ITR COLLABORATIVE RESEARCH
Machine Learning for Sequences and Structured Data: Tools for Non-Experts
Fernando Pereira, Andrew McCallum and John Lafferty
pereira@cis.upenn.edu, mccallum@cs.umass.edu, lafferty@cs.cmu.edu

Project Summary

This project proposes to dramatically improve the ability of people who are not experts in machine learning to design and automatically train models for analyzing and transforming sequences and other structured data such as text, signals, handwriting, and biological sequences. Working in the context of recent successes with conditional random fields (CRFs) and other conditional models of structured data, the proposed work will achieve its goals through scientific advances in model definition and combination, robust parameter estimation, and data-efficient training procedures, supported by an innovative compositional software architecture.

Intellectual merit: Automatic classification and function fitting (regression) are mature techniques used in many scientific, engineering, and national security applications. Free and commercial software packages allow novice users to train with reasonable confidence of attaining useful results, and to use these results in sophisticated predictive and decision-making systems. However, for sequence and structured data problems, while task-specific machine-learning software has become increasingly available (e.g., speech recognition and genomics software tools), there are no modular, easy-to-use packages for domain experts working in science and engineering problems such as predictive data modeling of spatio-temporal patterns of brain activity or extraction of conceptual and citation networks from the scientific literature, or applications in national and homeland security such as the extraction of social networks from the open Web, or dynamically recognizing suspicious temporal patterns in network logs.

The proposed work will make available for the first time a user-oriented software toolkit for integrated analysis and transformation of sequential and graph-structured data, which will be a major innovation in the data, models, and communications technical focus area. What makes this possible is the convergence of three scientific innovations in learning from structured data. First, powerful, trainable analyzers and transformers for sequences and other structured data can be built by combining simpler conditional models with general composition and product operations based on the theory of weighted automata. Second, a range of capacity-control techniques (feature induction, margin maximization, Bayesian automatic relevance determination) can be generalized to these complex models to control overfitting without the need for extensive hyperparameter adjustments. Third, the need for fully annotated training data can be reduced by combining partial evidence for multiple sequences into a single graph labeling problem. Together, these ideas provide a framework for flexibly specifying and learning parameters for multistep probabilistic transformations from complex data to their structured representations, and for designing and documenting efficient and usable software for model specification, training, combination, and application to real-world data.

Broader Impact: The successful realization of this research will shorten by five to ten years the usual time frame for such complex technology to have real impact in science, engineering, and national security. Progress in science and engineering increasingly demands more effective processing and combination of multiple data sources, ranging from the stored results of high-throughput analyzers to the rapidly growing mass of electronically available literature. The proposed tools will enable researchers to create software components for recognizing, extracting, and cross-linking patterns in sequential and graph-structured data. Security analysts will also benefit from the availability of the proposed tools for extracting relevant information from a variety of data streams, ranging from textual messages to network event logs. The PIs have extensive experience in creating modular software toolkits that have enhanced the productivity of researchers in text classification, statistical language modeling, information extraction, and speech recognition. Their experience will be critical in creating the proposed algorithms and software to satisfy the needs of scientific, engineering, and security communities.
1 The Scientific Opportunity: A Compositional Framework for Learning from Structured Data

Sequential data—text, sound, event logs, biological sequences—arise naturally in a wide variety of scientific, engineering, and intelligence questions. More generally, many important types of data can be viewed as graphs connecting basic data elements—handwriting and other visual data, alternatively spliced genes, gene networks, and linked data structures like the Web. From this point of view, sequential data is the special case in which the basic elements are arranged in a simple chain graph. The term structured data is used in this proposal to refer to these types of data.

Many important processes on structured data can be viewed as the computation of a labeling for the nodes or the edges of the underlying graph. For example, part-of-speech tagging of natural-language text can be seen as the labeling of nodes representing the successive words with part-of-speech labels. Accurate labeling depends not just on individual nodes but also on the contents and labels of nearby nodes, that is, the preceding and following words. Other processes on structured data are better understood as graph transformations. For example, handwriting recognition can be modeled as a cascade of graph transformations from a directed acyclic graph (DAG) representing alternative segmentations of handwriting to another DAG representing alternative transcriptions of the handwriting [28]. For conciseness, the term transducer, borrowed from formal-language theory [40], will be used in the rest of this proposal to refer to any of these labeling or transformation processes.

The modeling and learning of transducers is fundamentally different from that of the more familiar classification processes because the size of the output generally depends on the size of the input, and the choices made for each part of the output will in general depend on choices made for other parts of the output. However, with appropriate representation changes, a transducer can be seen as a collection of node classification tasks, where the choice of label for each node depends on the choices for its neighbors in the graph.

While particular transducer-learning tasks have been tackled with a variety of techniques, the underlying commonalities have for the most part not been exploited. The craft of applying machine learning to structured data is still much less accessible to scientists, engineers, and analysts than the better understood statistical learning techniques of classification and regression. The objective of the proposed research is to make transducer learning a generally applicable tool by achieving the following goals:

- Develop a compositional framework for specifying and training transducers building on the conceptual and practical successes of previous work on conditional random fields [24], weighted transducers [41], and graph-transformer networks [28].

- Make model selection and capacity control [55] for complex transducers, required for the models to generalize well from their training data, easier to carry out by non-experts by building on successful methods for feature induction, margin maximization, and automatic relevance determination.

- Alleviate the need for extensive, fully annotated training data by building on preliminary results that extend to transducers partially labeled data and active learning techniques that have proven successful for classification tasks.

- Implement the proposed methods in a software toolkit designed and documented for a broad technical audience, building on the experience of several large-scale software projects for text classification, language modeling, information extraction, and speech recognition [60, 34, 40].
• Work with a few selected end users in the areas of scientific text mining, brain imaging, bioinformatics and environmental datamining to field test and refine the software toolkit and associated methodology.

