Automatic Model Adaptation for Complex Structured Domains

Geoffrey Levine, Gerald DeJong, Li-Lun Wang, Rajhans Samdani,
Shankar Vembu, and Dan Roth

Department of Computer Science
University of Illinois at Champaign-Urbana
Urbana, IL 61801
{levine,dejong,lwang4,rsamdan2,svembu,danr}@cs.illinois.edu

Abstract. Traditional model selection techniques involve training all candidate models in order to select the one that best balances training performance and expected generalization to new cases. When the number of candidate models is very large, though, training all of them is prohibitive. We present a method to automatically explore a large space of models of varying complexities, organized based on the structure of the example space. In our approach, one model is trained by minimizing a minimum description length objective function, and then derivatives of the objective with respect to model parameters over distinct classes of the training data are analyzed in order to suggest what model specifications and generalizations are likely to improve performance. This directs a search through the space of candidates, capable of finding a high performance model despite evaluating a small fraction of the total number of models. We apply our approach in a complex fantasy (American) football prediction domain and demonstrate that it finds high quality model structures, tailored to the amount of training data available.

1 Motivation

We consider a *model* to be a parametrically related family of hypotheses. Having a good model can be crucial to the success of a machine learning endeavor. A model that is too flexible for the amount of data available or a model whose flexibility is poorly positioned for the information in the data will perform badly on new inputs. But crafting an appropriate model by hand is both difficult and, in a sense, self-defeating. The learning algorithm (the focus of ML research) is then only partially responsible for any success; the designer's ingenuity becomes an integral component. This has given rise to a long-term trend in machine learning toward weaker models which in turn demand a great deal of world data in the form of labeled or unlabeled examples. Techniques such as structural risk minimization which *a priori* specify a nested family of increasingly complex models are an important direction. The level of flexibility is then variable and can be adjusted automatically based on the data itself.

This research reports our first steps in a new direction for automatically adapting model flexibility to the distinction that seem most important given the data. This allows adaptation to the *kind* of flexibility in addition to the level of complexity. We are interested in automatically constructing generative models to be used as a computational

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part II, LNAI 6322, pp. 243–258, 2010.

proxy for the real world. Once constructed and calibrated, such a model guides decisions within some prescribed domain of interest. Importantly, its utility is judged only within this limited domain of application. It can (and will likely) be wildly inaccurate elsewhere as the best model will concentrate its expressiveness where it will do the most good.

We contrast model learning with classification learning, which attempts to characterize a pattern given some appropriate representation of the data. Learning a model focuses on how best to represent the data. Real world data can be very complex. Given a myriad of distinctions that are possible, which are worth noticing? Which interactions should be emphasized and which ignored? Should certain combinations of distinctions be merged so as to pool the data and allow more accurate parameter estimation? In short, what is the most useful model from a space of related models? The answer depends on 1) the purpose to which the model will be applied: distinctions crucial for one purpose will be insignificant in others, 2) the amount of training data available: more data will generally support a more complex model, and 3) the emergent patterns in the training data: a successful model will carve the world "at its joints" respecting the natural equivalences and significant similarities within the domain.

The conventional tools for model selection include the minimum description length principle [1], the Akaike information criterion [2], and the Bayesian information criterion [3], as well as cross-validation. The disadvantage of these techniques is that in order to evaluate a model, it must be trained to the data. For many models, this is time consuming, and so the techniques do not scale well to cases where we would like to consider a large number of candidate models.

In adapting models, it is paramount to reduce the danger of overfitting. Just as an overfit hypothesis will perform poorly, so will an overfit model. Such a model would make distinctions relevant to the particular data set but not to the underlying domain. The flexibility that it exposes will not match the needs of future examples, and even the best from such a space will perform poorly. To drive our adaption process we employ available prior domain knowledge. For us, this consists of information about distinctions that many experts through many years, or even many generations, have discovered about the domain. This sort of prior knowledge has the potential to provide far more information than can reasonably be extracted from any training set.

For example, in evaluating the future earnings of businesses, experts introduce distinctions such as the Sector: {*Manufacturing, Service, Financial, Technology*} which is a categorical multiset and Market Capitalization: {*Micro, Small, Medium, Large*} which is ordinal. Distinctions may overlap, a company's numeric Beta and its Cyclicality: {*Cyclical, Non-cyclical, Counter-cyclical*} represent different views of the same underlying property. Such distinctions are often latent, in the sense that they are derived or emergent properties; the company is perfectly well-formed and well-defined without them. Rather they represent conceptualizations that experts have invented. When available such prior knowledge should be taken as potentially useful. Ignoring it when relevant may greatly increase the required amount of data to essentially re-derive the expert knowledge. But blindly adopting it can also lead to degraded performance. If the distinction is unnecessary for the task or if there is insufficient data to confidently make use, performance will also suffer. We explore how the space of distinctions interacts with training data. Our algorithm conducts a directed search through model structures, and

performs much better than simply trying every possibility. In our approach, one model is trained and analyzed to suggest alternative model formulations that are likely to result in better general performance. These suggestions guide a general search through the full space of alternative model formulations, allowing us to find a high quality model despite evaluating only a small fraction of the total number.

