Detecting Feature Interactions from Accuracies of Random Feature Subsets

Thomas R. Ioerger
Department of Computer Science
Texas A&M University

Feature Interactions

- Intuitive definition - *when the relationship between one feature and the target concept explicitly depends on the value of another feature(s)*
- One of the most significant causes of difficulty for learning (data-mining) in real-world domains
- Example
  - daytime: negative correlation between cloudiness and air temperature
  - night: positive correlation (prevents cooling)
  - if is time ignored: patterns blur together
- Another Example: parity in Boolean concepts (e.g. $A \otimes$B)
- Causes "concept dispersion"
- interferes with biases of almost all learning algorithms

Feature Construction

- If feature interactions were known, new features could be constructed that combine them

  (`sky=cloudy` $\wedge$ `time-of-day=night`)

  compute for each example to help predict `temp`

- Examples of Feature Construction Systems
  - STAGGER (Schlimmer, 1987)
  - CITRE (Matheus, 1989)
  - MIRO (Drastal et al., 1989)
  - FRINGE (Pagallo and Haussler, 1990)
  - LFC (Ragavan and Rendell, 1993)
  - many more...

- Problem: *combinatorial complexity* of selecting features and operators to construct

An Empirical Method for Detecting Feature Interactions

- Idea: *Use performance of learning algorithm itself on subsets of features to detect interactions*
  1. run algorithm on random subsets of features
  2. identify when accuracy is higher than "expected"
  3. construct new features from interacting ones

- to be defined:
  - sampling strategy
  - "expected" accuracy of feature subsets
  - how to construct new features

- Similar to "wrapper" approaches for feature selection, except...
  - look for subset with most *unexpected* accuracy, instead of highest
  - construct new features once interactions detected

Identifying Interactions

- Empirical definition - compare observed accuracy of learning algorithm run on subset of features with what was "expected"

- What is accuracy should be expected?
  - simplistic: best individual feature in subset
  - better: combine accuracy of best feature with *classifier* built from rest
  - (assumes best feature is part of interaction)
  - *Bayesian ensemble* model

$$\mathsf{P}[L_{F \cup G}(x) = c(x)] =$$

$$\sum_{c_j, c_k} (\ max_{c_i}\ \mathsf{P}[L_F(x) = c_j \mid c(x) = c_i] \times$$

$$\mathsf{P}[L_G(x) = c_k \mid c(x) = c_i] \cdot \mathsf{P}[c(x) = c_i]\ )$$

- Final test
  1. run alg. on Best feature and Rest of features
  2. calculate expected accuracy for all
  3. run algorithm on all features in subset
  4. identify subset with max ($Observed - Expected$)

Sampling Strategy

- exhaustive search is exponential in order of interaction; ~3-way combinations at best can be tested
- test works only when *best* feature interacts - called the “apical” feature
- Idea: sample in stages
  - treat each feature as the apical one in turn
  - combine only with random subsets of *worse* features
- Key point: *including non-interacting features of lower accuracy usually doesn't affect test*
  - If $\{F_1, F_2\}$ interact, then $\{F_1, F_2, F_3\}$ should also have higher accuracy than expected (assuming $acc(F1) > acc(F2) > acc(F3)$)
  - Can sample larger subsets, fewer tests required
  - What is the probability of selecting 3 interacting features in random subset of 6?
  - How many times must repeat for 95% conf.?

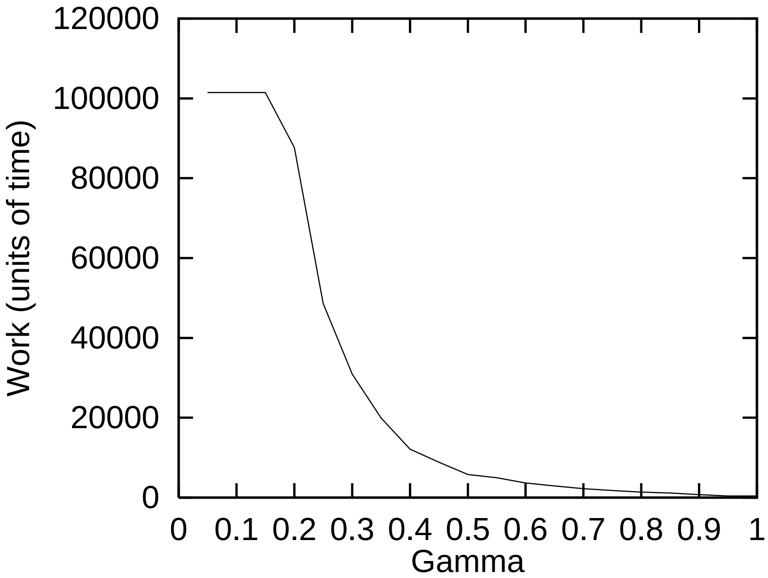
$$\text{Pr} = 1 - \left(1 - \frac{{}_{n-b}C_{k-b}}{{}_{n}C_{k}}\right)^r$$

- γ: defines size of subsets tested for interaction with each apical feature

- e.g. $\gamma = 0.5$ means combinations of 4 out of 8 lower-accuracy features are tested with 9th worst feature, etc.