2 The Need: Structured-Data Machine Learning Tools for Non-Experts

Statistical modeling tools are at the foundation of progress in science. From astronomy to biology, from climatology to psychology, automated classifiers such as decision trees and support vector machines have become the every day workhorses used to find patterns and make predictions. These methods, which assume that the items under analysis are independent and identically distributed (iid), have been implemented in numerous free and commercial software packages, and these packages have had tremendous impact by allowing scientists to train classifiers and regressors with reasonable confidence of attaining useful results.

Notably, these tools can be effectively used by scientists, engineers, analysts and educators who are not experts in statistical modeling. Because the statistical methods are sufficiently well understood, these standard tools can provide automatic, self-tuning front-ends—incorporating capabilities such as decision tree pruning, stepwise regression and cross-validation. The users, experts in their own disciplines rather than in statistical modeling, do not need to have deep understanding of these techniques, nor of regularization, numerical optimization methods, or efficient data representations. These easy-to-use toolkits, with wide applicability across many domains, serve as gathering places for statistical techniques that have arisen across many scientific fields over the past decades.

In contrast to tools for traditional iid data, machine learning methods for transducing structured data are not yet available in modular, easy-to-use form. Interest in data with structured dependencies, such as sequences, trees and other graphs, has risen sharply within the last decade. Text mining, Web mining, speech systems, computer vision, bioinformatics, network security, social network analysis, and fraud detection are just a few examples of domains in which relations among the various predicted variables must be captured for good performance. The following are several concrete applications in some of these areas that could benefit from advances in learning transducers for structured data:

• **Mining medical abstracts for disease-related genetic variations**: Research on particular diseases would benefit from up-to-date databases of genetic variations that have been found to be associated with the disease. A variation tagger would scan medical abstracts for mentions of genetic variation events associated with the disease. Individual taggers would recognize gene citations, variation types, variation locations, and phenotypic state changes; a second-level transducer would recognize meaningful combinations of those elements in variation event descriptions.

• **Aiding homeland security by helping analysts mine open source Web pages**: Classifying Web pages on a particular Web site labels the pages (nodes) with labels that depend not just on the page but also on the contents and labels of pages hyperlinked with the given page [52]. Here the labels of the pages are clearly not independent, and depend on the relations between pages across the entire site, as well as the relations with outside sites. This problem is naturally approached in a compositional framework: the classification of pages may depend on information such as named entities extracted in other pages in the domain; thus page-level classification is dependent upon entity-level extraction within pages.

• **Determining function in metabolic pathways**: Networks of documented biochemical reactions, built up over several years by biological scientists, encode a great deal of information. One version of this network can be represented as a graph whose vertices are genes, and whose edges indicate that the two genes have been analyzed as participating in successive biochemical reactions; the scientific challenge is to label the nodes in this gene graph according to biological function and, ultimately,
phenotypic traits. Recent experiments using support vector machines to predict the functional class of genes from the yeast *S. Cerevisiae* have been promising [57], but have not yet begun to fully exploit the structured nature of the data.

- **Protein structure prediction**: The problem of predicting protein side-chain conformation is a subtask of the fundamental challenge of protein folding and molecular design. In the side-chain prediction problem, the distinctive amino acid side-chains interact in a complex manner that depends on the geometry of the backbone. This is naturally modeled using a graph which captures the pairwise interactions between neighboring side-chains; the graph is relatively complex, with many short cycles. Recently, approximate inference methods for graphical models have been shown to be promising for predicting side-chain configurations, using standard physical models (van der Waals energy potentials) as “features” [59].

- **Mapping brain images to cognitive states**: Functional MRI (fMRI) brain imaging gives time series data that represents the activity in a collection of 2-dimensional slices of the brain. This activity is measured as a ratio of oxygenated to deoxygenated hemoglobin in the blood at the corresponding location in the brain. From labeled data collected by asking human subjects questions during fMRI scans, the scientific challenge is the difficult “inverse problem” of inferring cognitive states from the raw fMRI data in new samples. The problem has a natural time series structure that is currently being modeled with hidden Markov models [38]. However, the registration problem of mapping between different brains, as well as the high dimensionality and structure of the image data suggests that an approach based on transducers and conditional sequence models may be more effective.

- **Computer security and intrusion detection**: Intrusion detection is the problem of monitoring networks to discover unauthorized usage or other suspect behavior. The intrusion detection problem is extremely difficult when given only data in the form of computer traces, which have a natural sequential structure. Anomalous behavior can take many forms, including abuse of access by trusted users and automated attacks launched by outsiders [2]. Work in machine learning has already approached the detection task using HMMs and instance-based learning methods, transforming temporal sequences into points in a metric space [26, 27, 29]. However, further advances may come from a general framework based on transductions, conditional sequence models, and sequence kernels, in order to account for the multi-resolution nature of the problem, where attacks can range from seconds to months.

New methods for capturing structured, relational data have developed rapidly in the past several years, but these methods are complex and fragile, and significant scientific progress is required in order to enable the creation of general-purpose, easily used tools. These difficulties are described in the next section.

In the absence of such general-purpose tools, scientists are in some cases creating sub-optimal and domain-specific tools. For example, as noted above, current gene finders do not use a modular structure that allows the ready use of multiple information sources or alternative scoring models. In other cases, users are simply going without appropriate tools and incurring significant costs in manual labor. For example, the RTKNET aggregates information about pollution and environmental impacts from government and Web-based sources. Without having easily trainable tools for gathering and mining this information, they are attempting to extract this data by hand.

The lack of robust, capable and easily deployed tools for structured data causes not only inefficiency due to inaccuracy, but also inefficiency due to *missed scientific opportunities*. Because gene segmentation cannot be composed with EST data and comparative alignments, bootstrapping gene finders for newly sequenced genomes is much harder than it could be. Because not all environmental documents can be manually processed, some findings will be missed.
Having each domain-specific team hire its own machine learning expert is not a satisfactory solution. Not only would the cost be prohibitive, but there would not be enough experts to go around. More importantly, feature engineering and interactive data manipulation for hypothesis testing are extremely important parts of the scientific process, and only the domain expert has the ability to do this. It is therefore essential that the statistical modeling tools are directly usable by domain experts.

