
Heterogeneous Hidden Conditional Random Fields for Fixed-Length Sequence Classification

Keywords: [TODO: Keywords]

Abstract

For the problem of classifying fixed-length noisy observation sequences, we build an integrated model that combines sequence-wide and position-specific features to cooperatively denoise observations by sequential correlation and learn a sparse set of sequence positions that are significant in predicting the sequence class. We provide an efficient L_1 -regularized discriminative training algorithm for the model. Our experiments on synthetic data and real genomic cancer prediction data show that our method is superior, both in prediction accuracy and relevant feature discovery, to the common practice of preprocessing with a purely homogeneous sequence model and then learning with a purely non-sequential algorithm.

1. Introduction

For sequence classification problems, where each example has a large number of sequentially-correlated noisy inputs and one classification label for the whole sequence, models developed for speech or text tasks, such as Hidden Markov Models, generally learn a small number of position-invariant parameters to reduce model complexity and to accommodate the variable-length aspect of such data. In scenarios where the sequence length is fixed and position in the sequence is known to be significant for classification, these assumptions lead to a loss of information. Conversely, applying a conventional non-sequential machine learning algorithm directly on the noisy input features ignores the sequential information and is likely to suffer from overfitting.

One such scenario arises in predicting cancer survival

by genomic microarray measurement profiles. Each sequence represents a patient's genome, with input features obtained by high-throughput microarray probes at fixed genome positions. The number of probes is fixed, and individual probes are highly correlated due to probe overlap and genetic factors. For many cancers, disease development is believed to be associated with certain genes (at certain probe positions), but attempting to classify such sequences by a homogeneous sequence model can only capture genome-wide properties. Conversely, using the probe measurements directly for classification (e.g. with a support vector machine) is infeasible because probe measurements are highly noisy and also the training set size is typically much smaller than the number of features (typically tens versus thousands). Even though model complexity can be limited by regularization, it will be arbitrary compared to the sequential information already lost. Therefore, current common practice is to use a two-step process with such data, first de-noising using a simple sequence model [TODO: e.g. cite BioHMM, ChARM], and then applying a non-sequential classification algorithm on the de-noised features.

In this work, we describe a method that combines the sequential de-noising and the supervised classification aspects in one integrated architecture, so that they can cooperatively learn a better overall model, without loss of relevant information to either. We provide an efficient, regularized training algorithm that finds a sparse, interpretable solution.

2. Previous Work

[TODO: Talk about structured methods, how they are different]

3. Method

We uphold the hidden Markov assumption that the observed features are noisy representations of a set of correlated latent variables. In breast cancer classification by array CGH microarray profiles, for instance, this is

known to be true: the measured real values (microarray color intensity log-ratios) correlate linearly (albeit very noisily) to how many copies of a genes are present in the DNA, which is two copies for most genes in normal cells, but breast cancer cells often have continuous regions of copy number alterations. We assume the sequence label to be effected by a subset of the latent variables (which we want to discover as part of the learning process; e.g. oncogenes and tumor suppressors in cancer), as opposed to being directly related to the noisy observations. This conditional independence of the observed sequence and the label given all the latent variables is illustrated in Figure 1.

For training example $t \in 1, \dots, T$, let s^t be the sequence label, let x_i^t denote the observation and c_i^t the latent variable at position $i \in 1, \dots, N$.

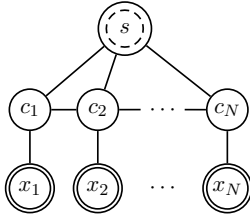


Figure 1. The Hidden Conditional Random Field model. The variables x_i are observed, c_i are hidden, and the sequence label s is only observed during training. An exponential model for $p(s, \mathbf{c}|\mathbf{x})$ is tuned to maximize the class-conditional likelihood $p(s|\mathbf{x})$ of training data.

