Capacity Control for Partially Ordered Feature Sets

Ulrich Rückert

International Computer Science Institute, 1947 Center Street. Suite 600, Berkeley, CA 94704, rueckert@icsi.berkeley.edu

Abstract. Partially ordered feature sets appear naturally in many classification settings with structured input instances, for example, when the data instances are graphs and a feature tests whether a specific substructure occurs in the instance. Since such features are partially ordered according to an "is substructure of" relation, the information in those datasets is stored in an intrinsically redundant form. We investigate how this redundancy affects the capacity control behavior of linear classification methods. From a theoretical perspective, it can be shown that the capacity of this hypothesis class does not decrease for worst case distributions. However, if the data generating distribution assigns lower probabilities to instances in the lower levels of the hierarchy induced by the partial order, the capacity of the hypothesis class can be bounded by a smaller term. For itemset, subsequence and subtree features in particular, the capacity is finite even when an infinite number of features is present. We validate these results empirically on three graph datasets and show that the limited capacity of linear classifiers on such data makes underfitting rather than overfitting the more prominent capacity control problem. To avoid underfitting, we propose using more general substructure classes with "elastic edges" and we demonstrate how such broad feature classes can be used with large datasets.

Key words: capacity control, partially ordered features, graph mining, QSAR

1 Introduction

In this paper we investigate classification with feature sets that are partially ordered. Such feature sets often appear naturally in learning settings with structured input objects. For instance, consider the task of learning whether or not a particular chemical compound inhibits tumor growth [3, 8, 2, 10]. A popular approach to this setting is to represent the compounds by molecular graphs and to generate Boolean features that test for the occurrence of certain substructures (e.g. subgraphs) in the molecular graph. In this setting each compound can be represented by a bit vector, where each bit indicates whether or not the corresponding subgraph appears in the compound's molecular graph. Such a representation can then be used with traditional linear classifiers such as the

ones output by support vector machines. The difference to most other classification settings, of course, is that the space of subgraphs is *partially ordered* via the "is subgraph of" relation. Consider, for instance, a data instance where the feature representing an aromatic ring is set to *true*. This means that all features that represent linear sequences of up to six carbon atoms connected with aromatic bonds must also be set to *true*. Consequently, the information provided by partially ordered feature sets is redundant by design.

The main question dealt with in this paper is how this redundancy affects the empirical risk minimization and capacity control behavior of a learning algorithm. In the first part of the paper we address these questions from a theoretical point of view. We show that the capacity of the class of linear classifiers (as measured by the VC-dimension) does not change, even when the features in the training data are totally ordered. However, if the underlying data distribution puts lower probabilities on data instances that are ordered in the later levels of the hierarchy induced by the partial order, one can find smaller upper bounds for the capacity of the class of linear classifiers. We show that distributions where the probability of observing an instance declines exponentially with the instance's level in the hierarchy can lead to settings, where the class of linear classifiers has finite capacity even in the presence of an infinite amount of features.

On the practical side we validate the theoretical results empirically on three datasets from quantitative structure-activity relationships. We show that adding more features does indeed not lead to overfitting for subsequence, subtree and subgraph features. Instead, we show that extending the feature set with more expressive substructures can improve predictive accuracy. This indicates that underfitting rather than overfitting is the prominent problem on datasets with partially ordered features. Finally, we investigate how an expressive and therefore large substructure feature class can be efficiently applied on large datasets.

2 Background

Before we can delve into the details, we need to introduce the used concepts and definitions. We assume an instance space $\mathcal{X}$ of possible objects and a binary output space $\mathcal{Y} := \{-1, 1\}$. We are given a set $\mathcal{F} = \{f_0, \ldots, f_m\}$ of $m+1$ substructure features, which are ordered by a (possibly partial) "is substructure of" relation $R \subset \mathcal{F} \times \mathcal{F}$ so that $(f_i, f_j) \in R$ whenever f_i is a substructure of f_j. We write $f_i(x) = 1$ or $f_i(x) = 0$ to express that the substructure for feature f_i is contained ($f_i(x) = 1$) or not contained ($f_i(x) = 0$) in object x. We also assume that the first feature f_0 represents the empty substructure, so that $f_0(x) = 1$ for all $x \in \mathcal{X}$. With this, each object $x \in \mathcal{X}$ can be represented by a $m+1$-dimensional binary vector $\boldsymbol{x} \in \{0, 1\}^{m+1}$. In the following, we will not distinguish between x as an object and $\boldsymbol{x}$ as a bit vector if the meaning is clear from the context. We write $f_i \preceq f_j$ to denote that $(f_i, f_j) \in R$. Naturally, the "is substructure of" relation R is transitive so that $f_i \preceq f_k$ whenever there is a f_j with $f_i \preceq f_j$ and $f_j \preceq f_k$. Also, whenever a substructure feature f_i is more

general than a feature f_j (i.e. $f_i \preceq f_j$), all objects $x \in \mathcal{X}$ with $f_j(x) = 1$ also have $f_i(x) = 1$. The relation R limits the number of bit vectors that can be used to represent the examples in $\mathcal{X}$. We denote the space of possible bit vectors by $\mathcal{X}_R := \{x \in \{0,1\}^{m+1} | \forall (f, f') \in R : f'(x) = 1 \rightarrow f(x) = 1\}$.

