Dynamic Structured Model Selection

David Weiss
University of Pennsylvania
Philadelphia, PA
djweiss@cis.upenn.edu

Benjamin Sapp
Google Inc.
Los Angeles, CA
bensapp@google.com

Ben Taskar
University of Washington
Seattle, WA
taskar@cs.washington.edu

Abstract

In many cases, the predictive power of structured models for for complex vision tasks is limited by a trade-off between the expressiveness and the computational tractability of the model. However, choosing this trade-off statically a priori *is suboptimal, as images and videos in different settings vary tremendously in complexity. On the other hand, choosing the trade-off dynamically requires knowledge about the accuracy of different structured models on any given example. In this work, we propose a novel two-tier architecture that provides dynamic speed/accuracy trade-offs through a simple type of introspection. Our approach, which we call dynamic structured model selection (DMS), leverages typically intractable features in structured learning problems in order to automatically determine' which of several models should be used at test-time in order to maximize accuracy under a fixed budgetary constraint. We demonstrate DMS on two sequential modeling vision tasks, and we establish a new state-of-the-art in human pose estimation in video with an implementation that is roughly* $23\times$ *faster than the previous standard implementation.*

1. Introduction

Computational budget constraints on inference in structured models for complex vision tasks force us to trade off between expressiveness and tractability of models. Choosing this trade-off statically for all images is suboptimal since images and image parts vary tremendously in complexity; choosing it dynamically requires meta-level assessment and prediction of performance of different models on given examples.

One concrete example of this tradeoff is managing the cost of low-level vision processing: dense evaluation of input features common in visual processing (e.g., normalized cut segmentation, optical flow, contour detection, etc.). In many object detection systems, the cost of computing these features is several times greater then the time spent on structured inference given the features. There is a tremendous variation of cost of low-level processing based on resolution and other accuracy parameters; choosing one global setting is often not sufficient for complex images and wasteful on simple ones. Another example is Markov order of temporal models for analyzing action in videos: in many parts of videos, simple zero or first-order models suffice, but complex cases require long-range, higher-order models.

We propose a novel two-tier architecture that provides dynamic speed/accuracy trade-offs through a simple type of *introspection*. The key idea is a division of labor between a hierarchy of models/inference algorithms (tier one) and meta-level model selector (tier two), which decides when to use expensive models adaptively, where they are most likely to improve the accuracy of predictions. The two tiers have complementary strengths: Tier one models provide increasingly accurate and more expensive inference over structured outputs. Tier two model-selectors use arbitrary sparsely-computed features and long-range dependencies, which would make inference intractable, in order to evaluate the outputs of the first tier and decide when to stop. While the first tier optimizes over a combinatorial set of possibilities using inference over densely computed features, the second tier simply evaluates proposals of the first. The advantage of this division is that both tiers are efficient and the second tier has more information than the first that allows it to reason about the success of the first.

We summarize the contributions of this work as follows:

- We propose *dynamic structured model selection* (DMS), a novel two-tier framework for creating faster and more accurate structured prediction systems. In section 3, we propose a simple greedy algorithm for maximizing test-time accuracy given a budgetary constraint.
- We apply our approach to two sequential modeling tasks: handwriting recognition (section 4.1) and articulated pose estimation in videos (section 4.2). On the handwriting recognition task, we use DMS to achieve a significant increase in accuracy over baseline while at the same time being nearly $3\times$ faster.
- On the pose task, we propose a novel sequence model

based re-ranker that utilizes the recently introduced model of [18] to achieve state-of-the-art accuracy on a benchmark dataset while being $23\times$ faster than the previous best method. We then apply DMS to achieve even faster times on a new, much larger benchmark dataset, reducing the re-ranking model runtime by a factor of $2\times$ with no decrease in accuracy for wrist localization.

2. Related work

Classifier cascades. A significant source of related work to the approach proposed here is the study of *classifier cascades*, in which a pre-determined series of classification models are sequentially applied to a test example. For binary classification problems with a highly skewed class distribution with very few true positives, these works typically obtain efficiency via an "early-exit" strategy: at test-time, simpler models may reject an example as negative without needing to evaluate the more complex models farther along the cascade. This approach has long been highly successful for object detection, using boosting methods to train the cascade of classifiers (e.g. [22]). However, feature computation cost was not incorporated specifically into the learning procedure until more recently (e.g., [17, 4, 7].) The most related recent work is [21], who define a reward function for multi-class classification with a series of increasingly complex models. Nonetheless, while the goals of these works are similar to ours–explicitly controlling feature computation at test time–none of the classifier cascade literature addresses either inference as a batch or the structured prediction setting.

