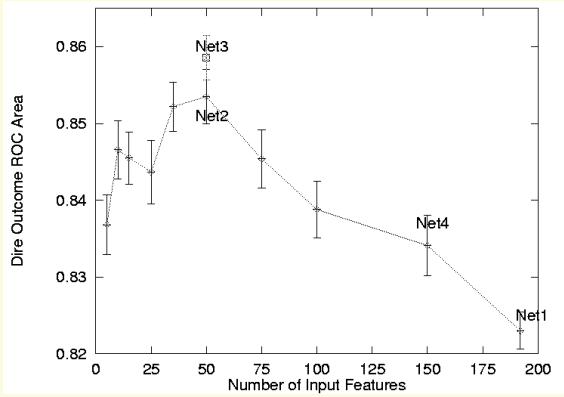
Special Topic:

Feature Selection

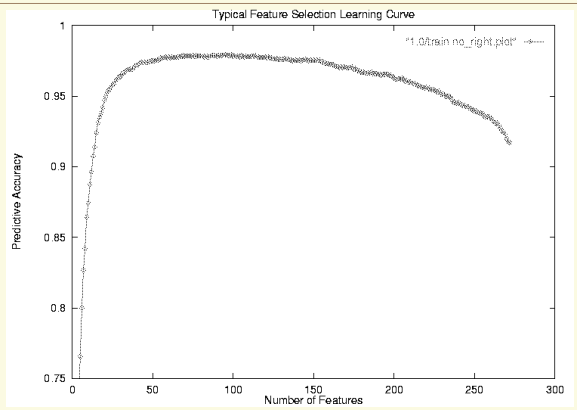
Anti-Motivation

- Most learning methods implicitly do feature selection:
  - decision trees: use info gain or gain ratio to decide what attributes to use as tests. many features don't get used.
  - neural nets: backprop learns strong connections to some inputs, and near-zero connections to other inputs.
  - kNN, MBL: weights in Weighted Euclidean Distance determine how important each feature is. weights near zero mean feature is not used.
  - SVMs: maximum margin hyperplane may focus on important features, ignore irrelevant features.
- So why do we need feature selection?

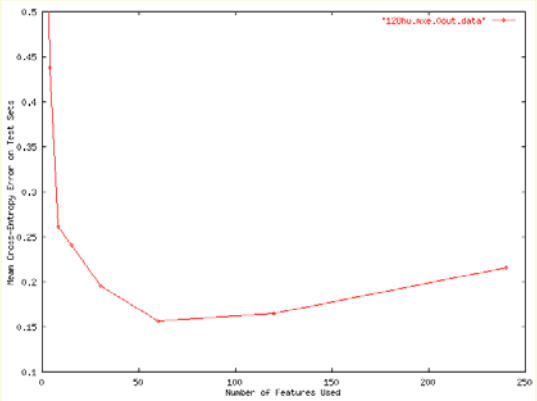
Motivation



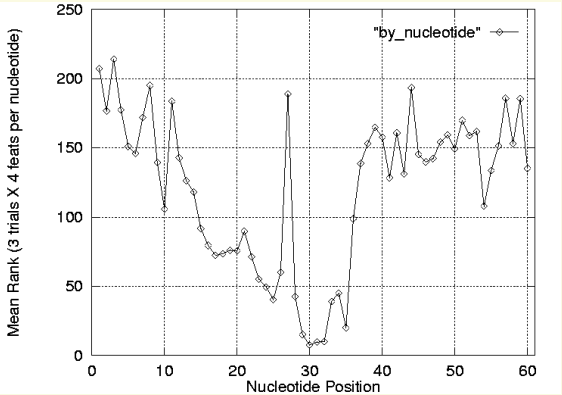
Motivation



Motivation



Motivation



Motivation: Ornithology

23,948 observations
- 12,849 presence
- 11,099 absence

What is important in this domain?