We follow the usual learning setting, where a data set $(X, Y) \in (\mathcal{X}_R \times \mathcal{Y})^n$ of n instances $(x_1, y_1), \ldots, (x_n, y_n)$ is drawn i.i.d. from a fixed but unknown distribution P. The task of the learning system is to find a linear classifier $w \in \mathbb{R}^{m+1}$, which minimizes the true error $\varepsilon_w := \mathbf{E}_{(x,y)\sim P}\, l(w^T x, y)$, where the loss $l(y, y') \rightarrow \mathbb{R}$ assigns some loss to each misclassification. Since P is unknown, a classifier's true error is unknown and practical learning algorithms deal with the empirical error $\hat{\varepsilon}_w = \frac{1}{n} \sum_{i=1}^{n} l(w^T x_i, y_i)$ as a computable substitute. Since the first feature f_0 represents the empty substructure and is always set to 1, the first component w_0 of the linear classifier is essentially a bias term, which controls the distance of the hyperplane induced by w to the origin.

3 Ordered Feature Sets

We are now in the position to formulate the main theoretical results. We first show that the capacity of the class of linear classifiers is the same as the capacity in the unrestricted case without a partial order. Thus, the introduction of a partial order on the features does not increase or decrease the capacity of linear classifiers for worst case distributions. In the second part we give upper bounds of the capacity for distributions where the probability of observing an instance declines with its level in the hierarchy induced by the partial order. We also show that an exponential decay in this probability can lead to linear classifiers having finite capacity, even though the number of features is infinite. This is the case for instance for subsequence and subtree features, but not for subgraph features.

3.1 Distribution-Independent Capacity

One of the main contributions of computational learning theory deals with estimates for the capacity of hypothesis classes. In the general case (i.e. without partially ordered features), it is well known that the hypothesis space of linear classifiers of size m has VC dimension m (see e.g. corollary 13.1 in [4]), giving rise to bounds of the form

$$\mathbf{Pr}\left[\varepsilon_w \leq \hat{\varepsilon}_w + O\left(\sqrt{\frac{m + \ln \frac{1}{\delta}}{n}}\right)\right] \geq 1 - \delta.$$

Thus, the "overfitting penalty" introduced by a hypothesis space of linear classifiers scales as $O(\sqrt{m/n})$ in the number of features. These bounds are tight up to the order of magnitude, that is, there are also lower bounds that scale with $O(\sqrt{m/n})$. Let us now consider the hypothesis space of linear classifiers on the *restricted* instance space $\mathcal{X}_R$, whose instances meet the constraints induced by the partial order R. Observe that the VC-dimension of the class of linear

classifiers decreases, if the data instances are chosen only from a d-dimensional subspace of $\mathcal{X}$ with $d < m$. Since $\mathcal{X}_R \subset \mathcal{X}$, the class of linear classifiers on $\mathcal{X}_R$ is smaller in the sense that its hypotheses needs to distinguish between a smaller number of instances. One could thus hope that the VC-dimension of linear classifiers on $\mathcal{X}_R$ decreases in a similar way, if the constraints imposed by R are strict enough. Unfortunately, this turns out to be not the case. The following theorem states this more formally:

Theorem 1. *Let R be an arbitrary partial order on the features $\{f_0, \ldots, f_m\}$, let $\mathcal{X}_R \subset \{0,1\}^{m+1}$ be the space of instances which are consistent with the order R and let $\mathcal{H}_R$ denote the hypothesis space of linear classifiers over $\mathcal{X}_R$. Then, $\mathcal{H}_\mathcal{R}$ has VC-dimension $m+1$.*

Proof. It is sufficient to show that there is no dataset of size $m+2$, which can be shattered and that there is a dataset of size $n \leq m+1$, which can be shattered by a linear classifier. The first statement follows directly from the fact that the VC-dimension of linear classifiers in $\mathbb{R}^m$ is also $m+1$. For the second statement, assume without loss of generality that the features $f_0, f_1, \ldots, f_m$ are ordered according to R. Then, select the set of examples $\{x_0, x_1, \ldots, x_m\}$, where $f_j(x_i) = 1$, if $f_j \preceq f_i$ and $f_j(x_i) = 0$ otherwise. The $(m+1) \times (m+1)$ training matrix X for this data set has the lower triangle set to zero and the diagonal set to one, that is, it is in upper diagonal form. A straightforward application of Gaussian elimination shows that the matrix has full rank. This means that there is a linear classifier for all 2^{m+1} possible target value assignments.

This means the VC bounds for the hypothesis space of linear classifiers with partially ordered features are essentially the same as the ones for unordered features. The lower bounds in chapter 14 of [4] ensure that there is no way to get significantly better guarantees than $O(\sqrt{m/n})$. This is quite remarkable, because we did not impose any restrictions on R. In particular, if R is a total order so that $f_i \preceq f_{i+1}$ for all $0 \leq i < m$, the instance space $\mathcal{X}_R$ contains only $m+1$ instances. Thus, the VC dimension $\mathcal{H}_R$ remains constant for any order R, regardless of whether R is a total order (and $|\mathcal{X}_R| = m+1$) or the empty order (and $|\mathcal{X}_R| = 2^{m+1}$). Even though the capacity of the hypothesis class remains constant, the best obtainable empirical risk does depend heavily on R. In particular, if R is a total order and the Bayes error is zero, it is easy to see that empirical risk minimization will find a w with $\hat{\varepsilon}_w = 0$. This is not true for non-total orders.

