Multiple-Instance Regression with Structured Data

Kiri L. Wagstaff
Jet Propulsion Laboratory
California Inst. of Technology
Pasadena, CA 91109
kiri.wagstaff@jpl.nasa.gov

Terran Lane
Dept. of Computer Science
University of New Mexico
Albuquerque, NM 87131
terran@cs.unm.edu

Alex Roper
California Inst. of Technology
1200 E. California Blvd.
Pasadena, CA 91125
alexr@ugcs.caltech.edu

Abstract

We present a multiple-instance regression algorithm that models internal bag structure to identify the items most relevant to the bag labels. Multiple-instance regression (MIR) operates on a set of bags with real-valued labels, each containing a set of unlabeled items, in which the relevance of each item to its bag label is unknown. The goal is to predict the labels of new bags from their contents. Unlike previous MIR methods, MI-ClusterRegress can operate on bags that are structured *in that they contain items drawn from a number of distinct (but unknown) distributions. MI-ClusterRegress simultaneously learns a model of the bag's internal structure, the relevance of each item, and a regression model that accurately predicts labels for new bags. We evaluated this approach on the challenging MIR problem of crop yield prediction from remote sensing data. MI-ClusterRegress provided predictions that were more accurate than those obtained with non-multiple-instance approaches or MIR methods that do not model the bag structure.*

1 Introduction

Classical supervised learning methods operate on individual items, each represented by a feature vector and assigned a label, which is either categorical (for classification) or real-valued (for regression). However, some learning problems do not fit this model. There are situations in which observations are instead *bags* of items, with a single label applied to the bag. These tasks require a *multiple-instance learning* (MIL) approach. For example, the problem that first motivated this area of research was to predict a drug's activity ("active" or "inactive") given observations of multiple structural conformations of the drug molecule [5]. Only some of the observed conformations contributed to the label of the molecule, but it was not known which ones were relevant.

Less work has been done in the area of *multiple-instance regression* (MIR). When the bags have real-valued labels, the goal is to construct a regression model that can predict a bag's label from its contents. Primary-instance regression [14, 4] assumes that a single item in each bag dictates the label. It can select primary instances for labeled bags, but it cannot make predictions for a new bag unless it is known *a priori* which item is the primary one. Other methods [13] assume that all items in the bag are relevant; they generate predictions for a new bag by predicting the outcome for each item inside the new bag and aggregating those results into a bag label. No existing methods have demonstrated the ability to identify which subset of items in a new bag are relevant.

The core assumption of existing approaches is that the bags themselves are *unstructured*: the items in the bag are drawn from a single, fixed distribution. For example, in drug activity prediction, all items in a bag are different conformations of the same molecule. However, in some domains (such as the remote sensing data we study in this paper), the bags are *structured*: they contain items drawn from a number of distinct underlying data distributions.

The main contribution of this paper is MI-ClusterRegress, an algorithm that learns a regression model from structured multiple-instance data. On the spectrum of data structure, from purely propositional (i.i.d. data described by a single table) to purely relational (highly auto-correlated data described by multiple arbitrary relations), "traditional" multiple-instance data is a step beyond propositional data in that it is described by two tables: one maps items onto bags, and the other maps bags onto labels. Our approach takes another step along this spectrum by also mapping items onto (hidden) cluster labels.

MI-ClusterRegress leverages the within-bag structure by modeling the distinct data components with a clustering step and then constructing local regression models for each component. MI-ClusterRegress includes a final model selection step that picks the best cluster/regression model.

This step essentially asks, "which cluster is most associated with the target value?" As a side effect of constructing the local models, MI-ClusterRegress also estimates the relevance of each data item with respect to the bag label. MI-ClusterRegress classifies previously unseen bags by mapping their contents onto the best-fit cluster selected during training. The result is fed into the regression function of the corresponding best-fit local model, yielding a scalar prediction for that bag.

A compelling and large-scale problem that requires multiple-instance regression is crop yield prediction from remote sensing data [16]. Early prediction of expected crop yields is a priority of the United States Department of Agriculture (USDA) for two major reasons: crop yield estimates can help establish pricing strategies, and an early warning of low yield can inform corrective crop management strategies (precision agriculture). Earth-orbiting instruments such as MODIS (the MODerate resolution Imaging Spectroradiometer) collect global observations often enough that they can be used to assess the current condition of crops and thereby inform a crop yield prediction model. The USDA tracks crop yield at the per-county level. This is a MIR problem because each county (bag) has a single label but contains thousands of pixel-level multispectral sensor observations, with no guidance as to which pixels are relevant to the output. The bags are structured in that they are composed of pixels drawn from a variety of underlying distributions (e.g., corn, wheat, forest, cities, water), while the bag label (e.g., corn yield) is associated with only one of those distributions.

Our experiments on synthetic data and on crop yield prediction confirm the need for explicitly modeling internal bag structure when it is present (Section 4). Compared to other learning methods on the same data, test error on the synthetic data is reduced by orders of magnitude, and error goes down by 1–10% when predicting crop yield.

We also demonstrate the ability to identify relevant items, a particularly useful advance in the crop yield prediction domain. Our approach can provide per-pixel crop maps, a data product not currently available from any source.

2 Related Work

A significant amount of work has been devoted to methods for multiple-instance classification, including axis-parallel rectangles to capture the target concept [5], diverse density [11], voting by the k nearest neighbor bags [17], support vector machines [2], and graph spectral methods [12]. Extensions beyond classification have progressed in two directions: MIL with real-valued labels and multiple-instance regression (MIR). As in MIL classification, the goal of MIL with real-valued labels is taken to be the identification of a target concept t. The real-valued bag labels are interpreted as the proximity (or similarity) of an item/bag to t. Amar et al. [1] extended the k nearest neighbor and diverse density approaches to bags with real-valued labels. Goldman and Scott [8] interpreted the bag label to be "the degree to which the example satisfies the target concept" and used axis-aligned rectangles to learn the target concept. While useful for identifying a target concept, none of these approaches are designed to model or learn general regression relationships.

