Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation

J. Zico Kolter
MIT CSAIL
kolter@csail.mit.edu

Tommi Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Abstract

This paper considers additive factorial hidden Markov models, an extension to HMMs where the state factors into multiple independent chains, and the output is an *additive* function of all the hidden states. Although such models are very powerful, accurate inference is unfortunately difficult: exact inference is not computationally tractable, and existing approximate inference techniques are highly susceptible to local optima. In this paper we propose an alternative inference method for such models, which exploits their additive structure by 1) looking at the observed difference signal of the observation, 2) incorporating a "robust" mixture component that can account for unmodeled observations, and 3) constraining the posterior to allow at most one hidden state to change at a time. Combining these elements we develop a convex formulation of approximate inference that is computationally efficient, has no issues of local optima, and which performs much better than existing approaches in practice. The method is motivated by the problem of *energy disaggregation*, the task of taking a whole home electricity signal and decomposing it into its component appliances; applied to this task, our algorithm achieves state-of-the-art performance, and is able to separate many appliances almost perfectly using just the total aggregate signal.

1 Introduction and Background

Factorial hidden Markov models (FHMMs) (Ghahramani and Jordan, 1997) are an extension of the basic hidden Markov model where several HMMs evolve independently in parallel, and the observed output is some joint function of all the hidden states. Due to their ability to capture complex aggregate signals via a compact structure, such models have found applications in areas as speech recognition (Virtanen, 2006), audio separation (Roweis, 2001), and pitch tacking (Bach and Jordan, 2005). However, despite their utility, accurate inference is difficult: exact inference requires enumerating an exponential number of states, and common approximate algorithms, (block) Gibbs sampling (Kim et al., 2011) or structured mean field methods (Ghahramani and Jordan, 1997), are highly susceptible to local optima in the likelihood function.

In this paper we consider the special case of *additive* FHMMs, where each HMM emits an (unobserved) real-valued output, and we observe the sum of these outputs.[1] The key algorithmic contribution of this paper is a method for exploiting the additive structure of the FHMM to develop an approximate inference procedure that vastly outperforms existing approaches. Our method considers models that represent both the total aggregate output and the difference between successive outputs, and also introduces a "robust" mixture component that can represent arbitrary additional signals (but with an ℓ_1-based regularity condition). By constraining the posterior to require that only one HMM change state any any given time, we develop an efficient convex quadratic programming relaxation of the inference problem. The resulting method works extremely well in practice, is computationally efficient (scales roughly linearly in the number of HMMs), can handle non-IID additive noise, and is free from local optima in the optimization procedure.

We focus on the application of electrical energy disaggregation, also called non-intrusive load monitoring (Hart, 1992), the task of taking a whole-home energy signal and breaking it down into its component appli-

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

[1] The original FHMM model (Ghahramani and Jordan, 1997) was of this form, but since many other aggregation functions have since been used and referred to as FHMMs, we explicitly denote this case the additive FHMM.

ances. Studies have shown that just presenting such a breakdown to users, so that a home owner can see precisely how much energy is being used by which appliance, can automatically lead to energy-saving behavior (Darby, 2006; Neenan and Robinson, 2009). Given the large percentage of energy that residential and commercial buildings consume, enabling such energy-conserving behavior is a key element to addressing energy sustainability problems. Although there has been much work on energy disaggregation (see e.g. (Ziefman and Roth, 2011) for a recent survey) it has largely focused simply upon classifying electrical events; early work looked only at power signals along with finite state machine models, (Hart, 1992), while later work has incorporated transient and harmonic information from very high-frequency sampling (Laughman et al., 2003; Gupta et al., 2010; Berges et al., 2010). Recent work has begun to look at more complex inference procedures for the actual disaggregation task, including FHMMs (with block Gibbs sampling) (Kim et al., 2011) and sparse coding methods (Kolter et al., 2010).

In comparison to past work, this paper makes several contributions from an application standpoint. First, unlike all past work we are aware of other than (Kim et al., 2011), we consider the *unsupervised* setting, where we do not have a priori knowledge of the appliances in the home; the inference procedure we develop, however, enables us to use much more complex device models than this past work. Our method effectively joins the two prior threads of work in energy disaggregation: we are able to capture general device states *and* capture the information in transient power signature, using a single unified model. Finally, our inference method significantly outperforms alternative inference methods for this task, and for several devices is able to achieve essentially perfect separation.

2 The Additive Factorial Model

The basic FHMM consists of several independent HMMs evolving in parallel, with the observation being a joint function of all hidden states; a graphical model representation is shown in Figure 1. Exact inference typically is not tractable in such a model, so approximate inference procedures are needed, such as block Gibbs sampling or structured mean field methods. In this paper we are interested in the special case of the FHMM where the output is an additive function of the different hidden states as in (Ghahramani and Jordan, 1997). The factors in the additive FHMM are

$$\begin{aligned} x_1^{(i)} &\sim \text{Mult}(\phi^{(i)}) \\ x_t^{(i)}|x_{t-1}^{(i)} &\sim \text{Mult}(P^{(i)}_{x_{t-1}^{(i)}}) \\ \bar{y}_t|x_t^{(1:N)} &\sim \mathcal{N}\left(\sum_{i=1}^{N} \mu^{(i)}_{x_t^{(i)}}, \Sigma\right) \end{aligned} \tag{1}$$

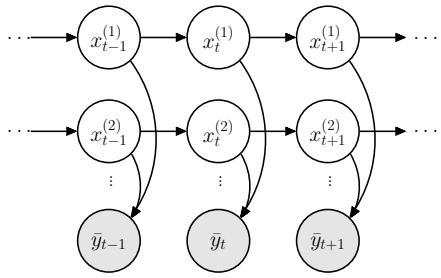

Figure 1: The FHMM model.

where $N \in \mathbb{Z}_+$ is the number of HMMs; $m_i \in \mathbb{Z}_+$ is the number of states in the ith HMM; $T \in \mathbb{Z}_+$ is the number of time steps; $x_t^{(i)} \in \{1, \ldots, m_i\}$ denotes the state of the ith HMM at time t; $\bar{y}_t \in \mathbb{R}^n$ is the observed aggregate output; $\mu_j^{(i)} \in \mathbb{R}^n$ is the mean of the ith HMM for state j; $\Sigma \in \mathbb{R}^{n \times n}$ is the observation variance; $\phi^{(i)} \in [0,1]^{m_i}$ is the initial state distribution for ith HMM; and $P^{(i)} \in [0,1]^{m_i \times m_i}$ is the transition matrix for ith HMM.

