Predicting Cassette Exons Using Transductive Learning Approaches

Ana Stanescu and Doina Caragea
Department of Computing and Information Sciences
Kansas State University, Manhattan KS, USA
{anas, dcaragea}@k-state.edu

Abstract—**Recent advances in biotechnology have resulted in large volumes of genomic and proteomic data leading to the emergence of numerous *in silico* methods for annotation, such as supervised machine learning approaches. Such algorithms, however, require large amounts of labeled data for training. In practice, labeled data is oftentimes limited because it is difficult to obtain. Therefore, semi-supervised machine learning is preferable, in which classifiers trained on limited amounts of labeled data can be improved by exploiting the large amounts of unlabeled data. In this work, we focus on transductive learning, a special case of semi-supervised learning. A semi-supervised algorithm builds an inductive model that generalizes well to new, unseen (test) instances. In contrast, during the training phase, a transductive algorithm has access to the (test) instances that need to be classified, allowing advantageous utilization of these points in order to reach the best separation function. Compared to learning a classifier for use with future data, cassette exon identification is a suitable application for transductive learning, since the goal is to annotate a sequenced genome for which a limited amount of labeled data is available. We study the applicability of three popular transductive techniques and their compatibility with various kernels to the binary DNA classification problem of cassette exon identification. The results of our experiments suggest that transductive learning is a useful approach for assisting genome annotation.**

I. Introduction

Supervised machine learning produces dependable classifiers when large amounts of labeled data are available for training. Because of expensive generation, however, labeled data is usually scarce. Unlabeled data is easier to obtain as a result of advancement in high throughput Next Generation Sequencing (NGS) technologies. This scenario, in which limited amounts of labeled data along with considerably larger amounts of unlabeled data are available, suggests the use of semi-supervised learning (SSL), which is a learning paradigm at the intersection of supervised and unsupervised learning. SSL requires a small amount of labeled data and larger amounts of unlabeled data in order to build classification tools that perform better than models trained only on labeled data. Improving supervised classifiers by leveraging unlabeled data is a very appealing concept, although it does not always work as intended: in practice, the unlabeled data can degrade a classifier [1], [2]. Understanding whether or not unlabeled data will enhance a supervised learning classifier for a particular problem is still the focus of ongoing research [3], [4].

978-1-4799-6926-5/15/$31.00 ©2015 IEEE

In a classic semi-supervised environment, a learner has access to labeled and unlabeled examples during the training phase, and the classifier must produce a classifier that can be used to predict the class of future data points not previously encountered. A subtype of SSL, called *transductive* learning, aims to classify unlabeled data without generalizing to other new, unseen examples. The goal of transduction is not to produce an *inductive* model (as in supervised and SSL), but to predict the labels of the unlabeled data to which the algorithm has access during the training phase. This may be an advantage for the algorithm, and transduction is sometimes viewed as an "easier" case of semi-supervised learning.

Theoretically, transduction is particularly suitable for genome annotation, in which a newly sequenced genome, ready to be annotated, is typically available up front, along with limited annotation. Vapnik introduced a popular large-margin transductive approach, known as Transductive Support Vector Machines (TSVM) [5]. TSVM has primarily been used for protein-related problems in bioinformatics [6]–[8], with a notable exception for promoter recognition [9].

One of the most popular graph-based transductive algorithms is Label Propagation (LP), proposed by [10], in which available labels are propagated across a graph, thereby resembling the Markov random-walk algorithm. LP was originally tested on the problem of recognizing handwritten digits, but it has also produced successful results on problems related to natural language processing (*e.g.*, word sense disambiguation). LP is one of the first methods to gain rapid popularity, and it remains in use as a baseline for derivations of graph-based algorithmic approaches.

A more recent transductive algorithm is the Adsorption algorithm, a graph-based approach first introduced by Baluja *et al.* [11] in the context of YouTube video recommendation. As a variation of "Adsorption", Talukdar and Crammer [12] proposed the "Modified Adsorption" algorithm (MAD) and used it for sentiment classification on Twitter data. Several other problems have been addressed using MAD [13], [14], but only a limited amount of work has been conducted on biology-related classification problems, with the exception of [15], who applied MAD to a gene prioritization problem. We believe that MAD's suitability for bioinformatics comes form the fact that it is scalable to accommodate the large amounts of data available in biology-related fields, and can also handle multiclass problems. The goal of this study is to increase understanding of the strengths and limitations of the three popular transductive learning algorithms (TSVM, LP, and MAD) for DNA sequence classification, with concrete applications to the problem of predicting a type of alternative splicing, specifically cassette exons.

Alternative splicing, a naturally-occurring phenomenon first observed in the late 1970s, increases proteome complexity in eukaryotes. Alternative splicing occurs after transcription. There several types of alternative splicing events, but in this work we focus on alternatively spliced exons, also called “cassette” or “skipped” exons. As illustrated in Figure 1, when transcribing DNA into mRNA, some exons, called “constitutive” exons, are always transcribed, while the “cassette” exons can be skipped in some isoforms.

The identification of alternative splicing events, in particular, “cassette” exons, is an essential step in the task of genome annotation and can be addressed by conducting wet-lab experiments. However, such experiments are time-consuming and require expert involvement, and unfortunately computational methods based on Expressed Sequence Tags (EST) and full length cDNA are still expensive because constructing them is difficult. Recently, RNA-Seq to genome alignments have emerged [16], [17], but are not accurate enough (*e.g.*, Cufflinks only detects 44% of true alternative splicing events, as shown in a recent study [18]).