Controlling test-time complexity. More directly related to the work in this paper is the work of [23, 6, 10]. [23] extend the idea of a classifier cascade to the structured prediction setting, wherein a series of increasingly complex structured models progressively refine the structured output space; the cascade is trained sequentially with the objective of retaining accuracy while maintaining sparsity in the output space for fast inference. Our approach is orthogonal to [23] in several significant ways; primarily, where [23] focus on learning the *models* efficiently, we focus on introducing additional meta-features to learn a *selector*. On the other hand, [6] also propose explicitly modeling the *value* of evaluating a classifier, but their approach requires modeling the entropy of a predicted class distribution and therefore does not apply to the structured setting in which there are exponentially many outputs. Also for multi-class, [11] apply an MDP approach to feature selection, while [10] apply imitation learning to speed up *inference* of a single syntatic parsing model; in contrast, we consider post-hoc evaluation of *multiple* structured models, and we learn a valuation function directly through regression instead of adopting reinforcement learning techniques. Furthermore, each of the above works analyze average per-example efficiency, whereas we propose Algorithm 1 for *batch* inference under an explicit budget. Finally, [3] propose an approximate dynamic program to determine *where* in video clips to apply an expensive algorithm for analysis (whereas we learn to *which* video clips to apply different models.)

Predicting model accuracy. The basic idea of predicting the accuracy of a model is not new. E.g. [1] attempt to predict various video analysis algorithm's performance (similar in spirit to the selector we propose), but based on measures of image quality rather than properties of model output. [9] propose an evaluator for human pose estimators, but only for single-frame images, and propose only learning "correct or not" coarse-level distinctions, whereas we attempt to predict a measure of the error of each model directly. In the speech community, [14] propose a parallel method of "dynamic model selection" in which several models are continually re-evaluated in an online fashion using a generative model, which is a very different setting than that we analyze here.

Pose estimation. There is considerable research into human pose estimation from 2D images; far more than we can review here. However, as state-of-the-art pose estimation can take upwards of several minutes per frame (e.g., [19, 13]) there is significantly less prior work on pose estimation in video clips. [2] use a single pictorial structure model for upper body in the different setting of extended signing sequences, where e.g. a static background over long periods leads to useful models of background. [16] propose a related approach to our method by stitching together N hypothesized poses per frame into video tracks, using $N = 300$ and evaluating their approach on 4 video sequences. In contrast, we use $N = 32$ proposals from [19] (assuming the scale and location of the person is known), learn additional sequence models using features computed over proposed tracks, and evaluate on hundreds of short clips from cinema. [19] tackles video clips from TV shows and is therefore the most relevant competitor to our approach, but (as we demonstrate) our method is both more accurate and an order of magnitude faster.

3. Dynamic Structured Model Selection

In this section, we introduce our approach to dynamic model selection and provide an overview of the algorithm. The core idea behind our approach is very simple: we learn to predict the *value* of choosing a more expensive model over a cheaper one, and we use predicted values to allocate computational resources at test time. In this fashion, we only apply the computationally expensive models to those examples where we will receive the most benefit.

Setup. We consider the problem of *structured prediction*, in which our goal is to learn a hypothesis mapping inputs $\mathbf{x} \in \mathcal{X}$ to outputs $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$, where $|\mathbf{x}| = \ell$ and $\mathbf{y}$ is a

Algorithm 1: Dynamic structured model selection.

input : Test set $\{\mathbf{x}^j\}_1^n$, hypotheses $h_1, \ldots, h_m$, costs $c_1, \ldots, c_m$, selector ν, and budget B.
output: Predictions $\mathbf{y}^1, \ldots, \mathbf{y}^n$.

Initialize $B' \leftarrow 0$, $\tau_j \leftarrow 1$, $\mathbf{y}^j \leftarrow h_1(\mathbf{x}^j)$, priority queue Q with priority-value pairs $\langle \nu(h_2, \mathbf{x}^j), j \rangle$;
while $B' < B$ *and* Q *is not empty* **do**
  Pop value j from Q with max priority;
  if $c_{\tau_j+1} \leq (B - B')$ **then**
    $\tau_j \leftarrow \tau_j + 1$;
    $B' \leftarrow B' + c_{\tau_j}$;
    $\mathbf{y}^j \leftarrow h_{\tau_j}(\mathbf{x})$;
    Insert $\langle \nu(h_{\tau_j+1}, \mathbf{x}^j), j \rangle$ into Q;
  end
end

ℓ-vector of K-valued variables, i.e. $\mathcal{Y}(\mathbf{x}) = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_\ell$ and each $\mathcal{Y}_i = \{1, \ldots, K\}$. We consider linear hypotheses of the form

$$h(\mathbf{x}; \mathbf{w}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \psi_{\mathbf{w}}(\mathbf{x}, \mathbf{y}), \tag{1}$$

where $\psi_{\mathbf{w}} = \mathbf{f}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{w}$ is a linear scoring function of a weight vector $\mathbf{w}$ and feature vector $\mathbf{f}(\mathbf{x}, \mathbf{y})$. Note that because there are exponentially many $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$, in order for the inference problem (1) to be computationally feasible, $\mathbf{f}$ is typically decomposed into a piece-wise sum over subsets of $\mathbf{y}$ (e.g. section 4 in this paper) so that (1) can be computed or approximated efficiently.