In contrast, MIR seeks to build a regression model that maps bags to real-valued outputs; there is no notion of a target concept t. Ray and Page [14] pioneered this area by developing a primary-instance regression (PIR) method. The PIR approach assumes that the label of a bag is determined by exactly one *primary instance* and that the rest of the items in the bag are noisy observations of the primary instance. PIR is an EM-based solution that alternately selects the most likely primary instance for each training bag and then to maximizes the fit of a linear regression through the primary instances. The learned model can only be applied to new bags if the primary instance for each one is known. Cheung and Kwok [4] and Ray [13] identified problem domains in which it is possible to assume that the primary instance is the one with the largest output value. For other domains, min, average, or sum are appropriate combining functions, and it is possible to learn which of these four functions applies to a given data set [13]. However, none of these functions models per-item relevance to the bag label, so the presence of irrelevant items will skew the results.

Methods that directly estimate item relevance include CH-FD for classification [6] and QPAP-Salience [16] for regression. These techniques use alternating optimization to iteratively estimate item relevances and coefficients for the learned models that predict bag labels. However, neither method can generalize to new bags, where both the bag labels and item relevances are unknown. CH-FD was evaluated by applying the learned classifier only to individual items, not bags. QPAP-Salience was not evaluated on new data.

This paper advances the state of the art by proposing a method that addresses both goals: assigning per-item relevances and building regression models that can generate predictions for new bags. These goals are achieved by explicitly modelling internal bag structure.

3 Multiple-Instance Regression

In the multiple-instance regression problem, we seek a function that maps bags to real values. In many cases, the bags are also structured: the bag contents are drawn from a variety of different underlying distributions, not all of which are relevant to the bag labels.

Algorithm 1 The MI-ClusterRegress$(\mathbf{D}, Y, k)$ algorithm

```
 1: Inputs: bag data $\mathbf{D} = \{\mathbf{B}^i\}_{i=1\cdots m}$, labels $Y$, number of clusters $k$
 2: Outputs: regression parameters $\psi'$ and cluster parameters $\theta'$ for the best local model
 3: $\mathbf{X} := \bigcup_{i=1\cdots m} \mathbf{B}^i$  // Concatenation of all items into single set, ignoring bag structure
 4: $\theta_{i=1\cdots k} :=$ Cluster$(\mathbf{X}, k)$  // Cluster all items into $k$ clusters
 5: for $i = 1$ to $m$ do
 6:     for $j = 1$ to $k$ do
 7:         $R :=$ Relevance$(\mathbf{B}^i, \theta_j)$  // Relevance vector for items in bag $i$ with respect to cluster $j$
 8:         $\widehat{B^i_j} := \mathbf{B}^i R$  // Exemplar for bag $\mathbf{B}^i$ in cluster $j$: weighted average of contents of $\mathbf{B}^i$
 9:     end for
10: end for
11: for $j = 1$ to $k$ do
12:     $\psi_j :=$ Regress$(\{\widehat{B^i_j}\}_{i=1\cdots m}, Y)$  // Regression model for cluster $j$
13: end for
14: $[\psi', \theta'] :=$ Select$(\{\psi_j, \theta_j\}_{j=1\cdots k}, \{\mathbf{B}^i\}_{i=1\cdots m}, Y)$  // Model selection: pick best local model to map all bags to all labels
```

3.1 Notation

We use lower-case italics to denote scalars (e.g., m, y^i), capital italics to denote vectors (B^i_j, Y), and bold capitals to denote matrices and sets ($\mathbf{D}$, $\mathbf{B}^i$). The ith column of the matrix $\mathbf{B}$ is written $B(i)$. Where possible, we use upper subscripts to indicate indices over bags ($\mathbf{B}^i$) and lower subscripts to indicate indices over clusters (θ_j, W_j). Finally, we use greek letters to denote parameter vectors (θ, ψ).

$\mathbf{B}^i$ denotes bag i from a data set $\mathbf{D}$, which is a collection of m bags. Each bag consists of $|\mathbf{B}^i|$ d-dimensional items, $B^i(j)_{j=1\cdots|B^i|} \in \mathbb{R}^d$. (Note that, although bags may contain different numbers of items, every item must be of the same dimension.) Thus, each bag can be thought of as a $d \times |\mathbf{B}^i|$ matrix. In the MIR framework, each $\mathbf{B}^i$ has an associated label, $y^i \in \mathbb{R}$.

3.2 The MI-ClusterRegress Algorithm

The core challenge of the MIR problem is that bags comprise variable numbers of items, while regression models map individual items to scalars. We solve this problem by reducing each bag to a fixed set of *exemplars* and then building traditional regression models over the exemplars. This approach is in contrast to methods that apply the regression model to individual items and then aggregate their predictions to produce a bag label [13, 4], and it is a generalization of methods that model bags with only one exemplar [6, 16].

The main assumption of the MI-ClusterRegress algorithm (Algorithm 1) is that the individual items are drawn (noisily) from a set of underlying *clusters* and that a bag's label is a function of one relevant cluster. For example, in our remote-sensing problem domain, clusters might correspond to "corn" and "wheat" crops as well as to cities, water, and other non-crop pixels. The yield of a single crop, such as corn, depends on all pixels in the corn cluster. Other labels for the same bag, such as wheat yield, may be determined by different clusters in the same set of observations.