yearseason	nearby_feeders	snow_cov_atleast	med_age_m	gcrh2311	gcsnow2910
dayselapsed	squirrels	snow_dep_atleast	med_age_f	gcrh2312	gcsnow2911
halfdays	cats	snow_crusty	households_sqmi	gcrh2313	gcsnow2912
latitude	dogs	precip_len_atleast	ave_hh_sz	gcskyc5001	gcsnow34
longitude	humans	effort_hrs_atleast	hsehld_1_m_hh	gcskyc5002	gcsnow3513
yard_type_garden	count_area_size	nlcd_water	hsehld_1_f_hh	gcskyc5003	gcsun5301
yard_type_landsca	housing_density	nlcd_developed	marhh_chd_hh	gcskyc5004	gcsun5302
yard_type_woods	high_feeders	nlcd_barren	marhh_no_c_hh	gcskyc5011	gcsun5303
yard_type_desert	fed_yr_round	nlcd_forested	mhh_child_hh	gcskyc5012	gcsun5304
yard_type_pavement	fed_in_jan	nlcd_shrubland	fhh_child_hh	gcskyc5013	gcsun5311
hab_dcid_woods	fed_in_feb	nlcd_orchards	families	gcwnd60b01	gcsun5312
hab_evgr_woods	fed_in_mar	nlcd_grasslands	ave_fam_sz	gcwnd60b02	gcsun5313
hab_mixed_woods	fed_in_apr	nlcd_cultivated	vacant_hu	gcwnd60b03	gctemp0301
hab_orchard	fed_in_may	nlcd_wetlands	owner_occ_hu	gcwnd60b04	gctemp0302
hab_park	fed_in_jun	elev_gt30	renter_occ_hu	gcwnd60b11	gctemp0303
hab_water_fresh	fed_in_jul	pop00_sqmi_mv	houden	gcwnd60b12	gctemp0304
hab_water_salt	fed_in_aug	white_pop00	elev_ned	gcwnd60b13	gctemp0311
hab_residential	fed_in_sep	black_pop00	gcfog4401	gcslvp6101	gctemp0312
hab_industrial	fed_in_oct	ameri_es_pop00	gcfog4402	gcslvp6102	gctemp0313
hab_agricultural	fed_in_nov	asian_pop00	gcfog4403	gcslvp6103	gctmp07a13
hab_desert_scrub	fed_in_dec	hawn_pi_pop00	gcfog4404	gcslvp6104	gctmp07b13
hab_young_woods	numfeeders_suet	other_pop00	gcfog4411	gcslvp6113	gcvis68a01
hab_swamp	numfeeders_ground	mult_race_pop00	gcfog4412	gcslvp6112	gcvis68a02
hab_marsh	numfeeders_hanging	hispanic_pop00	gcfog4413	gcslvp6111	gcvis68a03
hab_other	numfeeders_platfrm	males_pop00	gcprec0101	gcsnow1401	gcvis68a04
evgr_trees_atleast	numfeeders_humming	females_pop00	gcprec0102	gcsnow1402	gcvis68a11
evgr_shrbs_atleast	numfeeders_water	age_under5_pop00	gcprec0103	gcsnow1403	gcvis68a12
dcid_trees_atleast	numfeeders_thistle	age_5_17_pop00	gcprec0104	gcsnow1404	gcvis68a13
dcid_shrbs_atleast	numfeeders_fruit	age_18_21_pop00	gcprec0111	gcsnow1411	evgr_any_atleast
fru_trees_atleast	day1_am	age_22_29_pop00	gcprec0112	gcsnow1412	dcid_any_atleast
cacti_atleast	day1_pm	age_30_39_pop00	gcprec0113	gcsnow1413	

After feature selection …

```
Yearseason,
Dayselapsed,
Longitude,
hab_young_woods,
hab_swamp,
nearby_feeders,
Cats,
Dogs,
fed_in_feb,
Numfeeders,
Hanging,
day1_am,
snow_dep_atleast,
effort_hrs_atleast,
nlcd_barren,
nlcd_shrubland,
nlcd_grasslands,
pop00_sqmi_mv,
asian_pop00,
Gcprec0101,
Gcrh2313,
Gcskyc5002,
Gcwnd60b01,
Gcslvp6101,
Gcsnow1411,
Gcsnow3513,
gcsun5312
```

Brute-Force Approach

- Try all possible combinations of features
- Given N features, 2^N subsets of features
  - usually too many to try
  - danger of overfitting
- Train on train set, evaluate on test set (or use cross-validation)
- Use set of features that performs best on test set(s)

Two Basic Approaches

- Wrapper Methods:
  - give different sets of features to the learning algorithm and see which works better
  - *algorithm dependent*
- Proxy Methods (relevance determination methods)
  - determine what features are important or not important for the prediction problem without knowing/using what learning algorithm will be employed
  - *algorithm independent*

Wrapper Methods

- Wrapper methods find features that work best with some particular learning algorithm:
  - best features for kNN and neural nets may not be best features for decision trees
  - can eliminate features learning algorithm "*has trouble with*"
- Forward stepwise selection
- Backwards elimination
- Bi-directional stepwise selection and elimination

Relevance Determination Methods

- Rank features by information gain
  - Info Gain = reduction in entropy due to attribute

$$Entropy = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- Try first 10, 20, 30, …, N features with learner
- Evaluate on test set (or use cross validation)
- May be only practical method if thousands of attributes