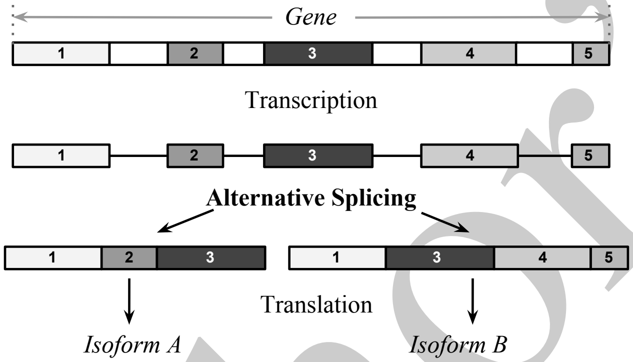

Fig. 1: Cassette versus Constitutive Exons: Exons 1 and 3 are “constitutive” since they appear in all isoforms, while exons 2, 4, and 5 are “cassette” exons, because they are excluded from some isoforms.

Supervised machine learning approaches have also been implemented for the problem of predicting alternative splicing events, including the prediction of cassette exons. In [19], the task is formulated as a binary classification problem, where the two classes are given by “cassette” (alternatively spliced) exons and “constitutive” exons (*i.e.*, exons that are always transcribed). In [20], the focus is on predicting alternative splicing events in humans. The authors used conserved information between human and mouse, upstream and downstream intronic sequence motifs, and length-based features in the learning process. Specialized biological kernels that model similarities between sequences have been used with SVM to predict alternative splicing [20], [21].

To the best of our knowledge, no study has compared transductive algorithms on a DNA sequence classification problem; therefore our research focuses precisely on this comparison. The contributions of this paper are threefold: (1) We study and compare three transductive algorithms based on two paradigms (large-margin and graph-based) in order to evaluate the algorithms' suitability for DNA sequence classification. More specifically, we use TSVM, LP, and MAD to predict cassette exons in *Caenorhabditis elegans*; (2) We experiment with various data representations and kernels to determine which of them exhibits stronger compatibility with transductive methods. We utilize an additive kernel comprised of the spectrum kernel or weighted degree kernel with shifts on the actual sequence, along with a linear kernel on sequence length features; (3) We study the effects of the amount of labeled data on the performance of the transductive algorithms considered.

The rest of the paper is organized as follows. In Section II, we review relevant and related works and present the context of our study, and explain the need for this research. We present the algorithms in Section III, and data and similarity measures used are described in Section IV. We enumerate research questions that we want to address and outline the experimental setting in Section V. The results are presented and discussed in Section VI. Finally, we present our conclusions in Section VII, where we also enumerate several directions we are interested in pursuing as future work.

II. Related Work

Transductive learning has been applied to a wide range of domains, including text classification, sentiment analysis, movie and video recommendation, natural language processing, image and phonetic processing, and prediction or diagnosis of various events in medical fields. In bioinformatics, transductive approaches have been successfully used primarily for protein-related problems.

Shin *et al.* [6] proposed a method for combining multiple graphs obtained from several independent and complementary sources of information. The resulting combined graph was used with spectral clustering to determine functional classes of yeast proteins, a multiclass prediction problem. Weston *et al.* [22] classified protein domains into SCP super families (SCP stands for Structural Classification of Proteins). The authors employed cluster kernels (bagged mismatch and neighborhood mismatch kernels) to utilize unlabeled data and labeled data. Kondratovich *et al.* [7] utilized TSVM for the problem of molecule activity prediction.

Comparative studies of transductive algorithms have been conducted for sentiment classification, including a recent study by Yong *et al.* [23] at the document level, for underresourced languages. The authors compared MAD and LP and ran experiments on datasets from three domains (hotels, notebooks, and books). The datasets consist of approximately 4,000 reviews, out of which a balanced subset of 300 comprised labeled instances, manually annotated in terms of sentiment polarity (150 positive and 150 negative reviews). Yong *et al.* [23] also decreased the amount of labeled data (from 300 instances to 20 instances) in order to assess the algorithms' behavior with various amounts of labeled data. Results showed that MAD outperformed LP. We conduct a similar study, but we compare TSVM, LP, and MAD on a biological (DNA) classification problem.

For DNA classification, purely SSL approaches, such as Expectation Maximization, Self-training, and Co-training, have been studied for the problem of predicting alternatively spliced exons [24] and acceptor splice sites [25], [26]. However, the collection of studies on purely transductive approaches is not as rich; here we mention a notable exception from Kasabov *et al.* [9], who used TSVM on the problem of promoter recognition in a multispecies dataset.

Because transductive learning algorithms rely on similarities, biological kernels are also relevant to our work. Specialized biological kernels have been proven to enhance classification capabilities of supervised large-margin classifiers, for protein related problems. For example, Kuang *et al.* [8] used SVM with profile-based string kernels from PSI-BLAST profiling for the problems of protein classification and detecting remote homology of proteins, in a supervised classification setting. Rangwala and Karypis [27] designed two classes of kernels, window-based and alignment-based, for SVM to be used for the problem of detecting remote homologs and identifying folds, respectively.

For supervised DNA sequence classification, Rätsch *et al.* [19] created a biological string kernel, called the weighted degree kernel with shifts, and used this kernel with SVM. We also employ this kernel in our study but in a transductive framework.

