

Evolving Feature Selection

Huan Liu, *Arizona State University*

Data preprocessing is an indispensable step in effective data analysis. It prepares data for data mining and machine learning, which aim to turn data into business intelligence or knowledge. Feature selection is a preprocessing technique commonly used on high-dimensional data. Feature selection studies how to select a subset or list of attributes or variables that are used to construct models describing data. Its purposes include reducing dimensionality, removing irrelevant and redundant features, reducing the amount of data needed for learning, improving algorithms' predictive accuracy, and increasing the constructed models' comprehensibility.

Feature selection is different from feature extraction (for example, principal component analysis, singular-value decomposition, manifold learning, and factor analysis), which creates new (ex-tracted) features that are combinations of the original features. Feature-selection methods are particularly welcome in interdisciplinary collaborations because the selected features retain the original meanings domain experts are familiar with. The rapid developments in computer science and engineering allow for data collection at an unprecedented speed and present new challenges to feature selection.

Wide data sets, which have a huge number of features but relatively few instances, introduce a novel challenge to feature selection. This type of data supports a vast number of models. Examples of such data are microarrays, transaction logs, and Web data. Unlabeled data presents another challenge; the lack of class labels compounds the already difficult problem of feature selection. The integration of domain knowledge in feature selection tends to be a perpendicular challenge. Feature-selection methods attempt to explore data's intrinsic properties by employing statistics or information theory. As in microarray data analysis, however, statistical significance might not directly translate to biological relevance. It's imperative to integrate both in selecting features to generate results meaningful to domain experts.

These six articles address emerging issues that concern evolving feature selection. Edward Dougherty addresses the daunting small-sample problem, while Jennifer Dy introduces ways to select features for unlabeled data. Kari Torkkola and Eugene Tuv advocate achieving stability of feature selection with ensemble methods. Hanchuan Peng, Chris Ding, and Fuhui Long apply dual criteria (minimum redundancy and maximum relevance) to selecting features for microarray data. Michael Berens and his colleagues show how to foster biological relevance in feature selection. Finally, George Forman reviews feature selection's status and points out the need for further research.

—Huan Liu

Feature-Selection Overfitting with Small-Sample Classifier Design

Edward R. Dougherty, *Texas A&M University*

High-throughput technologies facilitate the measurement of vast numbers of biological variables, thereby providing enormous amounts of multivariate data with which to model biological processes.¹ In translational genomics, phenotype classification via gene expression promises highly discriminatory molecular-based diagnosis, and regulatory-network modeling offers the potential to develop therapeutic strategies based on genomic decision making using classical engineering disciplines such as control theory.² Yet one must recognize the obstacles inherent in dealing with extremely large numbers of interacting variables in a nonlinear, stochastic, and redundant system that reacts aggressively to any attempt to probe it—a living system. In particular, large data sets may have the perverse effect of limiting the amount of scientific information that can be extracted, because the ability to build models with scientific validity is negatively impacted by an increasing ratio between the number of variables and the sample size. Our specific interest is in how this dimensionality problem creates the need for feature selection while making feature-selection algorithms less reliable with small samples.

Two well-appreciated issues tend to confound feature selection: *redundancy* and *multivariate prediction*. Both of these can be illustrated by taking a naïve approach to feature selection by considering all features in isolation, ranking them on the basis of their individual predictive capabilities, selecting some features with the highest individual performances, and then applying a standard classification rule to these features, the reasoning being that these are the best predictors of the class. Redundancy arises because the top-performing features might be strongly related—say, by the fact that they share a similar regulatory pathway—and using more than one or two of them may provide little added benefit. The issue of multivariate prediction arises because top-performing single features may not be significantly more beneficial when used in combination with other features, whereas features that perform poorly when used alone may provide outstanding classifi-

cation when used in combination. This situation can be dramatic in highly complex regulatory systems. Another impediment to feature selection concerns estimation. With small samples, the choice of error estimator can make a greater difference than the manner of feature selection.³

Overfitting

While redundancy, multivariate prediction, and error estimation can severely impact feature selection, my commentary here is aimed at the role of feature selection in *overfitting* the data and how this is exacerbated by high dimensionality and small samples.

A classification rule chooses a classifier from a family G of classifiers on the basis of the data. A classifier is optimal in G if its error, ϵ_G , is minimal among all classifiers in G . Since a designed classifier depends on the particular sample, it is random relative to random sampling. We would like the expected error, $\epsilon_{G,n}$, of the designed classifier, n denoting sample size, to be close to ϵ_G .