Each bag is assumed to contain items drawn from one or more of these clusters. After clustering all items together (Lines 3 and 4), we have (soft) assignments of each item in each bag to each cluster. Using these assignments, we construct per-bag exemplars for each cluster (5–10). The exemplar for cluster j within bag i, $\widehat{B^i_j}$, is the average of all items in bag i weighted by their respective memberships in cluster j (i.e., "relevances", formalized below), denoted by R. Given these exemplars, we construct a regression model for each cluster j using the cluster j exemplars from all bags (11–13). Finally, a model selection step identifies the regression model (i.e., cluster) that best captures the relationship between data and bag labels (14).

MI-ClusterRegress uses two black-box machine learning subroutines: Cluster, an unsupervised clustering algorithm, and Regress, a (possibly nonlinear) regression algorithm. For any given application, the experimenter can select different clustering and regression subroutines.

Clustering. The only requirement on Cluster is that it produce k generative cluster models, $\theta_{j=1\cdots k}$, that can assign likelihoods to individual items. We used EM-based Gaussian mixture models, but other unsupervised learners are possible. For example, Latent Dirichlet Allocation [3] would remove the need to specify the number of clusters, while a spatio-temporal clustering model [15] might take more advantage of the geographic nature of our crop-yield prediction task. Or, if we wished to assume no structure in the data, we could use a "null clusterer" that places all of the items into a single cluster. In that case, MI-ClusterRegress would reduce to simply aggregating each bag to its (weighted) mean instance and building a regression model over all the means, akin to QPAP-Salience [16].

Algorithm 2 The MI-ClusterPredict($\mathbf{B}, \theta', \psi'$) algorithm

```
Inputs: New bag $\mathbf{B}$, cluster parameters $\theta'$, regression model parameters, $\psi'$
Output: Prediction for $\mathbf{B}$: $\widehat{y}$
$R :=$ Relevance($\mathbf{B}, \theta'$)   // Per-item relevance
$\widehat{B} := \mathbf{B}R$   // Bag exemplar
$\hat{y} :=$ RegressPredict($\widehat{B}, \psi'$)   // Regression prediction
```

Regression. The regression learner, Regress, takes a set of bag exemplars and returns a set of regression parameters, ψ. It is applied, in turn, to the m bag exemplars for each cluster, yielding k different local regression models, $\psi_{j=1\cdots k}$. This learner can be a linear or nonlinear regression model. In this work, we employ a support vector regression (SVR) learner[1] with a grid search over kernels (linear and RBF) and possible values for C, the regularization parameter. For RBF kernels, we used Jaakkola's heuristic [10] to set the scaling parameter $\gamma = \frac{1}{m} \sum_{i=1}^{m} \min_{j \neq i} dist(\widehat{B}^i, \widehat{B}^j)$, where $dist$ is Euclidean distance.

Model Selection. Once the clusters have been identified, and a local model has been constructed for each one, MI-ClusterRegress must select the most appropriate model for predicting the bag labels provided. Model selection methods generally seek to trade off model complexity and generalization error. We have evaluated two model selection heuristics:

- **MSV:** Minimize the number of Support Vectors used
- **MTE:** Minimize the Training Error

The **MSV** heuristic attempts to minimize the learned model's complexity, as an approximation to minimizing $\|w\|^2$ (which in turn bounds the VC dimension of the model). In contrast, the **MTE** heuristic seeks the best empirical fit to the training data, ignoring generalization. One of the interesting results of our experiments is that the latter heuristic performs very well, despite the seeming risk of overfitting.

Prediction. The MI-ClusterPredict algorithm (Algorithm 2) assigns a predicted value, $\widehat{y}$, to a previously unlabeled bag. It takes as input the new bag and the local model cluster and regression parameters, θ' and ψ', that were picked in the model-selection step of MI-ClusterRegress. It computes the relevance of each item in the new bag to the local model cluster, and then constructs the corresponding weighted average exemplar for this bag. It employs a RegressPredict routine, corresponding to the Regress learner from MI-ClusterRegress, that uses previously learned regression parameters to predict a label for the exemplar.

Algorithm 3 The Relevance($\mathbf{B}, \theta_c$) subroutine

```
Inputs: Single bag $\mathbf{B}$, parameters for one cluster $\theta_c$
Output: Relevance column vector $R$
$r(i) = p(cl_i = c | B(i); \theta_c), \forall i$
$z := \sum_i r(i)$
$R(i) := r(i)/z, \forall i$   // Relevance of item $i$ to cluster $\theta_c$
```

Relevance. Both MI-ClusterRegress and MI-ClusterPredict calculate the relevance of a given item to a particular cluster (Algorithm 3). The generative model for a cluster c with parameter θ_c provides $p(B(i)|cl_i = c; \theta_c)$ where $cl_i = c$ means that item i was generated by cluster c. Via Bayes's rule, we can calculate $p(cl_i = c|B(i); \theta_c)$ (this is the usual "responsibility" computed in the the E step of an EM algorithm). We renormalize these values across the bag so that the sum of all relevances within a bag is 1. Thus, the relevance of an item, with respect to a cluster, is sensitive to its context, which is the rest of the bag's contents. The renormalization ensures that the exemplar for the cluster is a weighted average of the contents of the bag (i.e., it is an affine combination of the bag data).