For the dynamic model selection problem we consider here, we assume that we are given a *set* of models, $h_1, \ldots, h_m$, that require some amortized cost $c_1, \ldots, c_m$ to evaluate on any given example $\mathbf{x}$. Given a fixed ordering of the models, we define the *value* of evaluating model h_i on example $\mathbf{x}$,

$$V(h_i, \mathbf{x}, \mathbf{y}) = \mathcal{L}(h_{i-1}(\mathbf{x}), \mathbf{y}) - \mathcal{L}(h_i(\mathbf{x}), \mathbf{y}), \tag{2}$$

where $\mathcal{L}(\mathbf{y}, \mathbf{y}')$ is a non-negative loss function. Note that a positive value signifies a decrease in loss, while a negative value signifies an increase is loss. (While more expensive models usually increase accuracy on *average*, in practice we find that there are many examples where the more expensive features hurt performance.) Our proposed goal for meta-learning is to learn a *selector* $\nu(h_i, \mathbf{x})$ to approximate the value function.

Batch inference. Given a set of models $h_1, \ldots, h_m$, a selector ν, and a test set $\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$, we perform inference using Algorithm 1. The algorithm is very simple: we greedily optimize the total predicted value,

$$J(\tau_1, \ldots, \tau_n, \eta) = \sum_{j=1}^{n} \sum_{i=1}^{\tau_j} \nu(h_i, \mathbf{x}^j), \tag{3}$$

where τ_j is the stopping point on the j'th example. Note that even if all predicted $\nu(h_i, \mathbf{x}^j)$ are negative, Algorithm 1 continues to greedily choose more expensive models as long as budget is available.

Learning the selector. In order for Algorithm 1 to succeed, the selector ν must provide a useful estimate of the value V. We formulate the selector as a linear function of *meta-features* computed on the output of the models. The key idea is that, while the feature generating function $\mathbf{f}$ for a structured prediction model decomposes over subsets of $\mathbf{y}$ in order to maintain feasible inference, the meta-features ϕ need only be computed efficiently for the specific outputs $h_1(\mathbf{x})$ through $h_{i-1}(\mathbf{x})$. We provide detail on the specific meta-features used in each application in section 4.

We learn the selector by learning a weight vector β to approximate the value function. On the training set, we first compute the value for every model on every training example. We then minimize the following ℓ_2-regularized squared loss over a training set,

$$\frac{\lambda}{2}||\beta||_2^2 + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(V(h_i, \mathbf{x}^j, \mathbf{y}^j) - \beta^\top \phi(\mathbf{x}^j, h_{1:i-1}) \right)^2 \tag{4}$$

where the function ϕ is a function generating *meta-features* that takes all predictions $h_1, \ldots, h_{i-1}$ as input and λ is a regularization parameter chosen via cross validation.

Preventing overfitting. Some care is needed when learning the selector in order to avoid re-using the same training set for learning both the models and the selector; i.e. if h_i was trained on example $(\mathbf{x}^j, \mathbf{y}^j)$, we expect $\mathcal{L}(h_i(\mathbf{x}^j), \mathbf{y}^j)$ to be unrealistically low and thus the value may be unrepresentative of the test distribution. However, a simple N-fold cross-validation scheme suffices to prevent this. We train N different models $h_i^1, \ldots, h_i^N$ in the standard way and use the model not trained on example j when evaluating (4).

4. Application to Sequential Prediction

In this section, we discuss two applications of our dynamic structured model selection framework to computer vision problems: handwriting recognition and human pose estimation from 2D video. In both settings, DMS provides for a far more efficient structured prediction model.

Sequence model. In both settings, we use the following standard *linear-chain* structured prediction model. Given an input $\mathbf{x}$ of length ℓ, we wish to predict a sequence of discrete K-valued outputs $y_1, \ldots, y_\ell$, where $y_i \in \{1, \ldots, K\}$. For the handwriting recognition problem, each y_i corresponds to one letter of the written word; for human pose estimation, each y_i corresponds to one of K possible predicted poses. For any given sequence $y_1, \ldots, y_\ell$, the combined score of

the sequence is the sum of unary and pairwise potentials,

$$\psi_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}, y_i) \cdot \mathbf{w} + \sum_{i=2}^{n} \mathbf{f}(\mathbf{x}, y_{i-1}, y_i) \cdot \mathbf{w}, \quad (5)$$

where $\mathbf{w}$ is a (learned) weight vector and $\mathbf{f}$ is a feature generating function. At test time, we can efficiently make predictions using the Viterbi algorithm to find the state sequence that maximizes (5).