III. Transductive Approaches Studied

In this section we describe the types of methods compared in our study, with a focus on transductive learning and determining which algorithm produces the best results. Many popular transductive algorithms have different assumptions, but in this study we will focus on one margin-based algorithm in this study, namely TSVM (Section III-A) and two graph-based algorithms, LP (Section III-B) and MAD (Section III-C). Other transductive approaches such as Learning with Local and Global Consistency [28] and Label Matrix Normalization [29], did not produce satisfactory results on our data, and were therefore excluded from this paper.

A. Transductive Support Vector Machines (TSVM)

The TSVM algorithm [5] is an extension to the classical SVM algorithm. The "low density separation" assumption states that points residing in the same cluster share the same label and that the decision boundary should reside in a low density region, known as a large margin. This separating hyperplane maximizes the margin while minimizing the training error, as a penalty term for misclassification must be introduced for the non-linearly separable cases. Because TSVM optimization is an intractable problem, Joachims [30] proposed a solution resembling the classical "self-training" approach because it uses the completely supervised SVM built on the labeled data, and then "switches" labels of the unlabeled (test) data in order to optimize the objective function while consistently classifying the originally labeled examples. In other words, the new boundary must be consistent with the labeled data. In this paper, we use SVMLight [30] implementation of TSVM that was designed to accommodate problems with datasets of no more than a few thousand examples.

Similar to SVM, TSVM can benefit from the "kernel trick", in which the traditional dot product that appears in the original SVM optimization problem is replaced by a nonlinear kernel function, which provides an alternative to measuring the similarity between two instances. Instead of utilizing the dot product of the instances' vector representation, the kernel models different notions of similarity that are more appropriate for the problem studied. This kernelized version that transforms the representation of instances to a higher dimensional space allows customized solutions to calculate similarities between instances. We experiment with various sequence representations and similarity kernels, as explained in Section IV. The same representations and similarity kernels are used to build similarity (affinity) matrices for the graph-based approach.

B. Label Propagation (LP)

In graph-based methods, all available data, including labeled instances $\{(x_1, y_1), ..., (x_l, y_l)\}$ and unlabeled (or test) instances $\{(x_{l+1}, y_{l+1}), ..., (x_u, y_u)\}$ where usually $l \ll u$, are represented as nodes in an undirected graph. Formally, the graph is defined as $G = \{V, E, W\}$, where V represents the set of nodes (vertices), $E = V \times V$ is the set of edges that represents every pair of nodes, and W is the set of weights associated with the edges. Weights on the edges reflect the similarities between the connected nodes. The "smoothness" assumption of graph-based methods states that because nodes connected by a strong edge are very similar, the nodes are more likely to share the same label. LP [10] is a transductive algorithm that spreads labels of the originally labeled nodes throughout the graph in order to classify unlabeled nodes, which receive a class distribution in the form of "soft" labels (probabilities). The elements of the vector Y_v maintain the node's v prior class distribution, and are different from zero if the node is labeled, and null if the node is unlabeled. The second vector $\widehat{Y_v}$ is initialized to zero and its dimensions get assigned values for each class, as inferred by the algorithm. The smoothness assumption can be mathematically formulated as the optimization problem from Equation (1), where labels $\widehat{y}_i$ and $\widehat{y}_j$ of nodes v_i and v_j, respectively, should be similar for a large W_{ij} in order to minimize the function, while ensuring that the original labels are maintained.

$$\min \sum_{i,j} W_{ij}(\widehat{y}_i - \widehat{y}_j)^2, \ s.t. \ \widehat{Y}_l = Y_l. \tag{1}$$

The function from Equation (1) can be solved iteratively, using Algorithm 1, which utilizes the nodes' label distribution given in the form of a matrix $Y = (l + u) \times C$, where l represents the number of labeled examples, u is the number of unlabeled examples, and C is the number of classes. Next, a probabilistic transition matrix T is computed such that the probability of jumping from node i to node j is

$$T_{ij} = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}} \tag{2}$$

After the initialization of $\widehat{Y_v}$ with class labels for $\widehat{Y_l}$ and arbitrary values for $\widehat{Y_u}$, actual propagation occurs (line 4 in Algorithm 1). The algorithm continues with re-setting of the initial labels (line 5 in Algorithm 1) in order to reinforce the labels of the originally labeled training data. This operation is referred to as "clamping" of the labels. The iterations are then repeated until convergence (*i.e.*, until the propagation is complete and the labels do not vary much between iterations).

Algorithm 1 Label Propagation (LP)

```
Require: Similarity Graph $G = \{V, E, W\}$, Label Matrix $Y_v$
1: Compute $T = D^{-1}W$, where $D$ is diagonal degree matrix
2: Initialize $\widehat{Y_v} = Y_v$
3: repeat
4:     $\widehat{Y_v} = T\widehat{Y_v}$
5:     $\widehat{Y_v} = Y_v, (v \in V_l)$ "Clamp" the original labels
6: until $\widehat{Y_v}$ converges
7: return $\widehat{Y_v}$, the estimated probability distribution over the labels of vertex $v$
```

C. Modified Adsorption (MAD)

The original Adsorption [11] algorithm resembles the concepts of LP [10] and also [31]. MAD [12] can be considered a "random walk"-type approach that propagates labels throughout the graph in a more controlled manner, by the means of three probabilities: (1) injection probability, p_v^{inj}, which returns the initial Y_v label distribution of a node; (2) continuation probability, p_v^{cont}, that continues to propagate the label from v onto the next node v' with probability proportional with the similarity between the two nodes, given by:

$$Pr[v'|v] = \frac{W_{v'v}}{\sum_{u:(u,v)\in E} W_{v'v}} \tag{3}$$

and (3) termination (or abandonment) probability, p_v^{term} that terminates the propagation process for a node. The condition is that p_v^{inj}+p_v^{cont}+p_v^{term} = 1.

