# Learning Optimal Subsets with Implicit User Preferences

**Yunsong Guo** and **Carla Gomes**
Department of Computer Science
Cornell University
{guoys, gomes}@cs.cornell.edu

## Abstract

We study the problem of learning an optimal subset from a larger ground set of items, where the optimality criterion is defined by an unknown preference function. We model the problem as a discriminative structural learning problem and solve it using a Structural Support Vector Machine (SSVM) that optimizes a "set accuracy" performance measure representing set similarities. Our approach departs from previous approaches since we do not explicitly learn a pre-defined preference function. Experimental results on both a synthetic problem domain and a real-world face image subset selection problem show that our method significantly outperforms previous learning approaches for such problems.

## 1 Introduction

Many interesting problems can be abstracted as finding an optimal subset of items that maximizes the utility induced by a preference function defined over individual items or sets of items, given a larger ground set. Examples range from real-world scenarios, such as recommending movies from a list of titles or buying the weekly groceries from a supermarket among aisles of alternatives, to the more general binary knapsack and set covering problems. We note that selecting a subset of items from a larger ground set based on a preference function is different from ranking the items and selecting the top items to include in the subset, since choices may interact with each other. For example, when selecting what to take on a backpacking trip, a bottle of water can be most essential (thus ranked highest), but two bottles of water may be less preferred than a bottle of water and an apple. This example is also known as the *portfolio effect*. In addition, because items in the optimal subset are normally equally preferred, learning a total ranking of items may not be possible. The problems of preference representation have been studied in the literature [Brafman *et al.*, 2006a; desJardins and Wagstaff, 2005; Freund *et al.*, 2003; Domshlak and Joachims, 2005]. The problem of computing the optimal subset, assuming a completely known preference function, has also been studied [Binshtok *et al.*, 2007; Price and Messinger, 2005]. Recent work has also shown how, given a pre-defined parameterized preference function, the parameters can be automatically learned from historical data. A heuristic greedy method is then used to select an item subset from a larger ground set, trying to maximize the utility associated with the pre-defined (with learned parameters) preference function [desJardins *et al.*, 2006].

In our work we assume that for each ground set of items, a *single* most preferred optimal subset exists according to some (unknown) preference function, and we want to build a model that can predict the subset as close as possible to the optimal one. Given the wide range of (unknown) preference functions, it is difficult to define a priori the *right* parameterized family of functions. In order to circumvent this intermediate step, we discriminatively learn to predict the optimal subsets from historical datasets, using structural machine learning, without explicitly assuming a pre-defined parameterized preference function over sets of items. As no explicit preference function is assumed during learning, our performance evaluation is based on a set similarity measure. To the best of our knowledge, our approach is the first one to formulate this problem as the task of learning to predict an optimal subset that maximizes a set similarity measure, without an explicit preference representation. We solve the structural learning for the optimal subset selection problem using Structural Support Vector Machines, for which we guarantee that learning is performed in polynomial time and testing inference is exact, even with an exponential number of candidate subsets. Our experiments are conducted using both a synthetic dataset and a real-world face image dataset. A total of 8 subset selection tasks are used for the evaluation. Our method outperforms the previous approach significantly in all tasks in terms of several set similarity measures.

## 2 Optimal Subset Learning

### 2.1 Problem Formulation

Let $\mathcal{X}$ denote the space of possible ground sets, and $\mathcal{S}(x)$ the space of subsets given a ground set $x \in \mathcal{X}$. We assume all possible subsets are legitimate so $|\mathcal{S}(x)| = 2^{|x|}$. In addition, let $\mathcal{S}(\mathcal{X}) = \bigcup_{x \in \mathcal{X}} \mathcal{S}(x)$. An example instance is illustrated in Figure 1,where the ground set consists of six face images and the two optimal subsets correspond to two very different kinds of preferences on the same ground set.

