Group Norm for Learning Latent Structural SVMs

Daozheng Chen
UMD, College Park
dchen@cs.umd.edu

Dhruv Batra
TTI Chicago
dbatra@ttic.edu

Bill Freeman
MIT
billf@mit.edu

Micah K. Johnson
GelSight, Inc.
kimo@csail.mit.edu

Abstract

Latent variables models have been widely applied in many problems in machine learning and related fields such as computer vision and information retrieval. However, the complexity of the latent space in such models is typically left as a free design choice. A larger latent space results in a more expressive model, but such models are prone to overfitting and are slower to perform inference with. The goal of this paper is to regularize the complexity of the latent space and *learn* which hidden states are really relevant for the prediction problem. To this end, we propose regularization with a group norm such as ℓ_1-ℓ_2 to estimate parameters of a Latent Structural SVM. Our experiments on digit recognition show that our approach is indeed able to control the complexity of latent space, resulting in significantly faster inference at test-time without any loss in accuracy of the learnt model.

1 Introduction

Fully supervised algorithms are really the spoilt children of computer vision. We almost never have complete supervision – there are always some variables relevant to the problem that not annotated in our datasets. Latent variable models provide an ideal abstraction for such situations. They allow for modelling of interaction between the observed data (*e.g.* image features) and latent or hidden variables not observed in the training data (*e.g.* location of body parts). These hidden variables may help explain correlations in the features, provide a low-dimensional embedding of the input, or help with prediction. Training latent variable models, however, is notoriously problematic, since it typically involves a difficult non-convex optimization problem. Common algorithms for solving these problems, EM [5] and CCCP [8, 16, 18], are known to be highly sensitive to initialization and prone to getting stuck in a poor local optimum. Recently, Bengio *et al.* [2] and Kumar *et al.* [10] have presented a curriculum learning scheme that trains latent variable models in an easy-to-difficult manner, by initially pruning away difficult examples in the dataset.

Our goal is to study the modelling-optimization tradeoff in designing latent variable models for computer vision problems. From a modelling perspective, we would like to design ever more complex latent variables, *e.g.* capture location of parts, their scale, orientation, appearance. However, from an optimization perspective, complex models are more difficult to train than simpler ones, more prone to getting stuck in a bad local minima, resulting in poor generalization. In most existing models, the complexity of the latent variable space is typically left as a free design choice that is hand-tuned. Thus, the question we seek to answer is: Is there a principled way to *learn* the complexity of the latent space in a latent variable model?

In this paper, we propose the use of structured sparsity inducing norms like ℓ_1-ℓ_2 to estimate the parameters of a latent-variable model, thereby regularizing the complexity of the latent space. Structured sparsity inducing norms are generalization of the ℓ_1 norm and regularize solutions to be sparse in a structured way. Specifically, group ℓ_1-ℓ_2 norm behaves like an ℓ_1 norm at a group level and encourages groups of variables to be sparse. We divide the latent variable state space into different groups, among which the group norm is induced. Since the group norm encourages group-sparsity, this allows simultaneous parameter estimation as well as state selection. We apply our approach to

the recently proposed latent structural SVM (LSSVM) [16]. Our results on digit recognition show that our approach is indeed able to control the complexity of latent space, resulting in significantly faster inference at test-time without any loss in accuracy of the learnt model.

2 Related Work

Most relevant to our work are algorithms for discovering latent structure in latent variable models and other applications of structured sparsity inducing norms. These are both broad goals and cover a vast amount of literature. We mention the works most directly relevant to our approach.

Latent variable models have been used to model observations in both generative and discriminative settings. In the generative setting, the goal is to *explain* the data with a low-dimensional latent structure. Mixture models like Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) also have a long history in applications such as speech recognition [13].

More recently, a number of discriminative latent models such as Hidden Conditional Random Field (HCRFs) [15], Latent SVMs [8] and Latent Structural SVMs (LSSVMs) [16] have been proposed. These models have demonstrated success in a number of applications. They differ from generative models in the sense that the ultimate goal is prediction not explanation of the data.