LP and MAD differ from each other in (1) that MAD does not reinforce the initial class distribution carried by the training labeled data, thereby presumably dealing with potential noise in the original label data and (2) that MAD can express uncertainty regarding classification through the means of a dummy label that acts as an extra "class" initialized to zero in the beginning and later assigned the default abandonment probability when/if the label propagation is abandoned at a given training phase (iteration).

The actual class distribution of every node v is stored in Y_v, which is a $(C+1)$-dimensional row vector enhanced to hold the extra dummy variable ν. C is the number of classes. Similar to the notation from LP, the predicted (inferred) class distribution of every node is stored in $\widehat{Y_v}$. MAD also utilizes a $(C+1)$-dimensional row vector $\mathbf{r}$ whose elements are set to zero, except for the extra element holding the dummy label, which is set to 1 ($\mathbf{r}_l = 0$ for $l \neq \nu$, $\mathbf{r}_\nu = 1$).

MAD, an extensions to the original Adsorption algorithm, has a well-defined optimization function (Equation 4) that can be solved iteratively in matrix form using the Jacobi method (Algorithm 2). The first term of the cost function captures the constraint that the inferred labels should not significantly differ from the original labels. The second term ensures the "smoothness" assumption and the third term is a regularizer that discourages uncertainty. The importance of each term is controlled by three hyperparameters, μ_1, μ_2, and μ_3.

$$\begin{aligned} \min \sum_v [\mu_1 \sum_k p_v^{inj} (Y_{vk} - \widehat{Y_{vk}})^2 + \\ \mu_2 \sum_v \sum_j p_v^{cont} w_{vj} (\widehat{Y_{vk}} - \widehat{Y_{jk}})^2 + \\ \mu_3 \sum_k p_v^{term} (\widehat{Y_{vk}} - R_{vk})^2] \end{aligned} \tag{4}$$

Algorithm 2 Modified Adsorption (MAD)

```
Require: Similarity Graph $G = \{V, E, W\}$, Label Matrix $Y_v$, Probabilities $p_v^{inj}$, $p_v^{cont}$, $p_v^{term}$, $\forall v \in V$
Initialize $\widehat{Y_v} = Y_v$
repeat
    $D_v = \frac{\sum_u W_{uv} \widehat{Y_v}}{\sum u W_{uv}}$
    for $v \in V$ do
        $\widehat{Y_v} = p_v^{inj} \times Y_v + p_v^{cont} \times D_v + p_v^{term} \times \mathbf{r}$
    end for
until $\widehat{Y_v}$ converges
return $\widehat{Y_v}$, the estimated probability distribution over the labels of vertex $v$
```

In this work, we use the Junto implementation of LP and MAD, from *https://github.com/parthatalukdar/junto* and we maintain the default parameters. All three transductive approaches explored in this work require a similarity measurement, in the form of a kernel function, for each pair of instances, such as in the case of TSVM, or they require a similarity measurement in the form of a similarity matrix, as in the case of the graph-based MAD and LP algorithms.

IV. Data Representation and Similarity Measures

In our experiments, we use genomic data from the model organism *Caenorhabditis elegans* in our experiments. The dataset was published by Rätsch *et al.* [19] and it is publicly available at *http://people.kyb.tuebingen.mpg.de/raetsch/RASE.old/*. The dataset contains 3,018 nucleotide sequences of exons and adjacent introns, *i.e.*, each instance is in the form *left intron–exon–right intron*, as illustrated in Figure 2. Out of these 3,018 instances, 487 are labeled as alternatively spliced, meaning that the flanked exon is a cassette exon that can be skipped in some isoforms. The remaining 2,531 sequences are labeled as constitutive, meaning that the exon is present in all known isoforms. The data was labeled based on alignments between ESTs and genomic DNA.

Given the intron-exon-intron sequence, two types of features are readily available: (1) content-based features obtained directly from the DNA sequence, and (2) length-based numeric features obtained from the lengths of the exons and their flanking introns. Accordingly, two types of similarity scores can be captured by string kernels and numeric kernels, respectively. Because kernels are additive, these two scores can be added, to more accurately reflect the overall similarity between two instances. In our study, we experiment with three different ways for capturing content-based similarity at the sequence-level using string kernels, as described below. For lengths, we always use a linear kernel that computes the dot (inner) product between numeric features. Along with the dataset, Rätsch *et al.* [19] also made available length features associated with the instances.

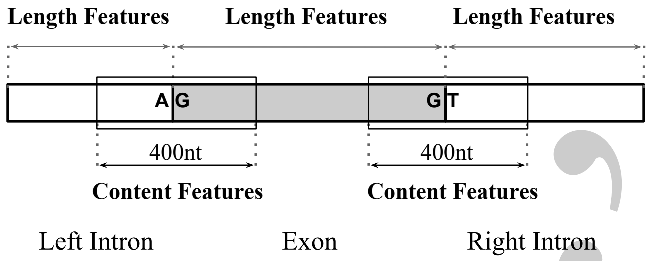

Fig. 2: Example of an instance from the dataset in the form of intron-exon-intron. Content-based features are generated from 400-nt windows around the splice sites, while length features are obtained from the lengths of the exons and flanking introns.