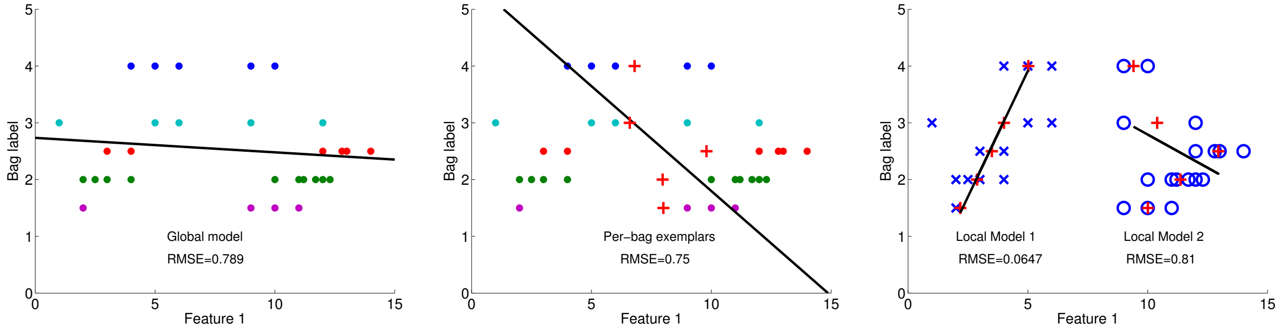
A key benefit of this approach is that, unlike previous MIR methods [14, 13, 4, 16], it is for the first time possible to identify relevant items in new bags, and therefore to achieve truly "multiple-instance" regression. Previous methods could make predictions for single items, but then were forced to either make fixed assumptions about which single item would dictate the label, or assume (or learn) which aggregator should combine all $|\mathbf{B}|$ predictions into a bag label. The critical difference is that MI-ClusterRegress is able to transfer what is learned on the training bags about relevance to the test bags, using the cluster models, and this provides the missing information needed to assign appropriate relevance values and generate the correct label.

3.3 Illustrative Example

Consider the five-bag data set shown in Figure 1(a). Each bag contains several one-dimensional items and a real-valued label. Each item has an unknown degree of relevance to the bag label. We plot these items with their single feature along the x axis, with all items from a given bag sharing their bag's label as the y value.

A non-multiple-instance approach would be to use all of the items as training data, assigning each item the label of its bag, and building a single regression model (Figure 1(a), marked "global model"). (For simplicity, we use linear models in this example.) This model provides a poor fit to the data, since it is required to model all of the items,

[1] We used the Matlab SVM Toolbox by S. Gunn [9], from http://www.isis.ecs.soton.ac.uk/resources/svminfo/.



(a) Global model: Map all items to their bag's label.

(b) Single exemplar: Replace each bag with its exemplar ('+').

(c) MI-CR: Create cluster 1 ('x') and cluster 2 ('o'), then create an exemplar ('+') for each cluster, in each bag.

Figure 1. A simple example showing how MI-ClusterRegress works. The data consists of five bags containing some number of one-dimensional items (a and b). After clustering, bag exemplars are shown as '+' (c). A single, global regression model (a) fits the data poorly. Reducing every bag to a single average exemplar and then fitting a single regression model to those exemplars (b) also yields a poor model. Finally, the combination of clustering, regression, and model selection permits the identification of the relevant subset of items in each bag (c), which are best fit by local model 1.

whether or not they are relevant to the bag labels. A second approach preserves the bags but does not model structure within the bags. It replaces each bag with a single exemplar that is the average of its contents, shown as '+' in Figure 1(b). A linear model is fit to the exemplars and their bag labels, providing a slightly better fit than the global model. This is the approach used by QPAP-Salience [16], which cannot be applied to new test bags unless the new item weights are known (or assumed).

MI-ClusterRegress instead identifies k local models. Given $k = 2$, MI-ClusterRegress first ignores the bag boundaries and clusters all of the one-dimensional data into two clusters. Figure 1(c) shows the cluster assignments by marking items with 'x' (cluster 1) and 'o' (cluster 2).[2] MI-ClusterRegress then computes two exemplars for each bag, one for each cluster, shown as '+' symbols in Figure 1(b).

Next, MI-ClusterRegress trains $k = 2$ regression models, one based on the bag exemplars for cluster 1, and the other based on the bag exemplars for cluster 2. In each case, the same bag labels are used as the regression targets. The regression model for cluster 1 obtains an RMSE of 0.0647, while cluster 2's model obtains an RMSE of 0.81. Selecting the model with minimum error (RMSE) items to cluster 1 as the best match, with a model that is far superior to either the global model or the single-exemplar approach that ignores internal bag structure. MI-ClusterRegress is able to generate predictions for new bags because the cluster models allow the assignment of relevance values even to unseen items.

4 Experimental Results

We have conducted experiments with synthetic data (known linear models) and with remote sensing data (to predict crop yield) to assess how well MI-ClusterRegress can both learn accurate regression models and identify the relevant items in each bag. All data sets used in these experiments are available at `http://harvist.jpl.nasa.gov/papers.shtml`.

4.1 Baseline Methods

We have identified two important baselines against which MI-ClusterRegress should be compared. The first posits that the multiple-instance regression problem can be solved as a regular supervised learning problem. The second suggests that a single exemplar per bag suffices to learn a good model; that is, the clustering step in MI-ClusterRegress is not needed. Together with MI-ClusterRegress, these represent a spectrum of assumptions about data structure: no bags → unstructured bags → structured bags.

Baseline 0: Global model (no bags). This baseline removes the bag structure by training a model on all of the items in the data set, giving each individual item the label of its bag. That is, it converts $\mathbf{D}$ into $\bigcup_i \mathbf{B}^i$ with a total of $n = \sum_i |\mathbf{B}^i|$ items and expands the vector of m bag labels Y to include $|\mathbf{B}^i|$ copies of y^i. Then, regular supervised learning methods can be used to learn a regression model mapping all input items to their labels. This approach assumes that that every item in a bag has the same relationship

[2] The cluster assignments here are hard-thresholded versions of the soft assignments produced by EM.

to the bag label, which can be modeled with a single global model. An example was shown in Figure 1(a).