In both kinds of models, the parameter learning problem is non-convex and solved with techniques like EM [5] and Concave-Convex Procedure (CCCP) [8, 16, 18] respectively. Note that for all the models above, the latent variables and their state space are predefined and fixed for specific applications. Our approach, on the other hand, aims for parameter estimation as well as *discovery* of meaningful latent variable states.

Related to this goal of discovery is the work of Chandrasekaran et al. [3], which attempts to identify graphical model structure assuming that latent and observed variables are jointly Gaussian. Our work is different in that we are interested in *prediction* via a sparse latent model and not identification of such a model. Moreover, we make no Gaussian assumptions, which may be infeasible for applications.

There is a fairly mature body of work on ℓ_1 regularization for sparse regression models [4, 7, 14]. Sparse coding with ℓ_1 regularization has been successfully used to solve many problems in compressed sensing [6] and signal processing [12]. Yuan and Lin [17] introduced group-norm regularization to allow parameter estimation as well as selection of certain groups of variables. More recently, Bach [1] proposed general sparsity inducing structured norms. To the best of our knowledge, this is the first work to use structured norms in the context of latent variable selection in LSSVMs.

Section 3 revisits the LSSVM model and describes our proposed group-norm modification. Section 4 describes how parameter learning can be performed in this new model. Finally, our experiments on digit recognition in Section 5 demonstrate that our approach is indeed able to control the complexity of latent space, resulting in significantly faster inference at test-time without any loss in accuracy of the learnt model.

3 Latent Structural SVM

Notation. For any positive integer n, let $[n]$ be shorthand for the set $\{1, 2, \ldots, n\}$. We denote training data as $\mathcal{D} = \{(x_i, y_i) \mid i \in [n]\}$, where $x_i \in \mathcal{X}$ is the (input) observed feature-vector and $y_i \in \mathcal{Y}$ is the (possibly structured) output label for the i^{th} sample. In addition, let $h_i \in \mathcal{H}$ denote the latent variable for the i^{th} sample. For example, in handwritten digit recognition, x_i can be the original digit image, y_i the true digit label and h_i the (deformation) rotation angle that must be corrected for before extracting features.

LSSVMs provide a linear prediction rule of the form $f_{\boldsymbol{w}}(x) = \operatorname{argmax}_{(y,h)\in\mathcal{Y}\times\mathcal{H}} \boldsymbol{w} \cdot \phi(x, y, h)$, where $\phi(x, y, h)$ is the joint feature vector that encodes the relationship between the input, hidden and output variables, and $\boldsymbol{w}$ is the model parameter vector. In digit recognition, this joint feature vector can be the vector representation of the image x rotated by the angle corresponding to h.

The parameter vector $\boldsymbol{w}$ is learned by minimizing the (regularized) risk on the training dataset $\mathcal{D}$. A user-specified risk function $\Delta(y_i, \hat{y}_i(\boldsymbol{w}), \hat{h}_i(\boldsymbol{w}))$ measures the loss incurred for predicting

$(\hat{y}_i(\boldsymbol{w}), \hat{h}_i(\boldsymbol{w}))$ for the i^{th} sample. Yu and Joachims [16] minimized an upper-bound on the risk and formulated the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}, \xi_i \geq 0} \quad & \Omega(\boldsymbol{w}) + \frac{C}{n} \sum_{i=1}^{n} \xi_i, \\ s.t. \quad & \max_{h_i \in \mathcal{H}} \boldsymbol{w} \cdot \Big(\phi(x_i, y_i, h_i) - \phi(x_i, \hat{y}_i, \hat{h}_i) \Big) \geq \Delta(y_i, \hat{y}_i, \hat{h}_i) - \xi_i, \\ & \forall (\hat{y}_i, \hat{h}_i) \in \mathcal{Y} \times \mathcal{H},\ i \in [n]. \end{aligned} \tag{1}$$

where, the regularization term is $\Omega(\boldsymbol{w}) = \Omega_{\ell_2}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$. We refer the reader to [16] for more details about this formulation.