Length features are obtained directly from [19] in which lengths of each upstream intron, exon and downstream intron (of every sequence in the set) were used to generate 30-dimensional logarithmically spaced vectors for a total of 90 features per instance, corresponding to the three lengths. The set of length features also includes 3-dimensional vectors that characterize the frame of the stop codon, resulting in 15 additional features for a total of 105 length features (LG) per instance. Labels of the instances were not used during the feature generation process. The following sections describe how we used the string kernels to capture DNA-level similarities.

A. Weighted Degree Kernel with Shifts (WDS)

The similarity between two DNA strings using the Weighted Degree kernel with Shifts **(WDS)** [19] is given by the count of co-occurrences of exact k-mers at correspondent (exact or shifted) positions in the sequences, where $k \in \{1..degree\}$, and whose weights are controlled by β coefficients, with β dependent on the size of k. In order to utilize WDS, the DNA sequences must have equal lengths. Because most splicing regulatory information is typically aggregated in the proximity of splice sites, the WDS is applied on 400nt windows centered around the acceptor and donor splice sites in regions upstream and downstream of the exon. The more sequence overlap that exists close to the splice site, the higher the score captured by the WDS. Leveraging the additive property of kernels, the two score values that correspond to donor and acceptor sites are then added; the combined kernel reflects the overall sequence similarity. For more details regarding WDS, the reader is referred to [19].

B. K-Spectrum Kernel (k-SK)

The k-Spectrum Kernel (k-SK) is a linear kernel introduced in [32] for strings; we combine it with the linear kernel for length features. The Spectrum Kernel, designed for protein classification using SVM, is similar in nature to the feature vector representation of sequences because it describes the content of a sequence, in terms of substring frequencies. However, it is ignorant to the order or position of such occurrences. In order to calculate the pairwise similarity of two instances (DNA strings), the k-SK uses all subsequences of a fixed length k that occur throughout the instance. If subsequences co-occur frequently throughout two DNA strings, their dot (inner) product under the kernel will be large. The intuition is that the more subsequences two DNA strings have in common, the more likely they are to be similar and share the same biological functions. Biological signals are relatively short, usually 6-14 nucleotides long. We use the Spectrum Kernel with length $k = 6$ denoted **6SK** because a majority of the biological motifs described next are 6-nucleotides long. Other studies of exonic splicing regulators have also focused on hexamers [33], [34].

C. Motif-Spectrum Kernel (MSK)

WDS and SK can be used if there is no prior knowledge about biologically significant motifs (that have an influence on the problem of interest) because WDS and SK use all possible occurrences of subsequences of variable length (in the case of WDS) or fixed length (in the case of SK) to compute similarities. In order to better understand how well "unbiased" kernels capture sequence similarity in a transductive framework, we use the Spectrum Kernel in a slightly different manner. Instead of using all occurrences of k-length subsequences, we use only a selected subset of motifs recognized to have biological significance, and we omit the rest of the subsequences. In other words, we only account for biological motifs, known as splicing regulators, established to work as signals responsible for the occurrence of alternative splicing and potentially result in good classification performance. We denote this kernel as *Motif*-Spectrum Kernel, **MKS**.

Biologically relevant signals, such as splicing regulators, can occur in exons and introns. The ones that occur in exons are called Exonic Splicing Enhancers (ESE), while those occurring in introns are called Intronic Regulatory Sequences (IRS). We use 45 ESE hexamers (6-nucleotide long) derived in [35] for the *Caenorhabditis elegans* dataset. The set of IRS motifs [36] was obtained using comparative genomics in nematodes based on the observation that intronic sequences that are relevant for alternative splicing are highly conserved among closely related species. In order to form the set of IRS motifs, we combined the upstream and downstream motifs and removed duplicate motifs, resulting in a total of 165 IRS motifs assumed to be informative for alternative splicing. The class label was not used in any of these procedures, and repetitive regions were not specifically addressed. A total of 205 biological motifs with variable lengths were present. Their usefulness in a purely semi-supervised framework was reported in [24], and we anticipate that its quality will also aid transduction.

V. Experimental Setup

In this work, we investigate the performance of transductive algorithms TSVM, LP, and MAD on the binary classification problem of predicting cassette exons. Our experimental setup is designed to address the following research questions:

1) What is the most effective transductive algorithm for the problem of identifying cassette exons based on DNA sequences?
2) How does the performance of the transductive algorithms vary with the amount of labeled data?
3) What is the most useful sequence representation and similarity measure (or kernel) when classifying instances transductively?

A. Evaluation

We used 5-fold cross-validation to avoid sampling bias and to be consistent with [19]. Furthermore, in order to use the tuned parameters of the Weighted Degree kernel with Shifts (WDS), we utilized identical splits from the supervised study conducted on the same dataset as [19]. In order to simulate a transductive environment, we deliberately hide some of the labels at random.

In general, the effect of the labeled data on the classification ability, in semi-supervised and transductive frameworks, is far more significant than the effect that the same amount of unlabeled data would have [30]. In order for the unlabeled instances to have an observable impact, they must significantly outnumber the labeled instances. Therefore, we limit the amount of labeled data to 20% of the total dataset, and the test (unlabeled) instances represent the remaining 80%. In order to observe variation in the algorithms' performance, we also decrease the labeled data from 20% (approximately 600 instances per fold, on average) to 5% (approximately 150 instances per fold, on average), by discarding some instances at random, while the test dataset remains the same 80% (approximately 2,415 instances per fold, on average).