The learning problem can be formulated as follows: we

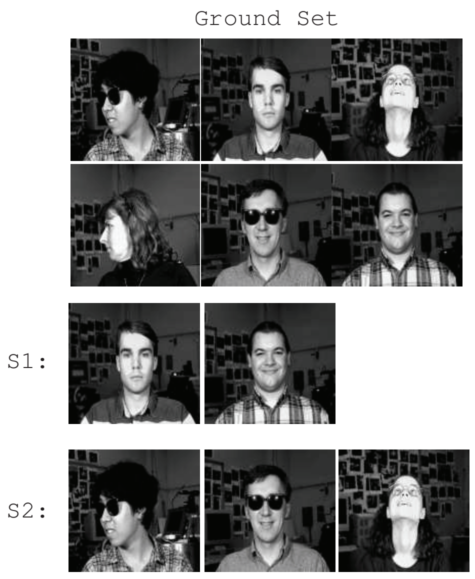

Figure 1: Optimal Subset Example
The two subsets correspond to two user preferences: $S1$-all face images that can be used as passport photos (looking straight and no sunglasses); $S2$-if there are more angry faces than happy faces, choose all angry faces, otherwise all happy faces that are with sunglasses or looking up.

are given $n$ supervised training sets ($x_i$,$y_i$), where $x_i$ is the $i^{th}$ ground set of items and $y_i$ is the optimal subset computed from some implicit user preferences according to the ground set $x_i$ with $y_i \subseteq x_i$; each item $I \in x_i$ is represented as a real valued feature vector; the goal is to learn a function F: $\mathcal{X} \rightarrow \mathcal{S}(\mathcal{X})$ which, given a new unseen ground set $x$, predicts the subset $\widehat{y}$ of $x$ that minimizes the expected value of a set similarity loss $\Delta(y, \widehat{y})$, where $y$ is the optimal subset computed from $x$ if we knew the underlying implicit user preferences. As the distribution of instances in $\mathcal{X}$ is largely unknown, we instead construct the function $f$: $\mathcal{X} \rightarrow \mathcal{S}(\mathcal{X})$ that minimizes the empirical risk

$$R^{\Delta}(f) = \frac{1}{n}\sum_{i=1}^{n} \Delta(y_i, f(x_i)) \tag{1}$$

where $\Delta : \mathcal{S}(\mathcal{X}) \times \mathcal{S}(\mathcal{X}) \rightarrow \mathbb{R}$ is the loss function that penalizes predicting $f(x_i)$ as the optimal subset $y_i$.

### 2.2 The Loss Function $\Delta$

We define the loss function $\Delta(\widehat{y}, y_i) = 1 - \sigma(\widehat{y}, y_i)$ with $\sigma(y_1, y_2)=\frac{|y_1 \cap y_2|}{\max(|y_1|,|y_2|)}$, assuming $\max(|y_1|, |y_2|)>0$. $\sigma(y_1, y_2)$ is our measure for set similarity which we refer to as *set accuracy*. Note that $\sigma(y_1, y_2)$ is the minimum of the precision and recall scores given sets $y_1$ and $y_2$. Averaged over multiple instances, set accuracy is upper bounded by the average precision and recall scores. This similarity measure differs from two other often used set similarity measures, namely the Jaccard Index defined as $J(y_1, y_2) = \frac{|y_1 \cap y_2|}{|y_1 \cup y_2|}$, and the Dice's coefficient $D(y_1, y_2) = \frac{2\times|y_1 \cap y_2|}{|y_1|+|y_2|}$, which can be shown to be equal to the F1-score in our problem setting.