3.1 Inducing Group Norm for State Learning

Let $\mathcal{H} = \{h^1, \cdots, h^P\}$ be the set of states which the latent variables can take. Such set of states, for example, can be the set of all possible rotation angles in digit recognition. Our goal is to regularize the complexity of the latent space and *learn* which hidden states are really relevant for the prediction problem. To this end, we consider replacing the ℓ_2-norm regularizer ($\Omega_{\ell_2}(\boldsymbol{w})$) in problem (1) with an ℓ_1-ℓ_2-norm to learn meaningful latent variable states.

We start with describing a modification to the linear prediction rule that makes it easier to encode the group structure of latent variables. Specifically, let the parameter $\boldsymbol{w}$ be partitioned into P groups. Each group corresponds to the parameters of a latent variable state we want to learn. Let the parameter vector for the p^{th} group be denoted by $w^p = [w_1^p, \cdots, w_{n_p}^p]$ and $\boldsymbol{w} = [w^1, \cdots, w^P]$ is the concatenation of such vectors from each group. Thus, the modified linear prediction rule is given by $f_{\boldsymbol{w}}(x) = \text{argmax}_{y \in \mathcal{Y},\, p \in [P]}\, w^p \cdot \phi(x, y, h^p)$. Note that with appropriate zero-padding of the features, this model is equivalent to the original linear model. With this representation, parameters for each state are represented separately and thus group ℓ_1 regularization is possible over the state space. We apply ℓ_1-ℓ_2-norm [17] in $\Omega(\boldsymbol{w})$ to perform this regularization.

$$\Omega(\boldsymbol{w}) = \Omega_{\mathcal{G}}(\boldsymbol{w}) = \sum_{p=1}^{P} \lambda_p \|w^p\|_2, \tag{2}$$

where $\lambda_p \geq 0$ is the regularization weight for group p. Within each group, ℓ_2-norm is used, which does not promote sparsity. At the group level, this norm behaves like the ℓ_1-norm and thus induces group sparsity, *i.e.* the parameters of some groups are encouraged to be set completely to zero. Uninformative states will thus have sparse learned parameters. This gives us a way to select most useful states for prediction and shrink the state space size. Note that this approach is not feasible when the latent space is structured (trees, etc) and thus exponentially large.

Putting equation (2) into problem (1), we have the formulation for our state learning problem. The next section gives a detailed description of our algorithm for solving this problem.

4 Alternating Coordinate and Subgradient Descent

Problem (1) can be rewritten as

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) = \min_{\boldsymbol{w}} \left[\Omega(\boldsymbol{w}) + \frac{C}{n} \sum_{i=1}^{n} \max\{0, f_i(\boldsymbol{w}) - g_i(\boldsymbol{w})\} \right], \tag{3}$$

where

$$f_i(\boldsymbol{w}) = \max_{(\hat{y}_i, \hat{h}_i) \in \mathcal{Y} \times \mathcal{H}} \Big[\boldsymbol{w} \cdot \phi(x_i, \hat{y}_i, \hat{h}_i) + \Delta(y_i, \hat{y}_i, \hat{h}_i) \Big] \tag{4}$$

$$g_i(\boldsymbol{w}) = \max_{h_i \in H} \boldsymbol{w} \cdot \phi(x_i, y_i, h_i) \tag{5}$$

Yu and Joachims [16] used the Concave-convex procedure (CCCP) [18] to minimize $L(\boldsymbol{w})$, while Felzenszwalb *et al.* [8] used Stochastic Subgradient Descent (SSD). Our approach is similar to that

Algorithm 1 Alternating coordinate and subgradient descent algorithm for parameter estimation

Input: $\mathcal{D} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, initialize $\boldsymbol{w}_0$, learning rate α_0, ϵ