Because our dataset is relatively imbalanced (with approximately 5 times more *"constitutive"* instances compared to *"cassette"* instances) – the accuracy of the predictions would not reflect the quality of the classifiers [37]. Therefore, we report the performance in terms of area under the Receiver Operating Characteristic curve (auROC) [38], averaged over 5 folds, and the afferent variance.

VI. Results

We present our results in Table I. The auROC values emphasized in bold font represent the best values obtained by an algorithm for a given amount of labeled data. The colored cells highlight values of the best result overall for a given amount of labeled data. In the first column, the percentages refer to the amount of labeled data used for training the algorithms. The three groups of experiments represent the performances of TSVM, LP, and MAD algorithms using each of the three data representations (and corresponding kernels): (1) the Weighted Degree kernel with Shifts (WDS) for the DNA sequence along with the Linear Kernel (LK) for the Length Features (LG), (2) the 6-Spectrum Kernel (6SK) capturing 6-mers along with the Linear Kernel (LK) for the Length Features (LG), and (3) the M-Spectrum Kernel (MSK) for the biologically relevant motifs and the Linear Kernel (LK) for the Length Features (LG).

We discuss the results by answering the research questions.

1) *What is the most effective transductive algorithm for the problem of identifying cassette exons based on DNA sequences?* Empirical results of our study are encouraging, showing that from limited amounts of labeled data, the performance of transductive classifiers reaches high auROC values (from 0.903 to 0.942 for various amounts of labeled data). These values are comparable to the ones from our previous study of purely semi-supervised algorithms for this problem [24], however, a direct comparison is not possible since the unlabeled and test sets differ in semi-supervised learning from transductive, where the unlabeled data is the actual test data to predict. Overall, TSVM performs better than MAD and LP, especially when trained on smaller amounts of labeled data (5% to 15%). However, MAD more advantageously utilizes the 20% labeled instances.

2) *How does the performance of the transductive algorithms vary with the amount of labeled data?* As expected, the amount of labeled data is a deciding factor for training quality classifiers, and auROC values for all algorithms generally increase with the increase in the amount of labeled data. The trends from our study are consistent with the trends reported on the task of sentiment classification [23].

For the 6-mers representation, MAD and LP recorded more rapid increases in performance from increasingly larger amounts of labeled data. The classification performance improved from 0.621 auROC in the case of 5% labeled data to 0.942 auROC in the case of 20% labeled for MAD, and from 0.615 auROC to 0.864 auROC in the case of LP. TSVM is not as sensitive to the amount of labeled data, and variations are not as abrupt as for graph-based approaches. However, for 6-mers and motifs, TSVM records a counterintuitive decrease in performance at 15% labeled data, most likely due to an erroneously found hyperplane, unrepresentative of the whole labeled data, also suggested by slightly higher variance. This is understandable since TSVM relies on support vectors found in the low density region, as opposed to graph-based methods that utilize a diffusion approach to propagate labels.

3) *What is the most useful sequence representation and similarity measure (or kernel) when classifying instances transductively?* WDS is particularly suitable for MAD and

	TSVM			LP			MAD		
	WDS+LK	6SK+LK	MSK+LK	WDS+LK	6SK+LK	MSK+LK	WDS+LK	6SK+LK	MSK+LK
	DNA+LG	6MERS+LG	Motifs+LG	DNA+LG	6MERS+LG	Motifs+LG	DNA+LG	6MERS+LG	Motifs+LG
5%	0.777±4.9E-4	0.614±3.9E-4	**0.903± 1.9E-4**	**0.800± 6.3E-4**	0.615± 4.6E-4	0.534±93.3E-6	**0.828±39.0E-4**	0.621± 4.9E-4	0.742±2.0E-4
10%	0.811±3.6E-4	0.652±4.5E-4	**0.916±59.6E-6**	**0.810±71.4E-6**	0.698± 6.3E-4	0.565±14.3E-6	**0.828± 7.9E-4**	0.729± 3.3E-4	0.781±4.3E-4
15%	0.838±3.7E-4	0.616±7.0E-4	**0.887± 1.3E-4**	0.801± 3.5E-4	**0.815±15.1E-4**	0.596±37.2E-6	0.830±37.9E-4	**0.873± 1.6E-4**	0.830±3.5E-4
20%	0.858±4.5E-4	0.700±3.4E-4	**0.926±94.8E-6**	0.814±14.1E-6	**0.864±70.9E-6**	0.612± 1.7E-4	0.888± 4.5E-4	**0.942±32.2E-6**	0.835±3.5E-4

TABLE I: AVERAGES OF AUROC VALUES OVER THE 5 FOLDS AND THE CORRESPONDING VARIANCE, WHILE VARYING THE AMOUNT OF LABELED DATA FROM 5% TO 20%, AND MAINTAINING A FIXED TEST SET OF 80%. THE ALGORITHMS ARE TRANSDUCTIVE SUPPORT VECTOR MACHINES (**TSVM**), LABEL PROPAGATION (**LP**), AND MODIFIED ADSORPTION (**MAD**). THE FIRST SIMILARITY MEASURE USED IS THE WEIGHTED DEGREE KERNEL WITH SHIFTS (**WDS**) FOR THE DNA SEQUENCE ALONG WITH THE LINEAR KERNEL (**LK**) FOR THE LENGTH FEATURES (LG). THE SECOND SIMILARITY MEASURE IS THE 6-SPECTRUM KERNEL (**6SK**) CAPTURING 6-MERS ALONG WITH THE LINEAR KERNEL (**LK**) FOR THE LENGTH FEATURES (LG). THE THIRD SIMILARITY MEASURE IS THE M-SPECTRUM KERNEL (**MSK**) FOR THE EXONIC SPLICING ENHANCERS AND INTRONIC REGULATORY SEQUENCES (MOTIFS) ALONG WITH THE LINEAR KERNEL (**LK**) FOR THE LENGTH FEATURES (LG). THE VALUES EMPHASIZED IN BOLD FONT REPRESENT THE BEST PERFORMANCE RECORDED BY AN ALGORITHM FOR A GIVEN AMOUNT OF LABELED DATA, AND THE COLORED CELLS HIGHLIGHT THE VALUES OF THE BEST RESULTS OVERALL, FOR A GIVEN AMOUNT OF LABELED DATA.