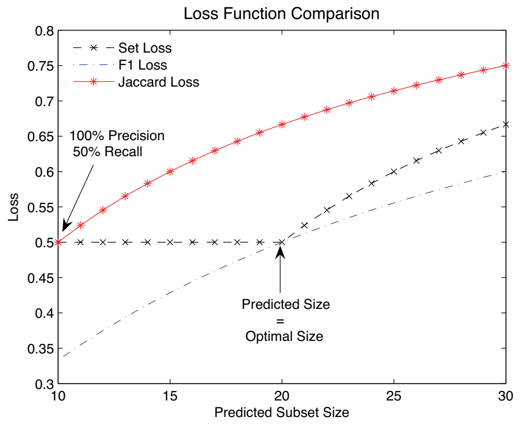

Figure 2: Loss Function Comparison

The differences of the loss functions are illustrated in Figure 2, where the true optimal subset size is 20, and the size of the overlap of the optimal and predicted subsets is a constant 10. Both Jaccard and F1 loss increases as the predicted size increases, while the set loss is indifferent until the size of the predicted subset reaches that of the optimal subset. We prefer set loss in our problem setting because it encourages the model to aggressively select more items into the subset to achieve a larger overlap between the predicted set and the optimal one when the sizes of two sets do not differ significantly. For example, consider the instance where the true optimal subset is {1,2,3} for a much larger ground set, and the two predicted subsets are $\widehat{y}_1 = \{1\}$ and $\widehat{y}_2 = \{1, 2, 4, 5, 6\}$. Intuitively we prefer $\widehat{y}_2$ over $\widehat{y}_1$ as it contains more information included in the optimal set. The Jaccard loss and F1-loss will have the same penalty on both predicted subsets, while set loss penalizes $\widehat{y}_1$ more. Nevertheless, we will also include the Jaccard Index and the F1-score as performance evaluation metrics in the experiment section.

### 2.3 Structural Learning Formulation

The optimal subset learning problem naturally falls into the structured learning regime, as the input and output are sets of objects that can interact with other objects in the ground set. We propose to solve it using structural SVM for optimal subset learning (SVM$_{OS}$), formulated as follows:

OPTIMIZATION PROBLEM SVM$_{OS}$

$$min_{w,\xi\geq 0} \quad \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \tag{2}$$

$$s.t. \quad \forall 1 \leq i \leq n, \forall \widehat{y} \in \mathcal{S}(x_i):$$

$$w^\top \phi(x_i, y_i) \geq w^\top \phi(x_i, \widehat{y}) + \Delta(\widehat{y}, y_i) - \xi_i \tag{3}$$

$\phi(x, y)$ in (3) is the feature map jointly defined by a ground set and a candidate subset. It is a well-known result in the multiclass setting that for any feasible solution $w$ and $\xi$, $\sum_{i=1}^{n} \xi_i$ is an upper bound on the training loss, independent of the prediction function $f$. The same result holds for our structural learning setting, as we can use simple arithmetic to show that

$$\sum_{i=1}^{n} \xi_i \geq \sum_{i=1}^{n} \Delta(f(x_i; w), y_i) \tag{4}$$

After the weight vector $w$ is optimized in the above quadratic formulation, the linear discriminant score for predicting subset $y$ as the optimal subset for ground set $x$ is $\mathcal{T}(x, y; w) = w^\top \phi(x, y)$, and the function $f$ we use to predict subsets for an arbitrary ground set is:

$$f(x; w) = \arg\max_{y \in \mathcal{S}(x)} \mathcal{T}(x, y; w) \tag{5}$$

## 2.4 Optimizing for SVM$_{OS}$

There are an exponential number of constraints ($\sum_{i=1}^{n} 2^{|x_i|}$) in (3), which is the main difficulty in solving SVM$_{OS}$ optimally. We use Algorithm 1, a cutting plane algorithm proposed in [Tsochantaridis *et al.*, 2005] to solve SVM$_{OS}$ within a tolerance $\varepsilon$ from optimality. Algorithm 1 iteratively adds the most violated constraint to the initially empty working set of constraints to be optimized.

Theoretically, given $\overline{R} = max_{i,y}\|\phi(x_i, y_i) - \phi(x_i, y)\|$, $\overline{\Delta} = \max_{i,y} \Delta(y_i, y)$ and a fixed $\varepsilon$, Algorithm 1 stops after adding at most $\max\{\frac{2n\overline{\Delta}}{\varepsilon}, \frac{8C\overline{\Delta}\overline{R}^2}{\varepsilon^2}\}$ constraints, with the solution $\varepsilon$-close to optimality. The proof can be found in [Tsochantaridis *et al.*, 2005].