Learning. Given a set of n training pairs $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^n$, we learn $\mathbf{w}$ by minimizing the following structured max margin objective,

$$\min_{\mathbf{w}} \frac{\lambda}{2} ||\mathbf{w}||_2^2 + \frac{1}{n} \sum_{j=1}^{n} H_{\mathbf{w}}^j, \quad (6)$$

where $H_{\mathbf{w}}^j$ is the structured hinge loss on the j'th example,

$$H_{\mathbf{w}}^j = \max_{\mathbf{y}} \psi_{\mathbf{w}}(\mathbf{x}^j, \mathbf{y}) + \Delta(\mathbf{y}^j, \mathbf{y}) - \psi_{\mathbf{w}}(\mathbf{x}^j, \mathbf{y}^j), \quad (7)$$

and where we set $\Delta(\mathbf{y}, \mathbf{y}')$ to be the Hamming loss to measure distance between state sequences. For the handwriting recognition task, we approximately optimize (6) using the structured perceptron algorithm, which has been shown to work well for this task [23]. For the video pose estimation task, we optimize (6) directly using the recent stochastic Frank-Wolfe block-coordinate descent method of [12], which we found to be more robust. In both cases, we choose λ and a stochastic stopping time through cross-validation using a development set.

4.1. Handwriting Recognition

We first apply our method to the handwriting recognition dataset of [20]. For our purposes, this dataset represents a "best case" type of scenario: while there are thousands of examples of handwritten words in the dataset, examples were generated by enlisting many different people to rewrite the same list of less than a hundred unique words. Therefore we expect high-order features (e.g., 5-grams) to be very informative, but these features are computationally infeasible to include in the linear-chain model directly. Instead, they are ideally suited as informative meta-features for the selector. In this way, we the selector is ideally suited to direct computation at test time, and a very fast and effective method is the result.

Models. We use three different models for the handwriting recognition problem, differing only in the unary term features of the sequence model. In each model, we have K^2 binary pairwise features $\mathbf{f}_{k,k'}(y_{i-1}, y_i) = \mathbf{1}[y_{i-1} = k, y_i = k']$ as well as a unary feature for every binary pixel activation in the 16×8 image. The second model h_2 computes a coarse Histogram of Gradients (HoG) in 3×3 bins and the third model h_3 additionally computes HoG in smaller 2×2 bins. Since the pixels are given as in the input, and HoG takes constant time for fixed input size, we have $c_1 = 0, c_2 = 1, c_3 = 1$.

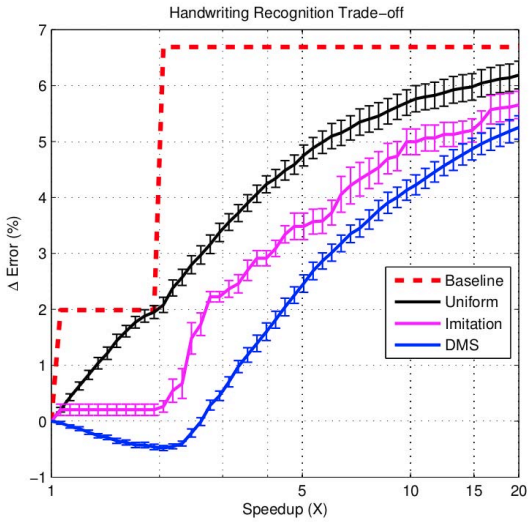

Figure 1. Trade-off on handwriting recognition task, displayed as a function of the efficiency speedup w.r.t the final model (Speedup) vs. the change in error rate w.r.t the final model (ΔError). To draw each curve, we sweep the budget B or tradeoff parameter η until we find a point with at least the target speedup and record the error rate. Our approach (DMS) significantly outpeforms imitation learning, yielding an error rate *below* that of the final model. The Uniform method consists of picking which element to expand uniformly at random until all examples use the same model, and the Baseline method consists of picking a single entire fixed stage of models *a priori*.

Selector Meta-features. We use two sets of meta-features for the selector. The first are computed from the output of $h_i(\mathbf{x})$, consisting of the relative difference in the scores of the top two outputs and the average of the mean, min, and max entropies of the marginal distributions predicted by h_i at each position in the sequence. The second set of meta-features count the number of times an n-gram was predicted in $h_i(\mathbf{x})$ that occured zero times in the training set, computed for $n = 3, 4, 5$. Both of these features take neglible time to compute compared to the HoG computation.