Although the additive model is a special case of the general FHMM, exact inference is still not tractable in the model, and standard mean-field methods are susceptible to local optima. To address these issues, with an eye towards formulating a convex approximate inference procedure, we propose two additions to the model that guide the posterior.

2.1 The Difference FHMM Model

In the additive FHMM above, it is natural to consider *differences* in the output signal: intuitively, in the case that only one HMM changes state at a given time, these differences correspond precisely to the state change of this one HMM and can thus help us to infer the hidden states. We will formalize this intuition in the next section, but this has been a common heuristic in algorithms for energy disaggregation: looking for sharp changes in the power signal and classifying these as a single device change. We encode these difference observations directly using a slight modification of the FHMM, which we refer to as the *difference FHMM*. The model is shown in Figure 2; the the $x_t^{(i)}$ variables and its factors are the same as for the FHMM, but the output at time t, denoted $\Delta\bar{y}_t$, is given by

$$\Delta\bar{y}_t|x_t^{(1:N)}, x_{t-1}^{(1:N)} \sim \mathcal{N}\left(\sum_{i=1}^{N}\left(\mu_{x_t^{(i)}} - \mu_{x_{t-1}^{(i)}}\right), \Sigma\right) \tag{2}$$

where we use the same parameters as for the additive FHMM described earlier (we will hereafter use the abbreviation $\Delta\mu_{j,k} = \mu_j - \mu_k$).

The additive FHMM and the difference FHMM defined here have complementary advantages and disadvantages. The additive FHMM captures well the total

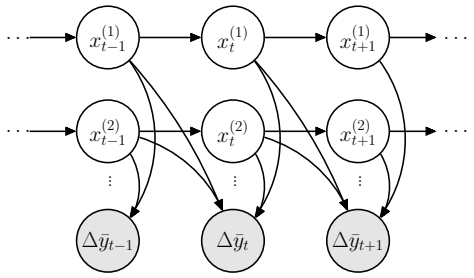

Figure 2: The difference factorial HMM model.

aggregate output of the system, but does not directly include information about the signal differences. In contrast the difference FHMM directly encodes these difference observations, but fails to model the total aggregate output, so that errors in the cumulative sum of this model can build up over time. Our final inference procedure will thus ultimately use both models, but will constrain the posteriors over the hidden states to agree between them.

2.2 Robust Mixture Components

The Gaussian likelihood term in the FHMM (both the additive and difference versions) makes the model sensitive to outliers. This is an especially troublesome problem for the energy disaggregation task, as there may be new devices or rarely used devices for which we have no model. We thus introduce a robust version of the additive mixture model, which includes a generic component that can take on arbitrary values, but which obeys certain regularity conditions. A particularly suitable choice of regularizer here is (1D) *Total Variation (TV) Regularization* (Rudin et al., 1992), a penalty on the ℓ_1 norm of the signal differences (or probabilistically, a Laplace prior on the differences); this prior encourages the generic mixture component to take on piece-wise constant values, which intuitively captures the typical nature of many devices.

To add this generic mixture component to the additive FHMM model, we extend (1) by introducing a signal $z_t \in \mathbb{R}^n$, with the prior

$$p(z_{1:T}) = \frac{1}{Z(\lambda, T)} \exp\left\{-\lambda \sum_{t=1}^{T-1} \|z_{t+1} - z_{t-1}\|_1\right\} \quad (3)$$

and modify the likelihood of the aggregate output to be

$$\bar{y}_t | x_t^{(1:N)}, z_t \sim \mathcal{N}(\sum_{i=1}^{N} \mu_{x_t^{(i)}}^{(i)} + \Sigma^{1/2} z_t, \Sigma) \quad (4)$$

where we scale z_t by $\Sigma^{1/2}$ because we are using a single penalty λ for all the dimensions of z_t, and we want to make its units commensurate with those of $\bar{y}_t$.

A similar robust version of the difference FHMM is possible, and is even simpler in this case, as the model directly represents the difference signal. We now introduce a signal $\Delta z_t \in \mathbb{R}^n$, with prior

$$p(\Delta z_{1:T}) = \frac{1}{Z(\lambda, T)} \exp\left\{-\lambda \sum_{t=1}^{T} \|\Delta z_t\|_1\right\} \quad (5)$$

and modify the likelihood of $\Delta\bar{y}_t$ to be

$$\Delta\bar{y}_t | x_t^{(1:N)}, x_{t-1}^{(1:N)}, \Delta z_t \sim \mathcal{N}\left(\sum_{i=1}^{N} \Delta\mu_{x_t^{(i)}, x_{t-1}^{(i)}} + \Sigma^{1/2} \Delta z_t, \Sigma\right) \quad (6)$$

For intermediate values of λ (i.e.,a λ must be large enough so that the model does not assign all the output to z_t but small enough so that $z_{1:T}$ is not a constant-valued), these additional mixture components are able to explain elements of the signal that are not well modeled by any of the HMMs.

3 Approximate MAP Inference

For the FHMM model and the proposed extensions, exact inference remains a computationally expensive procedure: we know of no way to perform exact inference short of enumerating all the states. Thus, in this section, we focus on methods for approximate inference, and approximate MAP inference in particular. By exploiting the additive structure of the FHMM, using both the additive and difference formulations, and by constraining the posterior, we develop a method that is computationally efficient, free of local optima, robust to unknown components in the signal, and which in practice works much better than existing approaches.