We lack general tools for modeling sequences, trees and graphs—an analogue to S (or R) [56] for structured data. Necessary to its creation is not only a gathering of existing techniques and some software engineering, but more importantly, fundamental scientific progress in the understanding of structured models, how to assemble solutions from pieces and how to make them robust in the hands of non-experts. These problems are discussed further in the next section.

3 The Problem: It is Difficult to Set Up and Train Models

In the tasks described in the previous section, structured data objects (sequences or graphs) are required to be transduced to outputs representing the structure or relevant content of the input. In general, a single input may yield several alternative outputs corresponding to alternative interpretations of the input. To select among the alternatives, the transducer may assign probabilities to alternative outputs, although non-probabilistic ranking is also possible as discussed in Section 4.

Free and commercial software packages for specifying and training transducers are restricted to particular application domains, and require a major investment of resources to be applied successfully even to their chosen domains. None provide a compositional language of trainable transducers that would support independent development of modules and their combination and retraining for new applications. None provide general capacity-control methods to avoid overfitting that do not require the user to explore painstakingly the feature and parameter space. None include techniques for taking advantage of partially-labeled data, or for systematically transferring the predictions on one dataset to similar datasets. These difficulties are now addressed in more detail.

The transducer for a complex task is typically composed from simpler transducers that extract and combine different kinds of information needed to construct the final result. For instance, information extraction may involve part-of-speech tagging, some form of (possibly shallow) parsing, entity and relation recognition stages, and a co-reference stage that merges multiple mentions of an entity and separate parts of the same relation instance. A particular choice of extracted entities and relations may be arrived at through different combinations of intermediate results, and the probability of that result is obtained by marginalizing over all intermediate alternative results. Therefore, a user needs a flexible, easy to understand framework for building up such complex transducers from the simpler transducers in each stage.

Training such complex models raises data, statistical, and algorithmic challenges. On the data side, obtaining consistent, complete, and accurate labeled data is more difficult than for classification tasks. Except in a few tasks in which annotated data is already created as part of a common human activity (for example, closed captions for video, translations), hand annotation with complex output labels (for example, entity tags, parse trees, annotated biological sequences) is laborious, difficult to verify, and expensive. Furthermore, high accuracy even in manual annotation is difficult to achieve. For example, in information extraction, human annotators can reach a high degree of agreement on the label for a particular word or phrase, but they have much greater difficulty in determining all of the words or phrases that must be labeled. In addition, complex annotations like parse trees are based on a variety of assumptions that may change as a project evolves. Therefore, it is likely that annotated data may be incomplete with respect to later project requirements. Finally, multiple types of annotation may be required for different stages of analysis or transformation, and it is difficult to use the different types of annotation together in training.

On the statistical side, the models under discussion are typically very complex [49], and thus vulnerable
to overfitting, especially given the difficulties in obtaining large training sets. Common schemes for alleviating parameter overfitting involve hyperparameters that must be adjusted by the user [7]. In addition, the number possibly relevant features in sequence or graph data is unbounded (since inputs themselves may be arbitrarily large), increasing the opportunities for overfitting. Finally, model structure itself, for example the number of states in finite-state sequence models, is difficult to determine for optimal generalization.

The situation for structured data today is similar to that for iid data problems ten years ago. While many successful uses of classification technology was made in several application areas, it was difficult for non-experts in the underlying machine learning algorithms to make use of the latest developments in techniques such as support vector machine classifiers. Today, this technology is being very widely deployed for iid prediction problems, and are no longer considered too complex for non-experts. As just one example, automatic classification has become so effective and easy to use that millions of email users rely daily on automatically-trained junk mail filters that perform very well and constantly adapt to the changing attributes of spam.

In the same way, sequence modeling technology based on hidden Markov models and related models already exists in some form in a variety of deployed systems, such as text mining systems, speech and handwriting recognizers, and gene finders. For instance, both text mining software and gene finders often use trainable probabilistic finite-state machines to recognize particular subsequences (e.g. person names, exons). However, these successes have come from highly dedicated individuals or teams of researchers who have had to become experts in both their applications areas and in the underlying machine learning techniques. Moreover, the underlying models and algorithms have been unnecessarily instantiated in application-specific ways that impede their convenient application to new areas. The current state of the art is considered too complex for non-experts to make effective use of it in new applications without a prohibitive investment of resources. The goal of the work outlined in this proposal is to accelerate the development of statistical learning methods for structured data so that these methods can more quickly and effectively be used by non-experts in a new generation of applications.

4 The Solution: Composable, Learnable Transducers for Structured Data

The previous sections discussed how standard methods based on “independent and identically distributed” assumptions are inadequate to meet the challenges of complex data analysis tasks in science, engineering, and intelligence gathering. It was also argued that an approach based on complex transducers for structured data is required. In this section further detail on the proposed solution is given, concentrating primarily on sequence data, understood to include the sequences themselves represented as chain graphs as well as graph structures derived from sequence data through factorial models, alternative interpretations of a sequence, or similarity links between sequences.

The proposed work, discussed in detail in Section 4.2 below, integrates and extends two naturally complementary frameworks for sequence processing developed by the PIs in the last decade: conditional random fields (CRFs) [24, 32] and weighted finite-state transducers (WFSTs) [40, 41]. CRFs provide a language for describing the probabilistic dependencies in a transducer, while WFSTs provide a language and algorithms for composing and optimizing transducers. As is already the case with CRFs, the combined framework facilitates the creation of rich models with millions of tunable parameters. Given the well-known tradeoffs between model complexity, training set size, and generalization error, effective use of the proposed modeling framework will require sophisticated ways of exploiting training data, including unlabeled or partially labeled data, and of reducing overfitting.
4.1 Background Work

4.1.1 Conditional Random Fields

Markov random field (MRF) models originated in statistical physics, and have been extensively used in image processing [16]. In statistical physics, a region of space of a physical entity (such as an atom) is conveniently represented by a vertex in a graph, and each vertex is labeled with a quantity that indicates some physical property such as the spin of a particle or a “field” at the position corresponding to that vertex. The undirected graphical models developed in statistics and artificial intelligence [12] are essentially the same class of models.