$t \leftarrow 0$
repeat
  for $i = 1$ to n **do**
    $g_i^* \leftarrow \max_{h_i \in \mathcal{H}} \boldsymbol{w}_t \cdot \phi(x_i, y_i, h_i)$, and obtain maximizer h_i^*
    $f_i \leftarrow \max_{(\hat{y}_i, \hat{h}_i) \in \mathcal{Y} \times \mathcal{H}} \boldsymbol{w}_t \cdot \phi(x_i, \hat{y}_i, \hat{h}_i) + \Delta(y_i, \hat{y}_i, \hat{h}_i)$, and obtain maximizer $(\hat{y}_i^*, \hat{h}_i^*)$
    if $f_i - g_i^* \leq 0$ **then**
      $m_i \leftarrow 0$
    else
      $m_i \leftarrow \phi(x_i, \hat{y}_i^*, \hat{h}_i^*) - \phi(x_i, y_i, h_i^*)$
    end if
  end for
  $\nabla L^* \leftarrow \nabla \Omega(\boldsymbol{w}_t) + \frac{C}{n} \sum_{i=1}^n m_i$
  $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \alpha_t \nabla L^*$
  $t \leftarrow t + 1$
until $|(L(\boldsymbol{w}_t) - L(\boldsymbol{w}_{t-1}))/L(\boldsymbol{w}_t)| < \epsilon$

of Felzenszwalb *et al*. [8] – we minimize the following upper bound on $L(\boldsymbol{w})$:

$$\min_{\boldsymbol{w}, \{h_i\}} L^*(\boldsymbol{w}, \{h_i\}) = \min_{\boldsymbol{w}, \{h_i\}} \left[\Omega(\boldsymbol{w}) + \frac{C}{n} \sum_{i=1}^n \max\{0, f_i(\boldsymbol{w}) - g_i(\boldsymbol{w}, h_i)\} \right], \quad (6)$$

where $$g_i(\boldsymbol{w}, h_i) = \boldsymbol{w} \cdot \phi(x_i, y_i, h_i) \quad (7)$$

Here $L^*(\boldsymbol{w}, \{h_i\})$ is the objective function with latent variables specified for the training data. Fixing the latent variables makes $L^*(\boldsymbol{w}, \{h_i\})$ convex in $\boldsymbol{w}$. Moreover, $L(\boldsymbol{w}) \leq L^*(\boldsymbol{w}, \{h_i\})$. In a manner similar to Felzenszwalb *et al*. [8], we follow an alternating coordinate descent and subgradient descent scheme. At iteration t, we first fix $\boldsymbol{w}_t$ and optimize $L^*(\boldsymbol{w}, \{h_i\})$ w.r.t. $\{h_i\}$, *i.e*. compute $h_i^* = \text{argmin}_{h_i} L^*(\boldsymbol{w}, \{h_i\}) = \text{argmax}_{h_i \in \mathcal{H}} \boldsymbol{w}_t \cdot \phi(x_i, y_i, h_i)$. Next, we fix $\{h_i^*\}$ and update $\boldsymbol{w}_t$ by taking a negative subgradient step $-\nabla L^*(\boldsymbol{w}, \{h_i^*\})$:

$$\nabla L^*(\boldsymbol{w}, \{h_i^*\}) = \nabla \Omega_{\mathcal{G}}(\boldsymbol{w}) + \frac{C}{n} \sum_{i=1}^n m_i(\boldsymbol{w}, h_i^*), \quad (8)$$

where

$$\nabla \Omega_{\mathcal{G}}(\boldsymbol{w}) = \left[\frac{w_1^1}{\|w^1\|_2}, \cdots, \frac{w_{n_1}^1}{\|w^1\|_2}, \cdots, \frac{w_1^P}{\|w^P\|_2}, \cdots, \frac{w_{n_P}^P}{\|w^P\|_2} \right], \quad (9)$$

and

$$m_i(\boldsymbol{w}, h_i^*) = \begin{cases} 0 & \text{if } f_i(\boldsymbol{w}) - g_i(\boldsymbol{w}, h_i^*) \leq 0 \\ \phi(x_i, \hat{y}_i^*, \hat{h}_i^*) - \phi(x_i, y_i, h_i^*) & \text{otherwise} \end{cases} \quad (10)$$

where $(\hat{y}_i^*, \hat{h}_i^*) = \text{argmax}_{(\hat{y}_i, \hat{h}_i) \in \mathcal{Y} \times \mathcal{H}} f_i(\boldsymbol{w}) = \text{argmax}_{(\hat{y}_i, \hat{h}_i) \in \mathcal{Y} \times \mathcal{H}} \left[\boldsymbol{w} \cdot \phi(x_i, \hat{y}_i, \hat{h}_i) + \Delta(y_i, \hat{y}_i, \hat{h}_i) \right]$. Algorithm 1 describes the entire algorithm. Following [9], we choose learning rate at iteration t to be $\alpha_t = \frac{1}{\eta_t + 1}$, where η_t is the number of times the objective value $L^*(\boldsymbol{w}, \{h_i^*\})$ has increased from one iteration to the next. This learning rate is effective in our experiment.