2 Preliminaries

To introduce our notation, we use as a running example the business earnings domain. Assume we predict future earnings with a linear function of N numerical features: f_1 to f_N. Thus, each prediction takes the form $\Phi \cdot F = \phi_1 f_1 + \phi_2 f_2 + ... + \phi_N f_N$. One possibility is to learn a single vector Φ. Of course, the individual ϕ_i's must still be estimated from training data, but once learned this single linear function will apply to all future examples. Another possibility is to treat companies in different sectors differently. Then we might learn one Φ for Manufacturing, a different one for Service companies, another for Financial companies, and another for Technology companies. A new example company is then treated according to its (primary) sector. On the other hand, perhaps Service companies and Financial companies seem to behave similarly in the training data. Then we might choose to lump those together, but to treat Manufacturing and Technology separately. Furthermore, we need not make the same distinctions for each feature. Consider f_i = unemployment rate. If there is strong evidence that the unemployment rate will have a different effect on companies based on their sector *and* size, we would estimate (and later apply) a specialized ϕ_i (for unemployment) based on sector and size together. Let D_i be the finest grain distinction for feature f_i (here, $Sector \times Size \times Cyclicity$). We refer to this set as the *domain of applicability* for parameter ϕ_i. Depending on the evidence, though, we may choose not to distinguish all elements D_i. The space of distinctions we can consider are the partitions of the D_i's. These form the alternative models that we must choose among.

Generally, the partitions of D_i form a lattice (as shown in Figure 1). This lattice, which we will refer to as Λ_{D_i}, is ordered by the *finer-than* operator (a partition P is said to be finer than partition P', if every elements of P is a subset of some element of P'). In turn, we can construct the Cartesian product lattice $\Lambda = \Lambda_{D_1} \times \Lambda_{D_2} \times ... \times \Lambda_{D_N}$. Note that Λ can have a very large number of elements. For example, if $|N|$ = 4, and $|D_i|$ = 4 for all i, then each lattice Λ_{D_i} has 15 elements and the joint lattice Λ has 15^4 = 50625 elements.

We formally characterize a model by (M, Θ_M), where $M = (P_1, P_2, ..., P_N)$ and $P_i = (S_{i,1}, S_{i,2}, ..., S_{i,|P_i|})$ is a partition of D_i, the domain of applicability for parameter type i. $\Theta_M = (\phi_{1,1}, \phi_{1,2}, ..., \phi_{1,|P_i|}, \phi_{2,1}, ..., \phi_{N,|P_N||})$, where each $\phi_{i,j}$ is the value of parameter i applicable to data points corresponding to $S_{i,j} \subseteq D_i$. We denote this, $c_i(x) \in S_{i,j}$, where c_i is a "characteristic" function. We refer to M as the *model structure* and Θ_M as the *model parameterization*.

Our goal is to find the trained model (M, Θ_M) that best balances simplicity with goodness of fit to the training data. For this paper, we choose to use the minimum description length principle [1]. Consider training data $X = \{x_1, x_2, ..., x_m\}$. In order

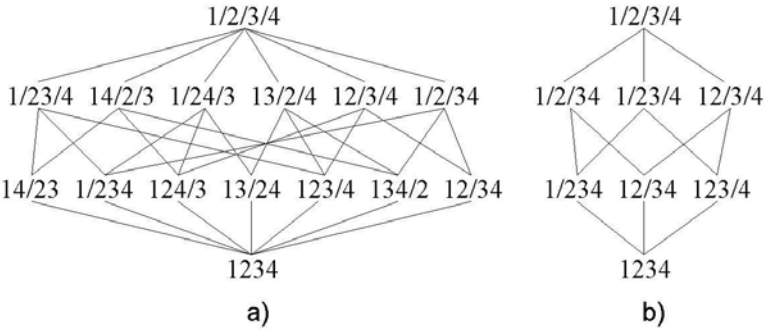

Fig. 1. Lattices of distinctions for four-class sets. If the classes are unstructured, for example the set of business sectors {Manufacturing, Service, Financial, Technology}, then we entertain the distinctions in lattice a). If the classes are ordinal, for example business sizes {Micro, Small, Medium, Large}, then we entertain only the partitions that are consistent with the class ordering, b). For example, we would not consider grouping small and large businesses while distinguishing them from medium businesses.

to evaluate the description length of the data, we use a two-part code combining the description length of the model and the description length of the data given the model:

$$L = DataL(X|(M, \Theta_M)) + ModelL((M, \Theta_M)) \tag{1}$$

for our approach we assume that $ModelL((M, \Theta_M))$ is a function only of the model structure M. Thus, adding or removing parameters affects the value of $ModelL$, but solely changing their values does not. We also assume that $DataL(X|(M, \Theta_M))$ can be decomposed into a sum of individual description lengths for each x_k. That is:

$$DataL(X|(M, \Theta_M)) = \sum_{x_k \in X} ExampleL(x_k|(M, \Theta_M)) \tag{2}$$

We assume that the function $ExampleL(x_k|(M, \Theta_M))$ is twice differentiable with respect to the model parameters, $\phi_{i,j}$, $1 \leq i \leq N$,$1 \leq j \leq |P_i|$.

Consider a model structure $M = (P_1, P_2, ..., P_N)$, $P_i = (S_{i,1}, S_{i,2}, ..., S_{i,|P_i|})$. We refer to a neighboring (in Λ) model $M' = (P'_1, P'_2, ..., P'_N)$ as a refinement of M if there exists a value j such that $P'_j = (S_{j,1}, ..., S_{j,k-1}, Y, Z, S_{j,k+1}, ..., S_{j,|P_j|})$, (Y, Z) is a partition of $S_{i,j}$, and $P'_i = P_i$ for all i $\neq$ j. That is, a refinement of M is a model which makes one additional distinction that M does not make.

Likewise, M' is a generalization of M if there exists a value j such that $P'_j = (S_{j,1}, ..., S_{j,k-1}, S_{j,k} \cup S_{j,k+1}, S_{j,k+2}, ..., S_{j,|P_j|})$, and $P'_i = P_i$ for all i $\neq$ j. That is, a generalization of M is a model that makes one less distinction than M.