Ornithology Example

		0.9948	houfin
0.6259	+longitude	0.1083	latitude
0.6863	+dayselapsed	0.0896	gctemp0302
0.7369	+yearseason	0.0846	gctemp0313
0.7506	+numfeeders_hanging	0.0841	gcsnow34
0.7594	+asian_pop00	0.0840	gctemp0312
0.7619	+nearby_feeders	0.0822	gcvis68a01
0.7642	+hab_young_woods	0.0788	gcsnow1401
0.7656	+pop00_sqmi_mv	0.0736	gctmp07a13
0.7682	+effort_hrs_atleast	0.0714	gctmp07b13
0.7691	+gcsnow3513	0.0701	gcfog4403
0.7699	+gcsnow1411	0.0679	gctemp0301
0.7703	+gcsun5312	0.0678	gcfog4401
0.7716	+hab_swamp	0.0672	gcrh2303
0.7722	+nlcd_barren	0.0661	gcsnow1413

Advantages of Feature Selection

- Improved accuracy!
- Less complex models:
  - run faster
  - easier to understand, verify, explain
- Feature selection points you to most important features
- Don't need to collect/process features not used in models

Limitations of Feature Selection

- Given many features, feature selection can overfit
  - consider 10 relevant features, and 10^9 random irrelevant features
- Wrapper methods require running base learning algorithm many times, which can be expensive!
- Just because feature selection doesn't select a feature, doesn't mean that feature isn't a strong predictor
  - redundant features
- May throw away features domain experts want in model
- Most feature selection methods are greedy and won't find optimal feature set

Current Research in Feature Selection

- Speeding-up feature selection (1000's of features)
- Preventing overfitting (1000's of features)
- Better proxy methods
  - would be nice to know what the good/relevant features are independent of the learning algorithm
- Irrelevance detection:
  - truly irrelevant attributes can be ignored
  - better algorithms
  - better definition(s)

Bottom Line

- Feature selection almost always improves accuracy on real problems
- Plus:
  - simpler, more intelligible models
  - features selected can tell you about problem
  - less features to collect when using model in future

Feature selection usually is a win.

Special Topic:

Missing Values

Missing Values Common in Real Data

- Pneumonia:
  - 6.3% of attribute values are missing
  - one attribute is missing in 61% of cases
- C-Section:
  - only about 1/2% of attribute values are missing
  - but 27.9% of cases have at least 1 missing value
- Ornithology:
  - 8% of attribute values are missing
  - 94.5% of cases have at least 1 missing value
- UCI machine learning repository:
  - 31 of 68 data sets reported to have missing values

"Missing" Can Mean Many Things

- MAR: "Missing at Random":
  - usually best case
  - usually not true
- Non-randomly missing
- Presumed normal, so not measured
- Causally missing:
  - attribute value is missing because of other attribute values (or because of the outcome value!)

Dealing With Missing Data

- Some learning methods can handle missing values
- Throw away cases with missing values
  - in some data sets, most cases get thrown away
  - if not missing at random, throwing away cases can bias sample towards certain kinds of cases
- Treat "missing" as a new attribute value
  - what value should we use to code for missing with continuous or ordinal attributes?
  - if missing causally related to what is being predicted?
- Impute (fill-in) missing values
  - once filled in, data set is easy to use
  - if missing values poorly predicted, may hurt performance of subsequent uses of data set

Imputing Missing Values

- Fill-in with mean, median, or most common value
- Predict missing values using machine learning
- Expectation Maximization (EM):
  - Build model of data values (ignore missing vals)
  - Use model to estimate missing values
  - Build new model of data values (including estimated values from previous step)
  - Use new model to re-estimate missing values
  - Re-estimate model
  - Repeat until convergence

Potential Problems

- Imputed values may be inappropriate:
  - in medical databases, if missing values not imputed separately for male and female patients, may end up with male patients with 1.3 prior pregnancies, and female patients with low sperm counts
  - many of these situations will not be so humorous/obvious!
- If some attributes are difficult to predict, filled-in values may be random (or worse)
- Some of the best performing machine learning methods are impractical to use for filling in missing values (neural nets)
- Beware of coding - reliably detect missing cases can be difficult

Research in Handling Missing Values

- Lazy learning:
  - don't train a model until you know test case
  - missing in test case may "shadow" missing values in train set
- Better algorithms:
  - Expectation maximization (EM)
  - Non-parametric methods (since parametric methods often work poorly when assumptions are violated)
- Faster Algorithms:
  - apply to very large datasets