Baseline 1: Single exemplars (unstructured bags). Another simple approach is to represent each bag with a single exemplar that is the (unweighted) average of the items in the bag. This baseline makes no attempt to model structure within a bag. It converts $\mathbf{D}$ into $\bigcup_i \left(\frac{1}{|\mathbf{B}^i|}\sum_j \mathbf{B}^i(j)\right)$, which contains m items, and uses Y unchanged. Again, regular methods can learn a regression model mapping the exemplars to the bag labels. This is the solution shown in Figure 1(b). It is akin to QPAP-Salience with the assumption that all items are equally relevant.

We cannot provide a quantitative comparison with primary instance regression, since there is no method for selecting the primary instance in new bags. In their experiments, Ray and Page assumed that the primary instance for each test bag was known *a priori* [14], and Cheung and Kwok [4] assumed that the primary instance was the item with the largest individual output, which is inappropriate for our data sets. Regardless, it seems likely that more than a single item is needed to predict labels for structured bags accurately, especially when predicting crop yield from remote sensing data.

4.2 Synthetic Data Experiments

To evaluate the MI-ClusterRegress approach in a controlled setting, we generated a series of one-dimensional synthetic data sets with known structure and models. We refer to the items generated by distribution c as "component" c. We created synthetic bags with 10 items per component, where the jth item in the cth component of the ith bag was drawn according to $B^i_c(j) \sim \mathcal{N}(\mu_c + f^i_c, \sigma_c)$. The cth mean, μ_c, was selected uniformly at random from the interval $[20c, 20c{+}10)$, and all models used a standard deviation of $\sigma_c = 3$. f^i_c was an offset applied to component c in bag i, randomly drawn $\sim \mathcal{N}(0, 4)$. This offset is a critical factor; if all items for component c were generated from μ_c alone, then the bags would be indistinguishable in terms of their contents and their labels, leaving no basis for generalization. The complete ith bag with k components is denoted $\mathbf{B}^{i,k}$.

We used a randomly selected linear model to generate the bag label based on the mean and offset for a randomly selected component r:

$$y^i = 6.34\,(\mu_r + f^i_r) - 2.24.$$

The bag labels y^i are correlated with the data from component 1, but not with the data in other components, since they have different offsets.

We created 20 bags per data set:

$$\mathbf{D}_k = \{\mathbf{B}^{i,k}\}_{i=1}^{20}, \qquad Y = [y^1 \ldots y^{20}]^\mathsf{T},$$

where $\mathbf{D}_k$ was "nested" inside $\mathbf{D}_{k+1}$ in that the bags in $\mathbf{D}_{k+1}$ use the data already generated from the first k components, plus an additional new component $k+1$, not associated with the bag label. Thus, as k increases, the relevant items within each bag are rarer and more difficult to distinguish. Since each bag consists of k components, each containing 10 items, data set $\mathbf{D}_k$ contains $200k$ total items.

Table 1. Test error (RMSE) on synthetic data as the number of components k increased, averaged over 10 trials. Results matching the oracle are in bold. Data set D_k had 20 bags, each with k components containing 10 items.

Data	MI-ClusterRegress			B0:	B1: No
set	MSV	MTE	Oracle	Global	structure
$\mathbf{D}_2$	**2.58**	**2.58**	2.58	221.61	55.04
$\mathbf{D}_3$	11.36	**7.67**	7.67	6271.24	76.81
$\mathbf{D}_4$	4.13	3.97	3.52	22.67	61.89
$\mathbf{D}_5$	3.87	**3.70**	3.70	3818.13	18.30
$\mathbf{D}_6$	**2.71**	2.92	2.71	2938.81	165.94
$\mathbf{D}_7$	7.16	5.06	5.05	3529.47	26.84
$\mathbf{D}_8$	5.31	5.31	4.44	230.76	160.08
$\mathbf{D}_9$	12.14	9.82	9.27	230.76	160.08
$\mathbf{D}_{10}$	**4.25**	**4.25**	4.25	140.75	39.70

The goal of these experiments was to assess 1) how accurately MI-ClusterRegress can construct and select a local model that predicts the bag labels, and 2) how sensitive MI-ClusterRegress is to an incorrect choice of k.

Methodology. For each experiment, we randomly split $\mathbf{D}_k$ into a training set (15 bags) and test set (5 bags). We further split the training set into base training (11 bags) and parameter tuning (4 bags) sets. We shifted and scaled Y to 0 mean and 1 standard deviation. We constructed each SVR model (used by MI-ClusterRegress or the baselines) with a linear kernel, error tolerance $\epsilon = 0.1$, and its own grid search over regularization parameter $C \in 10^{\{0,1,2,3,4,5\}}$ on the tuning set.

Results. We found that MI-ClusterRegress performed very well at this task, outperforming both baseline methods (Table 1). We also report the results obtained when selecting the model that minimizes *testing* error ("Oracle"), which provides a lower bound on MI-ClusterRegress's achievable error rate. This experiment did not aim to achieve zero test error (more than 15 training examples would be needed), but to compare performance against the baseline methods in a challenging (but artificial) setting.

The oracle's RMSE remained low for all values of k, while the baseline RMSE values were 1 to 3 orders of magnitude larger. The supervised approach of ignoring the bag boundaries (baseline 0) resulted in the worst performance;

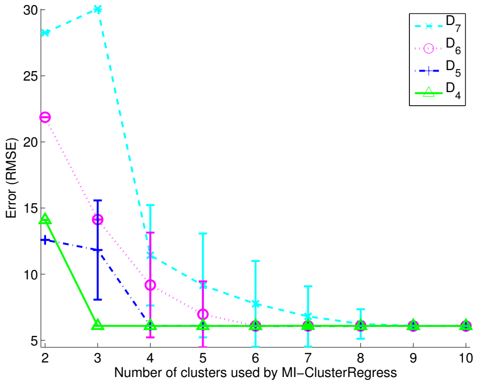


Figure 2. Test error (RMSE) for MI-ClusterRegress with the MTE heuristic when analyzing $\mathbf{D}_k$ and modeling the data with 2 to 10 clusters (using a single train/test split, averaged over 100 trials, with standard deviation bars shown). Using too large a value for k did not impact error, but too small a value for k did.

using a single exemplar to represent each bag (baseline 1) improved performance but fell short of MI-ClusterRegress.