If G and H are families of classifiers such that $G \subset H$, then $\epsilon_H \leq \epsilon_G$. However, the error relation need not hold for designed classifiers, where it may be that $\epsilon_{H,n} > \epsilon_{G,n}$. This is known as overfitting: the designed classifier partitions the feature space well relative to the sample data but not relative to the full distribution. Overfitting is ubiquitous for small samples. To mitigate overfitting, one can choose from smaller classifier families whose classifiers partition the feature space more coarsely. Using G instead of H , where $G \subset H$, reduces the *design cost*, $\epsilon_{G,n} - \epsilon_G$, relative to $\epsilon_{H,n} - \epsilon_H$ at the expense of introducing a *constraint cost*, $\epsilon_G - \epsilon_H$.

Consider a collection of classifier families, $G_1 \subset G_2 \subset G_3 \subset \dots$, for a fixed sample size n . A typical situation might be that, while the smaller families extensively reduce design cost, their constraint is excessive. Thus, we might expect the expected errors of the designed classifiers to fall as we utilize increasingly large families but then to begin to increase when the design cost grows too much. Applying this reasoning to a sequence, $x_1, x_2, \dots, x_d, \dots$, of features, we might expect at first a decrease in expected error as d increases and then subsequently an increase in error for increasing d . While this description is idealized and the situation can be more complex, it describes the *peaking phenomenon*. In this scenario, one would be interested in the optimal number of features.⁴

In practice, the features are not ordered, and the best feature set must be found from among all possible feature subsets. We confront a fundamental limiting principle: In the absence of countervailing distribution knowledge, to select a subset of k features from a set of features and be assured that it provides minimum error among all optimal classifiers for subsets of size k , all k -element subsets must be checked.⁵ Thus, we are challenged to find suboptimal feature-selection algorithms.

When used, a feature-selection algorithm is part of the classification rule. This is why feature selection must be included when using cross-validation error estimation. Feature selection yields classifier constraint, not a reduction in the dimensionality of the feature space relative to design. For instance, if there are d features available for linear dis-

When used, a feature-selection algorithm is part of the classification rule. This is why feature selection must be included when using cross-validation error estimation.

criminant analysis (LDA), when used directly, then the classifier family consists of all hyperplanes in d -dimensional space. But, if a feature-selection algorithm reduces the number of variables to $m < d$ prior to application of LDA, then the classifier family consists of all hyperplanes in d -dimensional space confined to m -dimensional subspaces. The dimensionality of the classification rule has not been reduced, but the new classification rule (feature selection plus LDA) is constrained. The issue is whether it is sufficiently constrained. Given 20,000 gene-expression levels as features, the new rule has significant potential for overfitting.

An illustration

For illustration, consider a d -dimensional model where the class conditional densities for 0 and 1 are uniformly distributed over the regions

$$D_0 = [0, a_1/2] \times [0, a_2] \times [0, a_3] \times \dots \times [0, a_d]$$

$$D_1 = [a_1/2, 1] \times [0, a_2] \times [0, a_3] \times \dots \times [0, a_d]$$

respectively, $a_1, a_2, \dots, a_d > 0$. This model is useful to illuminate the difficulty of small-sample feature selection for several reasons. First, for the first feature there is a perfect classifier (0 error) consisting of the single-point decision boundary, $x_1 = a_1/2$; for every other feature x_k , every classifier consisting of a finite number of splits of the interval $[0, a_k]$ has error 0.5; and so long as x_1 is not included, every feature set composed of any number of variables has error 0.5. Thus, in some sense, this corresponds to the easiest possible feature-selection problem. Second, it is not complicated by redundancy because all features are independent. Third, it is not complicated by multivariate prediction because optimal feature selection involves a single feature, x_1 . Finally, the problem is not necessarily mitigated by the common practice of commencing feature selection by throwing out those with low variance. As seen by the error formulas below, the variances of the features play no role, and one might eliminate the good feature if a_1 is small in comparison to a_2, a_3, \dots, a_d .

I now demonstrate that high dimensionality and low sample size can make finding the 0-error feature difficult even though all other features have error 0.5. We consider three single-feature classification rules of increasing complexity: a single split, up to two splits, and up to three splits of the interval. The probabilities of a random sample of size $2n$, equally split between the two classes, being perfectly separated by a single value, at most two values, and at most three values of x_k are

$$p_{1,n} = \frac{2(n!)^2}{(2n)!}$$

$$p_{2,n} = \frac{2n(n!)^2}{(2n)!}$$

$$p_{3,n} = \frac{2(n^2 - n + 1)(n!)^2}{(2n)!}$$

respectively. Since the feature distribution is uniform, all features are independent. Therefore, the separability or lack of separability of the data by features x_2, x_3, \dots, x_d constitutes a binomial distribution with $d - 1$

Table 1. Expected number of separating feature sets using one split.