In the context of machine learning, it is possible to create a graph with a node for each example, and with edges between examples that are similar to each other. This point of view plays two closely related roles in the work under discussion. First, each position in a sequence is seen as an example to be labeled, and adjacent positions are linked because adjacent labels are correlated. Second, position $i$ in sequence $x$ and position $j$ in sequence $x'$ may be linked because the neighborhoods of $i$ in $x$ and of $j$ in $x'$ are similar according to some similarity metric (kernel) and therefore the labels for those positions should be correlated. The first perspective was developed in the previous work on CRFs, while the second has been developed in work on exploiting unlabeled data. The two came together in recent preliminary research that this proposal plans to build on. Details on these related strands are discussed later in this subsection and in Section 5.

Customarily, random fields are based on generative models having two components, one involving the “smoothness” of the labels with respect to the underlying graph structure, another attempting to capture how well the labels match the data. In order to have computational tractability, the data items (e.g., the pixels in image processing) are conditionally independent given the labels. This same assumption is made in hidden Markov models (HMMs) and stochastic grammars. Conditional random fields (CRFs) [24] are random fields that are globally conditioned on the input data to be labeled. Rather than building a generative model of latent variables and data, a conditional random field assigns a conditional probability $p(y|x)$ to the labels $y$ given the input $x$; a CRF does therefore not model the marginal distribution of the data $p(x)$. This allows greater context-sensitivity in a CRF without sacrificing the efficiency in the underlying statistical inference algorithms that is achieved by strong independence assumptions in, for example, HMMs. Our practical successes with CRFs are discussed in Section 5; several independent research groups have built on these ideas with new training methods and applications [10, 22, 53].

4.1.2 Model Structure and Combination

Weighted finite-state transducers were developed by one of the PIs and collaborators at AT&T as a compositional framework for multi-stage tasks in speech and text processing [45, 40, 41]. The framework has been successfully implemented in a software toolkit that is used in fielded speech recognition systems at companies including AT&T and ScanSoft, and for research purposes at many universities. In this toolkit, individual stages were trained separately with specialized methods and then combined using a variety of optimization techniques to minimize the search space over possible outputs. Beyond its original application to speech recognition, the weighted transducer framework has been found to be very effective for building complex transducers in machine translation [18, 48, 21]. These ideas also influenced the development at AT&T of graph transformer networks for handwriting recognition [28]. A graph transformer is a weighted transducer over DAGs in which the weights are parameterized functions, for example multilayer perceptrons. A composition of graph transformers representing processing stages can be trained globally by gradient propagation. Finally, recent results show that weighted transducers can be efficiently trained with partially labeled data with a powerful generalization of the Baum-Welch algorithm [14].
4.2 Research Directions

As outlined earlier, the overall goal of the proposed research is to create an integrated toolkit for learning from structured data that will be usable effectively by domain experts. To do this successfully, a suitable software architecture must be chosen and implemented, and scientific progress is needed on several specific fronts outlined earlier.

4.2.1 Model Structure, Inference Algorithms, and Basic Training

Our experience with previous text and speech processing toolkits [8, 41] shows that users need a modeling framework that allows complex models to be built from simpler, possibly reusable parts using transparent composition operations. The proposed framework will be built from:

- An inventory of deterministic feature-extraction steps that map a sequence to a sequence of feature vectors representing local properties of the sequence at each position.
- For kernel-based models, a base set of sequence kernels that evaluate the local similarity of two sequence positions.
- Feature- and kernel-based CRFs specified in a language that allows for a variety of state structures, including factorial structures, with parameter tying.
- Composition operators for building multi-stage transducers by combining feature extractors, kernels, and individual CRFs.

The main algorithmic challenge in implementing the framework is to achieve efficient inference — computing the most likely output and model expectations involving intermediate and final results — in composed models. The existing techniques for WFSTs will be very helpful here, but their use requires a careful characterization of the feature extractors and kernels that can be efficiently implemented as weighted transducers for composition with the CRF steps [11].

A second algorithmic challenge is the choice of approximate inference methods for intractable models, in particular factorial models, and how that choice interacts with the basic composition machinery.

As is the case in WFSTs, all algorithms generalize naturally to DAG data that can be taken as representing alternative interpretations of a sequence. In addition, search and pruning operations can be readily expressed in terms of search on weighted DAGs [30, 42, 39].

For basic training, we will develop a suitable generalization of CRF training algorithms to composed transducers, building techniques developed earlier for graph transformer networks [28] and WFSTs [14].

4.2.2 Model Selection and Capacity Control

One of the most delicate and time-consuming aspects of configuring a new model is feature engineering. It often takes considerable expertise not only in the task at hand, but also in the machine learning methods’ overfitting behavior in order to add just the right features—not too many features, yet features of the right kind. Automatic feature induction methods allow novice users to indiscriminantly add the “kitchen sink” worth of features, and let the system determine which features to include or exclude, and also which feature conjunctions to use. Our preliminary results with feature induction show that it indeed remains robust when the user indiscriminantly provides a very large number of candidate features, and can reduce error by 40% [32]. There is, however, considerable further work to be done. First, current results use a poor mean field approximation; we will experiment with improved approximations. Second, the search through the space of possible feature conjunctions is currently quite time-consuming; we will explore more efficient ways of...
searching through this space, such as pre-computing statistics about features that frequently co-occur, and using ideas from optimization methods, e.g. [5], to learn distributions over good conjunctions to consider by using “features of features.” Finally, we will explore the use of these methods to learn not only new feature conjunctions in input space, but also to add new edges among the predicted variables—that is, automatic model structure learning—which will be especially helpful for novices who may have little experience in specifying relations among the inputs.