3.2 Distribution-Dependent Capacity

The results in the preceding section indicate that there are worst-case distributions where the partial ordered feature sets do not decrease the capacity of the class of linear classifiers. However, there may very well be distributions that lead to smaller capacity estimates. In the following we show that this is indeed the case. More specifically, we introduce a distribution-based quantity that can be

used to upper-bound the capacity of the class of linear classifiers with partially ordered feature sets.

We begin with a few definitions. Since the features are partially ordered, they can be categorized by level. More formally, for a given feature f_i let the *level* $\lambda(f_i)$ denote the largest $k \in \mathbb{N}$ so that there is a sequence $i_1, i_2, \ldots, i_k$ of size k with $f_{i_1} \prec \ldots \prec f_{i_k}$. Similarly, the level $\lambda(x) := \max_{f \in \mathcal{F}}\{\lambda(f) | f(x) = 1\}$ of an instance x is the largest level of the features that are set to one by x. The probability $\mathbf{Pr}[f(x) = 1] = \mathbf{E}[f(x)]$ that a feature f is set to one decreases with its level. In fact, if $f_i \preceq f_j$ and $\mathbf{E}[f_i(x)] = \mathbf{E}[f_j(x)]$, then we know that the two features f_i and f_j are equivalent, because $f_i(x) = f_j(x)$ for all instances $x \in \mathcal{X}$. This means that we can remove a feature f_i from the training data whenever it is more general than another feature f_j and $\mathbf{E}[f_i(x)] = \mathbf{E}[f_j(x)]$. Removing the feature does not affect the capacity of the hypothesis class of linear classifiers, because for every classifier w that assigns a non-zero weight to f_i there is an equivalent classifier w', which simply transfers the weight from f_i to the equivalent feature f_j. Thus, we can assume without loss of generality that $\mathbf{E}[f_i(x)] > \mathbf{E}[f_j(x)]$ whenever $\lambda(f_i) < \lambda(f_j)$. With this, let $\lambda_i := \max_{x \in \mathcal{X}_R}\{\mathbf{Pr}[X = x] | \lambda(x) \geq i\}$ denote the maximum probability of obtaining an example of level at least i and let $d_i := |\{f \in \mathcal{F} | \lambda(f) = i\}|$ denote the number of features of level i. We can now state the following two results. The first one gives an upper bound of the capacity of the class of linear classifiers for the zero-one loss, whereas the second one deals with loss functions that are Lipschitz with Lipschitz constant L.

Theorem 2. *Let R be a partial order on a feature space $\mathcal{F}$, let $\mathcal{X}_R := \{x \in \{0,1\}^m | \forall (f, f') \in R : f'(x) = 1 \rightarrow f(x) = 1\}$ be an partially ordered instance space, let P be a probability distribution on $\mathcal{X} \times \mathcal{Y}$ and let ε_w and $\hat{\varepsilon}_w$ be based on the zero-one loss. Define*

$$D_R := \sum_{i=1}^{n} \log \left[\sum_{j=1}^{k} \lambda_j \exp\left(\frac{k}{i} d_j\right) \right]$$

Then it holds for all linear classifiers $w \in \mathbb{R}^m$ that

$$\Pr\left[\varepsilon_w \leq \hat{\varepsilon}_w + \sqrt{\frac{2D_R + \log\frac{1}{\delta}}{n}} + \sqrt{\frac{\log\frac{1}{\delta}}{2n}} \right] \geq 1 - \delta$$

A more straightforward bound can be achieved, if the loss function is continuous and Lipschitz. Let $\mathcal{B}_2^m := \{w \in \mathbb{R}^m | \|w\|_2 \leq 1\}$ denote the m-dimensional unit ball for the 2-norm.

Theorem 3. *Let R, $\mathcal{X}_R$, P be as above, but let ε_w and $\hat{\varepsilon}_w$ be based on a Lipschitz loss $l_L : \mathbb{R} \rightarrow [0,1]$ with Lipschitz constant L. Then it holds for all linear classifiers $w \in \mathcal{B}_2^m$ that*

$$\Pr\left[\varepsilon_w \leq \hat{\varepsilon}_w + 2L\sqrt{\frac{\sum_{i=1}^{k} d_i \lambda_i}{n}} + \sqrt{\frac{8\log\frac{2}{\delta}}{n}} \right] \geq 1 - \delta$$

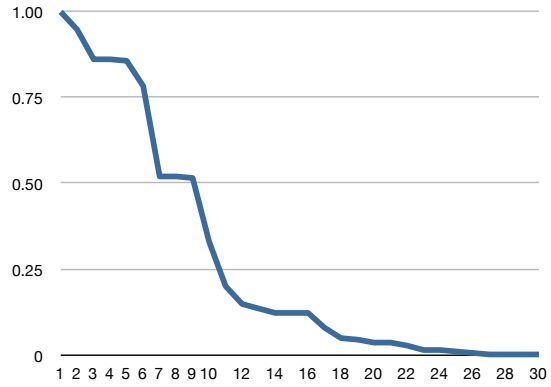

Fig. 1. The maximum probability of observing an instance for each level on the NCTRER dataset.