Imitation learning baseline. We compare to an alternative method for dynamic model selection inspired by imitation learning methods for feature selection [8]. For this baseline, we first pick a trade-off parameter η, and then for each example $(\mathbf{x}^j, \mathbf{y}^j)$ in the training set independently decide the optimal stopping point,

$$\tau_j^\star = \operatorname*{argmin}_{\tau} \mathcal{L}(h_\tau(\mathbf{x}^j), \mathbf{y}^j) + \eta \cdot c_\tau. \quad (8)$$

These stopping points define an optimal policy $\pi^\star(i, \mathbf{x}^j) = \mathbf{1}\left[\tau_j^\star < i\right]$, where the policy $\pi^\star(i, \mathbf{x}^j)$ is 1 if computation should continue on example $\mathbf{x}^j$ after model i and 0 otherwise. We then learn an approximate policy $\pi(i, \mathbf{x}^j)$ using

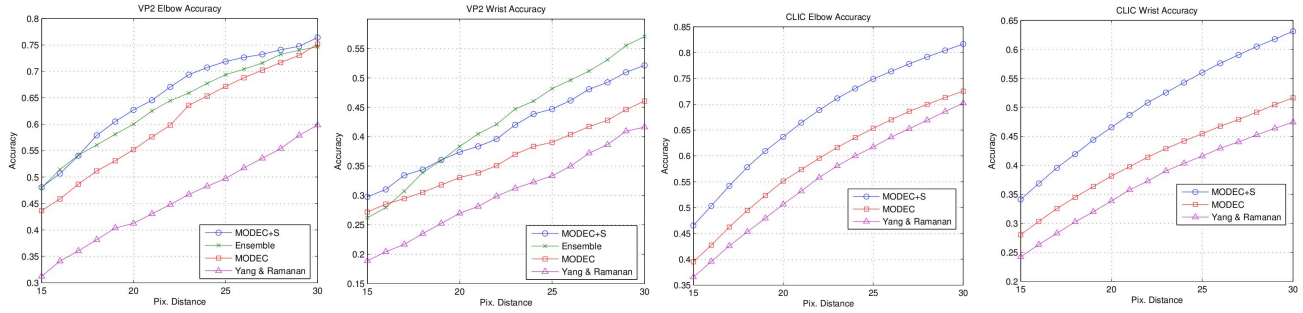

Figure 2. Exceeding state-of-the-art on VideoPose2 [19] and CLIC datasets. The MODEC+S model matches the much slower ensemble approach of [19] (Ensemble) in elbow accuracy and exceeds it in wrist accuracy (at high precisions), and provides a significant boost in performance over MODEC. However, MODEC is still competitive and more accurate than previous state-of-the-art [24] (Yang & Ramanan) on both datasets.

a linear SVM classifier trained with the same meta-features as the selector uses; we generated training data points by sampling all trajectories generated by the optimal policy on the training set. Note that for each η, we obtain one error/cost trade-off point. For all experiments, we swept η across many values to generate all possible unique optimal policies for each fold of cross validation.

Results. We visualize the trade-off between error rate and computation time on the handwriting recognition task is given in Figure 1. All results are plotted in terms relative to the third, most expensive model. Our approach significantly outperforms imitation learning, and both imitation learning and our approach provide a significant increase in efficiency over choosing one of the models *a priori* or uniformly at random. Most significantly, the model/selector approach leads to a predictor that is more accurate than the final model (due to choosing the most accurate models first) while yielding a roughly 2.5$\times$ speedup compared to the most expensive model. Note that we also computed a different uniform baseline where examples were advanced to the next model uniformly at random without ensuring that all examples reached the same stage; we found this performed equivalent or worse than the baseline shown.

Imitation learning vs. DMS. Besides the improvement in accuracy and speedup, there are several practical advantages of DMS over the imitation learning baseline. Unlike DMS, each choice of η yields a different policy; in order to sweep a curve, we must re-run learning for every point we wish to generate on the curve. Furthermore, imitation learning as defined using equation 8 does not guarantee that computation over a batch of test examples will run within a fixed budget; each choice of η yields a fixed trade-off that will approximately run at some budget that is a function of the interaction between computation and accuracy on the given training set.

4.2. Human Pose Estimation in Video

Our approach to video pose estimation can be summarized as follows. We propose a simple bigram linear-chain model, one per arm. The model consists of 32 states per frame. We adapt the efficient and current state-of-the-art MODEC pose model [18] to generate the states: each state corresponds to the highest scoring prediction of one of the 32 MODEC sub-models. Next, given the set of states for each clip in our training database, we learn to predict a path through the states using high level features such as color and flow consistency. The resulting model is both more accurate than previous state-of-the-art and is roughly 23$\times$ faster. We then apply dynamic model selection to choose the features in the sequence models on-the-fly for even greater efficiency gains on a new, large dataset.