Even with feature induction in place, CRFs may still overfit. A prior over feature weights, such as the Gaussian prior [7] helps somewhat, but it does not seem to effectively control capacity even as its variance hyperparameter is tuned. Alternative methods for capacity control include $L_1$-shaped priors to encourage sparse parameterization, (as opposed to the $L_2$-shaped prior), automatic relevance determination [44], and other Bayesian methods, such as expectation propagation [37]. Other methods include conjugate priors which automatically adjust to different dynamic ranges when the input features are in different numeric ranges—a good property if users are not expected to think deeply about feature engineering. As discussed in the next section, maximum margin approaches are an interesting alternative to weight priors, but their cost grows quadratically with the total length of the training data, and they may not offer significantly better generalization performance for larger training sets.

4.2.3 Alternative Loss Functions

Conditional random fields have primarily been used within a standard statistical framework where the loss function employed for training is the (regularized) log-likelihood of the data. However, alternative loss functions may be desirable for different applications, or because of convenient mathematical properties. As one well known example, the so-called “hinge loss” of support vector machines [55] can result in sparse solutions, with few support vectors, leading to attractive mathematical guarantees on the generalization performance of the underlying classifiers.

Recently, the analogue of such alternative loss functions has been proposed in the framework of random field models of structured data [53]. However, current understanding of the impact of alternative loss functions for structured models is limited. Indeed, an understanding of the issues even in the much simpler case of iid data is only recently emerging.

One of the most fundamental properties a statistical learning procedure is consistency. That is, in the limit of infinite training data, does the classifier achieve the optimal Bayes error rate? This question has only been recently resolved for surrogate loss functions for large margin classification [61, 3]. It is therefore of significant interest to explore the related questions for structured classification using CRFs and related methods such as max-margin Markov networks [53]. To formulate the consistency problem for conditional graphical models, we need two things: a notion of admissibility, and a result that says admissibility implies convergence to minimum $\ell$-loss, where $\ell$ is the error metric (such as 0-1 loss) that is actually of interest in the application.

At the same time as investigating such foundational questions, we propose to make the use of alternative loss functions available to users within the software toolkit that will be developed as part of this project. The goal is ultimately to enable users to be able to easily specify (surrogate) loss functions for training, without being experts in the technical details, and for the system to validate whether or not the loss function is admissible, resulting in a consistent classification procedure. While this is only recently (with the past two years) feasible for large margin classification for iid data, our aim is to achieve this level of understanding for structured data classification as well.
4.2.4 Incorporating Kernels into Structured Models

The use of Mercer kernels for transforming linear classification and regression schemes into nonlinear methods is a fundamental idea, which was recognized early in the development of the perceptron, splines, and support vector machines [1, 17, 4]. The recent resurgence of activity on kernel methods in the machine learning community has led to the further development of this important technique, demonstrating how kernels can be key components in tools for tackling nonlinear data analysis problems and integrating data from multiple sources.

Kernel methods can typically be viewed either in terms of an implicit representation of a high dimensional feature space, or in terms of regularization theory and smoothing [46]. Recent work by the PIs [19] has introduced kernel methods that can be applied to discrete, categorical data, in particular when the data lies on a graph. This work has already been used in biological problems where genes can be represented in a graphical structure [57, 25]. However, our understanding of kernel methods for structured data such as sequences is just beginning.

More recently, the PIs have introduced an extension of conditional random fields that permits the use of implicit features spaces through Mercer kernels, in contrast to previous work on CRFs that made use of explicit feature spaces [23]; related ideas appear in [53]. In particular, this recent work gives an extension of the classical “representer theorem” [17], showing how kernel CRFs can be motivated by regularization theory. The resulting techniques combine the strengths of hidden Markov models, or more general graphical models, kernel machines, and standard discriminative linear classifiers including logistic regression and SVMs. This framework is very new, and much work remains to be done to better understand and exploit the use of kernel methods for structured, non-iid data problems. In particular, sequence and directed graph kernels [31, 11, 47, 51] provide a powerful way of representing local sequence properties, and their efficient use in kernel CRFs through weighted transducer techniques deserves exploration.

4.2.5 Active Learning

Rather than requiring data to be labeled in advance, in practice, it may often make sense to utilize active learning. That is, we might allow the learning algorithm to pick a set of unlabeled instances to be labeled by a domain expert, which will then be used as (or to augment) the labeled data set. If we have to label a few instances for semi-supervised learning, it may be attractive to let the learning algorithm tell us which instances to label, rather than selecting them randomly. Limiting the range of query selection to the unlabeled data set results in a method known as pool-based active learning or selective sampling.

There has been a great deal of research in active learning. For example, queries may be selected to minimize the version space size for support vector machines [54]; to minimize the variance component of the estimated generalization error [9]; or to solve disagreements among a committee of classifiers [15]. However, most of the active learning methods do not take further advantage of the large amount of unlabeled data once the queries are selected, with a few exceptions including the use of EM with unlabeled data integrated into active learning [35], and using semi-supervised learning method during training [43]. In addition to this body of work from the machine learning community, there is a large literature on the closely related topic of experimental design in statistics, especially from a Bayesian perspective [6].

Recent work has shown how conditional random fields can be useful in the setting of active learning. In brief, this work allows one to efficiently estimate the expected generalization error after querying a point, which leads to a better query selection criterion than naively selecting the point with maximum label ambiguity. Then, once the queries are selected and added to the labeled data set, the classifier can be trained using both the labeled and remaining unlabeled data. Preliminary results using this framework are quite promising [63].
4.2.6 Software Architecture

As shown by the AT&T FSM software library [40], compositionality is the key property that allows the creation of complex transducers from simple, easy to understand and specify parts. In the FSM library, extremely powerful transducers can be created by applying general operations (concatenation, sum, iteration, composition) to primitive transducers. The language formed by these operations has a simple mathematical meaning, but their implementation exploits special cases for efficiency in subtle ways. Furthermore, the simple mathematical semantics opens up the opportunity for meaning-preserving optimizations that allow efficient creation and application of very complex transducers. A compositional architecture also supports the reuse of previously developed models, and rapid modeling innovation as users find new ways of combining models.