3.1 Exact MAP Inference

We begin by considering optimization approaches to exact MAP inference. To define notation, we perform optimization over the variables

$$\mathcal{Q} = \left\{Q(x_t^{(i)}) \in \mathbb{R}^{m_i}, Q(x_{t-1}^{(i)}, x_t^{(i)}) \in \mathbb{R}^{m_i \times m_i}\right\}. \quad (7)$$

These variables (in the exact MAP case) are indicators, so that $Q(x_t^{(i)})_j = 1 \Leftrightarrow x_t^{(i)} = j$ and $Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} = 1 \Leftrightarrow x_{t-1}^{(i)} = j, x_t^{(i)} = k$. In the following, all summations or constraints over t are assumed to range from $t = 1, \ldots, T$ (or $2, \ldots, T$, as appropriate), over i are assumed to range from $i = 1, \ldots, N$, and over j, k are assumed to range from $j, k = 1, \ldots, m_i$. For the robust mixture components, we also optimize over the variables $z_t, \Delta z_t \in \mathbb{R}^n$ for the additive and difference formulations respectively.

The first constraint on $\mathcal{Q}$ is that it must be locally consistent and must represent a valid distribution locally, (i.e., it must lie within the *local marginal polytope* (Wainwright and Jordan, 2008)) which for this

problem takes the form

$$\mathcal{L} = \left\{ \mathcal{Q} : \begin{array}{l} \sum_j Q(x_t^{(i)})_j = 1 \\ \sum_k Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} = Q(x_{t-1}^{(i)})_j \\ \sum_k Q(x_{t-1}^{(i)}, x_t^{(i)})_{k,j} = Q(x_t^{(i)})_j \\ Q(x_t^{(i)})_j,\ Q(x_{t-1}^{(i)}, x_t^{(i)})_{k,j} \geq 0 \end{array} \right\} \quad (8)$$

MAP inference can now be cast as optimizing the log-likelihood of the model, subject to the constraint that $\mathcal{Q} \in \mathcal{L}$ and that all the variables in $\mathcal{Q}$ take on binary values (a constraint which we abbreviate $\mathcal{Q} \in \{0,1\}$). Due to the Gaussian likelihood term, these problems take the form of mixed-integer quadratic programs (MIQPs). For the additive FHMM model, this leads to the optimization problem

$$\begin{aligned} &\textbf{minimize over } \{\mathcal{Q} \in \mathcal{L} \cap \{0,1\},\ z_{1:T}\}, \\ &\frac{1}{2}\sum_t \left\| \bar{y}_t - \Sigma^{1/2} z_t - \sum_{i,j} \mu_j^{(i)} Q(x_t^{(i)})_j \right\|_{\Sigma^{-1}}^2 \\ &+ \sum_{t,i,j,k} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} (-\log P_{k,j}^{(i)}) \\ &+ \lambda \sum_t \|z_t - z_{t-1}\|_1. \end{aligned} \quad (9)$$

Similarly, exact MAP inference for the difference FHMM model is given by the MIQP

$$\begin{aligned} &\textbf{minimize over } \{\mathcal{Q} \in \mathcal{L} \cap \{0,1\},\ \Delta z_{2:T}\}, \\ &\frac{1}{2}\sum_t \left\| \Delta\bar{y}_t - \Sigma^{1/2}\Delta z_t - \sum_{i,j,k} \Delta\mu_{k,j}^{(i)} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} \right\|_{\Sigma^{-1}}^2 \\ &+ \sum_{t,i,j,k} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} (-\log P_{k,j}^{(i)}) + \lambda \sum_t \|\Delta z_t\|_1. \end{aligned} \quad (10)$$

Naturally, solving these problems exactly is difficult, as they involve non-convex integer constraints, and so some form of relaxation is necessary to obtain a tractable formulation. A typical method for converting binary integer problems such as (9) and (10) into tractable optimization problems is to drop the constraint that $\mathcal{Q}$ take on integer values, and solve the resulting, now convex, quadratic programs (this is usually stated as relaxing that constraint that the variables be in $\{0,1\}$ to the constraint that they be in $[0,1]$, but the later is already implied by the local marginal polytope in our case). Unfortunately, the quadratic term in the optimization problems can make this relaxation perform poorly in practice: this term typically encourages the $Q(x_t^{(i)})$ variables to take on non-integral values, and the resulting solution can have little correspondence to the MAP solution. However, as we show in the next section, by constraining the posterior in the difference FHMM formulation, we can transform (10) to a mixed-integer *linear* program, which is much more amenable to convex relaxation.[2]

3.2 The One-at-a-time Condition and LP Relaxations

As described intuitively above, a natural condition to impose upon the posterior distribution over states is that at most one HMM changes state at any given time, a requirement we refer to as the one-at-a-time condition. Indeed, if we consider each HMM in our model to be a sufficiently discretized approximation to a continuous time HMM, then this condition holds with high probability.[3] Formally,

Proposition 1. *For each HMM i, let $P^{(i)}$ be the discretization of a continuous time Markov chain with incidence matrix $Q^{(i)} \in \mathbb{R}^{m_i \times m_i}$ and discretization interval $\Delta t \in \mathbb{R}_+$. For total time $t_f \in \mathbb{R}_+$, this results in an FHMM with $t_f/\Delta t$ time steps, and we let A denote the event that there is some time where two HMMs change state simultaneously*

$$A = \{\exists t, i \neq j : x_t^{(i)} \neq x_{t+1}^{(i)}, x_t^{(j)} \neq x_{t+1}^{(j)}\}. \quad (11)$$

Then

$$p(A) \leq O(\Delta t). \quad (12)$$

The proof and definitions for a continuous time Markov process are in Appendix A. Note the one-at-a-time condition is not imposed directly by modifying the FHMM model (for example, by adding a latent variable that determines which of the HMMs change); this would create an undesired dependence between all the hidden states in subsequent time steps and complicate inference. Rather this condition is imposed as a constraint on the class of allowable posterior distributions, similar in spirit to the *posterior regularization* framework (Ganchev et al., 2010), except that we are looking only at the inference task rather than a joint EM task, and the specific constraint we consider here is of course quite different.

The set of distributions that obey the one-at-a-time condition can be expressed formally as the constraint that the off-diagonal elements of $Q(x_{t-1}^{(1:N)}, x_t^{(1:N)})$ sum

[2] The same can be done in principle for the additive model, but would require us to introduce exponentially-many marginal variables.