MODEC Proposals. A key relevant modeling innnovation in the MODEC method is the joint learning of a mixture of 32 articulated part-based models. Each mixture component, or *mode*, represents a different canonical pose. To generate our 32 states, we find the argmax arm configuration for each of the 32 modes in MODEC. Note that MODEC models each arm as a separate pose model, but chooses a single mode for each arm based on a combined compatiblity score between the two poses; for our purposes, we ignore the compatibility score and take the 32 separate predictions for each arm independently. Experimentally, we find that with 32 states per arm, at least one state is typically very close to the true arm pose for a given image (i.e. on the VideoPose2 [19] dataset, greater than 80% of elbows and wrists are within 20 pixels of the ground truth, on average.)

MODEC+S sequence model. Given a video sequence, we generate 32 states for each arm for each frame independently using the MODEC model. The problem then becomes selecting which of the 32 poses for each arm and frame to choose. We apply a standard linear-chain bigram model for this task. Let y_i be state at frame i; for each assignment to y_i we have a corresponding MODEC argmax pose on the i'th frame, which we denote $p_i(y_i)$. For

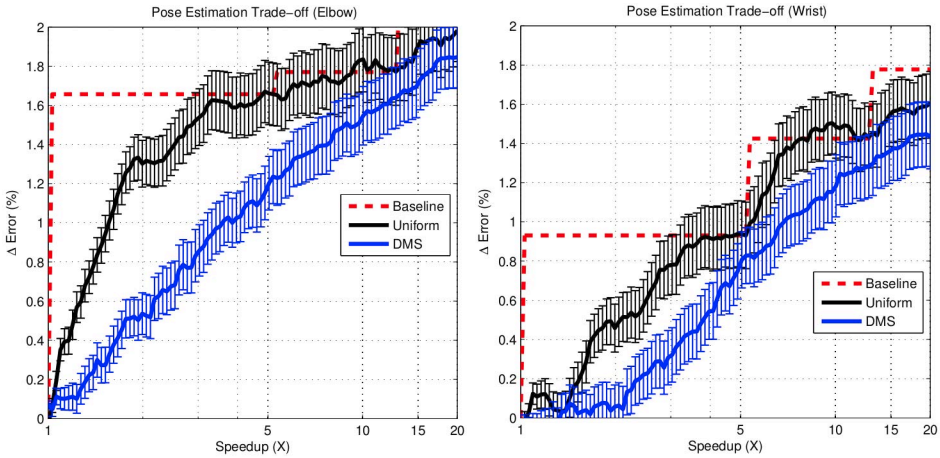

Figure 3. Dynamic model selection on the CLIC dataset. See caption of Figure 1 for explanation of axes/baselines. DMS provides a significant increase in speedup with very little accuracy cost compared to picking elements uniformly at random; e.g. for elbows, a $2\times$ speedup can be obtained for hardly any accuracy cost, while a $5\times$ speedup with DMS can be obtained for the same accuracy of a $3\times$ speedup when picking uniformly at random.

each state y_i in the i'th frame we allow transitions from the 5 closest states y'_{i-1} in the previous frame, as measured by the distance in joint configuration space $||p_i(y_i) - p_{i-1}(y'_{i-1})||_2$. This yields a total of 160 possible transitions for each frame. At test time, we can efficiently make predictions using the Viterbi algorithm to find the state sequence that maximizes (5) with practically neglible runtime due to the tiny size of the state and transition space. We call our approach MODEC+S.

Learning. We learn the scoring functions ψ from a set of n training examples as follows. Given a video clip $\mathbf{x}^j$ and a labeled pose p_i^j for each frame i in $\mathbf{x}^j$, we define the ground truth state y_i^j to be the state with the closest pose $y_i^j = \text{argmin}\, ||p_i(y_i) - p_i^j||_2$. This results in the training set $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^n$ which we use as the basis for the rest of our analysis.