3 Model Exploration

The number of candidate model structures in Λ explodes very quickly as the number of potential distinctions increases. Thus, for all but the simplest spaces, it is computationally infeasible to train and evaluate all such model structures.

Instead, we offer an efficient exploration method to explore lattice Λ in order to find a (locally) optimal value of (M, Θ_M). The general idea is to train a model structure M, arriving at parameter values Θ_M, and then leverage the differentiability of the description length function to estimate the value for other model structures, in order to direct the search through Λ.

3.1 Objective Estimation

Note that at convergence, $\frac{\partial L}{\partial \phi_{i,j}} = 0$ for all $\phi_{i,j}$. As $ModelL$ is fixed for a fixed M, and data description length is the sum of description lengths for each training example we have that

$$\sum_{x_k \in X} \left(\frac{\partial ExampleL(x_k|(M, \Theta_M))}{\partial \phi_{i,j}} \right) = 0 \tag{3}$$

Recalling that $\phi_{i,j}$ is applicable only over $S_{i,j} \subseteq D_i$, this can be rewritten:

$$\sum_{w \in S_{i,j}} \sum_{x_k \ s.t. \ c_i(x_k)=w} \left(\frac{\partial ExampleL(x_k|(M, \Theta_M))}{\partial \phi_{i,j}} \right) = 0 \tag{4}$$

Note that the inner summation (over each w) need not equal zero. That is, the training data may suggest that for class $w \in D_i$, parameter ϕ_i should be different than the value $\phi_{i,j}$. However, because the current model structure does not distinguish w from the other elements of $S_{i,j}$, $\phi_{i,j}$ is the best value across the entire domain of $S_{i,j}$.

In order to determine what distinctions we might want to add or remove, we consider the effect that each parameter has on each fine-grained class of data. Let $w \in S_{i,j} \subseteq D_i$ and $v \in S_{g,h} \subseteq D_g$. Let $w \wedge v$ denote the set $\{x_k | c_i(x_k) = w \wedge c_g(x_k) = v\}$. We define the following quantities:

$$d_{\phi_{i,j},w} = \sum_{x_k \ s.t. \ c_i(x_k)=w} \left(\frac{\partial ExampleL(x_k|(M, \Theta_M))}{\partial \phi_{i,j}} \right) \tag{5}$$

$$d_{\phi_{g,h},v} = \sum_{x_k \ s.t. \ c_g(x_k)=v} \left(\frac{\partial ExampleL(x_k|(M, \Theta_M))}{\partial \phi_{g,h}} \right) \tag{6}$$

$$dd_{\phi_{i,j},\phi_{g,h},w,v} = \sum_{x_k \in w \wedge v} \left(\frac{\partial^2 ExampleL(x_k|(M, \Theta_M))}{\partial \phi_{i,j} \partial \phi_{g,h}} \right) \tag{7}$$

The first two values are the first derivatives of the objective with respect to $\phi_{i,j}$ and $\phi_{g,h}$ for the examples corresponding to $w \in D_i$ and $v \in D_g$ respectively. The third equation is the second derivative taken once with respect to each parameter. Note that the value is zero for all examples other than those in $w \wedge v$. Consider the model M^* that makes every possible distinction (the greatest element of lattice Λ). Computed over all $1 \leq i, g \leq N$, $w \in D_i$, $v \in D_j$, these values allow us to construct a second order Taylor expansion polynomial estimation for the value of $L((M^*, \Theta_{M^*}))$ for all values of Θ_{M^*}.

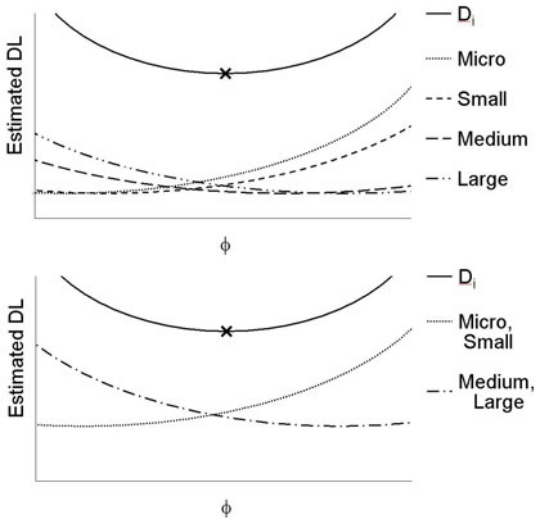

Fig. 2. Example polynomial estimation of description length considering distinctions based on business size. For no distinctions, description length is minimized at point x. However, Taylor expansion estimates that the behavior of micro and small businesses is substantially different than medium or large businesses. This suggest that the distinction {{Micro, Small}, {Medium, Large}} should be entertained if the expected reduction in description length of the data is greater than the cost associated with the additional parameter.