The proofs are in the Appendix. The results essentially state that the capacity of the class of linear classifiers can be upper-bounded depending on how λ_i and d_i scale for increasing levels. The bound is small, either if there are only a limited number of features of higher levels (i.e. d_i is small for large i), or if the probability of encountering instances of higher levels is small (i.e. λ_i is small for higher levels). As explained above, the sequence $\lambda_1, \lambda_2, \ldots$ is strictly decreasing, but the extent of the decay depends on the distribution. In practice, applications with structured examples and substructure features often lead to distributions where the level probabilities features exponential decay. For instance, we plot the level sequence for the NCTRER dataset in figure 1. The figure shows that the probability decreases approximately exponentially.

The number d_i of features per level, though, grows usually exponentially. If a learning system makes use of all possible substructure features for each level, an exponential decay in the level probabilities does therefore not automatically guarantee a small or even finite learning capacity. In the following we give capacity estimates for the case where the decay in the level probabilities is exponential. In particular we assume that the level probabilities can be upper-bounded by a decay that is exponential in a constant α:

$$\lambda_i \leq \alpha^i$$

3.3 Capacity Estimates for Various Partial Orders

Given the upper bound in Theorem 3 we can consider $C := \sum_{i=1}^{k} d_i \lambda_i$ as a capacity measure for the class of linear classifiers on partially ordered feature sets. This enables us to investigate the capacity estimates for the partial orders induced by some popular substructure classes. Let us begin with the total order

$R_t := \{(f_i, f_j)|i \leq j\}$. While this order will be rarely encountered in practical applications, it is interesting from a theoretical perspective, because it is the order that puts the strongest constraints on the features. Obviously, the level of feature f_i is simply i so that $C_{R_t} = \sum_{i=1}^{k} \alpha^i$. This is a geometric series, and a basic analysis confirms that

$$C_{R_t} = \frac{1-\alpha^{k+1}}{1-\alpha} \leq \frac{1}{1-\alpha}$$

This means that the capacity of the hypothesis class of linear classifiers with totally ordered features is bounded by a term of order $O(1/(1-\alpha))$, which is independent of the number of features. Thus, empirical risk minimization is consistent even for instance spaces with an infinite amount of totally ordered features.

As a slightly more complicated case, we consider the setting where the examples are sets of items. Here, we have a set $O = \{o_1, \ldots, o_k\}$ of items and the instance space $\mathcal{X}$ is the power set of O, so that the instances and features are represented by subsets of O. A feature assigns the label 1 to an example, whenever the item set associated with the feature is a subset of the item set associated with the example. It is easy to see that the level of a feature is simply the number of items in its associated item set. Since there are $\binom{k}{i}$ possible item sets with i items, the decay capacity can be computed as:

$$C_{R_I} = \sum_{i=0}^{k} \binom{k}{i} \alpha^i = (1+\alpha)^k \leq e^{k\alpha}$$

Depending on the size of α, this is an exponential improvement over the VC-bound, which upper-bounds the capacity of the class of linear classifiers by $O(2^k)$, because there are 2^k possible features. This result is also applicable to SVM-classification with polynomial kernels on binary data. If one selects a polynomial kernel of degree t for SVM training on data with k binary (i.e. zero-one-valued) features, the kernel-induced feature space is equivalent to a feature space containing all itemset features of size at most t. In this case the decay capacity is $\sum_{i=0}^{t} \binom{k}{i} \alpha^i$.

In the next step, we handle the case where the examples are strings over an alphabet $\mathcal{A} = \{a_1, \ldots, a_h\}$ containing h characters. Here, it is a natural choice to use substring features, which are partially ordered by the "is substring of" order R_S. More precisely, we assume that each feature is associated with a string and the feature assigns the value 1 to an instance, if this string is a substring of the instance. Even though there is a potentially infinite number of instances and features, the analysis is particularly easy. It is clear that the level of a string is just its length. Also, there are h^l different strings of length l. That means that the decay capacity for linear classifiers with partial order R is:

$$C_{R_S} = \sum_{i=1}^{\infty} (h\alpha)^i$$

This is again a geometric series. If $\alpha < \frac{1}{h}$, the series converges and the capacity of the class of linear classifiers can be bounded by $\frac{1}{1-h\alpha}$. Since this quantity does not depend on the number of features, one can have finite capacity even with an infinite amount of features.

As a more complicated partial order, we consider the classification setting where the features are represented by labeled rooted trees, where the labels are taken from label set $\mathcal{L}$ of size l. Here, the features are ordered by some form of subtree isomorphism. Let T_i denote the number of rooted unlabeled trees with i vertices. It is well known that the fraction $\frac{T_i}{T_{i-1}}$ converges as $i \to \infty$ and that it converges to the limit $c_T := \lim_{i\to\infty} \frac{T_i}{T_{i-1}} = 2.955765\ldots$ from below [7]. Thus, we can use c_T^i as a crude upper bound for the number of unlabeled rooted trees with i vertices. Since there are l^i ways to assign labels, we get the following upper bound for the "is subtree of" order R_T:

$$C_{R_T} \leq \sum_{i=1}^{\infty} (lc_T\alpha)^i$$

If $\alpha < \frac{1}{c_T l} \approx \frac{0.3383\ldots}{l}$, the decay complexity of labeled rooted tree classifiers with the subtree order R_T can be upper-bounded as follows:

$$C_{R_T} \leq \frac{1}{1 - lc_T\alpha}$$

It is remarkable that this upper bound differs only in a comparably modest constant from the one for strings.