The MTE model selection heuristic performed better than the MSV heuristic for half of the k values, while MSV outperformed MTE only once ($k = 6$). When there are multiple local models with the same number of support vectors, MSV simply chooses the first one, which can lead to poor generalization. A more precise min-complexity heuristic could improve this performance. In several, but not all, cases, the heuristics matched the oracle's performance. The low RMSE across all k values confirms that MI-ClusterRegress is robust even in the presence of a large number of irrelevant items (e.g., for $k = 10$, each bag had 200 items but only 10 relevant ones), if there is a common structure to leverage.

What local models were learned? A frequently selected MI-ClusterRegress model (by both heuristics and the oracle) was $y = 5.99x + 5.40$, which is close although not identical to the "true" generating model, $y = 6.34x - 2.24$. In comparison, example models learned by Baseline 0 and 1 were (e.g., for $k = 3$) $y = 0.09x + 151.54$ and $y = 4.83x - 71.52$; both are very far off from the true model.

Since we may not always know the exact number of components in the data, we analyzed MI-ClusterRegress's sensitivity to the choice of k, the number of clusters to model. Figure 2 shows the results when MI-ClusterRegress, using the MTE heuristic, was applied to $\mathbf{D}_k$ for $k \in [4, 7]$. For each data set, we tried setting $k = 2$ to 10, only one of which fit the true number of components in the data set. To control the results, we used a single train/test split, but ran MI-ClusterRegress 100 times, each with a different EM initialization; any variation is due to changes in the clustering results, not the data set.

The results are striking. If the value chosen for k was larger than the true number of components, the error rate was not affected; while the clusters may have been overly split, enough "signal" remained that MI-ClusterRegress could choose a cluster and construct a good local regression model for it. However, if the value chosen for k was much smaller than the true number of components, the error increased dramatically. The clusters merged the relevant items with irrelevant ones, and as a result the local models could not accurately fit the data.

We conclude from these experiments that 1) modeling internal bag structure, when present, is critical for multiple-instance regression, and 2) if the number of components is not known, we recommend selecting a conservatively large value for k.

4.3 Predicting Crop Yield

A challenging MIR problem is the task of predicting crop yield early in the growing season, based on remote sensing observations. Each year, the USDA reports the average yield per acre, for each county in the U.S., for a variety of crops. Predicting the expected yield, especially when done early in the growing season, can inform agricultural market decisions as well as crop management strategies (irrigation, pest control, fertilizer, etc.).

Data Description. The MODIS instruments onboard the Terra and Aqua satellites provide repeat coverage of the entire United States every 1-2 days. We downloaded the 8-day aggregate product which provides the best (cloud-free) observation of each pixel every 8 days. For each pixel, we obtained a time series of observations, enabling us to generate a new yield prediction every 8 days. Each pixel consisted of observations at 250-m resolution, in the red (620–670 nm) and near infrared (841–876 nm). We combined these two values into a single index called NDVI (Normalized Difference Vegetation Index) as follows:

$$NDVI = \frac{NIR - RED}{NIR + RED}.$$

NDVI is known to provide a good indication of vegetation abundance and health. Further, NDVI is particularly good for identifying pixels that contain crops: they show a characteristic NDVI peak leading up to the time before harvest. Non-agricultural regions, in contrast, have flat NDVI profiles over the course of the year.

The crop yield data is also public data and comes from the USDA National Agricultural Statistics Service (NASS). We have created data sets that contain the remote sensing

Table 2. Counties (bags) reporting yield for corn and wheat in California (USDA-NASS).

Training				Test
2001	2002	2003	2004	2005
17	16	18	15	13

and crop yield data for California over five years (2001–2005). Table 2 shows the total number of counties that reported yield values for two crops we studied, corn and wheat.

This problem is challenging because the ratio of relevant observations (pixels) to the desired predictions (county-level crop yield) is so low. Each county contains thousands to hundreds of thousands of pixels, and we do not know which pixels (or even how many) contain the crop of interest. This problem lends itself well to the multiple-instance setting and provides both an important real-world application for this work and a particularly challenging "stress test" for any regression method. For the purposes of this study, we randomly sub-sampled the remote sensing observations for each county to obtain 100 pixels for each bag; ultimately, we aim to use more efficient EM and SVR implementations and analyze all pixels in every county.

Methodology. We trained models on observations from four years (2001–2004) with a 25% subset for parameter tuning, and tested the models by predicting the yield for 2005. To make a prediction at day T (ranging from 8 to 360, in increments of 8 days), we trained with pixels consisting of the sequence of NDVI observations taken every 8 days since the beginning of its year, up to day T. For example, a prediction at day 80 in 2005 is made by a model created from observing data from day 1 to 80 in 2001 through 2004 and matching them to the yield observed in those years. Again, we shifted and scaled Y and used the grid search previously described, except that the search included linear and RBF kernels, with the RBF γ parameter set using Jaakkola's heuristic [10]. Since we do not know *a priori* how many components exist in this data, we tried several different values for k, the number of clusters. We found that the best results were obtained for k larger than 20; given the results on the synthetic data, we selected a conservatively large value of $k = 30$.

