For  $j:=1$  to  $N$  do  $c'_i[j] := \text{IntervalLabelOf}(c_i[j])$ ;

    For  $j:=1$  to  $m$  do begin

$V := \text{ClusterIntervals}(p_j, c'_i)$  ;

        Record vector  $V$  in column  $2 + j$  of  $CM_i$ ;

    end;

end;

return  $CM_1, \dots, CM_k$ ;

**Fig. 2.** Contextual Analysis Algorithm

the results of the ANOVA. The set of intervals for  $c_i$  are labelled from  $1, \dots, r_i$ . We then create  $c'_i$  by mapping each value of  $c_i$  to its corresponding interval label.

**Cluster the intervals: Use of ANOVA** The attribute  $c'_i$  is now used to find clusters of intervals that model the effects of  $c_i$  on each  $p_j$ . We use the ANOVA approach as a basis for the search for these clusters. The ANOVA is a remarkably robust and efficient technique to handle large volume of data.

As shown in Figure 2, the procedure that clusters the intervals is repeated for each pair of  $c_i$  and  $p_j$ . The procedure starts by performing an ANOVA to test whether  $c'_i$  has an overall effect on  $p_j$ . If there is no effect then it assigns all intervals to a unique cluster (i.e. cluster #1). If there is an effect then the procedure performs pairwise mean comparisons (with a  $t$  test equivalent to the Fisher's protected LSD test). If means of  $p_j$  over intervals  $l$  and  $m$  are different (at a level of 0.05) then it assigns intervals  $l$  and  $m$  to two different clusters, otherwise it assigns these intervals to the same cluster. As output, the procedure returns a 1 by  $r_i$  vector  $V$  that describes the clustering obtained for the intervals of  $c_i$ .

## 2.2 Normalization

The normalization step leads to a new set of performance attributes  $\{p'_1, \dots, p'_m\}$  for which the effects described in  $CM_1, \dots, CM_k$  are reduced. The normalization algorithm is summarized in Figure 3. The basic idea is very simple: we normalize each performance value  $p_j[i]$  ( $j = 1, \dots, m$  and  $i = 1, \dots, N$ ) by adding to it a *context penalty number*, denoted by  $\delta$ .  $\delta$  is defined as the difference between the expected value of  $p_j[i]$  (noted  $\beta$ ) and the overall mean of  $p_j$  (noted  $\bar{p}_j$ ).  $\beta$  combines the effects of the contextual attributes by averaging the  $\bar{p}_{j,clu_l}$  for  $l = 1, \dots, k$ , where  $\bar{p}_{j,clu_l}$  denotes the mean of  $p_j$  in cluster  $clu_l$ . In the current version of the algorithm, we consider all attributes as equally important during the computation of  $\beta$ . However, we think that a faster convergence rate could be obtained by the use of a weighting average that would take into account the relative importance of each contextual attribute on  $p_j$ . The information contained in the  $CM_i$  seems to have some potential to weigh the contextual attributes, but this issue hasn't been investigated yet.

**Algorithm Normalize**

*Input:* i) A dataset as described in Algorithm RemoveContextEffects.

ii) A set of  $k$  matrices  $CM_1, \dots, CM_k$  generated by ContextAnalysis.

*Output:* A (partially) normalized set of performance attributes  $\{p'_1, \dots, p'_m\}$ .

For  $i:=1$  to  $N$  do begin

  For  $j:=1$  to  $m$  do begin

    Determine context status  $C$  for the value  $p_j[i]$ ;

    Compute  $\beta$ , the expected value of  $p_j$  given  $C$ ;

$\delta := \beta - \bar{p}_j$  {  $\delta$  is the *contextpenalty* number and  $\bar{p}_j$  is the mean of  $p_j$ . }

$p'_j[i] := p_j[i] + \delta$ ;

  end;

end;

return the new set of performance values  $\{p'_1, \dots, p'_m\}$ ;