Features. As in the handwriting recognition task, we use a fixed hiearchy of features to create a series of four increasingly complex base models. The first model uses unary features consisting of a prior term and the normalized MODEC score for each mode, and binary features consisting of a (mode,mode) transition prior and several kinematic terms (angular joint and limb velocities and x, y joint location velocities). The second model adds an image-dependent pairwise term, the χ^2-distance between color histograms of the predicted arm locations from one frame to the next. The third model adds an image-dependent unary term; each image is quickly segmented into superpixels using [5], and we compute the intersection-over-union (IoU) score between the predicted arm rectangles and superpixels selected by the rectangles. Finally, the fourth model computes a very fast and coarse optical flow using [15]; we obtain an estimate of the *foreground* flow by subtracting the median flow outside the target bounding box. We then compute a flow-based pairwise feature as follows: for each predicted arm location in the first frame, we shift each arm pixel by its estimated flow to produce a predicted arm location in the next frame, and compute the IoU between the flow-shifted arm and each possible predicted arm location in the next frame. Although the computational cost depends on the size of the input images, for the CLIC dataset (described below), we compute the per-frame amortized costs of each model (in seconds)[1] to be $c_1 = 0, c_2 = 0.41$, $c_3 = 1$, and $c_4 = 5.2$ (optical flow is by far the most expensive feature of MODEC+S).

Meta-features. We use similar meta-fetures as in the handwriting recognition setting (n-gram occurences and distribution and entropy of the sequence model marginals) with one set of additional image-dependent features. For every n-gram in the image, where $n = 2k$, we compute the mean and max χ^2 distance between a center frame predicted arm location and the k frames before and after. The feature is then the average number of times these distances exceed 0.5, indicating a significant difference between the predicted arm color of the center frame and the surround frames. For $k = 1, 2, 3$, these features yield a total of 40 features for the selector. Finally, for computing metafeatures at model level i, we also compute the features of level $i - 1$ and the change in these features from $i - 1$ to i. The per-frame amortized cost of evaluation is 0.014 seconds.

A new dataset. We introduce a new publicly available dataset, Clips Labeled in Cinema (CLIC). This dataset consists of 362 annotated clips from the same movies as the Frames Labeled in Cinema (FLIC) dataset from [18], for a total of roughly 15,000 individual frames. The annotations

[1] All computation was carried out on a AMD Opteron 4284 CPU @ 3.00 GHz with 16 cores.

were obtained from a Amazon Mechanical Turk crowdsourcing annotation tool. Due to the difficulty of labeling an entire video sequence, we labeled a significantly larger pool of video clips, but kept only those frames where inter-annotator agreement was within the 90th percentile of the distribution of agreement across all frames in all video clips. Each clip is between 10 and 61 frames in length, with a median clip length of 46 frames. Although all of the data will be publicly available, for our experiments in the following, we selected half of the clips that contained the most arm motion to provide for a more challenging dataset.

4.2.1 Evaluation of MODEC+S

Experimental setup. Before applying DMS, we evaluated the utility of the final, most complex MODEC+S model for articulated pose estimation in video. We first compared to the approach of [19] on the VideoPose2 (VP2) dataset [19], which represents state-of-the-art in human pose estimation in challenging videos. (We also compare to state-of-the-art single frame inference, as represented by [24]). For the VP2 dataset we used the same train/test partitioning of the VP2 dataset as [19]. We next compared our approach on the new CLIC dataset. Due to excessive runtime, it was infeasible to apply [19] to this much larger dataset, so we compared to single-frame methods [18] and [24]. To evaluate on CLIC, we divided the dataset into two halves so that different movies were contained in different halves. Since our new dataset CLIC uses the same movies as the FLIC dataset in [18], we re-trained MODEC on the corresponding movies from FLIC contained in each half and tested MODEC on the other half. To train and test MODEC+S, we used these test evaluations of MODEC as input, again training on one half of the data and testing on the other. Note that for all experiments, we ignored the human detection problem and fed all algorithms pre-localized and scaled images using the annotations in the data, although only MODEC and to choose [19] explicitly take advantage of this fact.

Results. The results are summarized in Figure 2, and qualitative results are presented in Figure 5. On VP2, MODEC+S achieves similar or better accuracies to [19], but whereas the downloadable code package for [19] took 367 seconds/frame of computation time, MODEC+S takes only 16 seconds/frame, a 22.9× speedup. In particular, the new model is significantly more accurate on the wrist than either the previous best or the single-frame MODEC model. However, even the single-frame MODEC is close to state-of-the-art (after smoothing), which is even faster than our approach, and [24] is faster still, though not as competitive. On CLIC, MODEC+S dominates the other methods by an even more significant margin, and all accuracies are generally higher than VP2 for all methods.

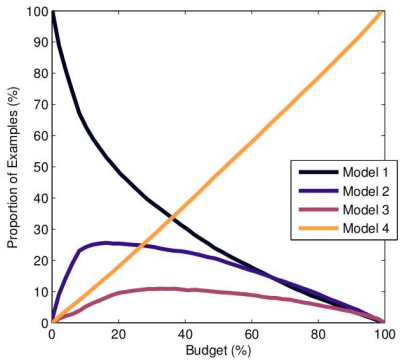

Figure 4. Expansion distribution of DMS on the CLIC dataset. For each % of budget used, the distribution of stopping points for the batch examples between the four possible models is shown. As can be seen, our approach quickly begins using the most expensive model in order to obtain higher accuracy for less overall computational cost.