We compared the performance of MI-ClusterRegress on this data set to Baseline 1, which does not model internal bag structure. We did not apply Baseline 0 to this data because we identified a more accurate baseline, and because Baseline 0 takes orders of magnitude longer to run than any other algorithm we tested (it must train multiple regression models, each time using the full data set, rather than one exemplar per bag). Instead, we used a domain-specific baseline for this problem. As in many time series prediction domains, a default prediction that the next target value will be the same as the last one observed can be surprisingly accurate. **Baseline 2: Last Year's Yield** predicts that the yield this year will be identical to last year's yield.

The magnitude of the target (yield) varies depending on the crop. For example, in 2005 corn yield ranged between 146.3 and 211.8 bushels per acre, but wheat yield ranged between 49.3 and 98.5 bushels per acre. Therefore, to permit comparisons across different crops, we evaluated the crop yield predictions in terms of *relative error*:

$$E_{rel}(y, \hat{y}) = \left| \frac{y - \hat{y}}{y} \right|,$$

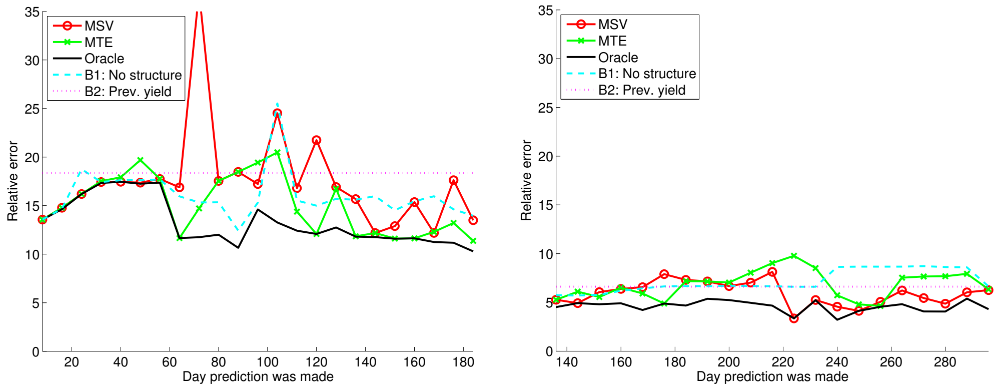
where y is the true value and $\hat{y}$ is the predicted value.

Results. We found that MI-ClusterRegress yielded results superior to both baselines. Figure 3 shows the relative error in predictions made every 8 days during 2005 for both wheat and corn. Each plot covers only the growing season for the crop selected. The typical planting dates in California are May 15 (day 135) for corn and Dec. 15 (day -16) for wheat; harvest dates are Oct. 20 (day 293) and July 1 (day 181), respectively.

In general, prediction error decreased as more observations were made (later in the year). This trend was stronger for wheat than for corn predictions. The corn yield errors were low throughout the year, and always lower than the wheat yield errors, suggesting that the corn signal was more evident in the remote sensing data. The "oracle" results show the error obtained if the model selection step precisely identified the best-fit cluster to model the crop chosen. This error dropped as low as 10.7% at day 88 for wheat (3 months before harvest) and 3.4% at day 224 for corn (2.5 months before harvest).

In terms of model selection, we found that MTE significantly outperformed MSV (and baseline 2) in predicting wheat yield (paired t-test, 95% conf.), while MSV was better than MTE (and both baselines) for predicting corn yield (paired t-test, 95% conf.). We suspect that a domain-specific heuristic that prefers clusters with exemplars that show the characteristic time profile of an agricultural crop [7], even if the time of harvest is not known, would do even better.

The best possible MI-ClusterRegress results consistently beat both baselines (by 1-10% relative error, statistically significant with 95% conf.), confirming our hypothesis that modeling internal bag structure is critical for this application. Baseline 1, which does not model internal bag structure, was competitive with the best model selection heuristic results for wheat but significantly worse than the model selection results for corn, as noted above. Baseline 2 (using last year's yield) does not depend on the observations and so had a constant error rate throughout the year of 18.3% for



(a) Relative error in wheat yield predictions

(b) Relative error in corn yield predictions

Figure 3. Relative error in predicting wheat and corn yield for California, as a function of day of year. Each plot shows prediction error over the growing season for the crop in question. The models were trained on data from 2001–2004 (k = 30) and tested on data from 2005.

wheat and 6.6% for corn. The model selection heuristics were significantly better for both crops.

We also examined the cluster exemplars selected by MI-ClusterRegress (data not shown). These exemplars represent NDVI time series for single categories (e.g., one for wheat pixels, one for corn pixels, etc.) We found that the exemplars for the corn and wheat clusters generally reflected the planting, maximum foliation, and harvest dates for each crop, despite no access to this domain knowledge.

Figure 4 demonstrates just one benefit of estimating individual item relevances. Figure 4(a) is a Google satellite map[3] of a randomly selected county with overlaid road and city features. Figure 4(b) shows the estimated wheat relevance values for the entire county (all 69,578 pixels) at day 72 (March 12) in 2005, based on the MTE local model. The lower-left corner of the relevance map is white (low relevance) and corresponds directly to the I-5 corridor and the non-irrigated, semi-arid foothills terrain. In the middle of the county is a large patch of agricultural land, identified clearly as high-relevance in Figure 4(b). Although MI-ClusterRegress was given no prior knowledge of crops, terrain, or spatial or temporal locality, it successfully identified semantically meaningful crop structure. To our knowledge, this is the first system to automatically identify individual crop locations from satellite images without any supervisory crop-type labels.