[3] We could alternatively use a continuous time FHMM model, where we assume that observations are generated (at least) whenever one of HMMs changes state; in this case the one-at-a-time condition holds with probability one. We use the discrete-time formulation in this paper for simplicity of the presentation.

to at most one. That is,

$$\mathcal{O} = \left\{ \mathcal{Q} : \sum_{i,j,k \neq j} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} \leq 1 \right\}. \tag{13}$$

Imposing this constraint implies that at most *one* of the $\Delta\mu_{k,j}^{(i)} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k}$ terms in each norm in (10) is non-zero; further, if exactly one of these terms is non-zero then $\Delta z_t = 0$, as we enforce the one-at-a-time condition over the generic mixture component as well, and so the quadratic term in (10) can be replaced by a linear term. Alternatively, if *none* of the off-diagonal terms are non-zero then the quadratic term is just $\|\Delta\bar{y}_t - \Sigma^{1/2}\Delta z_t\|_{\Sigma^{-1}}^2$, which (together with the $\lambda\|\Delta z_t\|_1$ term) can be optimized analytically to give the *Huber loss function*

$$\begin{aligned} D(y, \lambda) &= \min_z \{\|y - z\|_2^2 + \lambda\|z\|_1\} \\ &= \sum_{\ell=1}^{n} \min\left\{ \frac{1}{2} y_\ell^2, \max\left\{ \lambda|y_\ell| - \frac{\lambda^2}{2}, \frac{\lambda^2}{2} \right\} \right\} \end{aligned} \tag{14}$$

Thus, exact MAP inference for the difference FHMM model (for the posterior constrained to the set $\mathcal{O}$) can be written as the mixed-integer linear program (see Appendix B for the full derivation)

$$\begin{aligned} &\textbf{minimize over } \{\mathcal{Q} \in \mathcal{L} \cap \mathcal{O} \cap \{0,1\}\}, \\ &\frac{1}{2} \sum_{t,i,j,k \neq j} \left\| \Delta\bar{y}_t - \Delta\mu_{k,j}^{(i)} \right\|_{\Sigma^{-1}}^2 Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} \\ &+ \frac{1}{2} \sum_t D(\Sigma^{-1/2}\Delta\bar{y}_t, \lambda) \left(1 - \sum_{i,j} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,j} \right) \\ &+ \sum_{t,i,j,k} Q(x_t^{(i)}, x_{t+1}^{(i)})_{j,k} (-\log P_{k,j}^{(i)}) \end{aligned} \tag{15}$$

Dropping the integral constraint in (15) leads to a convex LP. While solutions to the LP need not be integer valued, in practice they often are: the objective term and simplex constraints in (15) alone result in separable optimization problems that would always have integer solution, and the only constraints that may cause non-integral solutions are the consistency terms in the local marginal polytope, which typically leads to a very small number of non-integral assignments.

3.3 Joint Approximate Inference

At this point we have two potential models for performing inference which capture different elements of the signal: the total aggregate output and the difference signal. To exploit the benefits of each model we combine these optimization problems (after relaxing the integer constraint) into a single joint problem;

Input: $\bar{y}_{1:T} \in \mathbb{R}^n$, aggregate output signal; $\mu_{1:m_i}^{(1:N)} \in \mathbb{R}^n$, state means for N HMMs; $\Sigma_1, \Sigma_2 \in \mathbb{R}^{n \times n}$, $\lambda_1, \lambda_2 \in \mathbb{R}_+$, covariance and regularization parameters

minimize over $\{\mathcal{Q} \in \mathcal{L} \cap \mathcal{O}, z_{1:T}\}$

$$\begin{aligned} &\frac{1}{2} \sum_t \left\| \bar{y}_t - \Sigma_1^{1/2} z_t - \sum_{i,j} \mu_j^{(i)} Q(x_t^{(i)})_j \right\|_{\Sigma_1^{-1}}^2 \\ &+ \frac{1}{2} \sum_{t,i,j,k \neq j} \left\| \Delta\bar{y}_t - \Delta\mu_{k,j}^{(i)} \right\|_{\Sigma_2^{-1}}^2 Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} \\ &+ \frac{1}{2} \sum_t D(\Sigma_2^{-1/2}\Delta\bar{y}_t, \lambda_2) \left(1 - \sum_{i,j} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,j} \right) \\ &+ \sum_{t,i,j,k} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} (-\log P_{k,j}^{(i)}) \\ &+ \lambda_1 \sum_t \|z_t - z_{t-1}\|_1 \end{aligned}$$

Output: $\hat{y}_{1:T}^{(1:N)}$, predicted individual HMM output

$$\hat{y}_t^{(i)} = \sum_j \mu_j^{(i)} Q(x_t^{(i)})_j$$

Figure 3: The AFAMAP algorithm.

equivalently, we are performing inference using both models, but constrain the posterior over the hidden variables in both cases to *agree* (note that we don't require the z_t variables to agree, but rather they act to make each model robust to outliers). This leads to the final optimization problem shown in Figure 3, and we refer to the resulting algorithm as AFAMAP (Additive Factorial Approximate MAP).

The AFAMAP optimization problem is a convex quadratic program, so could in principle be solved using off-the-shelf libraries. However, the total variation term in particular is poorly handled by general solvers. Fixing $\mathcal{Q}$, however, we are left with a simple proximal TV regularization problem, which is efficiently solved using recent methods (Barbero and Sra, 2011). Thus, we employ alternating minimization, using a generic QP solver (we use the CPLEX solver in our implementation) to solve the AFAMAP problem for fixed $z_{1:T}$, then using the custom solver referenced above to find $z_{1:T}$ given a fixed $\mathcal{Q}$. A major advantage to the AFAMAP algorithm is that the objective is *jointly* convex $\mathcal{Q}$ and $z_{1:T}$ and so this procedure will find the global optimum; if we try to incorporate a robust mixture component using standard MAP inference, some variational approximation would be necessary. Also crucial to obtaining good performance is to properly exploit sparsity: if a transition matrix $P^{(i)}$ is