Given the observations \mathbf{x} for an example, we use an exponential model for the conditional probability of the other variables:

$$p_{\theta}(s, \mathbf{c}|\mathbf{x}) = \frac{1}{Z_{\theta}(\mathbf{x})} \exp(\boldsymbol{\theta} \cdot \mathbf{f}(s, \mathbf{c}, \mathbf{x})) \quad (1)$$

where $Z_{\theta}(\mathbf{x})$ is a normalization factor, $\boldsymbol{\theta} = (\boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\omega})$ are the model parameters, and \mathbf{f} is a set of features, such that:

$$\begin{aligned} \boldsymbol{\theta} \cdot \mathbf{f}(s, \mathbf{c}, \mathbf{x}) = & \boldsymbol{\rho} \cdot \sum_{i=2}^N \mathbf{f}_{pair}(c_{i-1}, c_i, s) \\ & + \sum_{i=1}^N \boldsymbol{\lambda}_i \cdot \mathbf{f}_{local}(c_i, s) \\ & + \boldsymbol{\omega} \cdot \sum_{i=1}^N \mathbf{f}_{obs}(c_i, x_i) \end{aligned} \quad (2)$$

The pairwise features \mathbf{f}_{pair} and the corresponding parameters $\boldsymbol{\rho}$ model the sequence-wide correlation of adjacent nodes for each class, the local features \mathbf{f}_{local}

and their parameters $\boldsymbol{\lambda}_i$ model the correlation of latent variable c_i and the label s , and the observation features \mathbf{f}_{obs} and their parameters $\boldsymbol{\omega}$ model the correlation of latent variable c_i and its noisy observation x_i .

For discrete latent variables and class label, the feature functions \mathbf{f}_{pair} and \mathbf{f}_{local} are typically defined to be 1 for a particular combination and 0 otherwise. The pairwise parameters $\boldsymbol{\rho}$ then correspond to (unnormalized) log-probabilities of a homogeneous HMM's hidden state transitions. For real valued observations, $\mathbf{f}(c, x)$ can be defined as $(1, x, x^2)$ if $c = c'$ (and zeros otherwise) for each latent variable value c' , the sufficient statistics for Gaussian distributions.

The position-dependent local parameters, which make the model heterogeneous, allow the model to interpolate between a homogeneous sequence-wide hypothesis and one that ignores correlations. If all local parameters are made zero, the model is a fully homogeneous random field, and classification only depends on sequence-wide stability of latent state. Conversely, if they are unconstrained and allowed to overpower the pairwise component, classification will depend almost fully on them, and the model will be akin to logistic regression. In our model, we constrain the L_1 norm of the local parameters $\boldsymbol{\lambda}$ to adjust this tradeoff, which also encourages sparsity and results in an interpretable solution.

3.1. Training

The model is trained discriminatively, minimizing the conditional negative log-likelihood of labels over the empirical distribution $\tilde{p}(\mathbf{x}, s)$ of the training data:

$$L_{\theta} = - \sum_{\mathbf{x}, s} \tilde{p}(\mathbf{x}, s) \log p_{\theta}(s|\mathbf{x}) \quad (3)$$

subject to the regularization constraint $\|\boldsymbol{\lambda}\|_1 \leq \beta$.

We use a gradient-based procedure to solve the optimization problem. The partial derivative of the objective loss with respect to any parameter θ_k is:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_k} = & \sum_s \tilde{p}(s) \sum_{\mathbf{c}} p_{\theta}(s, \mathbf{c}|\mathbf{x}) f_k \\ & - \sum_{\mathbf{x}, s} \tilde{p}(\mathbf{x}, s) \sum_{\mathbf{c}} p_{\theta}(c|\mathbf{x}, s) f_k \\ = & \mathbb{E}_{\tilde{p}(s)p_{\theta}(s, \mathbf{c}|\mathbf{x})}[f_k] - \mathbb{E}_{\tilde{p}(\mathbf{x}, s)p_{\theta}(c|\mathbf{x}, s)}[f_k] \end{aligned} \quad (4)$$

Although $p_{\theta}(s|\mathbf{x})$ in (3) and the expectations in (4) call for marginalizing $p_{\theta}(s, \mathbf{c}|\mathbf{x})$ (1) over the exponentially

many value combinations of the latent variables \mathbf{c} , a dynamic programming solution exists, similar to the forward-backward procedure for HMMs, scaling linearly with sequence length.

3.2. Gradient LASSO

To satisfy the regularization constraint $\|\boldsymbol{\lambda}\|_1 \leq \beta$, we incorporate the Gradient LASSO algorithm (Kim & Kim, 2004), with a minor modification.