4.2.2 Evaluation of Dynamic Model Selection

Experimental setup. We evaluated our dynamic model selection framework on the CLIC dataset with 200 random partitions of the dataset. For each partition, we used 70% of the data for training, 10% as development, and 20% as test. Within the training set, we ran 3-fold cross-validation to generate model predictions for learning the selector as described in section 3. When learning the selector, we focused on minimizing wrist test error, counting as an error any frame that the wrist was not localized to within 20 pixels. We also smoothed the predictions of each model before passing them to the selector since we found this improved the overall accuracy of the system.

Results. The results are shown in Figure 3. For wrist localization, our approach was able to obtain a 2× speedup for little to no accuracy cost, and maintain a significant speedup compared to the uninformed model selection baseline. For elbow localization, our approach yields a speedup of 5× at the same accuracy cost as an uninformed 2× speedup, a significant improvement. Note that these improvements include the slight additional cost of evaluating the DMS meta-features. We also investigated whether or not Algorithm 1 was choosing models to use by simply choosing the cheapest first (Figure 4), which we found not to be the case.

5. Conclusion

We presented *dynamic structured model selection* (DMS), a simple but powerful meta-learning algorithm that leverages typically intractable features in structured learning problems in order to automatically determine which of several models should be used at test-time in order to maximize accuracy under a fixed budgetary constraint. In two domains, we found significant improvements in accuracy and efficiency compared to alternative or uninformed ap-

Figure 5. Qualitative results on the CLIC dataset. Shown are the predictions of the 4 base models (blue, cyan, yellow, red, respectively). The optical flow based features (red) are often times significantly more accurate than the other features.

proaches. We also established a new state-of-the-art in human pose estimation in video with an implementation that is 23× faster than the previous standard implementation.

Future work. Our results suggest that two-tier re-ranking style approaches are indeed a powerful technique to increase the efficiency and discriminative power of structured prediction systems. Nonetheless, the DMS approach outlined here could be improved in several key ways: in future work, we intend to explore optimal strategies for constructing a model set, unordered model sets, and better greedy batch inference strategies.

References

[1] A. Bedagkar-Gala and S. Shah. Joint modeling of algorithm behavior and image quality for algorithm performance prediction. In *BMVC*, 2010.

[2] P. Buehler, M. Everingham, D. Huttenlocher, and A. Zisserman. Upper body detection and tracking in extended signing sequences. *IJCV*, 95:180–197, 2011.

[3] D. Chen, M. Bilgic, L. Getoor, and D. Jacobs. Dynamic processing allocation in video. *PAMI*, 33(11), 2011.

[4] M. Chen, Z. Xu, K. Weinberg, O. Chapelle, and D. Kedem. Classifier cascade for minimizing feature evaluation cost. In *AISATATS*, 2012.

[5] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004.

[6] T. Gao and D. Koller. Active classification based on value of classifier. In *NIPS*, 2011.

[7] A. Grubb and D. Bagnell. Speedboost: Anytime prediction with uniform near-optimality. In *AISTATS*, 2012.

[8] H. He, H. Daumé III, and J. Eisner. Imitation learning by coaching. In *NIPS*, 2012.

[9] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C.V.Jawahar. Has my algorithm succeeded? An evaluator for human pose estimators. In *ECCV*, 2012.

[10] J. Jiang, A. Teichart, H. Daumé III, and J. Eisner. Learned prioritization for trading off accuracy and speed. In *NIPS*, 2012.

[11] S. Karayev, T. Baumgartner, M. Fritz, and T. Darrell. Timely object recognition. *NIPS*, 2012.

[12] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *ICML*, 2013.

[13] L. Ladický, P. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *CVPR*, 2013.

[14] P. Lanchantin and X. Rodet. Dynamic model selection for spectrael voice conversion. In *Interspeech*, 2010.

[15] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.

[16] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011.

[17] V. Raykar, B. Krishnapuram, and S. Yu. Designing efficient cascaded classifiers: tradeoff between accuracy and cost. In *SIGKDD*, 2010.

[18] B. Sapp and B. Taskar. MODEC: Multimodal decomposable models for human pose estimatino. In *CVPR*, 2013.

[19] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *CVPR*, 2011.

[20] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*, 2003.

[21] K. Trapeznikov and V. Saligrama. Supervised sequential classification under budget constraints. In *AISTATS*, 2013.

[22] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 57(2):137–154, 2002.

[23] D. Weiss and B. Taskar. Structured prediction cascades. In *AISTATS*, 2010.

[24] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.