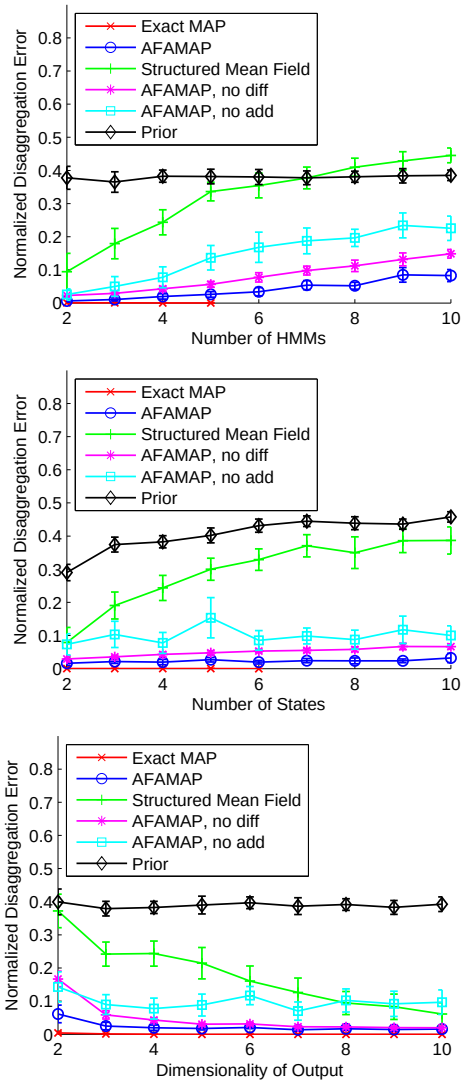

Figure 4: Performance varying the number of HMMs (top), the number of states (middle), and the dimensionality of the output (bottom).

sparse, then we only need to represent those values of $Q^{(i)}(x_{t-1}^{(i)}, x_t^{(i)})$ corresponding to non-zero entries.

Given a (possibly non-integral) solution of the optimization problem, we could potentially refine it using branch and bound approaches, randomized rounding, or run exact MAP inference only over the non-integral states. In practice, however, since the LP objective encourages integer-valued solutions, simply predicting the mean output works well, and we use this approach in AFAMAP.

4 Experiments on Synthetic Data

To measure the effectiveness of our inference algorithm with respect to different elements of the problem setting, we evaluate the AFAMAP algorithm versus several potential competitors on a simple synthetic data set.[4] For state spaces that are small enough we run exact MAP inference. We evaluate the above-mentioned structured variational mean field (SMF) algorithm (Ghahramani and Jordan, 1997), which approximates the FHMM posterior by N decoupled HMMs, and iteratively improves the log-likelihood. We also evaluate the AFAMAP algorithm with one of the two components removed (i.e., with $\Sigma_1 = \infty$ or $\Sigma_2 = \infty$). While the objective of SMF is slightly different, as it is computing a full distribution over hidden states rather than just a MAP estimate, in virtually all the cases we consider (both here and in the next section) the posterior is sufficiently peaked that the outputs are usually nearly integer valued. Finally, as a baseline we compare to a method that simply sets the expectation of all states to the stationary distribution of the Markov chain. To evaluate the methods, we use the normalized *disaggregation error*, which directly measures how well the models recovered the individual HMM outputs: given true output $y_t^{(i)}$ and predicted output $\hat{y}_t^{(i)}$, this is defined as

$$\sqrt{\left(\sum_{t,i} \left\| y_t^{(i)} - \hat{y}_t^{(i)} \right\|_2^2\right) / \left(\sum_{t,i} \left\| y_t^{(i)} \right\|_2^2\right)} \qquad (16)$$

Figure 4 shows the performance of the different algorithms varying the number of HMMs, states per HMMs, and the dimensionality of the output. In all cases AFAMAP outperforms all alternative approaches (except for exact MAP inference when applicable). The failure of the SMF approach is particularly evident here: because the algorithm assigns states to HMMs one at a time, it often will explain some portion of the aggregate signal using the incorrect state; even using annealing-based approaches (inflating the variance in early iterations, which we do for all SMF experiments), it is often difficult to get out of this local optimum. The AFAMAP algorithm, in contrast, is a convex method and thus has no problems of local optima; furthermore, training using both the difference and additive models outperforms either in isolation.

Our second set of experiments evaluates the robust mixture component. Here we add a random walk

[4] Experimental details: To create the data, we generate N "cyclic" HMMs, each with m states, and an n dimensional output; the initial state distribution is uniform over all states, the transition matrix is generated by $P_{j,j}^{(i)} \propto 30$, $P_{\mathrm{mod}(j+1,k+1)+1,j}^{(i)} \propto U[1,2]$ and the mean of each HMM is $(\mu_j^{(i)})_\ell \sim U[0,2]$. For each HMM we sample $T = 500$ time steps, and observe the sum of all the individual outputs plus Gaussian noise sampled from $\mathcal{N}(0, 0.01I)$. For all experiments we use $N = 4$, $m = 4$, and $n = 4$, except if one of these parameters is being varied in the experiment. Regularization parameters were set to be on the order of the parameters of the true model. MATLAB code for the experiments are included with the paper.

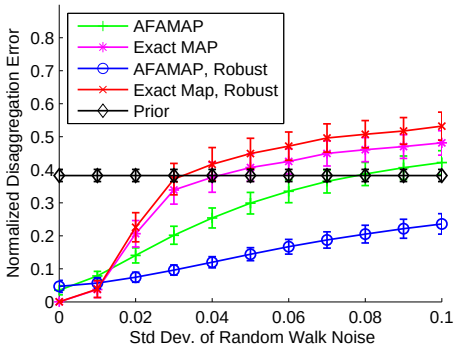

Figure 5: Performance of algorithms, with and without robust mixture component, with random walk noise of different levels added to the output.

signal to the output, $\bar{y}_t = \bar{y}_t + w_t$ where $w_t \sim \mathcal{N}(w_{t-1}; \sigma^2 I)$, and varied σ from 0 to 0.1. Figure 5 shows the performance of "exact" MAP and the AFAMAP algorithm, with and without the robust component. This experiment highlights the benefit of the convex formulation: for the exact MAP formulation, we need some way of including the TV mixture component, since joint MAP inference over $x_{1:T}^{(1:N)}$ and $z_{1:T}$ is not tractable; in this example, we used simple alternating maximization which leads to similar issues as with SMF: initial iterations "lock" the HMM into bad state guesses, and the resulting optimization optimization over the robust mixture cannot correct this. Alternatively, the convex objective of AFAMAP is able to estimate the noise component.