$$\widehat{DataL}(X|(M^*,\Theta_{M^*})) = DataL(X|(M,\Theta_M)) + \sum_{\substack{1\le i\le N \\ w\in D_i}} (\phi^*_{i,w} - \hat{\phi}_{i,j_w}) \times d_{\phi_{i,j_w},w} + \sum_{\substack{1\le i\le N \\ w\in D_i}} \sum_{\substack{1\le g\le N \\ v\in D_g}} (\phi^*_{i,w} - \hat{\phi}_{i,j_w})(\phi^*_{g,v} - \hat{\phi}_{g,h_v}) \times \frac{dd_{\phi_{i,j_w},\phi_{g,h_v},w\wedge v}}{2} \tag{8}$$

where $\hat{\phi}_{i,j_w}$ is the value of $\phi_{i,j}$, where $w \in S_{i,j}$. Note that this polynomial is the same polynomial that would be constructed from the gradient and Hessian matrix in Newton's method. By minimizing this polynomial, we can estimate the minimum L for M^*. More generally, we can use the polynomial to estimate the minimum L for any model stucture M' in Λ. Suppose we wish to consider a model that does not distinguish between classes w and $w' \in D_i$ with respect to parameter i. To do this, we enforce the constraint $\phi_{i,w} = \phi_{i,w'}$, which results in a polynomial with one fewer parameters. Minimizing this polynomial gives us an estimate for the minimum value of $DataL$ of the more general model structure. In this manner, any model structure can be estimated by placing equality constraints over parameters corresponding to classes not distinguished. A simple one dimensional example is detailed in Figure 2.

We can then estimate the complete minimum description length M':

$$\min_{\Theta_{M'}} \widehat{L}((M', \Theta_{M'}) = ModelL(M') + \min_{\Theta_{M'}} \widehat{DataL}((M', \Theta_{M'})) \tag{9}$$

3.2 Theoretical Guarantees

When considering alternative model structures, we are guided by estimates of their minimum description length. However, if the domain satisfies certain criteria, we can compute a lower bound for this value, which may result in greater efficiency.

Consider model formulation (M, Θ_M); let values $d_{\phi_{i,j},w}$ be computed as described above.

Theorem 1 *Consider maximal model (M^*, Θ_{M^*}). Assume $DataL(X|(M^*, \Theta_{M^*}))$ is twice continuously differentiable with respect to elements of Θ_{M^*} Let $H(\Theta_{M^*})$ be the Hessian matrix of the $DataL(X|(M^*, \Theta_{M^*}))$ with respect to Θ_{M^*}. If $y^T H(\Theta_{M^*})y \geq b > 0 \; \forall \; \Theta_{M^*}, y$ st. $||y||_2 = 1$, then*

$$\begin{aligned} \overline{DataL}(X|(M^*, \Theta_{M^*})) &= DataL(X|(M, \Theta_M)) \\ &+ \sum_{\substack{1 \leq i \leq N \\ w \in D_i}} \left((\phi^*_{i,w} - \hat{\phi}_{i,j_w}) \times d_{\phi_{i,j_w},w} + (\phi^*_{i,w} - \hat{\phi}_{i,j_w})^2 \times \frac{b}{2} \right) \end{aligned} \tag{10}$$

is a lower bound polynomial on the value of $DataL(X|(\Phi_{M^*}, \Theta_{M^*}))$.

Proof. The Hessian of $\overline{DataL}(X|M^*, \Theta_{M^*}))$ with respect to Θ, $\overline{H}(\Theta_{M^*})$, is equal to b times the identity matrix. Thus $\forall \; \Theta_{M^*}, y$ st. $||y||_2 = 1$, $y^T \overline{H}(\Theta_{M^*})y = b$. Let $z = ((\phi^*_{i,w_1} - \hat{\phi}_{i,j_{w_1}}), ..., (\phi^*_{N,w_{|D_N|}} - \hat{\phi}_{i,j_{w_{|D_N|}}}))$, and let $y = \frac{z}{|z|}$. By Taylor's Theorem,

$$\begin{aligned} DataL(X|(M^*, \Theta_{M^*})) &= DataL(X|(M, \Theta_M)) \\ &+ \sum_{\substack{1 \leq i \leq N \\ w \in D_i}} \left((\phi^*_{i,w} - \hat{\phi}_{i,j_w}) \times d_{\phi_{i,j_w},w} + \frac{(\phi^*_{i,w} - \hat{\phi}_{i,j_w})^2 y^T H(\Theta'_{M^*})y}{2} \right) \end{aligned} \tag{11}$$

for some Θ'_{M^*} on the line connected Θ_M and Θ_{M*}. Thus, by our assumptions on the Hessian matrix we know that,

$$\begin{aligned} DataL(X|(M^*, \Theta_{M^*})) &\geq DataL(X|(M, \Theta_M)) \\ &+ \sum_{\substack{1 \leq i \leq N \\ w \in D_i}} \left((\phi^*_{i,w} - \hat{\phi}_{i,j_w}) \times d_{\phi_{i,j_w},w} + \frac{(\phi^*_{i,w} - \hat{\phi}_{i,j_w})^2 b}{2} \right) \end{aligned} \tag{12}$$

□

When the condition holds, this derivation allows us not only to lower bound the data description length of M^*, but the length for any model structure in Λ. In the same manner as above, placing equality constraints on sets of $\phi^*_{i,j}$'s results in a lower order

polynomial estimation for $DataL$. In the same format as Equation 9, we can compute an optimistic lower bound for any model's value of L. $b > 0$ is satisfied for all cases where the objective function is strongly convex, however, the value is data and model format sensitive, so we can not offer a general solution to compute it.

Note that the gap between the estimated lower bound on $\min_{\Theta'_M} L(M', \Theta'_M)$ and its actual value will generally grow as the first derivative of $DataL(X|(M', \Theta_{M'}))$ increases. That is, we will compute more meaningful lower bounds of $\min_{\Theta'_M} L(M', \Theta'_M)$ for models whose optimal parameter values are "close" to our current values.