**Algorithm 1** Cutting plane algorithm to solve SVM$_{OS}$

```
1:  Input: $(x_1, y_1), \ldots, (x_n, y_n), C, \epsilon$
2:  $P_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
3:  Initialize w
4:  repeat
5:    for i = 1,. . .,n do
6:      $H(y) \equiv \Delta(y, y_i) + \mathcal{T}(x, y; w) - \mathcal{T}(x, y_i; w)$
7:      compute $\widehat{y} = \arg\max_{y \in \mathcal{S}(x_i)} H(y)$
8:      compute $\xi_i = \max\{0, \max_{y \in P_i} H(y)\}$
9:      if $H(\widehat{y}) > \xi_i + \varepsilon$ then
10:       $P_i \leftarrow P_i \cup \{\widehat{y}\}$
11:       $w \leftarrow$ optimize the dual of SVM$_{OS}$ over
          $P = \cup_i P_i$
12:     end if
13:   end for
14: until no $P_i$ has changed in the current iteration
```

Consequently, the polynomial runtime of Algorithm 1 is ensured if the "most violated" constraint, among the exponential number of constraints, i.e.,

$$\arg\max_{\widehat{y} \in \mathcal{S}(x)} \mathcal{T}(x, \widehat{y}; w) + \Delta(\widehat{y}, y) \tag{6}$$

can be computed in polynomial time.

In our problem setting, each item $I$ is represented as a real valued vector $v(I) \in \mathbb{R}^k$. We define the feature map to be:

$$\phi(x, y) = \sum_{I \in y} v(I) - \sum_{I \in x-y} v(I) \tag{7}$$

If item $I$ is selected in subset $y$, it contributes $w^\top v(I)$ to $\mathcal{T}(x, y; w)$, otherwise its contribution is $-w^\top v(I)$. For a given training set ($x$,$y$) and arbitrary $\widehat{y} \in \mathcal{S}(\mathcal{X})$, we define $a = |y \cap \widehat{y}|$, the size of overlap between subset $y$ and $\widehat{y}$; and $b = |\overline{y} \cap \overline{\widehat{y}}|$, the size of overlap of complement of $y$ and $\widehat{y}$ in $x$; and $n = |x|$, the size of the ground set. We can show:

$$\Delta(\widehat{y}, y) = 1 - \frac{a}{max(|y|, n - |y| + a - b)} \tag{8}$$

and solve (6) optimally in polynomial time O($|x|^2$) using Algorithm 2 in [Joachims, 2005] as follows: since both $a$ and $b$ can only take values in {0,. . .,n}, $\Delta(\widehat{y}, y)$ can have at most $O(n^2)$ different values. In addition, if $a$ and $b$ are fixed, we can construct $\widehat{y}$ to maximize $\mathcal{T}(x, \widehat{y}; w)$ in (6) easily: we rank all items in $x$ according to the score $w^\top v(I)$. Select $a$ highest scored items in $y$ to include in $\widehat{y}$, and do not select any of the $b$ lowest scored items in $x - y$ in $\widehat{y}$. For the rest $n - a - b$ items, choose to include the item in $\widehat{y}$ if and only if it is not in $y$. By iterating through legitimate values of $a$ and $b$, we can find the most violated constraint in $O(n^2)$ time with efficient implementation. During the testing phase, the optimal subset that maximizes equation (5) is $\{I \in x\colon w^\top v(I) > 0\}$.

## 3 Preference Formalism

So far we have concentrated on the structural learning algorithm, while being vague about the definition of preferences. Naturally, a preference $\mathfrak{p} : \mathcal{X} \to \mathcal{S}(\mathcal{X})$ is a function that maps any ground set in $\mathcal{X}$ to its optimal subset in $\mathcal{S}(\mathcal{X})$. Without loss of generality, we also assume that given a subset $y \in \mathcal{S}(x)$, it is easy to test if $y = \mathfrak{p}(x)$. In fact, we will show that the decision version of computing the optimal subset given the preference $\mathfrak{p}$ and a ground set $x$ is **NP-complete** in the following.