5 Electrical Energy Disaggregation

As mentioned above, the primary application we are concerned with in this paper is the task of electrical energy disaggregation, separating a whole-home energy signal into its component appliances. Although the focus of this paper is the inference procedure, briefly, the setup is as follows. Using a high-resolution A/D device, we logged whole-home aggregate power consumption for two weeks. We also logged power consumption at the circuit level to obtain ground truth labels (this was used only for validation, and not for training the models). To reduce the signal size, we used a greedy variant of total variation regularization to approximate the power signal as piecewise constant.[5] Using the total power signal, we extract all snippets of data where consumption increases over some threshold then eventually returns to its original level (i.e., where some device comes on and turns off); some of these are indecipherable due to the aggregation, but especially for short device durations there are occasional snippets of individual devices. We model all these snippets as "empirical" HMMs (means equal to

[5]All data is available at `http://redd.csail.mit.edu`.

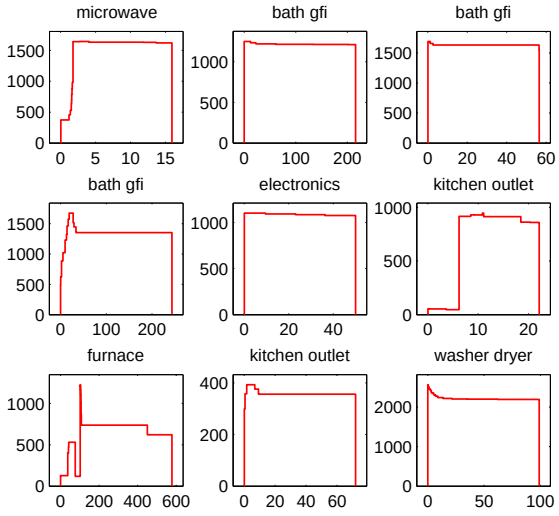

Figure 6: Extracted signatures for nine appliance types, labeled by device circuit, with time in seconds on the x-axis and power on the y-axis. Each state of the device HMM corresponds to a constant-power segment in the plot, with length equal to that state's expected duration.

the observed power output, and transitions probabilities set based upon the amount of time spent at each power level) and looked at all pairwise probabilities between them (the probability of one snippet generating another); using the k-nearest-neighbor graph induced by these probabilities, we ran spectral clustering to group devices together. This resulted in selecting nine "prototypical" motifs, shown in Figure 6, that occurred frequently in the data, and which correspond very clearly to different devices.

Treating these nine extracted motifs as the HMMs in a factorial model, we ran the AFAMAP to separate the appliances in the aggregate power signal over the entire two week period. For all methods, regularization parameters (λ and Σ) were fit using one day of the data. Table 1 shows the performance of the methods for each device. Because we do not know the true contribution of the device (just the circuit contribution), we report precision and recall metrics at the circuit level: recall measures what portion of a given circuit's energy is correctly classified, while precision measures, of the energy assigned to a circuit, how much truly belonged to that circuit. Because there are some portions of the energy that are correctly left unassigned (because we have no model of that device), we would not expect to achieve 100% recall, and we place higher emphasis on the precision metric.

For all the circuits in the home, AFAMAP vastly outperforms the SMF method on this problem. In particular, for five of the nine devices considered, AFAMAP

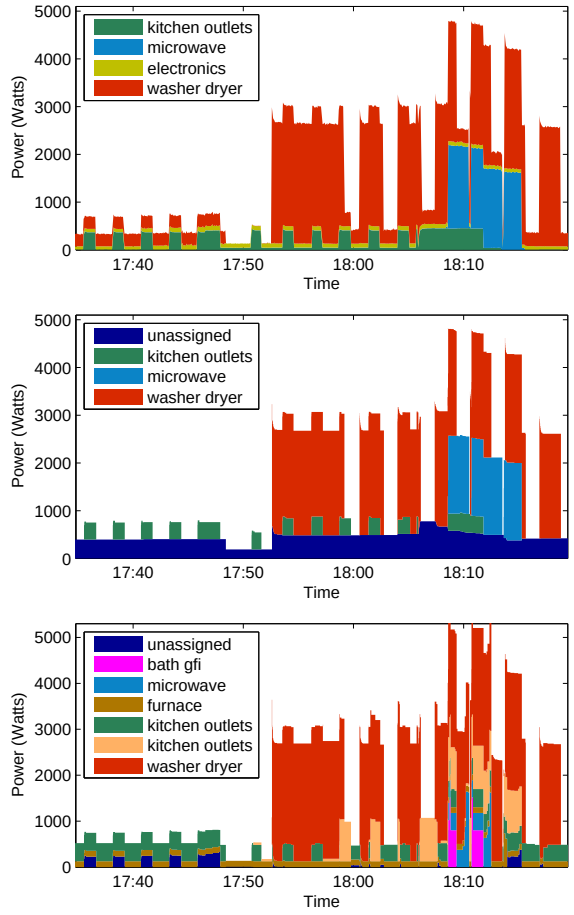

Figure 7: True (top), AFAMAP predicted (middle), and SMF predicted (bottom) breakdown by device for a short portion of power data.

has a precision in its assignments of more than 90%; this starts to reach the order of the accuracy in sensing between the two data sources, and a visual inspection of the signals confirms that AFAMAP is assigning these devices essentially perfectly. AFAMAP does make errors in the remaining four devices, which upon inspection is due to the fact that appliances on multiple circuits have signatures very similar to these; the five devices identified correctly have distinct patterns that AFAMAP can identify even when they occur in aggregated form. In contrast, the SMF algorithm essentially fails completely on this task: either the algorithm predicts a great deal of energy incorrectly, or it assigns very little energy to a circuit.

The difference between the AFAMAP and SMF algorithms is equally striking if we look visually at the predicted separations of the two methods. Figure 7 looks at a short period of time and shows the true appliance breakdown along with the predictions of AFAMAP and SMF. AFAMAP correctly identifies the washer/dryer, microwave, and kitchen outlets during this time (with minor errors), and (correctly) uses the robust TV mixture component to capture the remaining power, for which it does not have a model. In contrast, SMF assigns the power to the washer/dryer (almost) correctly, but completely fails to properly assign the other devices. This is indicative of the method's tendency to "lock in" to variable assignments; indeed, we tuned SMF substantially to get even this result: optimizing the HMMs in a random order, for example, typically misses the washer/dryer entirely.