Finally, let us investigate the setting, where examples and features are connected graphs and the features are partially ordered according to the "is subgraph of" order R_G. Here, a graph with i edges has level i. The number of graphs with i edges and at most l different node labels can be upper-bounded by $li!(l+1)^i$. If there are at most k levels, the capacity can thus be upper-bounded by

$$C_{R_G} \leq \sum_{i=1}^{k} i!l\left((l+1)\alpha\right)^i$$

Unlike the previous substructure classes, this bound does not converge for $k \to \infty$. In fact it is easy to see that the number of graphs with i edges grows at least with $o((i/c)!)$ for some constant c. That means that an exponential decay of the level probability is not sufficient to enforce a finite capacity bound in the limit.

4 Experiments

Theoretical results are interesting for determining worst case capacity estimates and investigating the asymptotical consistency of learning systems with partially ordered feature sets. However, in practical applications one is much more interested in finding average case capacity estimates, which can be used to avoid over-

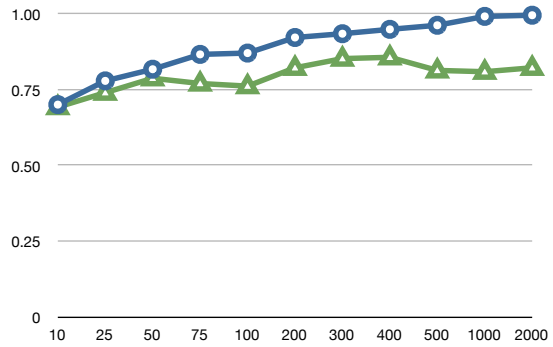

Fig. 2. Training accuracy (circles) and predictive accuracy (triangles) for the NCTRER dataset with an increasing number of features.

and underfitting. If the capacity of the learning system's hypothesis class is too large, it might overfit on the training data. In this case, the training error will be near zero, but the validation error is worse than necessary. If, on the other hand, the learner's capacity is too small, the system might induce classifiers with high training and validation error. The theoretical results in the preceding section indicate that the capacity of the space of linear classifiers can be considerably smaller than it is the case with non-ordered features. In the following we investigate the overfitting behavior in quantitative structure-activity relationships. Here, the learning system is given a training set containing the molecular structure of compounds as labeled graphs. The task is to induce a model that can predict some biological or chemical endpoint such as tumor growth inhibition or a compound's ability to pass the blood-brain barrier. We used the three datasets from [8]. The NCTRER dataset [5] deals with the prediction of binding activity of small molecules at the estrogen receptor. It contains 232 molecules. The Yoshida dataset [12] consists of 265 molecules classified according to their bioavailability. The third dataset classifies 415 molecules according to the degree to which they can cross the blood-brain barrier (BBB) [6].

4.1 Overfitting

For the first experiment we followed the standard substructure feature generation methodology (see e.g. [8, 2]) and implemented a frequent subgraph mining tool similar to gSpan [11]. The system recursively generates all subgraphs occurring in at least one graph of the training database. However, in contrast to gSpan, it discards all substructures whose instantiation vector is a duplicate of an existing subgraph, that is, a substructure which occurs in exactly the same graphs as an already generated subgraph. We used the tool to generate all possible subsequences, subtrees, and subgraphs for the three datasets. The NCTRER

Table 1. Predictive accuracy of a SVM on an increasing number of subsequence, subtree and subgraph features.

Dataset	Number of Features	Seq	Trees	Graphs	Elastic Graphs
NCTRER	500	81.0	81.9	81.5	79.7
	1000		82.3	81.0	82.3
	2000		82.3	82.3	83.6
	3000				85.8
	4000				84.9
	5000				86.2
	6000				86.2
Yoshida	500	63.8	67.5	66.0	64.9
	1000	64.5	69.8	69.1	68.7
	2000		67.5	66.8	64.2
	3000		69.1	67.9	63.4
	4000		67.9	69.1	64.5
	5000		67.5	67.9	63.4
	6000				67.9
	7000				69.1
	8000				68.3
BBB	500	76.1	74.0	77.6	75.4
	1000	75.7	74.5	76.1	78.1
	2000	76.6	74.5	73.7	79.3
	3000		74.2	74.0	81.0
	4000		75.2	74.2	81.2
	5000		76.1	75.9	81.2
	6000		75.2	75.7	81.4
	7000		73.7	74.0	81.7
	8000				81.9
	9000				83.4
	10000				81.9
	11000				81.7

dataset contains 463 subsequences, 1822 subtrees and 1897 subgraphs. We sort the features by level and plot training accuracy and predictive accuracy of a support vector machine (with $C = 1$) induced on an increasing subset of the features in figure 2. The plot shows no significant overfitting; even though the training accuracy reaches 100%, the predictive accuracy as measured by tenfold cross-validation does not decrease very much. Similar plots can be generated for the yoshida and BBB datasets. Obviously, overfitting is less of an issue as compared to many other datasets. This is remarkable when one considers that the substructure features lead to training data that has many more features than examples.