3.3 Model Search

Given a trained model, using the techniques described above, we can estimate and lower bound the value of L and estimate the optimal parameter settings for any alternative model, M'. We offer two general techniques to search through the space of model structures. In the first approach, we maintain an optimistic lower bound on the value $min_{\Theta_{M'}} L((M', \Theta_{M'}))$ for all M'. At each step, we select for training the model structure M' with the lowest optimistic bound for L. After training, we learn it's optimal parameter values, Θ'_M, and associated description length $L((M', \Theta'_M))$. We then use Equation 10 to generate the lower bounding Taylor expansion polynomial around $\Theta_{M'}$. This polynomial is then used to update the optimistic description lengths for all alternative models (increasing but never decreasing each bound). We proceed until a model M' has been evaluated whose description length is within ϵ of the minimum optimistic bound across all unevaluated models. At this point we adopt model M'.

Of course, the number of such models grows exponentially with the number of example classes. Thus, even maintaining optimistic bounds for all such models may be prohibitive. Thus, we present an alternative model exploration technique that hill-climbs in the lattice of models. In this approach, we iterate by training a model M, and then estimate $\widehat{L}((M', \Theta_{M'}))$ only for model structures M' that are neighbors (immediate generalizations and specializations) of M in lattice Λ. These alternative model structures are limited in number, making estimation computationally feasible, and similar to the current trained model. Thus, we expect the optimal parameter settings for these models will be "close" to our current parameter values, so that the objective estimates will be reasonably accurate. At this point, we can transition to and evaluate model M' with the lowest estimated value of $\widehat{L}((M', \Theta_{M'})$, from the neighbors of M. This cycle repeats until no neighboring models are estimated to decrease the description length, at which point the evaluated model with minimum L is adopted. For the complex fantasy football domain presented in the following section, the number of models in Λ is computationally infeasible, and so we use the alternative greedy exploration approach.

4 Fantasy Football

Fantasy football [4] is a popular game that millions of people participate in each fall during the American National Football League season. The NFL season extends 17 weeks, in which each of the 32 "real" teams plays 16 games, with one bye (off) week. In fantasy football, participants manage virtual (fantasy) teams composed of real players,

and compete in virtual games against other managers. In these games, managers must choose which players on their roster to make active for the upcoming week's games, while taking into account constraints on the maximum number of active players in each position. A fantasy team's score is then derived from the active players' performances in their real-world games. While these calculations vary somewhat from league to league, a typical formula is:

$$\begin{aligned} FantasyPoints = &+\frac{RushingYards}{10} + 6 \times RushingTouchDowns \\ &+\frac{ReceivingYards}{10} + 6 \times ReceivingTouchDowns \\ &+\frac{PassingYards}{25} + 4 \times PassingTouchDowns - 1 \times PassingInterceptions \\ &+3 \times FieldGoalsMade + ExtraPointsMade \end{aligned} \tag{13}$$

The sum of points earned by the active players during the week is the fantasy team's score, and the team wins if its score is greater than its opponent's. Thus, being successful in fantasy football necessitates predicting as accurately as possible the number of points players will earn in future real games.

Many factors affect how much and how effectively a player will play. For one, the player will be faced with a different opponent each week, and the quality of these opponents can vary significantly. Second, American football is a very physical sport and injuries, both minor and serious, are common. While we expect an injury to decrease the injured player's performance, it may increase the productivity of teammates who may then accrue more playing time.

American football players all play a primary position on the field. The positions that are relevant to fantasy football are quarterbacks (QB), running backs (RB), wide receivers (WR), tight ends (TE), and kickers (K). Players at each of these positions perform different roles on the team, and players at the same position on the same NFL team act somewhat like interchangeable units. In a sense, these players are in competition with each other to earn playing time during the games, and the team exhibits a preference over the players, in which high priority players (starters) are on the field most of the game and other players (reserves) are used sparingly.

4.1 Modeling

Our task is to predict the number of points each fantasy football player will earn in the upcoming week's games. Suppose the current week is week w (let week 1 refer to the first week for which historical data exists, not the first week of the current season). In order to make these predictions, we have access to the following data:

- The roster of each team for weeks 1 to w
- For each player, for each week 1 to $w - 1$ we have:
  - The number of fantasy points that the player earned, and
  - The number of plays in which the player actively participated (gained possession of or kicked the ball)

– For each player, for each week 1 to w we have the players pregame injury status

If we normalize the number of plays in which a player participated by the total number across all players at the same position on the same team, we get a fractional number which we will refer to as *playing time*. For example, if a receiver catches 6 passes in a game, and amongst his receiver teammates a total of 20 passes are caught, we say the player's playing time = .3. Injury statuses are reported by each team several days before each game and classify each player into one of five categories:

1. Healthy (H): Will play
2. Probable (P): Likely to play
3. Questionable (Q): Roughly 50% likely to play
4. Doubtful (D): Unlikely to play
5. Out (O): Will not play

In what follows we define a space of generative model structures to predict fantasy football performance. The construction is based on the following ideas. We assume that each player has two inherent latent features: *priority* and *skill*. Priority indicates how much they are favored in terms of playing time compared to the other players at the same position on the same team. Skill is the number of points a player earns, on average, per unit of playing time. Likewise, each team has a latent skill value, indicating how many points better or worse than average the team gives up to average players. Our generative model assumes that these values are generated from Gaussian prior distributions $N(\mu_{pp}, \sigma^2_{pp})$, $N(\mu_{ps}, \sigma^2_{ps})$, and $N(\mu_{ts}, \sigma^2_{ts})$ respectively.