Table 1: Per-circuit performance for AFAMAP and SMF for two weeks of data. Performance is reported as precision/recall, and bold entries denote statically significant better performance on both metrics. More than one device above can belong to a single circuit.

Circuit	AFAMAP	SMF
1 Microwave	**97.5% / 66.1%**	96.8% / 4.1%
2 Bath GFI	**82.7% / 70.8%**	50.1% / 9.1%
3 Electronics	41.6% / 0.8%	41.4% / 0.3%
4 Kitch. Out. 1	37.5% / 12.9%	10.2% / 47.8%
5 Furnace	**91.7% / 70.8%**	12.6% / 15.3%
6 Kitch. Out. 2	45.2% / 16.0%	13.3% / 24.8%
7 Wash/Dryer	98.8% / 73.6%	89.3% / 76.7%
Total	**87.2% / 60.3%**	35.5% / 45.1%

6 Conclusion

Although the additive factorial HMM model has large representation power, its applicability has likely been greatly reduced by the difficulty of inference in models with a substantial number of HMMs. By exploiting the additive structure of the problem and by constraining the set of allowed posteriors, the AFAMAP algorithm can accurately perform inference on such tasks, opening the door to many more potential applications. While this paper focused specifically on the inference problem and the application to energy disaggregation, it would be natural to investigate the combination of learning and inference using this procedure, as well as look at using the method as an initialization for other inference procedures. From an application standpoint, many extensions look promising as well. The unsupervised learning procedure we described briefly is currently limited to finding devices with relatively short time scales; but by applying this approximate inference and iteratively looking at the "unassigned" portions of energy, it may be possible to successively build models for more and more devices in the home. Combined with joint inference and "hard EM" learning procedures (EM but using MAP inference) these methods have the potential to achieve the eventual goal of disaggregating virtually all the appliances in a home without supervision.

References

Bach, F. R. and Jordan, M. I. (2005). Discriminative training of hidden markov models for multiple pitch tracking. In *Proceedings of the International Conference on Accoustics, Speech, and Signal Processing*.

Barbero, A. and Sra, S. (2011). Fast newton-type methods for total variation regularization. In *Proceedings of the International Conference on Machine Learning*.

Berges, M., Goldman, E., Matthews, H. S., and Soibelman, L. (2010). Enhancing electricity audits in residential buildings with non-intrusive load monitoring. *Journal of Industrial Ecology: Special Issue on Environmental Applications of Information and Communications Technology*, 14(5):844–858.

Darby, S. (2006). The effectiveness of feedback on energy consumption. Technical report, Environmental Change Institute, University of Oxford.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial hidden markov models. *Machine Learning*, 27:245–273.

Gupta, S., Reynolds, S., and Patel, S. N. (2010). ElectriSense: Single-point sensing using EMI for electrical event detection and classification in the home. In *Proceedings of the Conference on Ubiquitous Computing*.

Hart, G. (1992). Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12).

Kim, H., Marwah, M., Arlitt, M., Lyon, G., and Han, J. (2011). Unsupervised disaggregation of low frequency power measurements. In *Proceedings of the SIAM Conference on Data Mining*.

Kolter, J. Z., Batra, S., and Ng, A. Y. (2010). Energy disaggregation via discriminative sparse coding. In *Neural Information Processing Systems*.

Laughman, C., Leeb, S., and Lee (2003). Advanced non-intrusive monitoring of electric loads. *IEEE Power and Energy*.

Neenan, B. and Robinson, J. (2009). Residential electricity use feedback: A research synthesis and economic framework. Technical report, Electric Power Research Institute.

Roweis, S. T. (2001). One microphone source separation. In *Neural Information Processing Systems*.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phyica D*, 60:259–268.

Virtanen, T. (2006). Speech recognition using factorial hidden markov models for separation in the feature space. In *Procedings of Interspeech*.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Ziefman, M. and Roth, K. (2011). Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics*, 57(1):76–84.

A Proof of Proposition 1

A continuous time Markov chain with m states is defined via an *incidence matrix* $Q \in \mathbb{R}^{m \times m}$. Assuming the system is is state j at real-valued time $t \in \mathbb{R}_+$, we sample a *duration* from an exponential distribution with rate $-Q_{jj}$. We then transition to state $x(t+dt) = k$ with probability $-Q_{k,j}/Q_{j,j}$ (which implies that Q must have $\sum_{k \neq j} Q_{k,j} = -Q_{j,j}$). That is

$$dt|x(t) = j \sim \text{Exp}(-Q_{j,j})$$
$$x(t+dt) \sim \text{Mult}(q), \quad q_k = \begin{cases} \frac{-Q_{k,j}}{Q_{k,k}} & k \neq j \\ 0 & k = j \end{cases} \tag{17}$$

For some time interval Δt, the discretized transition matrix, defined as

$$P(\Delta t)_{k,j} = p(x(t+\Delta t) = k | x(t) = j), \tag{18}$$

is given by

$$P(\Delta t) = \exp(\Delta t Q) \tag{19}$$

where $\exp(\cdot)$ here is the matrix exponential defined as

$$\exp(X) = \sum_{i=0}^{\infty} \frac{1}{i!} X^i \tag{20}$$

We now prove Proposition 1 using these definitions.

Proof. We first note that for $r = -\min_j Q_{j,j}$ (the largest rate parameter for the exponential distributions)

$$\min_j P(\Delta t)_{j,j} \geq 1 - r\Delta t. \tag{21}$$

This follows immediately from the matrix inequality $\exp(X) \succeq I + X$ which in turn follows from the eigenvalue representation of matrix exponentiation

$$\exp(X) = U \exp(\Lambda) U^{-1} \tag{22}$$

where $X = U \Lambda U^{-1}$ is the eigenvalue decomposition of X. Since Λ is diagonal, the matrix exponential is simply the elementwise exponentiation of each entry on the diagonal, and the matrix inequality follows by

applying the basic inequality $e^x \geq 1+x$ for each of the diagonal entries.