For the proposed work, the main objects will be elementary trainable CRFs, feature extractors, and eventually kernels, that can be composed into sophisticated multistage pipelines as discussed above. Such a compositional framework fits very well into an object-oriented design, in which particular types of basic transducers and transducer combinators are organized in a class hierarchy rooted in a general transducer abstraction. Particular properties, for instance trainability, are represented by additional abstractions (in Java, interfaces). Processes that involve transducers, such as creation from external representations, batch or active training, or application, will be implemented as objects that encapsulate appropriate process parameters (for instance, data preprocessing for training, or instance selection policy for active learning).

A compositional architecture is also an excellent foundations for a scripting language in which complex transducers can be specified and operated much more concisely than in the implementation language (Java). Early in the project, we will determine whether to use a custom scripting language, or a general-purpose object-oriented or functional language like Python or Scheme. The scripting language will be critical in making the toolkit accessible to a wider audience in the same way as the concise, high-level language of MATLAB was critical in making sophisticated matrix computations available to a wide audience of scientists and engineers.

A graphical front-end will also be created for facilitating interactive data labeling and active learning, as well as fast prototyping of simpler applications.

In addition to the AT&T FSM library discussed above, the PIs have led or played major roles in the development of several widely used software packages for text and speech processing, including the Bow text classification package [33], the Lemur language modeling toolkit [60], and the MALLET machine learning library [34]. Their extensive experience with these systems will inform the detailed development of the proposed software.

4.3 Targeted Applications

Experimental work will be important in our research program, in order to help drive the development of tools and software for structured classification, as well as to validate that the tools are indeed accessible to non-experts. We will pursue applications related to brain imaging, biological sequence modeling, and text datamining.

4.3.1 fMRI Time Series Analysis

As discussed briefly above, functional MRI brain imaging is a revolutionary technology that has the potential to transform both medicine and scientific research. Typical fMRI data gives a time series that represents the activity in a collection of 2-dimensional slices of the brain, as measured by levels of oxygenated hemoglobin at various locations called “voxels,” which are roughly 3x3x5 mm regions. Brain scans of approximately 15,000 voxels are obtained at a rate of between one and two images per second. This data represents a time
Figure 1: CRF classification on 5 protein fragments (PDB ID top to bottom: 1S01, 1RHD, 1RBP, 1R092, and 1PYP). The first row of each protein shows the true labels as coil (red), beta sheet (green), and helix (blue). In the second row, R,G,B color components reflect the posterior probabilities. The third row shows errors made by the CRF with black dots. Although the accuracy is not yet satisfactory, greater accuracy is achieved at transition points between biologically interesting structures than is achieved with support vector machines used with heuristic “sliding windows.”

series “movie” of the brain activity, where each image is a highly structured object, and where the images are highly correlated in time. From labeled data collected by asking human subjects questions during fMRI scans, or by having them perform specific tasks, the scientific challenge is the difficult “inverse problem” of inferring cognitive states from the raw fMRI data in new samples.

Using standard static classifiers that treat voxels as iid (specifically, Gaussian naive Bayes), researchers have already been relatively successful in predicting high level cognitive states such as “the subject is reading an ambiguous sentence” versus “the subject is reading an unambiguous sentence” [38]. Current work at Carnegie Mellon is beginning to investigate the sequential nature of the time series data through the use of hidden Markov models with Gaussian or mixture of Gaussian emission models, where each voxel is modeled as conditionally independent given the HMM state.

In this preliminary work the classifiers have been subject-specific, due to the difficult problem of mapping between physiologically different brains in different subjects. In order to model the time series nature of the data as well to develop cross-subject methods, it will be natural to use the kernel CRF methods developed recently by the PIs, together with transducers that incorporate mappings across subjects. The strong independence assumptions made by the HMM model, for example that the voxels are conditionally independent given the state, are relaxed under the CRF model, which is a conditional model of the state given the entire image.

4.3.2 Biological Sequence Analysis

Computational biology is an important application area were structured prediction problems arise naturally and frequently. As noted earlier, current gene finders are based on a single level hidden Markov model (HMM) that comingles information relevant to transcription and translation. We plan to develop more biologically realistic gene finders using transducers modeling each biological subprocess, incorporating partial information about the genome, and other sources of information such as ESTs.

In a similar direction, we have already obtained preliminary results using kernel CRFs for protein secondary structure prediction. This is the task of mapping primary sequences of amino acids onto a string of secondary structure assignments, such as helix, sheet, or coil. It is widely believed that secondary structure can contribute valuable information to discerning how proteins fold in three dimensions. We compare kernel conditional random fields, estimated using clique selection, against support vector machine classifiers, with both methods using kernels derived from position-specific scoring matrices (PSI-BLAST profiles) as input features.

Secondary structure prediction is a problem that has been extensively studied for more than 20 years; yet the task remains difficult, with prediction accuracies remaining low. The major bottleneck lies in beta-
sheet prediction, where there are long range interactions between regions of the protein chain that are not necessarily consecutive in the primary sequence. Our experimental results indicate that kernel CRFs have the potential to lead to progress on this problem, where the state of the art has been based on heuristic “sliding window” methods. However, our results also suggest that the improvement will be limited by the lack of a good kernels—kernel design is a challenge that may best be tackled by biologists and machine learning researchers working together.

CRFs have only slightly lower overall error rates, but further information can be obtained by studying transition boundaries, for example, the transition from “coil” to “sheet.” From the point of view of structural biology, these transition boundaries may provide important information about how proteins fold in three dimensions. From a computational and modeling perspective, they may indicate positions where most secondary structure prediction systems fail. While this is a very difficult subproblem, CRFs significantly increase the accuracy over SVMs at transition points.