**Definition.** SUBSET COMPUTATION PROBLEM (SCP)
*Input: ground set x, preference* $\mathfrak{p}$*, an item I*$\in$ *x*
*Decision Question: Is I*$\in$ $\mathfrak{p}$*(x)?*

**Lemma.** *SCP is* **NP***-complete*.

*Proof.* This can be shown by reducing the binary knapsack problem to SCP. The details are omitted due to space constraints. □

Therefore, for the experiment dataset construction, we further restrict that given preference $\mathfrak{p}$ and ground set $x$, the optimal subset can be efficiently computed, for example in low order polynomial time. This restriction on the preference is for computational purposes of the experiment dataset, but it is not a limitation of our learning algorithm. If an oracle exists to produce the optimal subset given ground set $x$ and unrestricted preference $\mathfrak{p}$, SVM$_{OS}$ is able to learn to predict the optimal subset.

## 4 Related Work

[Domshlak and Joachims, 2005] learns a linear discriminant ordinal utility function from a set of qualitative preference statements by a non-parametric model similar to hard margin SVM ordinal regression. The obtained utility function essentially specifies an item-wise preference partial ordering. In addition to the fact that our model is based on structural learning, their focus is not on learning an optimal subset.

[Brafman *et al.*, 2006a] proposed a general way to specify preferences over sets of items. It first defines item properties from the attribute values of an item. The set properties are defined based on item properties and the set preferences are specified over set property values using TCP-nets [Brafman *et al.*, 2006b] that yield a partial ordering. While the emphasis of [Brafman *et al.*, 2006a] is the representation of set preferences, in our study we emphasize not the exact representation of set preferences, but the discriminative model in order to learn the most preferred set of items to maximize a set similarity measure, as we assume preferences are implicit and can take various forms.

[Binshtok *et al.*, 2007] extends the work of [Brafman *et al.*, 2006a]. They concentrate on the computational aspect of the NP-hard optimal subset selection after the set preferences are given in the formalism of [Brafman *et al.*, 2006a]. They showed that they can improve the heuristic search over the property value method first pursued in [Brafman *et al.*, 2006a], by carefully designed pruning mechanisms that efficiently rid many sub-optimal property combinations. They also proposed a new search over sets heuristic that deals with more general preference models. Their work clearly differs from ours, as no learning is involved and the preference model is pre-defined.

[Yue and Joachims, 2008] learns to predict a subset of documents that is as diversified as possible using structured SVM. The diversity measure is defined as a linear function using a structured feature space that encodes the benefit of covering individual words. In our paper we deal with a more general problem by optimizing the set accuracy metric, as diversity can be defined as a particular preference.

[desJardins and Wagstaff, 2005] represents the preference over sets of items by "depth" preferences and "diversity" preferences. The objective value of a given preference and a subset of items is a linear combination of its "depth" value and "diversity" value. While "depth" can be decomposed into a weighted sum of functions on item features, "diversity" is measured with a combination of item features and subset features. They also proposed several greedy methods to find optimal subsets with known preferences, and compared the objective value of subsets found by greedy and exhaustive search methods which showed that the greedy methods can find near optimal subsets much more efficiently than the exhaustive search. The preference representation is considered more specific than [Brafman *et al.*, 2006a] by [Binshtok *et al.*, 2007].

[desJardins *et al.*, 2006] extends [desJardins and Wagstaff, 2005], in which a preference learning mechanism (denoted DDpref) is introduced to learn the preference parameters for the model proposed in [desJardins and Wagstaff, 2005]. DDpref learns depth preferences through kernel density estimation (KDE) [Parzen, 1962] and diversity preferences by maximum *a posteriori* estimation. The experiments showed how DDpref improves the objective value of learned preferences as training set size increases. To our knowledge DDpref is the only existing method that learns user preferences from sets of objects and can predict optimal subsets for new ground sets of items. Therefore, we will compare our experimental results with DDpref's.