Consider the performance of player i on team t in week w. We model the playing time and number of points earned by player i as random variables with the following means:

$$\overline{PlayingTime}_{i,w} = \frac{e^{pp_i + injury(i,w)}}{\sum_{j \in R_t, pos(j)=pos(i)} e^{pp_j + injury(j,w)}} \tag{14}$$

$$\overline{Points}_{i,w} = PlayingTime_{i,w} \times ps_i \times ts(opp(t,w), pos(i)) \tag{15}$$

where R_t is the set of players on team t's roster, $pos(i)$ is the position of player i, $injury(i,w)$ is a function mapping to the real numbers that corresponds to the player's injury's effect on his playing time. We assume, then, that the actual values are distributed as follows:

$$PlayingTime_{i,w} \sim N(\overline{PlayingTime}_{i,w}, \sigma^2_{time}) \tag{16}$$

$$Points_{i,w} \sim N(\overline{Points}_{i,w}, PlayingTime_{i,w}\sigma^2_{points}) \tag{17}$$

We do not know *a priori* what distinctions are worth noticing, in terms of variances, prior distributions, and injury effects. For example, do high priority players have significantly higher skill values than medium priority players? Does a particular injury status have different implications for tight ends than for kickers? Of course, the answer to these questions depends on the amount of training data we have to calibrate our model. We utilize the greedy model structure exploration procedure defined in Section 3.3 to answer these questions. For this domain, we entertain alternative models based on the following parameters and domains of applicability:

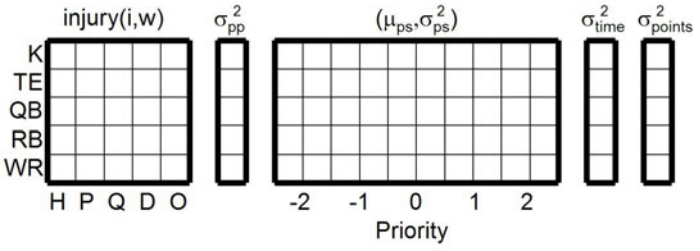

Fig. 3. The space of model distinctions. For each parameter, the domain of applicability is carved up into one or more regions along the grid lines, and each region is associated with a distinct parameter value.

1. $injury(i, w) : D_1 = Position \times InjuryStatus$
2. $\sigma^2_{pp} : D_2 = Position$ (μ_{pp} is arbitrarily set to zero)
3. $(\mu_{ps}, \sigma^2_{ps}) : D_3 = Position \times Priority$
4. $\sigma^2_{time} : D_4 = Position$
5. $\sigma^2_{points} : D_5 = Position$

Figure 3 illustrates the space of distinctions. We initialize the greedy model structure search with the simplest model, that makes no distinctions for any of the five parameters.

Given a fixed model structure M, we utilize the expectation maximization [5] procedure to minimize $DataL(X|(M, \Theta_M)) = -log_2 P(X|(M, \Theta_M))$. This procedure alternates between computing posterior distributions for the latent player priorities, skills, and team skills for fixed Θ_M, and then re-estimating Θ_M based on these distributions and the observed data. In learning these values, we limit the contributing data to a one year sliding window preceding the week in question. Additionally, because players' priorities change with time, we apply an exponential discount factor for earlier weeks and seasons. This allows the model to bias the player priority estimates to reflect the players' current standings on their team. We found that player and team skill features change little within the time frame of a year, and so discounting for these values was not necessary.

$ModelL((M, \Theta_M))$, the description length of the model, has two components, the representation of the model structure M, and the representation of Θ_M. We choose to make the description length of M constant (equivalent to a uniform prior over all model structures). The description length of Θ_M scales linearly with the number of parameters. Although in our implementation, these values are represented as 32-bit floating point values, 32 bits is not necessarily the correct description length for each parameter as it fails to capture the useful range and grain-size. Therefore, this parameter penalty, along with the week and year discount factors, are learned via cross validation.

5 Experiments

A suite of experiments demonstrates the following: First, given an amount of training data, the greedy model structure exploration procedure suitably selects a model (of the

appropriate complexity), to generalize to withheld data. Second, when trained on the full set of training data, the model selected by our approach exceeds the performance of suitable competitors, including a standard support vector regression approach and a human expert.

We have compiled data for the 2004-2008 NFL seasons. As the data must be treated sequentially, we choose to utilize the 2004-2005 NFL season data for training, 2006 data for validation, and the 2007-2008 data for testing each approach.

First we demonstrate that, for a given amount of training data, our model structure search selects an appropriate model structure. We do this by using our validation data to selecting model structures based on various amounts of training data, and then evaluate them in alternative scenarios where different amounts of data are available.

Due to the interactions of players and teams in the fantasy football domain, we cannot simply throw out some fraction of the players to learn a limited-data model. Instead, we impose the following schema to learn different models corresponding to different amount of data. We randomly assign the players into G artificial groups. That is, for G = 10, each group contains (on average) one tenth of the total number of players. Then, we learn different model structures and parameter values for each group, although all players still interact in terms of predicted playing time and points are as describe in Equation 14.

For example, consider the value μ_{ps}, the mean player skill for some class of players. Even if no other distinctions are made (those that could be made based on position or priority), we learn G values for μ_{ps}, one for each group, and each parameter value is based only on the players in one group. As G increases, these parameters are estimated based on fewer players. As making additional distinctions carries a greater risk of over-fitting, in general, we expect the complexity of the best model to decrease as G increases.

In order to evaluate how well our approach selects a model tailored to an amount of training data, we utilize the 2006 validation data to learn models for each of G_{train} = 1, 4, and 16. In each case we learn G_{train} different models (one for each group). Then for each week w in 2007-2008, we again randomly partition the players, but into a different number of groups, G_{test}. For each of the G_{test} groups, we sample at random a model structure uniformly from those learned. Then, model parameters and player/team latent variables are re-estimated using EM with data for the one year data window leading up to week w, for each of the G_{test} models. Finally, predictions are made for week w and compared to the players' actual performances. We repeat this process three times for each (G_{train}, G_{test}) pair and report the average results. We also report results for each value of G_{test} when the model structure is selected uniformly at random from the entire lattice, Λ.