**Fig. 3.** Normalization Algorithm

### 3 Experimental results

The data used in our experimentation comes from the Auxiliary Power Unit engines of a fleet of Airbus A-320. The dataset consists of 31059 cases of 23 attributes (2 symbolic, 19 numeric, and 2 for date and time of the event). The task was to develop classifiers to predict starter motor failures. Domain experts have identified three performance attributes for the starter problem: *STA*, *EGP*, and *NPA*. These attributes are all affected by variations in the environment (pressure, temperature, altitude, etc.) We used our normalization approach to normalize these performance attributes according to the remaining 18 contextual attributes (2 symbolic and 16 numeric). As usual in diagnosis problem, early predictions would have been appreciated, but the client insisted on the fact that false alarms (i.e. prediction of a problem when there is no problem) should be avoided as much as possible. To address these issues, we have selected four targets for prediction: 45 days, 30 days, 15 days, and 10 days. For each of these targets, we have developed four classifiers by respectively using: 1) the initial 21 attributes, 2) the three initial performance attributes before normalization, 3) the three normalized attributes obtained using formulas provided by the manufacturer of the APUs, and 4) the three normalized attributes obtained from our approach. Table 1 presents the average accuracies and average number of false alarms (obtained from two-fold cross-validation experiments) of the classifiers developed.

Attributes used	Task 45 days		Task 30 days		Task 15 days		Task 10 days	
	Acc	F.Alarm	Acc	F.Alarm	Acc	F.Alarm	Acc	F.Alarm
21 APU att.	94.9	141.5	96.0	131.0	97.8	74.5	98.6	31.5
3 init. perf. att.	94.3	81.5	96.1	80.0	97.9	57.0	98.6	27.5
3 Manuf. norm. att.	94.2	144.0	96.2	78.0	98.0	51.0	98.5	40.0
3 new norm. att.	94.3	85.0	96.0	61.0	98.0	47.5	98.5	35.0

**Table 1.** Accuracies and number of false alarms (false positive errors) for the runs for the different tasks

Since the four target tasks use different data, one should only compare results within one column. Comparing just the accuracies for different classifiers is not of great help for two reasons: i) the accuracies of the classifiers in each task are

very close, and ii) no classifier consistently outperformed the others over the four learning tasks. The inadequacy of simple accuracies for this problem can be explained by the fact that our data sets are imbalanced[1]. In terms of number of false alarms, values obtained from our approach performed very well. In two cases (30 and 15 days), they have lead to the minimum number of false alarms. In the other two cases (45 and 10 days), the numbers of false alarms with the new values were close to the minimum number obtained. It is also interesting to note that the classifier obtained with values from our approach generates fewer false alarms than the one obtained with the manufacturer's formulas in all four tasks.

## 4 Discussion and conclusion

It is important to note that the approach presented in this paper is less sensitive than ANOVA regarding the violation of the following assumptions: i) normality of the data, ii) equal variances for the different groups, and iii) independent error components. In fact, the only impact of a violation of these assumptions will be on the convergence rate. When these assumptions do not hold, we are likely to increase the number of errors during pairwise mean comparisons which will results in non-optimal normalization in each iteration of the approach. As a consequence, the overall process will be slower.

In this paper, we describe a pre-processing method that cancels the effects of contextual attributes. This method includes two algorithms: one for contextual analysis and the other for normalization. Evaluating our approach, we have developed classifiers for prediction tasks. Results showed that the number of false alarms would be substantially lower when we used our normalized attributes instead of using the ones obtained from manufacturer-supplied formulas.

We believe that our approach has a lot of potential for performing advanced data analysis in context sensitive domains where the class attribute is not known ahead of time. The approach could be applied for timely prediction of failures that in most cases is very expensive to deal with. Finally, our approach can also be used for dimensionality reduction so that data analysis is performed more precisely.

**Acknowledgments** The authors would like to thank Air Canada for providing the data and the very useful feedback on this research. The first author is partially supported by the Natural Sciences and Engineering Research Council of Canada.

## References

1. Kubat, M., Holte, R. C., Matwin, S.: Machine Learning for the Detection of Oil Spills in Satellite radar Images Machine Learning Journal, Special Issue on Applications of ML. (1998) to appear.
2. Letourneau, S., Matwin, S., Famili, F.: A Normalization Method for Contextual Data: Experience From A Large- Scale Application. TR-98-02, [www.csi.uottawa.ca/~stan/public\\_html/techrep/TR-98-02.ps](http://www.csi.uottawa.ca/~stan/public_html/techrep/TR-98-02.ps).
3. Taylor, C., Nakhaeizadeh, G.: Learning in Dynamically Changing Domains: Theory Revision and Context Dependence Issues. Proceedings of ECML-97. (1997) 353-360.
4. Turney, P, Halasz, M: Contextual Normalization Applied to Aircraft Gas Turbine Engine Diagnosis. Journal of Applied Intelligence 3 (1993) 109-129