4.3.3 Environmental Datamining

RTKNET is an online service providing access to various government databases—mostly those dealing with pollution. It provides free access to numerous databases, text files, and conferences on the environment, housing, and sustainable development. With the information available on the service, users can identify specific factories and their environmental effects, analyze reinvestment by banks in their communities, and assess people and communities affected. RTKNET was started in 1989 in support of the Emergency Planning and Community Right to Know Act (EPCRA), which mandated public access to the Toxic Release Inventory. In the past it was funded by various government agencies through a previous partner, Unison Institute, and foundations. RTKNET has been praised by Presidents, by the Environmental Protection Agency (EPA), by community groups across the country, and by many others.

We have been in contact with Richard Puchalsky, who works for RTKNET. Its data is currently gathered and integrated by hand from the Web and other sources. This process involves significant labor, and lack of human resources often means that data is missing. They are interested in using automatic tools to extract information from text, but do not have the machine learning expertise to successfully configure such tools themselves. We will work with them—not providing them with pre-trained solutions, but providing them with the easily-trained tools to create their own solutions—using them as a test-case for our the usability of our system by non-experts. It is interesting to note that this work at RTKNET corresponds significantly to the text data mining needs of agencies involved in homeland security.

5 Previous Results from NSF-Funded Work

One important motivation for the present proposal is the accelerating accumulation of theoretical, algorithmic, and experimental results on individual aspects of learning models of structured data, which taken together, strongly suggest that the time has arrived for developing an integrated toolkit for a broader audience. Among the results that support this optimism, the following advances have come from NSF-funded research by the PIs.

5.1 Applications of Conditional Random Fields

- Large-scale models for text: Using carefully-designed trained algorithms based on state-of-the-art convex optimization procedures, a CRF model achieved the best accuracy of any single model on a standard shallow-parsing task [49].

12
• **Extraction from biomedical text**: Using feature induction (described below) and the fast trainer discussed above, we have achieved promising results in extracting gene and protein (82.4% $F_1$ measure) and genetic variation (80% $F_1$ measure) mentions from biomedical abstracts [36].

• **Gene finder**: We are developing a new gene finding algorithm that combines evidence associated to transcription, splicing, and translation into a single semi-Markov CRF model trained with the voted perceptron algorithm [10]. We have recently achieved competitive results, which we are improving for publication.

5.2 New Model Structures

• **Dynamic CRFs**: Many sequence-processing problems are solved by chaining errorful subtasks. For example, in information extraction one often performs part-of-speech tagging and then shallow parsing as pre-processing steps before the main extraction task. In such an approach, however, errors early in processing nearly always cascade through the chain, causing errors in the final output. Recently we have begun work on addressing this problem by representing the multiple label sequences in a single graphical model, explicitly modeling limited dependencies between them [50]. For example, factorial structure models the dependencies between cotemporal labels, allowing information to flow between the subtasks in both directions. Using these models to perform joint part-of-speech tagging and noun-phrase segmentation, we are able to reduce by half the amount of training data required to reach a certain level of precision and recall. Significant open research questions remain, including work in training methods that maximize the marginal probability of only a subset of the factors rather than the full joint, work in more complex structures including hierarchical DCRFs (analogous to hierarchical HMMs) and DCRFs with memory of arbitrarily distant events, and methods for automatically learning factorial structure, which will make these models more accessible to novice users.

• **Kernelization of CRFs**: In recent work [23] we have developed an extension of conditional random fields that permits the use of implicit features spaces through Mercer kernels. Such an extension is motivated by the significant body of recent work that has shown kernel methods to be extremely effective in a wide variety of machine learning techniques; for example, they enable the integration of multiple sources of information in a principled manner. Our introduction of Mercer kernels into CRFs is also motivated by the problem of semi-supervised learning. The resulting framework has been applied to the problem of protein secondary structure prediction, the task of mapping primary sequences of amino acids onto a string of secondary structure assignments, such as helix, sheet, or coil.

5.3 Capacity Control and Model Selection

• **Feature induction**: A new training algorithm automatically builds new features from conjunctions of existing features, and has given improved results in named entity extraction. It is similar to techniques that will be used to learn data mining patterns that would improve extraction, and vice-versa [32].

• **Predictive confidence assessment**: By combining the exact calculation of marginals with meta-classifiers, we are able to provide better estimates of empirical uncertainty, helping to make output more interpretable [13, 20].

5.4 Using Unlabeled Data

• **Conditional Gaussian random fields**: Recent work shows how semi-supervised learning can be effectively carried out using a type of conditional random field based on joint Gaussian models [62].
Using a “relaxation” from a discrete to a continuous sample space, the most probable configuration of
the field is unique, is characterized in terms of harmonic functions, and has a closed form solution that
can be computed using matrix methods or loopy belief propagation [58]. These models also support a
natural combination of active learning and semi-supervised learning, by allowing the efficient estimation
of the expected generalization error after querying a point, which leads to a better query selection
criterion than naively selecting the point with maximum label ambiguity. Preliminary results using
this framework are promising [63].

6 Work Plan, Evaluation, and Dissemination

6.1 Work Plan

The proposed software will be developed in Java as an extension of the MALLET open source machine-
learning toolkit [34], which already has initial implementations of CRFs, an efficient training algorithm, ba-
sic capacity control through user-specified priors, and feature induction. In our experience, Java’s strengths
as a portable, object-oriented, strongly typed language with automatic memory management make it prefer-
able to lower-level, somewhat more efficient languages for complex machine-learning tasks.

The initial work plan is outlined below. However, individual tasks may be rearranged with input from
users, especially external user groups. The project budget includes a programmer at the University of Penn-
sylvania, who will be in charge of the scripting layer, documentation, bug tracking, software releases, and
Web-based information dissemination tasks specified in the work plan.

We will integrate research and education not only through the training of our graduate students, but also
through interdisciplinary work and training with students in the target fields (including cognitive science,
biology, and public policy). These students will be excellent test-case examples of the “domain-expert”
scientists whose research our tools are designed to accelerate.