LP when learning from limited amounts of labeled data and somewhat useful for TSVM when additional labeled data is available. The 6SK is most appropriate for MAD, which, compared to all three algorithms, seems to be least susceptible to noise, indicated by the fact that when using 6-mers, which probably contain more noisy features than the other representations, MAD achieves better results than TSVM and LP. MKS (biological motifs) along with the length features are the most helpful for TSVM, possibly because TSVM is able to locate a more accurate hyperplane in the space rendered by informative features (*i.e.*, biological motifs established as relevant to alternative splicing) since they are fewer than the 6-mers, which render data to a much higher dimensional space, thereby increasing the difficulty in identifying a good separation.

For 6-mers, TSVM records its worst performance as it is unable to find a correct separating hyperplane in the space generated by these features, possibly due to an unnecessarily high dimensionality (20 times higher than the motifs; 4.2K 6-mers vs 210 motifs). Because MAD has more more features available in the 6-mers set, a greater amount of common information could be propagated among the instances. However, if some of the information in the 6-mers set is noisy, the labeling becomes erroneous, since strong edges could connect positive instances to negative instances. This can potentially occur for small amounts of labeled data (*e.g.*, 5% and 10%). However, for relatively larger amounts of labeled data, (*e.g.*, 15% and 20%), the 6-mers can propagate the labels more accurately. For LP, the best performance is recorded for 6-mers, when the algorithm is presented with relatively larger amounts of labeled data (15% and 20%).

As opposed to TSVM, MAD records unsatisfactory results from MSK (the motif representation), possibly due to the fact that there are only 210 motifs available, and they don't capture overall sequence similarity as well as the set of all 6-mers used by the 6KS, or the various-length matches captured within close proximity of the splice sites by the WDS. Furthermore, a smaller set of motifs could lead to higher-degree nodes which are discouraged in MAD, hence the correct label is not propagated along the connected nodes. For LP, the motif representation is the least compatible.

VII. Conclusions

In this study, we investigate the applicability of transductive approaches to DNA sequence classification. The case study of our work is the problem of discriminating between cassette (or alternatively spliced) and constitutive exons. Experimental results suggest that transductive learning is a useful approach for addressing DNA sequence classification tasks, but we should note that it may be possible to observe different trends for different problems.

We found that biologically relevant features are better exploited by the discriminative nature of the TSVM algorithm, which is able to find a good separation boundary in the space defined by biological motifs. However, when such features are unavailable, the k-Spectrum Kernel is more appropriate for graph-based approaches if a reasonable amount of labeled data is available. Although the best classification performance came mostly from TSVM, this is not a feasible solution when managing massive amounts of data, comprised of more than a few thousand instances. However, MAD is particularly suitable for "big data" and could solve problems posed by larger datasets. Similar to previously reported results [23], MAD outperformed LP on all cases.

In future work, we plan to address other DNA sequence classification problems and evaluate graph-based algorithms on more ample datasets (with hundred thousands instances).

Acknowledgment

The computing for this project was performed on the Beocat Research Cluster at KSU, which is funded in part by NSF grants ACI-1341026, CNS-1126709, CNS-1006860, and EPS-0919443.

References

[1] C. Catal and B. Diri, "Unlabelled extra data do not always mean extra performance for semi-supervised fault prediction," *Expert Systems*, vol. 26, no. 5, pp. 458–471, 2009.

[2] Y.-f. Li and Z.-h. Zhou, "Towards making unlabeled data never hurt," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 1081–1088.

[3] A. Singh, R. Nowak, and X. Zhu, "Unlabeled data: Now it helps, now it doesn't," in *Advances in Neural Information Processing Systems*, 2009, pp. 1513–1520.

[4] Y. Wang and S. Chen, "Safety-aware semi-supervised classification," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 11, pp. 1763–1772, Nov 2013.

[5] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 2.

[6] H. Shin, K. Tsuda, and B. Schölkopf, "Protein functional class prediction with a combined graph," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3284–3292, 2009.

[7] E. Kondratovich, I. I. Baskin, and A. Varnek, "Transd. SVM: Promising Approach to Model Small and Unbalanced Datasets," *Molecular Informatics*, vol. 32, no. 3, pp. 261–266, 2013.

[8] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie, "Profile-based string kernels for remote homology detection and motif extraction," *Journal of bioinformatics and computational biology*, vol. 3, no. 03, pp. 527–550, 2005.

[9] N. Kasabov and S. Pang, "Transductive support vector machines and applications in bioinformatics for promoter recognition," in *Neural Networks and Signal Processing. Proceedings of the 2003 International Conference on*, vol. 1, 2003, pp. 1–6.

[10] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," CMU-CALD-02-107, Carnegie Mellon University, Tech. Rep., 2002.