We expect that if our model structure selection technique behaves appropriately, for each value of G_{test}, performance should peak when $G_{train} = G_{test}$. For cases where $G_{train} < G_{test}$ the model structures will be too flexible for the more limited parameter estimation data available during testing, and performance will suffer due to overfitting. On the other hand, when $G_{train} > G_{test}$, the model structures cannot appreciate all the patterns in the calibration data. The root mean squared error of each model for each test grouping is shown in Figure 4. In fact, for each value of G_{test} we see that

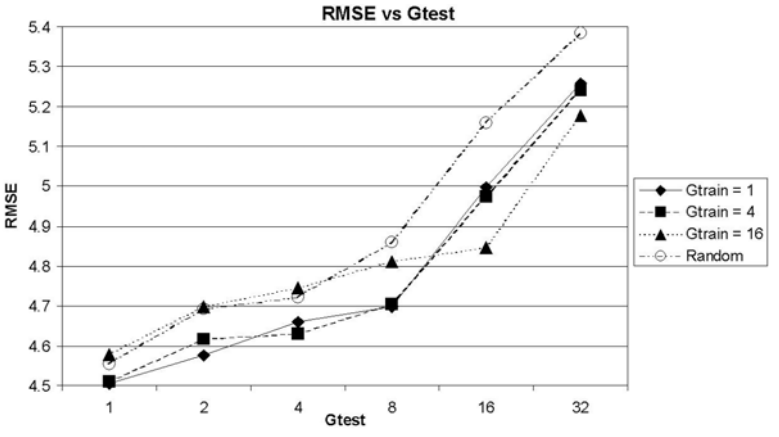

Fig. 4. Root mean squared errors for values of G_{test}. Model structures are learned from the training data for different values of G_{train} or sampled randomly from Λ.

performance is maximized when $G_{train} = G_{test}$, suggesting that our model structure selection procedure is appropriately balancing flexibility with generalization, for each amount of training data.

Figure 5 shows the model structure learned when G_{train} = 1, as well as a lower-complexity model learned when G_{train} = 16. For G_{train} = 1, the model structure selection procedure observes sufficient evidence to distinguish σ^2_{time} and σ^2_{points} with respect to each position. The model makes far more distinctions for high priority players than their lower priority counterparts. This is likely due to two reasons. First, the elite players' skills are further spaced out than the reserve level players, whose skills are closer to average and thus more common across all players. Second, because the high-priority players play more often than the reserves, there is more statistical evidence to justify these distinctions. The positions of quarterback, kicker and tight end all have the characteristic that playing time tends to be dominated by one player, and the learned model structure makes no distinction for the variance of priorities across these positions. Finally, the model does not distinguish the injury statuses *healthy* and *probable*, nor does it distinguish *doubtful* and *out*. Thus, *probable* appears to suggest that the player will almost certainly participate at close to his normal level, and *doubtful* means the player is quite unlikely to play at all.

In general, models learned for G_{train} = 16 contain fewer overall distinctions. In this case the model is similar to its G_{train} = 1 counterpart, except that it makes far fewer distinctions with regard to the priority skill prior.

Finally, we compare the prediction accuracy of our approach to those of a standard support vector regression technique and a human expert. For the support vector regression approach we use the LIBSVM [6] implementation of ϵ-SVR with a RBF kernel.

Consider the prediction for the performance of player i on team t in week w. We train four SVR's with different feature sets, starting with a small set of the most informative features and enlarging it to include less relevant teammate and opponent features. The

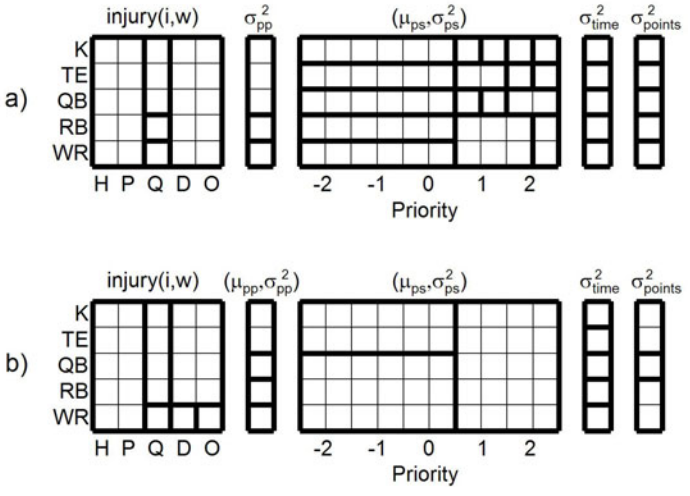

Fig. 5. Model structure learned for a) G_{train} = 1, and b) G_{train} = 16. Distinctions made with respect to 1) injury weight, 2) priority prior variance, 3) skill prior mean/variance, 4) playing time variance, and 5) points variance are shown in bold.

first SVR (SVR_1) includes only the points earned by player i in each of his games in the past year. Bye weeks are ignored, so f_1 is the points earned by player i in his most recent game, f_2 corresponds to his second most recent game, etc. For SVR_2, we also include in the feature set player i's playing time for each game, as well as his injury status each game (including the upcoming game). SVR_3 adds the points, playing times, and injury statuses for each teammate of player i at the same position each game. Finally, SVR_4 adds for teams that player i has played against in the last year, as well as his upcoming opponent, the total number of fantasy points given up by the team for each of their games in the data window. At each week w, we train one SVR for each position, using one example for each player at each week y, $w - h \leq y \leq w - 1$ (an example for week y has features based on weeks $y - h$ to y). All features are scaled to have absolute range [0,1] within the the training examples. We utilize a grid search on the validation data to choose values for ϵ, γ, and C.