5 Conclusions and Future Work

In this paper, we presented a new approach to multiple-instance regression that explicitly models internal bag structure. After constructing a local regression model for each component in the data set, MI-ClusterRegress selects the best-fit model and uses it to predict labels for new bags. We found consistent evidence, based on synthetic data and remote sensing data, that the presence of internal structure requires an approach such as MI-ClusterRegress. The crop yield predictions significantly out-performed other learning approaches. The primary area for future work is the development of better model selection heuristics, particularly ones that incorporate domain knowledge. We also consider it important to evaluate other models for internal bag structure, particularly those that relax the Gaussian distribution assumption.

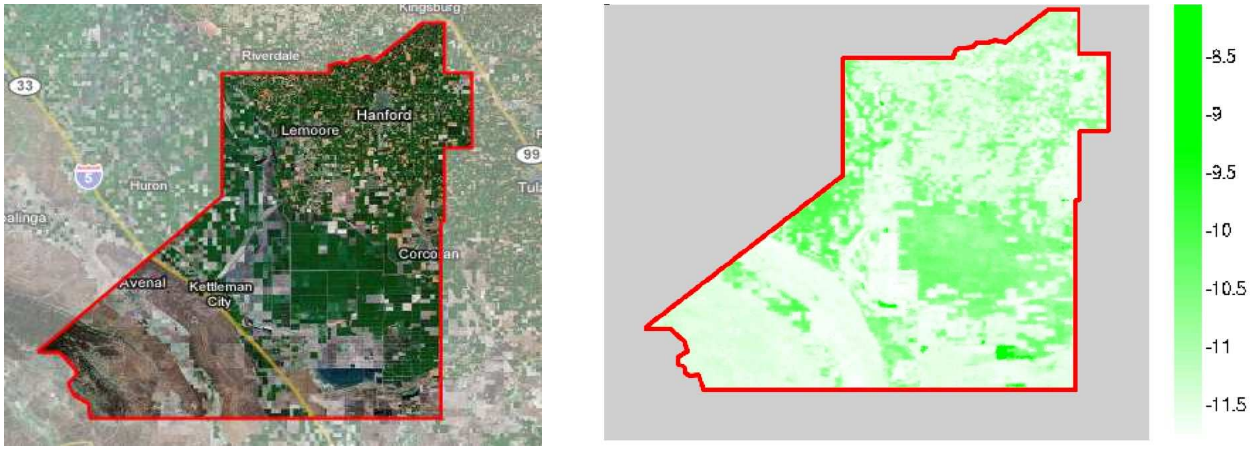
6 Acknowledgments

We gratefully acknowledge the support of a grant from NASA's AIST Program, NSF grants #ITR-0325329 and #IIS-0705681, and NIMH grant #1R01MH076282. Thanks to John Burge for the original idea to consider clustering the entire data set together, to Soumya Ray for his insightful comments about structured bags, and to Amy McGovern for feedback on draft versions of this work. We also thank the MODIS team and the NASS for the data we used. This work was partly carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] R. A. Amar, D. R. Dooly, S. A. Goldman, and Q. Zhang. Multiple-instance learning of real-valued data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 3–10, 2001.

[3] http://publicrecords.onlinesearches.com/maps/map-of-Kings-County-California.htm



(a) Google image of Kings County, CA

(b) MTE wheat relevance plot near peak of wheat growth (day 72)

Figure 4. Example per-pixel relevance values obtained from multiple-instance regression (MTE heuristic) on remote sensing data for a randomly chosen county (Kings County, CA). Darker green shades indicate higher relevance for predicting wheat (log relevance scale).

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, pages 561–568. MIT Press, 2003.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan. 2003.

[4] P.-M. Cheung and J. T. Kowk. A regularization framework for multiple-instance learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 193–200, 2006.

[5] T. G. Dietterich, R. H. Lathrop, and Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71, 1997.

[6] G. Fung, M. Dundar, B. Krishnapuram, and R. B. Rao. Multiple instance learning for computer aided diagnosis. In *Advances in Neural Information Processing Systems 19*, pages 425–432, 2006.

[7] G. L. Galford, J. F. Mustard, J. Melillo, A. Gendrin, C. C. Cerri, and C. E. P. Cerri. Wavelet analysis of MODIS time series to detect expansion and intensification of row-crop agriculture in Brazil. *Remote Sensing of the Environment*, 112(2):576–587, 2008.

[8] S. A. Goldman and S. D. Scott. Multiple-instance learning of real-valued geometric patterns. *Annals of Mathematics and Artificial Intelligence*, 39(3):259–290, 2003.

[9] S. R. Gunn. Support vector machines for classification and regression. Technical Report ISIS-1-98, Image Speech and Intelligent Systems Research Group, University of Southampton, 1998.

[10] T. Jaakkola, M. Diekhans, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 149–158, 1999.

[11] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems 10*, pages 570–576. MIT Press, 1998.

[12] R. Rahmani and S. A. Goldman. MISSL: Multiple-instance semi-supervised learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 705–712, Pittsburgh, PA, June 2006.

[13] S. Ray. *Learning from Data with Complex Interactions and Ambiguous Labels*. PhD thesis, University of Wisconsin, Madison, 2005.

[14] S. Ray and D. Page. Multiple instance regression. In *Proceedings of the 18th International Conference on Machine Learning*, pages 425–432, 2001.

[15] J. F. Roddick and K. Hornsby, editors. *Temporal, Spatial, and Spatio-Temporal Data Mining*. Springer, 2001.

[16] K. L. Wagstaff and T. Lane. Salience assignment for multiple-instance regression. In *Proceedings of the ICML 2007 Workshop on Constrained Optimization and Structured Output Spaces*, 2007.

[17] J. Wang and J. D. Zucker. Solving the multiple-instsance learning problem: A lazy learning approach. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1119–1125, 2000.