Table 1: Dataset Size of Each Preference Learning Task

| Experiment | n | m | card. | k | fea. |
|---|---|---|---|---|---|
| Face Images | 200 | 100 | 100 | variable | 960 |
| Toy Blocks | 100 | 100 | 100 | 20 | 4 |

## 5 Experiment

### 5.1 Dataset Description

The main experiments are based on the CMU face image dataset we obtained through [Asuncion and Newman, 2007]. This data set contains face images of 20 different people with various poses (straight, left, right or up); expressions (neutral, happy, sad or angry), and eyes (open or with sunglasses). Each person has 28 to 32 quality face images. Each greyscale image contains 32*30 pixels and a pixel is represented using its integer intensity in the range [0,255]. To construct the experiment dataset for optimal subset learning, we separate the 20 people into two disjoint groups of size 12 and 8 to construct the training and testing sets. In order to generate one training example, we randomly sample 100 images of the 12 people, and compute the optimal subset using the preferences we will describe soon. The test set is generated in the same manner on the other 8 peoples' images. 200 training sets and 100 test sets, each with one ground set and an optimal subset, are generated for every preference task 1-5. A total of $2 * 10^4$ images (with repetitions) are used to construct the training set for each task.

The synthetic dataset was first used in [desJardins *et al.*, 2006]. This dataset concerns learning optimal subsets of toy blocks. Each toy block is described using 4 integer values to represent its size, number of sides, color and bins. For task 6-8, we obtained 100 training sets and 100 testing sets using the block generator from [desJardins *et al.*, 2006]. Each training and test set contains 100 blocks as the ground set, and the size of the optimal subsets are fixed at 20 due to the requirement of DDpref. The scale of the blocks dataset exceeds that used in [desJardins *et al.*, 2006].

Table 1 gives the summary of the experiment dataset size of a single preference learning task in terms of the number of training sets ($n$), the number of test sets ($m$), the number of items in each train/test set ($card.$), the size of optimal subsets ($k$) and the number of features to describe an item in the set ($fea.$).

The different underlying preferences to generate the dataset following our preference formalism in section 3 are listed below, with the first 5 preferences on the face images dataset and the last 3 on the toy blocks dataset:

1. (Independent of ground set) Choose all images that are open, straight, and happy or neutral.
2. (Dependent) If the ground set has more images with people wearing sunglasses than open face, choose all left, otherwise choose all right.
3. (Dependent) If the ground set has fewer happy faces than sad ones, chose happy images; otherwise choose sad faces.

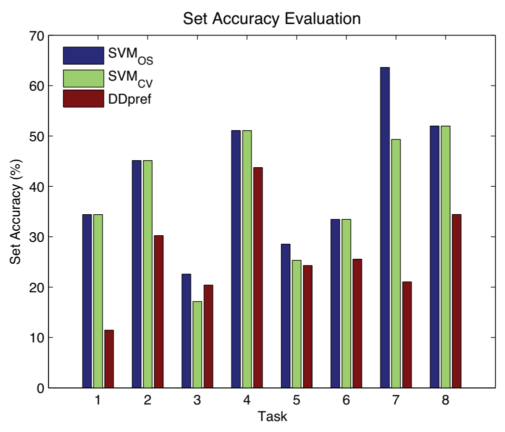

Figure 3: Set Accuracy Comparison

4. (Dependent, reverse of task 2) If the ground set has more left than right images, choose all open, otherwise choose all sunglasses
5. (Complex Dependent) If open and happy images consist of more than 30% of ground set, and sad and sunglass images are fewer than open and happy, choose all sad, otherwise all angry images.
6. (Blocks *Child*) Want to choose a variety of multi-colored, medium-sized blocks with few sides for a child to grasp.
7. (Blocks *Mosaic*) Want to create a mosaic with blocks of various shapes, with emphasis on simple, small blocks.
8. (Blocks *Tower*) Want to build a uniform tower with large, similar-sided blocks of the same color.