[11] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly, "Video suggestion and discovery for youtube: taking random walks through the view graph," in *Proceedings of the Seventeenth International Conference on World Wide Web*. ACM, 2008, pp. 895–904.

[12] P. P. Talukdar and K. Crammer, "New regularized algorithms for transductive learning," in *Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2009, pp. 442–457.

[13] K. Kirchhoff and A. Alexandrescu, "Phonetic classification using controlled random walks," in *INTERSPEECH*, 2011, pp. 2389–2392.

[14] Y. Liu and K. Kirchhoff, "Graph-based semi-supervised learning for phone and segment classification," in *INTERSPEECH*, 2013, pp. 1840–1843.

[15] L. De Baets, "Identifying novel neuroblastoma oncogenes using machine learning," Master's thesis, Department of Information Technology, Faculty of Engineering and Architecture, Universiteit Gent, 2014.

[16] P. Bonizzoni, R. Rizzi, and G. Pesole, "ASPIC: a novel method to predict the exon-intron structure of a gene that is optimally compatible to a set of transcript sequences," *BMC Bioinformatics*, vol. 6, no. 1, pp. 1–16, 2005.

[17] H. Lu, L. Lin, S. Sato, Y. Xing, and C. J. Lee, "Predicting functional alternative splicing by measuring rna selection pressure from multigenome alignments," *PLoS Computational Biology*, vol. 5, p. e1000608, 12 2009.

[18] N. Deng and D. Zhu, "dsplicetype: A multivariate model for detecting various types of differential splicing events using rna-seq," in *Bioinformatics Research and Applications*. Springer, 2014, pp. 322–333.

[19] G. Rätsch, S. Sonnenburg, and B. Schölkopf, "RASE: recognition of alternatively spliced exons in *c. elegans*," in *Proceedings of 13th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, vol. 21, 2005, pp. 369–377.

[20] G. Dror, R. Sorek, and R. Shamir, "Accurate identification of alternatively spliced exons using support vector machine," *Bioinformatics*, vol. 21, no. 7, pp. 897–901, Apr. 2005.

[21] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Scholkopf, and G. Ratsch, "Support vector machines and kernels for computational biology," *PLoS computational biology*, vol. 4, no. 10, p. e1000173, Oct 2008.

[22] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble, "Semi-supervised protein classification using cluster kernels," *Bioinformatics*, vol. 21, no. 15, pp. 3241–3247, 2005.

[23] Y. Ren, N. Kaji, N. Yoshinaga, and M. Kitsuregawa, "Sentiment classification in under-resourced languages using graph-based semi-supervised learning methods," *IEICE Transactions on Information and Systems*, vol. 97, no. 4, pp. 790–797, 2014.

[24] A. Stanescu, K. Tangirala, and D. Caragea, "Predicting alternatively spliced exons using semi-supervised learning," *I. J. Data Mining and Bioinformatics*, In press.

[25] A. Stanescu and D. Caragea, "Semi-supervised self-training approaches for imbalanced splice site datasets," in *Proceedings of The Sixth International Conference on Bioinformatics and Computational Biology, BICoB 2014*, 2014, pp. 131–136.

[26] ——, "Ensemble-based semi-supervised learning approaches for imbalanced splice site datasets," in *Proceedings of the Sixth IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2014*, 2014, pp. 432–437.

[27] H. Rangwala and G. Karypis, "Profile-based direct kernels for remote homology detection and fold recognition," *Bioinformatics*, vol. 21, no. 23, pp. 4239–4247, 2005.

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in neural information processing systems*, vol. 16, no. 16, pp. 321–328, 2004.

[29] F. Li, G. Li, N. Yang, F. Xia, and C. Yu, "Label matrix normalization for semi-supervised learning from imbalanced data," *New Review of Hypermedia and Multimedia*, 2013.

[30] T. Joachims, "Svmlight: Support vector machine," *SVM-Light Support Vector Machine http://svmlight.joachims.org, University of Dortmund*, vol. 19, no. 4, 1999.

[31] X. Zhu, Z. Ghahramani, J. Lafferty *et al.*, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, vol. 3, 2003, pp. 912–919.

[32] C. S. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classif." in *Pacific Symposium on Biocomputing*, vol. 7, 2002, pp. 566–575.

[33] W. G. Fairbrother, R.-F. Yeh, P. A. Sharp, and C. B. Burge, "Predictive identification of exonic splicing enhancers in human genes," *Science*, vol. 297, no. 5583, pp. 1007–1013, 2002.

[34] X. Wang, K. Wang, M. Radovich, Y. Wang, G. Wang, W. Feng, J. R. Sanford, and Y. Liu, "Genome-wide prediction of cis-acting rna elements regulating tissue-specific pre-mrna alternative splicing," *BMC genomics*, vol. 10, no. Suppl 1, p. S4, 2009.

[35] J. Xia, D. Caragea, and S. J. Brown, "Prediction of alternatively spliced exons using support vector machines," *I. J. Data Mining and Bioinformatics*, vol. 4, no. 4, pp. 411–430, Jul. 2010.

[36] J. L. Kabat, S. Barberan-Soler, P. McKenna, H. Clawson, T. Farrer, and A. M. Zahler, "Intronic alternative splicing regulators identified by comparative genomics in nematodes," *PLoS computational biology*, vol. 2, no. 7, p. e86, 2006.

[37] F. J. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proceedings of the 15th International Conference on Machine Learning*, ser. ICML '98. Morgan Kaufmann Publishers Inc., 1998.

[38] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 299–310, 2005.