4.2 Elastic Subgraph Feature Generation

As overfitting is apparently not a big problem, one might suspect that the SVM is actually underfitting. To investigate this question we need a new feature generation mechanism that leads to feature sets with higher capacity than the existing ones. The theorems in section 3.2 indicate that one should look for feature sets whose λ_is do not decrease too fast. This means we would like to use substructure occurrence tests, where even large substructures are still likely to appear in many instances. In order to do so, we extend the class of subgraph features by allowing for *elastic edges*. More precisely, we introduce a new "elastic" edge label in each subgraph's edge label set. Whenever a subgraph pattern with such an elastic edge is tested for occurrence in a graph in the database, the elastic edge matches with any path containing only edges, which are not already matched with another edge in the subgraph pattern. Figure 3 illustrates this concept. Here, the substructure pattern on the right contains an elastic edge between the vertices e and f. This pattern matches with the instance graph on the left, because the elastic edge can be matched to the path from vertex a over b and c to vertex d. On the other hand, consider the case where one extends the pattern with another vertex, which is connected to the vertex e and has label C. The resulting pattern does not occur anymore in the graph. This is because the new edge can only be matched to the edge between a and b, but the elastic edge is not allowed to match with the path from a to d, if one of the edges on the path is already used in a different match.

We extended the subgraph mining tool to also generate subgraph structures which contain a limited number of elastic edges. While it is feasible to compute all non-duplicate subtree or subgraph features for the three datasets, the number of non-duplicate subgraph patterns with elastic edges is way too large to generate all of them. We therefore restricted the maximum size of the elastic subgraph patterns to eight for the NCTRER dataset, and five for the BBB and yoshida datasets. Table 1 shows the predictive accuracies of a SVM (with C=1) as estimated by tenfold cross-validation for subsequence, subtree, subgraph and subgraph with one elastic edge patterns depending on the number of used features. Elastic subgraph patterns outperform the other pattern languages by approximately four percent on the NCTRER dataset and over five percent on the BBB dataset. On the yoshida dataset, trees, graphs and elastic graph feature give approximately the same predictive accuracy. All three datasets show better accuracies than the best ones reported in [8]. These results indicate that underfitting was indeed a problem on two of the three datasets.

4.3 Feature Generation for Large Datasets

For the third experiment, the goal was to investigate how learning linear classifiers with a broad class of substructure features can be made efficient on a large dataset. The main problem here is that the considerations in section 4.1 indicate that one should use broad substructure classes with many general features to avoid underfitting. Unfortunately, the number of substructures in such classes is

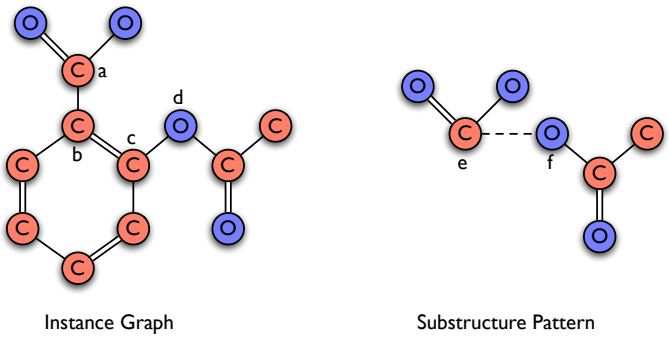

Fig. 3. The dashed edge in the substructure pattern is an elastic edge, which matches the path from vertex a over b and c to vertex d in the instance graph.

way too large to enumerate them exhaustively as it was possible in the preceding experiments. Consider the NCI DTP Human Tumor Cell Line Screen dataset [9]. The dataset contains 34748 compounds, which were tested for their ability to inhibit tumor growth. Mining for all non-duplicate subsequences of only up to four edges leads to over 10,000 features. Clearly, the database is too large to allow for exhaustive enumeration of all existing subsequence features and working with all subtree or subgraph features is clearly not feasible. To avoid the generation of all substructure features, we resort to a heuristic feature search algorithm inspired by the feature generation method presented in [8]. Instead of using a combinatorial search approach with an index structure (which would be too large for the NCI dataset), we perform a simple beam search. The algorithm starts with substructures of size one (i.e. single vertices) and iteratively extends the most promising candidate in the current beam with a new edge. The search heuristic is based on the *class-correlated dispersion score* as described in [8], but features an exponential rather than a quadratic penalty for features with high similarity to an existing feature:

$$h(s') := \sum_{i=1}^{m} \exp[\frac{c}{n} s_i^T s'] - m \exp[\frac{c}{n} t^T s']$$

Here, s' is the $-1/+1$-valued n-dimensional instantiation vector of the new feature candidate, the s_i are the instantiation vectors of the m existing features, and t is the target class vector. We generated one hundred substructure features for subsequences, subgraphs and subgraphs with one elastic edge. We then learned a linear classifier from a training set consisting of two thirds of the dataset and evaluated the classifier on the remaining third. The feature generation, learning and evaluation took 28 minutes on a 1 GHz Athlon 5200 computer for the sequences features, 68 minutes for the subgraphs feature set and 129 minutes for subgraphs with one elastic edge. The classifier achieved a predictive accuracy of 64.4% with subtree features, 64.1% with subgraph features and 63.3% with

subgraphs with one elastic edge, so underfitting seems not to be a big issue here. It is unclear whether this is a limitation of the feature generation method or a fundamental property of the data generation process.