## 5.2 Set Accuracy Result

The results are summarized in Figure 3. All performance measures are averaged over the 100 test sets for each task. For the first five tasks on face images, SVM$_{OS}$ reports the best set accuracy results by varying the $C$ parameter from $10^{-6}$ to $10^{4}$. We also included the results where the $C$ parameter is chosen from 5-fold cross validation on the training set (denoted by SVM$_{CV}$). For DDpref, we report the best results obtained from extensively tuning the $\alpha$ and $K$ parameters, where $\alpha$ is the tradeoff constant between diversity and depth preferences and $K$ is the required size of predicted subsets. The difference in performance between SVM$_{OS}$ and DDpref for all tasks except task 3 is significant at the 0.01 level with a signed rank test. Difference in task 3 is significant at the 0.1 level. SVM$_{OS}$ outperforms DDpref in all five tasks, while SVM$_{CV}$ outperforms DDpref other than task 3, in which the cross validation happens to find a substantially suboptimal $C$ setting. For other tasks, the cross validation on the training set often finds the best parameter.

For the three toy block tasks, we supplied DDpref with the true optimal subset size $K$=20 as it is a constant for the generation of the dataset. Thus, DDpref will always predict a subset with the same cardinality as the true optimal subset. All performance differences are significant at the 0.01 level.

Table 3: More Evaluation Metrics on Blocks Dataset

| Task | SVM$_{OS}$ | | | | DDpref | |
|---|---|---|---|---|---|---|
| | Prec. | Rec. | F1 | Jacc. | Prec/Rec/F1 | Jacc. |
| 6 | 33.44 | 49.75 | 39.80 | 26.02 | 25.55 | 16.05 |
| 7 | 64.63 | 74.05 | 68.59 | 52.95 | 21.04 | 12.44 |
| 8 | 52.03 | 73.85 | 60.73 | 44.48 | 34.42 | 23.63 |

Again, both SVM$_{OS}$ and SVM$_{CV}$ predicted better subsets than DDpref in terms of the set accuracy measure.

## 5.3 More Evaluation Metrics

**Results on Precision, Recall, F1 Score and Jaccard Index**

Other than the proposed set accuracy measure, we also evaluated the experiment performance of SVM$_{OS}$ and DDpref using the other metrics discussed in section 2.2 to provide a more complete view, namely the precision and recall accuracy, the F1-score and Jacaard Index. All performance measures are calculated using the optimal subset and the predicted one from each test set, and are averaged over 100 test sets. Tables 2 and 3 present the results when the best set accuracy is achieved for both SVM$_{OS}$ and DDpref.

We can conclude from Table 2 that SVM$_{OS}$ outperforms DDpref in all four metrics. The only exception is task 3, where DDpref achieves better recall and F1-score.

In Table 3 the precision, recall and F1 scores for DDpref are exactly the same in each task, as DDpref always predicts the subset with the same cardinality as the optimal subset for a fixed $K$=20. SVM$_{OS}$ achieves higher scores for all metrics in the three tasks.

We note that both DDpref and SVM$_{OS}$ also learn a preference ranking order over items. In this paper, we focus on the performance of predicting the optimal subset and therefore do not compare the ranking performance, which can be defined separately.

**Scalability Comparison**

We study the scalability behavior by comparing the training time for both SVM$_{OS}$ and DDpref. Apart from the fact that SVM$_{OS}$ is implemented in C++ and DDpref in Java, we observed that SVM$_{OS}$ is orders of magnitude faster than DDpref during training. Although the training time depends on the particular parameter settings, in general SVM$_{OS}$ usually requires less than 10 minutes to train while the training time required by DDpref can be more than 10 hours on a machine with a 3.8GHz Intel CPU and 8GB memory. Figure 4 shows the detailed training time against different training set sizes for task 5, where the parameters are set to provide the best set accuracy. Both methods appeared to scale linearly in our experiments. During the testing phase, SVM$_{OS}$ has a runtime linear in the number of test cases, and DDpref uses a greedy method to select the optimal subset from the learned preferences. Both SVM$_{OS}$ and DDpref are efficient in the testing phase in terms of runtime.