We also compare our accuracy against statistical projections made by the moderator of the fantasy football website (www.fftoday.com) [7]. These projections, made before each week's games, include predictions on each of the point earning statistical categories for many of the league's top players. From these values, we compute a projected number of fantasy points according to Equation 13. There are two caveats, the expert does not make projections for all players, and the projected statistical values are always integral, whereas our approach can predict any continuous number of fantasy points. To have a fair comparison, we compare results based only on the players for which the expert has made a prediction using the normalized Kendall tau distance. For this comparison, we construct two orderings each week, one based on projected points, the other based on actual points. The distance is then the number of disagreements between the

Table 1. Performance of our approach versus human expert and support vector regressors with various feature sets

	All Data		Expert Predicted Data	
	RMSE	Normalized Kenall Tau	RMSE	Normalized Kenall Tau
Our Approach	4.498	.2505	6.125	.3150
Expert	N/A	N/A	6.447	.3187
SVR_1	4.827	.2733	6.681	.3311
SVR_2	4.720	.2674	6.449	.3248
SVR_3	4.712	.2731	6.410	.3259
SVR_4	4.773	.2818	6.436	.3323

two orderings, normalized to the range [0,1] (0 if the orderings are the same, 1 for complete disagreement). By considering only the predicted ordering of players and not their absolute projected number of points, the expert is not handicapped by his limited prediction vocabulary. We compute the Kendall tau distances for each method each week, and present the average value across all weeks 2007-2008.

Table 1 shows that our approach compares favorably with both the SVR and the expert. Again, note that because of the constrained vocabulary in which the expert predicts points, the final column is the only completely fair comparison with the expert. Of the candidate SVR feature sets, SVR_2 (with player i's points, playing times, and injury statuses) and SVR_3 (adding teammates' points, playing times, and injury statuses) perform the best.

6 Related Work

Our work on learning model structure is related to previous work on graphical-model structure learning, including Bayesian networks. In cases where a Bayes net is generating the data, a greedy procedure to explore the space of networks is guaranteed to converge to the correct structure as the number of training cases increases [8]. Friedman and Yakhini [9] suggest exploring the space of Bayes nets structures using simulated annealing and a BIC scoring function. The general task of learning the best Bayesian Network according to a scoring function that favors simple networks is NP-hard [10]. For undirected graphical models such as Markov Random Fields, application of typical model selection criteria is hindered by the necessary calculation of a probability normalization constant, although progress has been made on constrained graphical structures, such as trees [11,12]. Our approach differs most notably from these in that we not only consider the relevancy of each feature, but the possible grouping of that feature's value. We also present a global search strategy for selecting model structure, and our approach applies when variables are continuous and interactions are more complex than a Bayesian network can capture.

Another technique, reversible jump Markov chain Monte Carlo [13], generalizes Markov chain Monte Carlo to entertain jumps between alternative spaces of differing dimensions. Using this approach, it is possible to perform model selection based on the posterior probability of models with different parameter spaces. The approach

requires that significant care be taken in defining the MCMC proposal distributions in order to avoid exorbitant mixing times. This difficulty is magnified when the models are organized in a high-degree fashion, as is the case for our lattice.

7 Conclusion

In this paper, we present an approach to select a model structure from a large space by only evaluating a small number of candidates. We present two search strategies, one global strategy guaranteed to find a model within ϵ of the best scoring candidate in terms of MDL. The second approach hill climbs in the space of model structures. We demonstrate our approach on a difficult fantasy football prediction task, showing that the model selection technique appropriately selects structures for various amounts of training data, and that the overall performance of the system compares favorably with the performance of a support vector regressor as well as a human expert.

Acknowledgments

This work is supported by an ONR Award on "Guiding Learning and Decision Making in the Presence of Multiple Forms of Information."

References

1. Grunwald, P.: The Minimum Description Length Principle. MIT Press, Cambridge (2007)
2. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
3. Schwarz, G.E.: Estimating the dimension of a model. Annals of Statistics 6(2), 461–464 (1978)
4. ESPN: Fantasy football, `http://games.espn.go.com/frontpage/football` (Online; accessed 15-April-2008)
5. Hogg, R., McKean, J., Craig, A.: Introduction to Mathematical Statistics. Pearson Prentice Hall, London (2005)
6. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
7. Krueger, M.: Player rankings and projections - ff today, `http://www.fftoday.com/rankings/index.html` (Online; accessed 8-April-2008)
8. Chickering, D.: Optimal structure identification with greedy search. Journal of Machine Learning Research 3, 507–554 (2002)
9. Friedman, N., Yakhini, Z.: On the sample complexity of learning bayesian networks. In: The 12th Conference on Uncertainty in Artificial Intelligence (1996)
10. Chickering, D.: Large-sample learning of bayesian networks is np-hard. Journal of Machine Learning Research 5, 1287–1330 (2004)
11. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14(3), 462–467 (1968)
12. Srebro, N.: Maximum likelihood bounded tree-width markov networks. Artificial Intelligence 143, 123–138 (2003)
13. Brooks, S., Giudici, P., Roberts, G.: Efficient construction of reversible jump markov chain monte carlo proposal distributions. Journal of the Royal Statistical Society (65), 3–55 (2003)