5 Conclusion

In the preceding sections we investigated classification with linear classifiers and partially ordered feature sets. Learning with partially ordered feature sets differs from other settings in that the partial order induces redundancy in the training and test data. From a theoretical point of view, this does not necessarily affect the over- or underfitting behavior of a learning system, because the VC-dimension of the class of linear classifiers remains the same for worst-case data distributions. However, if the data distribution features a sufficiently steep decline in the probability of observing features of higher level, the capacity of the learning system can be upper-bounded by a smaller term. This means that overfitting is less of an issue for linear classifiers on those distributions. We evaluated this theoretical result on three datasets and found that subsequence, subtree and subgraph features did indeed not show typical overfitting behavior. Instead, we were able to extend the class of subgraph features towards subgraphs with elastic edges. These patterns are more likely to occur in higher levels and thus increase the capacity estimate. Practical experiments confirmed that this extended class of subgraph features avoids underfitting and increases predictive accuracy on two of the three datasets. Finally, we showed how classification with such large substructure feature classes can be implemented efficiently on large datasets.

The work raises a couple of interesting questions. On the theoretical side one could look for lower bounds that quantify to which degree the presented upper bounds are tight and investigate how the results apply, if one uses support vector machines with non-linear kernels. On the practical side, it would be interesting to obtain more insights on the actual under- or overfitting behavior of common data (for instance with regard to the study in [2]) and how the present results apply to other partially ordered feature sets, for example in natural language processing.

6 Appendix

6.1 Proof of Theorem 2

Proof. For a data sample $S = (X_1, \ldots, x_n)$ and a vector of Rademacher variables $\sigma = (\sigma_1, \ldots, \sigma_n)^T$ (where σ_i has value $+1$ or -1 with probability 0.5) define $V(S, \sigma) = \sup_w [\frac{1}{n} \sum_{i=1}^{n} \sigma_i \operatorname{sgn}(w^T x_i)]$. First of all, we consider the conditional expectation $\mathbf{E}[V(S, \sigma)|S]$. Let $v := (\operatorname{sgn}(w^T x_1), \ldots, \operatorname{sgn}(w^T x_n))^T$ denote the vector of predictions for a fixed training sample S and a fixed linear classifier w. Changing the value of a Rademacher variable σ_i changes the value of $\frac{1}{n} v^T \sigma$ by at most $\frac{2}{n}$. Thus, for a fixed data set S, one can apply McDiarmid's inequality

to bound the probability that a random Rademacher vector disagrees with v by more than fraction a fixed $r > 0$:

$$\mathbf{Pr}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i \operatorname{sgn}(w^T x_i) \geq r \middle| S\right] \leq e^{-\frac{1}{2}r^2 n} \tag{1}$$

Now, let $\lambda(x) := \max\{\lambda(f)|f(x) = 1\}$ denote the level of example x. We can assume without loss of generality that the instances in S are sorted by level. Define $S_i := \{x|\lambda(x) = i\}$ and $n_i := |S_i|$, so that $\sum_{i=1}^{k} n_i = n$. Sauer's lemma states that there are at most $\sum_{j=0}^{d_i}\binom{n_i}{j} \leq (n_i+1)^{d_i}$ ways, in which the different w can assign class labels to the instances in S_i. This means the number of possible class vectors induced by the w is at most $\prod_{i=1}^{k}(n_i+1)^{d_i}$. Taking the union bound over all possible class label assignments in (1) yields:

$$\begin{aligned}
\mathbf{Pr}[V(S,\sigma) \geq r|S] &= \mathbf{Pr}\left[\sup_w \left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i v_i\right] \geq r \middle| S\right] \\
&\leq e^{-\frac{1}{2}r^2 n}\prod_{i=1}^{k}(n_i+1)^{d_i} \\
&\leq e^{-\frac{1}{2}r^2 n}\exp\left[\sum_{i=1}^{k} d_i \log(n_i+1)\right] \\
&\leq e^{-\frac{1}{2}r^2 n}\exp\left[\sum_{i=1}^{k} d_i \sum_{j=1}^{n_i}\frac{1}{j}\right] \\
&\leq e^{-\frac{1}{2}r^2 n}\exp\left[\sum_{i=1}^{n} d_{\lambda(x_i)}\frac{k}{i}\right] \\
&\leq e^{-\frac{1}{2}r^2 n}\prod_{i=1}^{n} e^{\frac{k}{i}d_{\lambda(x_i)}}
\end{aligned}$$

Taking the expectation on both sides yields:

$$\begin{aligned}
\mathbf{Pr}[V(S) \geq r] &= e^{-\frac{1}{2}r^2 n}\prod_{i=1}^{n}\mathbf{E}\left[e^{\frac{k}{i}d_{\lambda(x_i)}}\right] \\
&\leq e^{-\frac{1}{2}r^2 n}\prod_{i=1}^{n}\sum_{j=1}^{k}\lambda_j e^{\frac{k}{i}d_j}
\end{aligned}$$