Year 1

• Define initial system requirements, software architecture, core components, and utilities.
• Implement weighted transducer composition, composable CRFs, and composable feature extractors.
• Implement capacity control by automatic relevance determination.
• Select and implement a scripting language for specifying CRF model forms, feature extractors, com-
position pipelines, feature induction schemas, and capacity control methods.
• Create a Web-based bug reporting, announcement, and project discussion site. Depending on the
requirements and available facilities, this may be at a University of Pennsylvania site or at a public
open source development site like SourceForge.
• Identify and enlist two early adopter user groups outside the project team.
• Document and disseminate release 0.1.

Year 2

• Collect detailed feedback from internal users and external early adopter groups to prioritize improve-
ments and extensions.
• Implement basic kernels, kernel CRFs, and maximum-margin kernel training.
• Design and implement a global training procedure for composed models.
• Design and implement approximate inference and training procedures for models with loopy structures, in particular factorial models.
• Design and implement a training algorithm that can take advantage of unlabeled data in both batch and active learning settings.
• Extend scripting layer to support the representations and algorithms developed in Year 2.
• With the help of the user community, develop, document and disseminate at least three comprehensive case studies on model specification, training data selection, capacity control method selection, and model evaluation.
• Identify and enlist a third external early adopter group.
• Document and release version 0.5.

Year 3

• Collect detailed feedback from user community to identify usability and performance issues and prioritize final year improvements.
• Design and implement graphical user interface for simplified model specification, training and evaluation workflow, and result display.
• Extend training procedures to suitable families of admissible loss functions, and provide convenient scriptable means of specifying and checking loss functions.
• Extend inference and training algorithms to two-level graphs representing individual sequences and their similarity links, supporting learning from collections of partially labeled sequences.
• Document and release version 1.0.
• Conduct final project workshop with external user and project member presentations, including planning for ongoing development, application, and dissemination of the project software.

6.2 Evaluation

Both objective performance measures and more subjective user satisfaction measures will be used:

• To verify correctness and basic functionality, develop and periodically run unit and regression tests.
• Track accuracy and efficiency improvements with benchmark tasks on standard data sets.
• Track accuracy and efficiency improvements on user data sets as the software is updated.
• Survey users periodically about main bottlenecks to more effective and broader usage.
• Measure bug fix turnaround from bug tracker.
• Measure software downloads and traffic in discussion groups.
• Track publication of results achieved with the proposed software.

6.3 Preservation, Documentation, Dissemination

All of the software, documentation, technical papers, and discussions will be maintained on a project Web server backed up using the best practices, including offsite media storage, in use at the University of Pennsylvania. The repository will also be replicated at least one other project university, expected currently to be the University of Massachusetts. After project completion, we will seek continuing resources to ensure that the project software and documentation continue to be maintained and publicly available.
7 Coordination Plan

7.1 Roles of the Principal Investigators

Prior to this project, the PIs have collaborated closely on several projects. They developed the CRF framework jointly. Lafferty and Pereira are co-PIs on an NSF-funded project on biological sequence analysis. McCallum and Pereira collaborate on text information extraction and the development of the MALLET toolkit (partially funded by DARPA). The PIs will work closely on the main scientific and practical goals of the project, but existing collaborations and individual expertise suggest the following broad division of labor:

- Lafferty will direct outreach efforts to users in imaging and biological sequence analysis. He will direct research activities concerning inference and kernel methods for graph structured data, and software implementation of learning algorithms.

- McCallum will direct coordination and integration with the MALLET toolkit. He will direct research activities concerning capacity control methods and structure learning. McCallum will also direct outreach efforts to users in science and national security.

- Pereira will manage the main software, documentation, and code repository at the University of Pennsylvania, where the project’s programmer will be located. He will direct research efforts on model composition, scripting front-end, and software architecture. He will also direct outreach efforts to users in medicine and biology not otherwise supported by Lafferty or McCallum.

7.2 Coordination Mechanisms

The project will rely on the following coordination tools:

- Shared software, documentation, and data repository with source code control and versioning, discussion groups, and bug tracker.

- Project programmer for coordinating bug tracking, software updates, major releases, and general Web-mediated project communication and information dissemination. Included in the U. of Pennsylvania budget.

- Conferencing software for monthly PI meeting.

- Graduate student visits to other sites lasting from a few weeks to a semester, depending on project details and availability. Included in the travel budget items for the three collaborating institutions.

- Project meetings at the main conferences in the field (NIPS, ICML, UAI).

- Annual joint meeting with project members and users. Partially covered by the collaborating institutions’ travel budget items.
References


Project Personnel and Collaborators

Project Personnel


Collaborators

Steve Abney, U. of Michigan
Andrew Bagnell, CMU
William Bialek, Princeton University
David Blei, Berkeley
Avrim Blum, CMU
Rich Caruana, Cornell
Huan Chang, unaffiliated
William Cohen, CMU
David Cohn, Google
Bruce Croft, U. Mass, Amherst
Ido Dagan, Bar-Ilan University
Mary Dalrymple, Kings College, London
Dayne Freitag, Burning Glass Technologies
Alan Frieze, CMU
Alex Gamburd, Stanford
Zoubin Ghahramani, Gatsby Institute
Donald Hindle, AnswerLogic
Julia Hirschberg, Columbia University
Thomas Hofmann, Brown
John Lamping, Google
Lillian Lee, Cornell University
Andrej Ljolje, AT&T Labs
David McAllester, Toyota Technological Institute
Tom Mitchell, CMU
Mehryar Mohri, AT&T Labs
Andrew Ng, Stanford
Kamal Nigam, Intelliseek
Dallan Quass, BYU
Jason Rennie, MIT
Michael Riley, Google
Roni Rosenfeld, CMU
Dan Rockmore, Dartmouth
Nicholas Roy, CMU
Vijay Saraswat, Penn State University
Kristie Seymore, Panscient
Yoram Singer, Hebrew University
Amit Singhal, Google
Naftali Tishby, Hebrew University
Lyle Ungar, U. Pennsylvania
Paul Viola, Microsoft
Steve Whittaker, University of Sheffield
Cheng Zhai, University of Illinois, Urbana-Champaign