5 Experiment

We now demonstrate the efficacy of our approach in the context of handwritten digit recognition. We follow closely the experimental setup of Kumar *et al*. [10], who proposed a LSSVM approach for this problem. Each digit is represented as a vector x of grayscale values at pixels. The goal is to predict the label of the digit, $y \in \mathcal{Y} = \{0, 1, \cdots, 9\}$. It is well-known that the accuracy can be greatly improved by explicitly modeling the deformations present in each image. Kumar *et al*. [10] model rotations as a hidden variable taking values in a set of 11 angles uniformly distributed from

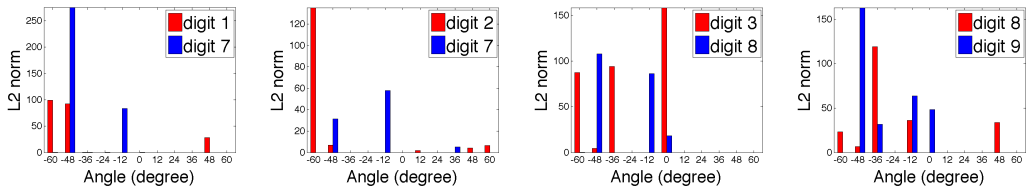

Figure 1: ℓ_2 norm of the parameter vectors for different angles over the 4 digit pairs.

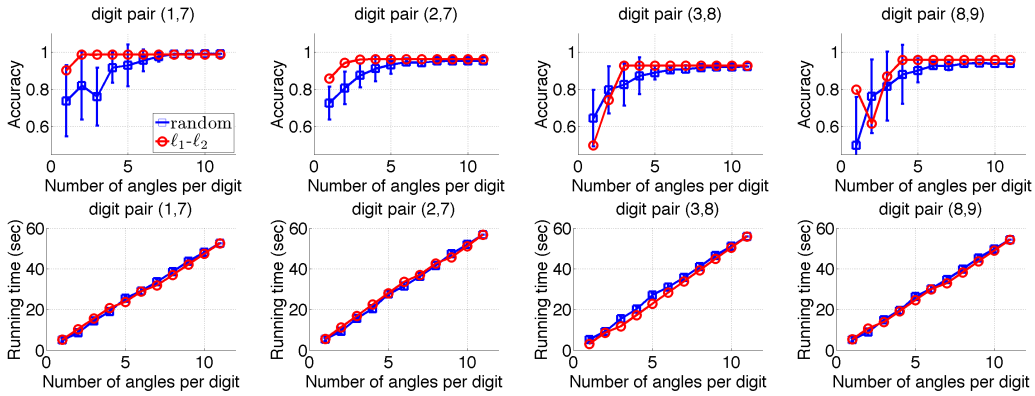

Figure 2: Comparison of prediction accuracy vs. angle budget and running time vs. angle budget of our approach and random selection. Curves for random selection are shown with standard deviation over 10 trials. The standard deviation for runtime plots is very small. We can see that our approach outperforms random selection of angles and is able to quickly achieve comparable accuracy as a complete model (using all angles).

$-60°$ to $60°$. We show that using our approach, only a few rotations are needed to achieve the recognition accuracy of using the full set of angles.

The joint feature vector $\phi(x, y, h) = [\mathbf{0}_{y(m+1)}\,;\, \theta_h(x)\ 1\,;\, \mathbf{0}_{(9-y)(m+1)}]$, where $\theta_h(x)$ is the image rotated by the angle specified by h, and then strung into a vector. To adapt our approach to this framework, we let the angle $h \in \mathcal{H} = \{h_0, h_1, \cdots, h_{10}\}$, where $\mathcal{H}$ is the set of 11 angles the digit can rotate, and induce a group-norm over the parameters corresponding to each angle.