Setting

$$r := \sqrt{\frac{2\sum_{i=1}^{n}\log\left[\sum_{j=1}^{k}\lambda_j \exp\left(\frac{k}{i}d_j\right)\right] + \log\frac{1}{\delta}}{n}}$$

yields that with probability larger than $1-\delta$ it holds that

$$V(S) \leq \sqrt{\frac{2\sum_{i=1}^{n} \log\left[\sum_{j=1}^{k} \lambda_j \exp\left(\frac{k}{i} d_j\right)\right] + \log\frac{1}{\delta}}{n}}$$

Taking the union bound with theorem 5 (b) in [1] yields the result.

6.2 Proof of Theorem 3

Proof. For a sequence of n Rademacher variables $\sigma_1, \ldots, \sigma_n$ define

$$R_n(\mathcal{X}) := \mathbf{E}\left[\sup_{w \in \mathcal{B}_2^m} \left| \frac{2}{n} \sum_{i=1}^{n} \sigma_i l\left(y_i, w^T x_i\right) \right| \right]$$

The result follows from theorem 8 in [1] and the fact that $R_n(\mathcal{X}) \leq 2L\sqrt{\frac{\sum_{i=1}^{k} d_i \lambda_i}{n}}$. To see this, observe that

$$\begin{aligned}
R_n(\mathcal{X}) &\leq \mathbf{E}\left[\sup_{w \in \mathcal{H}} \frac{2}{n} \sum_{i=1}^{n} \sigma_i l(y_i, w^T x_i)\right] \\
&\leq 2L\,\mathbf{E}\left[\sup_{w \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i w^T x_i\right] \qquad (2) \\
&\leq 2L\,\mathbf{E}\left[\sup_{w \in \mathcal{H}} \|w\|_2 \left\| \frac{1}{n} \sum_{i=1}^{n} \sigma_i x_i \right\|_2\right] \qquad (3) \\
&\leq 2L\,\mathbf{E}\left[\sqrt{\frac{1}{n^2} \sum_{k=1}^{m} \sum_{i,j=1}^{n} \sigma_i \sigma_j f_k(x_i) f_k(x_j)}\right] \\
&\leq 2L \sqrt{\frac{1}{n^2}\,\mathbf{E}\left[\sum_{k=1}^{m} \sum_{i=1}^{n} f_k(x_i)^2\right]} \qquad (4) \\
&\leq 2L \sqrt{\frac{1}{n} \sum_{k=1}^{m} \mathbf{E}\left[f_k(x)\right]} \\
&\leq 2L \sqrt{\frac{\sum_{i=1}^{k} d_i \lambda_i}{n}}
\end{aligned}$$

Here, (2) is due to theorem 12 in [1], (3) is an application of Hölder's inequality, while (4) follows from the concavity of the square root and the independence of the Rademacher variables.

References

1. Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2003.
2. Björn Bringmann, Albrecht Zimmermann, Luc De Raedt, and Siegfried Nijssen. Don't be afraid of simpler patterns. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18-22, 2006, Proceedings*, volume 4213 of *Lecture Notes in Computer Science*, pages 55–66. Springer, 2006.
3. Mukund Deshpande, Michihiro Kuramochi, and George Karypis. Frequent substructure-based approaches for classifying chemical compounds. *Data Mining, IEEE International Conference on*, 0:35, 2003.
4. Luc Devroye, Laszlo Györfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability)*. Springer, New York, February 1996.
5. H. Fang, W. Tong, L.M. Shi, R. Blair, R. Perkins, W. Branham, B.S. Hass, Q. Xie, S.L. Dial, C.L. Moland, and D.M. Sheehan. Structure-activity relationships for a large diverse set of natural, synthetic, and environmental estrogens. *Chemical Research in Toxicology*, 14(3):280–294, 2001.
6. Hu Li, Chun Wei Yap, Choong Yong Ung, Ying Xue, Zhi Wei Cao, and Yu Zong Chen. Effect of selection of molecular descriptors on the prediction of blood-brain barrier penetrating and nonpenetrating agents by statistical learning methods. *Journal of Chemical Information and Modeling*, 45(5):1376–1384, 2005.
7. Richard Otter. The number of trees. *The Annals of Mathematics*, 49(3):583–599, 1948.
8. Ulrich Rückert and Stefan Kramer. Optimizing feature sets for structured data. In Joost N. Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *Machine Learning: ECML 2007, 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007, Proceedings*, volume 4701 of *Lecture Notes in Computer Science*, pages 716–723. Springer, 2007.
9. A.B. Teicher, editor. *The NCI Human Tumor Cell Line (60-Cell) Screen*, pages 41–62. Humana Press, second edition, 1997.
10. Nikil Wale, Ian A. Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.*, 14(3):347–375, 2008.
11. Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, page 721, Washington, DC, USA, 2002. IEEE Computer Society.
12. F. Yoshida and J. Topliss. QSAR model for drug human oral bioavailability. *J. Med. Chem.*, 43:2575–2585, 2000.