Now, for a single time step let A^0 be the event that only zero or one of the N HMMs change state. Using the bound above with $r = p - \min_{i,j} Q^{(i)}_{j,j}$ defined over all the HMMs with Δt sufficiently small such that $r\Delta t < 1$,

$$\begin{aligned} p(A^0) &\geq (1-r\Delta t)^N + Nr\Delta t(1-r\Delta t)^{N-1} \\ &\geq (1-r\Delta t)^N + Nr\Delta t(1-r\Delta t)^N \\ &= (1+Nr\Delta t)(1-r\Delta t)^N. \end{aligned} \tag{23}$$

Then probability of the one-at-a-time condition holding for all $t_f/\Delta t$ time steps is just $P(\neg A) \geq P(A^0)^{t_f/\Delta t}$ and

$$\log P(\neg A) \geq \frac{t_f}{\Delta t}\left(\log(1+Nr\Delta t) + N\log(1-r\Delta t)\right). \tag{24}$$

Using the Taylor approximation

$$\log(1+x) = x - \frac{1}{2}x^2 + O(x^3) \tag{25}$$

we have that

$$\begin{aligned} &\log(1+Nr\Delta t) + N\log(1-r\Delta t) \\ &= Nr\Delta t - \frac{1}{2}N^2r^2\Delta t^2 - Nr\Delta t - \frac{1}{2}Nr^2\Delta t^2 + O(\Delta t^3) \\ &= -\frac{1}{2}(N+N^2)r^2\Delta t^2 + O(\Delta t^3). \end{aligned} \tag{26}$$

Combining this with (24) gives

$$\log P(\neg A) \geq -\frac{1}{2}t_f(N+N^2)r^2\Delta t + O(\Delta t^2). \tag{27}$$

Finally, using the inequality $e^x \geq 1+x$, we have

$$\begin{aligned} P(A) &= 1 - P(\neg A) \\ &= 1 - \exp(\log P(\neg A)) \\ &\leq -\log P(\neg A) \\ &\leq \frac{1}{2}t_f(N+N^2)r^2\Delta t + O(\Delta t^2) = O(\Delta t) \end{aligned} \tag{28}$$

□

While we focus on the dependence on Δt in our presentation of the theorem, it is worth noting that the dependence is also quadratic in the number of HMMs, N. Thus, for large numbers of concurrent HMMs (where the probability of changing state is not overly small), we may still need a fairly high sampling frequency.

B Derivation of the MILP for Difference FHMMs

Here we present a detailed description of how we arrive at the mixed-integer linear programming formulation to MAP inference in the difference FHMM (15), by considering the mixed-integer quadratic program (10) along with the one-at-a-time constraint (13). As mentioned in the text, enforcing the one-at-a-time constraint implies that at most one term in the summation

$$\sum_{i,j,k} \Delta\mu^{(i)}_{k,j} Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k} \tag{29}$$

is non-zero.

First suppose there is one HMM i at time t with an off-diagonal entry $Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k}$, $j \neq k$. Then the error in approximating $\Delta\bar{y}_t$ will be

$$\begin{aligned} &\left\|\Delta\bar{y}_t - \Delta\mu^{(i)}_{k,j} Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k}\right\|^2_{\Sigma^{-1}} \\ &= \left\|\Delta\bar{y}_t - \Delta\mu^{(i)}_{k,j}\right\|^2_{\Sigma^{-1}} Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k} \end{aligned} \tag{30}$$

since $Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k} = 1$ by the integrality constraint, and since $\Delta\mu^{(i')}_{j',j'} = 0$ for all other (diagonal) entries $Q(x^{(i')}_{t-1}, x^{(i')}_t)_{j',j'} = 1$. Furthermore, by the one-at-a-time condition, the $\Delta z_t = 0$ when $Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k} = 1$. Thus, the quadratic error term

$$\frac{1}{2}\left\|\Delta\bar{y}_t - \Sigma^{-1/2}\Delta z_t - \sum_{i,j,k} \Delta\mu^{(i)}_{k,j} Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k}\right\|^2_{\Sigma_{-1}} \tag{31}$$

is equivalent in this case to the linear error term

$$\frac{1}{2}\sum_{i,j,k\neq j} \left\|\Delta\bar{y}_t - \Delta\mu^{(i)}_{k,j}\right\|^2_{\Sigma_{-1}} Q(x^{(i)}_{t-1}, x^{(i)}_t)_{j,k} \tag{32}$$

as desired, and the ℓ_1 term $\|\Delta z_t\|_1 = 0$, so does not affect the optimization problem.

Alternatively, suppose that no HMM changes state at time t. Then the error in approximating $\Delta\bar{y}_t$ is simply

$$\frac{1}{2}\|\Delta\bar{y}_t - \Sigma^{1/2}\Delta z_t\|^2_{\Sigma^{-1}} = \frac{1}{2}\|\Sigma^{-1/2}\Delta\bar{y}_t - \Delta z_t\|^2_2 \tag{33}$$

This term plus the ℓ_1 penalty on Δz_t can be optimized analytically, and has the well-known "soft-thresholding" solution,

$$\arg\min_z \frac{1}{2}\|y-z\|^2_2 + \lambda\|z\|_1 = \text{sign}(y)\max\{|y| - \lambda, 0\} \tag{34}$$

where all the operations in the right hand side are applied elementwise to the entries of y. This solution

attains objective value

$$\begin{aligned}
&\sum_{i=1}^{n} \frac{1}{2}(y_i - \text{sign}(y_i)\max\{|y_i| - \lambda, 0\})^2 + \lambda \max\{y_i - \lambda, 0\} \\
&= \sum_{i=1}^{n} \min\{\frac{1}{2}y_i^2, \frac{1}{2}\lambda^2\} + \lambda \max\{|y_i| - \lambda, 0\} \\
&= \sum_{i=1}^{n} \min\left\{\frac{1}{2}y_i^2, \max\left\{\lambda|y_i| - \frac{\lambda^2}{2}, \frac{\lambda^2}{2}\right\}\right\} \\
&= D(y, \lambda)
\end{aligned} \tag{35}$$

Thus, the error (31) is equivalent in this case to the error term

$$D(\Sigma^{-1/2}\Delta\bar{y}_t)\left(1 - \sum_{i,j,k\neq j} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k}\right) \tag{36}$$

since $\sum_{i,j,k\neq j} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} = 0$ when no HMMs change state. Combining (32) and (36) give the MILP formulation (15) as desired, and have eliminated the Δz_t variable from the optimization problem.