Gradient LASSO is an interior point method for optimizing a differentiable function subject to L_1 constraints. It maintains an explicitly sparse current solution, alternating between a coordinatewise gradient step, which may add a new non-zero parameter, and a multivariate gradient step over the non-zero parameters, which may make one of them zero. The constraints are always kept satisfied, by starting inside the constraint simplex and bounding step sizes. When the current parameters satisfy the constraint by equality and local gradient descent is about to violate it, the gradient is projected onto the boundary, and linearity of L_1 constraint boundaries make line search along the boundary possible.

Our version of Gradient LASSO (summarized in Algorithm 1) differs slightly from the original presented by Kim and Kim (2004); in the deletion step, if the current solution is not on the constraint boundary, we use a less conservative maximum step size Δ to accelerate learning.

3.3. Unconstrained Parameters

The unregularized parameters our model $(\boldsymbol{\rho}, \boldsymbol{\omega})$ are optimized after each two-step Gradient LASSO iteration, using the gradient-based L -BFGS algorithm (Nocedal, 1980), a limited-memory quasi-Newton method for unconstrained optimization, while the regularized parameters $\boldsymbol{\lambda}$ are kept unchanged.

Note that the unconstrained optimization step causes the constrained problem objective $L(\boldsymbol{\lambda})$ to change between iterations, and therefore the optimality of its current solution. The two-step Gradient LASSO algorithm, by adding newly relevant features and deleting obsolete features as necessary, is able to robustly cope with this concept drift without compromising sparsity, which would not have been possible with strictly growing or shrinking algorithms.

[subsection: include gradient lasso, describe my modification, give triple-iteration details]

Algorithm 1 Gradient LASSO

Objective: $\min L(\boldsymbol{\lambda})$ s.t. $\|\boldsymbol{\lambda}\| \leq \beta$

repeat

Addition step:

 Compute gradient $\nabla = (\partial L / \partial \lambda_1, \dots, \partial L / \partial \lambda_d)$

 Choose coordinate $k = \arg \max_i |\nabla_i|$

$h_k = -\beta \text{sgn}(\nabla_k)$; $h_i = 0$ for all $i \neq k$

$\hat{\alpha} = \arg \min_{\alpha \in [0, 1]} L((1 - \alpha)\boldsymbol{\lambda} + \alpha \mathbf{h})$

$\boldsymbol{\lambda} \leftarrow (1 - \hat{\alpha})\boldsymbol{\lambda} + \hat{\alpha} \mathbf{h}$

Deletion step:

 Compute gradient $\nabla = (\partial L / \partial \lambda_1, \dots, \partial L / \partial \lambda_d)$

 Let $\boldsymbol{\sigma} = \{i : \lambda_i \neq 0\}$

 Let $p = \langle \nabla, \mathbf{s} \rangle$ where $s_i = \text{sgn}(\lambda_i)$

$h_{j \notin \boldsymbol{\sigma}} = 0$

$h_{j \in \boldsymbol{\sigma}} = \begin{cases} -\nabla_j + ps_j / |\lambda_j| & \text{if } p < 0 \text{ and } \|\boldsymbol{\lambda}\|_1 = \beta \\ -\nabla_j & \text{otherwise} \end{cases}$

$\Delta = \begin{cases} \min_{j \in \boldsymbol{\sigma}} \{-\lambda_j / h_j : \lambda_j h_j < 0\} & \text{if } \|\boldsymbol{\lambda}\|_1 = \beta \\ (\beta - \|\boldsymbol{\lambda}\|_1) / \|\mathbf{h}\|_1 & \text{if } \|\boldsymbol{\lambda}\|_1 < \beta \end{cases}$

$\hat{\alpha} = \arg \min_{\alpha \in [0, \Delta]} L(\boldsymbol{\lambda} + \alpha \mathbf{h})$

$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \hat{\alpha} \mathbf{h}$

until converged

4. Experimental Results

5. Discussion

[Future work includes generative model]

References

- Kim, Y., & Kim, J. (2004). Gradient lasso for feature selection. *ICML '04: Proceedings of the 21st International Conference on Machine Learning* (p. 60). New York, NY, USA: ACM.
- Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35, 773–782.