## 6 Conclusion

In this paper we propose a new approach for the optimal subset selection problem. We model the problem as a structural

Table 2: Precision, Recall, F1-Score and Jaccard Index Performance on Face Dataset in Percentage

| Task | $\text{SVM}_{OS}$ | | | | DDpref | | | |
|---|---|---|---|---|---|---|---|---|
| | Prec. | Rec. | F1 | Jacc. | Prec. | Rec. | F1 | Jacc. |
| 1 | 35.98 | 51.39 | 40.59 | 27.69 | 11.50 | 19.77 | 14.54 | 8.45 |
| 2 | 45.91 | 70.88 | 54.12 | 38.95 | 30.40 | 48.99 | 37.52 | 23.26 |
| 3 | 23.75 | 35.17 | 27.07 | 16.67 | 20.40 | *42.47* | *27.56* | 16.18 |
| 4 | 69.95 | 51.19 | 58.43 | 42.12 | 49.00 | 48.49 | 48.74 | 31.79 |
| 5 | 28.54 | 60.92 | 37.79 | 24.23 | 24.33 | 52.24 | 33.20 | 20.38 |

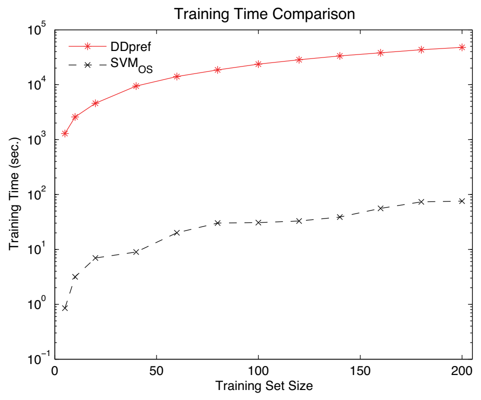

Figure 4: Scalability of $\text{SVM}_{OS}$ and DDpref

learning problem that maximizes set similarity between the predicted subsets and the optimal ones. While previous work requires an explicit preference representation, our discriminative learning model $\text{SVM}_{OS}$ only assumes the existence of implicit user preferences. Our model therefore does not need the intermediate step of learning the parameters of a pre-defined preference function. Experiment results on a variety of the subset selection tasks have shown that $\text{SVM}_{OS}$ provides high quality testing results compared with a recent learning model.

## Acknowledgments

We thank Thorsten Joachims for the discussions on structured learning methods. We also thank the reviewers for their comments and suggestions. This research was supported by AFOSR grant FA9950-08-1-0196, NSF grant 0713499, and NSF grant 0832782.

## References

[Asuncion and Newman, 2007] A. Asuncion and D.J. Newman. UCI machine learning repository, www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[Binshtok *et al.*, 2007] Maxim Binshtok, Ronen I. Brafman, Solomon E. Shimony, Ajay Martin, and Craig Boutilier. Computing optimal subsets. In *Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2007.

[Brafman *et al.*, 2006a] R. I. Brafman, C. Domshlak, S. E. Shimony, and Y. Silver. Preferences over sets. In *Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2006.

[Brafman *et al.*, 2006b] Ronen I. Brafman, Carmel Domshlak, and Solomon E. Shimony. On graphical modeling of preference and importance. *Journal of AI Research*, 25, 2006.

[desJardins and Wagstaff, 2005] Marie desJardins and Kiri L. Wagstaff. Dd-pref: A language for expressing preferences over sets. In *Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2005.

[desJardins *et al.*, 2006] Marie desJardins, Eric Eaton, and Kiri L. Wagstaff. Learning user preferences for sets of objects. In *Proceedings of the International Conference on Machine Learning*, pages 273–280, 2006.

[Domshlak and Joachims, 2005] C. Domshlak and T. Joachims. Unstructuring user preferences: Efficient non-parametric utility revelation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 169–177, 2005.

[Freund *et al.*, 2003] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[Joachims, 2005] T. Joachims. A support vector method for multivariate performance measures. In *International Conference on Machine Learning (ICML)*, pages 377–384, 2005.

[Parzen, 1962] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[Price and Messinger, 2005] Bob Price and P. R. Messinger. Optimal recommendation sets: Covering uncertainty over user preferences. In *In Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 541–548, 2005.

[Tsochantaridis *et al.*, 2005] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.

[Yue and Joachims, 2008] Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural svms. In *International Conference on Machine Learning (ICML)*, 2008.