- determining sampling schedule for limited CPU time

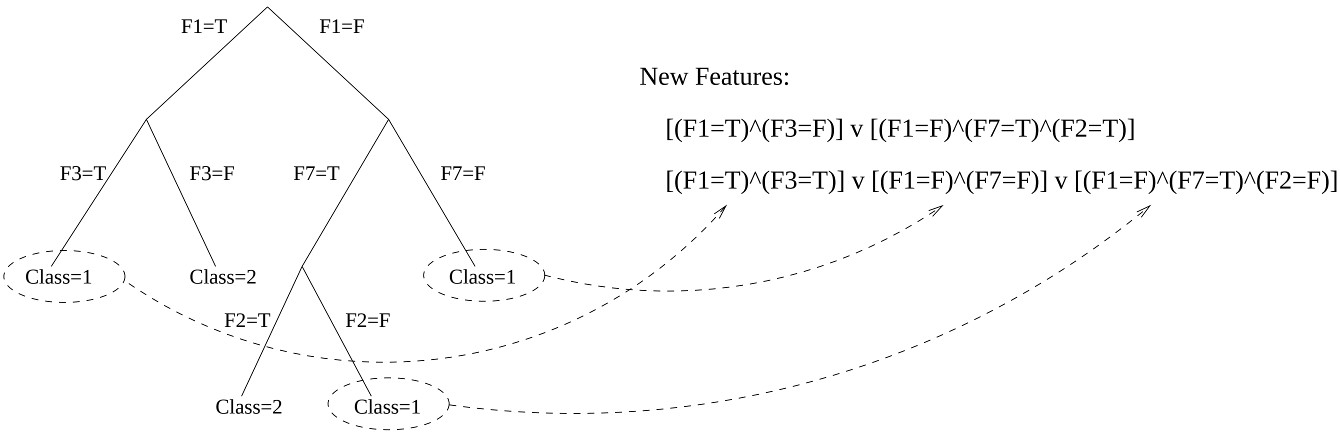
  1. pick initial value for γ
  2. calculate number of subsets to sample for each apical feature
  3. estimate run-time (for many runs of learner)
  4. adjust γ to be smallest that allows computation to be completed within specified time



- tradeoff: more CPU time allows greater number of smaller subsets to be tested for interactions – *context restriction*

Constructing New Features from Interactions

- build decision tree with interacting features
- extract Boolean expressions for paths from root down to same-class leaf nodes
- combine into DNF expressions
- refine by greedily adding or dropping terms to maximize information gain



Results

- Experimental setup
  - run on *Voting*, *Census*, and *Heart* databases
  - learning algorithm: C4.5
  - divide data randomly into 1/3 for detected interactions, 1/3 for constructing new features, 1/3 for testing
  - generate sampleing schedule estimated to take $\sim$ 15 min on a typical workstation
  - evaluate significance of increased accuracy due to new features by paired T-test on testing set

Voting database

<table>
<tr><th>acc w/o</th><th>acc with</th><th>diff</th><th>Z</th></tr>
<tr><td colspan="4">(EL-SALVADOR=?) or (EL-SALVADOR=N) or ((EL-SALVADOR=Y) and (SYNFUELS=Y))</td></tr>
<tr><td>81.7%</td><td>87.1%</td><td>5.4%</td><td>4.91</td></tr>
<tr><td colspan="4">(MX-MISSLE=Y) or (MX-MISSLE=?) or ((MX-MISSLE=N) and (SYNFUELS=Y))</td></tr>
<tr><td>80.8%</td><td>84.6%</td><td>3.8%</td><td>3.29</td></tr>
<tr><td colspan="4">(EL-SALVADOR=?) or (EL-SALVADOR=N) or ((EL-SALVADOR=Y) and (BUDGET=Y or ?))</td></tr>
<tr><td>80.2%</td><td>86.2%</td><td>6.0%</td><td>3.26</td></tr>
</table>

Heart Disease database

acc w/o	acc with	diff	Z
(NUM-COLOR-VESSELS=0) or ((NUM-COLOR-VESSELS=1) and (SEX=Fem))			
70.4%	78.8%	8.4%	4.25
((EX-ANGINA=T) and (CHEST-PAIN=None or Asymp)) or ((EX-ANGINA=F) and (COLORED-VESSELS$\leq$1))			
69.7%	74.8%	5.1%	2.13

Census database

acc w/o	acc with	diff	Z
(RELATIONSHIP$\neq$Husband) or ((RELATIONSHIP=Husband) and (YEARS-EDUCATION$\leq$9))			
75.2%	78.4%	3.2%	2.23
(MARITAL-STAT$\neq$MarriedToCivilian) or ((MARITAL-STAT=MarriedToCivilian) and (YEARS-EDUCATION$\leq$9))			
76.6%	80.3%	3.7%	2.50
(CAPITAL-GAIN$\leq$5013) and (CAPITAL-LOSS$\leq$1741)			
76.9%	80.4%	3.5%	5.4

Discussion

- could use Bonferroni adjustment to correct for extensive re-use of training data to evaluate subsets

- another learning algorithm besides C4.5 be used to test for feature interactions – would interactions be the same?

- can a more efficient search be designed?

- can features be constructed differently?

- can domain knowledge be used to bias the search based on beliefs about feature relationships?

Conclusions

1. Testing for feature interactions can be done empirically by running learning algorithm on feature subsets

2. Use Bayesian ensemble estimate to calculate expected accuracy of subset from best individual feature and classifier built from rest

3. Can set up a "stratified" sampling strategy to adjust to amount of CPU time given

4. Uses combinatorial trick to evaluate smaller number of larger size subsets for each apical interaction