We choose MNIST dataset [11] and compute exactly the same features as in Kumar et al. [10]. We use PCA to project each image to a 10 dimensional feature vector. We perform binary classification on four difficult digit pairs (1-7, 2-7, 3-8, 8-9). We vary the number of angles chosen for each digit from 1 to 11. For each angle budget, we select angles for a digit based on the magnitude of the ℓ_2-norm of the parameter vector corresponding to that angle. Angles with higher magnitude will be chosen first. As a baseline, we compare our approach to random angle selection. Given an angle budget, we randomly select a subset of angles. Only this subset of angles is used with the LSSVM trained model of Kumar et al. [10]. We perform 10 trials and take the average prediction accuracy to report results. We use $\lambda_p = 1$ for each group in our approach. We tried different values of C, and the prediction accuracies were fairly similar. We set $C = 1$.

Figure 1 shows the ℓ_2-norms of the parameter vectors for different angles in the 4 digit-pair experiments. Figure 2 shows how the prediction accuracy and feature computation time varies as angle budget increases. The feature computation time, which is proportional to the final prediction time, includes rotation time and PCA projection time.

We note a few key observations. First, in our approach, the ℓ_2-norms of the weight vector for many angles completely zero out, and only a subset of angles actually remain to contribute to final prediction. In the end, the trained model essentially selects 5, 8, 7, and 9 angles in total for digit pairs 1-7, 2-7, 3-8, and 8-9 respectively. This is a significant reduction from the hidden space of 22 angles per digit pair using the original model in Kumar et al. [10]. Second, our approach gives very similar prediction accuracy compared to the original approach with a full set of angles. Third, due

to the sparse solution of our model, using a maximum of 3, 4, 3, and 4 angles respectively for digit pair 1-7, 2-7, 3-8, and 8-9, we can achieve prediction accuracy similar to that using the full set of angles. However, using random selection, we need much higher number of angles per digit. Fourth, The running time increases roughly linearly in both approaches as angle budget increases. This is because the time to rotate an image and perform PCA for each angle is about the same. Overall, this shows that our method results in significantly faster inference at test-time without any loss in accuracy of the learnt model.

6 Conclusion

We address the problem estimating the parameters of an LSSVM model as well as discovering meaningful states for the latent variables. This allows us to control the model complexity and speed up inference time. We used ℓ_1-ℓ_2-norm regularization to approach this problem. Our experiments on handwritten digit recognition show that our approach is able to effectively reduce the size of latent variable state space and thus reduce the inference time with no loss of accuracy compared to using the full latent state space. In the future, we plan to investigate latent state learning with structured latent variables, where the state space may be exponentially large.

Acknowledgments

This work was supported by funds from Google Grants and from Shell Research.

References

[1] F. Bach. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems*, 2010. 2

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009. 1

[3] V. Chandrasekaran, P. Parrilo, and A. Willsky. Latent variable graphical model selection via convex optimization. In *48th Annual Allerton Conference on Communication, Control, and Computing*, 2010. 2

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43:129–159, 2001. 2

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977. 1, 2

[6] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 2

[7] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–451, 2004. 2

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010. 1, 2, 3, 4

[9] T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. Dual decomposition for parsing with non-projective head automata. In *Conference on Empirical Methods in Natural Language Processing*, 2010. 4

[10] P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 2010. 1, 4, 5

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, 2009. 2

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., 1990. 2

[14] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996. 2

[15] S. B. Wang, A. Quattoni, L.-P. Morency, and D. Demirdjian. Hidden conditional random fields for gesture recognition. In *IEEE conference on Computer Vision and Pattern Recognition*, 2006. 2

[16] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *International Conference on Machine Learning*, 2009. 1, 2, 3

[17] M. Yuan, M. Yuan, Y. Lin, and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006. 2, 3

[18] A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15:915–936, 2003. 1, 2, 3