$2^n(d-1)$	1,000	5,000	20,000
10	8	40	159
12	2	11	43
14	1	3	12
16		1	3
18			1
20			

Table 2. Expected number of separating feature sets using at most two splits.

$2^n(d-1)$	1,000	5,000	20,000
10	40	198	793
12	13	65	260
14	4	20	81
16	1	12	50
18		2	7
20		1	2

Table 3. Expected number of separating feature sets using at most three splits.

$2^n(d-1)$	1,000	5,000	20,000
10	167	833	3,333
12	67	336	1,342
14	25	125	501
16	9	44	177
18	3	15	60
20	1	5	20

trials. Thus, the expected number, $N_{k,d,n}$, of separating features is $E[N_{k,d,n}] = (d-1)p_{k,n}$ for $k = 1, 2$, and 3 splits.

The danger of obtaining a poor feature set with high-dimensional data sets and small samples is seen in tables 1 through 3, which give $E[N_{k,d,n}]$ for large values of $d-1$ and small sample sizes. And this is for a situation in which every poorly selected feature has error 0.5! More tables giving $E[N_{k,d,n}]$ are provided at www.ee.tamu.edu/~edward/feature_overfitting.

In conclusion, note that in high-dimension, small-sample settings, a key difficulty is the masking of good feature sets by bad ones. The result can be false-negative reasoning where one wrongly concludes that no good feature sets exist simply because they cannot be found. Owing to its importance for high-throughput technologies, feature

selection is receiving much attention with many schemes being proposed. It seems incumbent on those proposing algorithms that limitations be investigated at the outset to see under what conditions one can reasonably expect satisfactory results.

References

1. J. Chen et al., "Grand Challenges for Multimodal Bio-Medical Systems," *IEEE Circuits and Systems*, vol. 5, no. 2, 2005, pp. 46–52.
2. E.R. Dougherty and A. Datta, "Genomic Signal Processing: Diagnosis and Therapy," *IEEE Signal Processing*, vol. 22, no. 1, 2005, pp. 107–112.
3. C. Sima et al., "Impact of Error Estimation on Feature-Selection Algorithms," *Pattern Recognition*, vol. 38, no. 12, 2005, pp. 2472–2482.
4. J. Hua et al., "Optimal Number of Features as a Function of Sample Size for Various Classification Rules," *Bioinformatics*, vol. 21, no. 8, 2005, pp. 1509–1515.
5. T. Cover and J. Van Campenhout, "On the Possible Orderings in the Measurement Selection Problem," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 7, no. 9, 1977, pp. 657–661.

Feature Selection for Unlabeled Data

Jennifer G. Dy, *Northeastern University*

Technological advances such as the Internet, hyperspectral imagery, microarrays, and digital-storage-capacity increases have contributed to the existence of large volumes of data. One way to extract information and knowledge from data is through clustering or unsupervised learning.

Creating effective algorithms for unsupervised learning, or learning from unlabeled data, is important because large amounts of data preclude humans from manually labeling the categories of each instance. In addition, human labeling is expensive and subjective. So, much existing data is unlabeled.

Unsupervised learning, or cluster analysis, aims to group similar objects. A metric or a probability model typically defines *similarity*. These measures depend highly on the features representing the data. Many clustering algorithms assume that domain experts have determined relevant features. But not all features are important; some

might be redundant or irrelevant. And the presence of many irrelevant features can even misguide clustering results. Moreover, reducing the number of features increases comprehensibility and ameliorates the problem of some unsupervised-learning algorithms failing with high-dimensional data.

Let's say we apply k -means with Euclidean distance as a measure for dissimilarity to cluster the data. For a finite amount of data, high dimensions lead to a sparse data space, and most of the data points will look equally far. For probability-based clustering algorithms, high dimensions mean more parameters to predict (that is, we need more data points to obtain accurate estimates). These clustering methods wouldn't work well in high dimensions.

To deal with high dimensionality, we can perform either feature transformation or feature selection. Principal component analysis, factor analysis, projection pursuit, and independent component analysis are examples of transformation methods, which involve transformations of the original variable space. In this article, I talk about selecting subsets of the original space. Subset selection is desirable in some domains that prefer the original variables so as to maintain these features' physical interpretation. In addition, feature transformation algorithms require computation or collection of all the features before dimension reduction can be achieved. In contrast, feature selection algorithms require computation or collection of only the selected feature subsets after the feature subsets are determined.

Carla Brodley and I define the goal of feature selection for unsupervised learning as finding the smallest feature subset that best uncovers "interesting natural" groupings (clusters) from data according to the chosen criterion.¹ Unlike supervised learning, which has class labels to guide the feature search, in unsupervised learning, we must define what interesting and natural mean in the form of criterion functions. The problem is that no global consensus exists on how to define interestingness. Moreover, different feature subspaces reveal different cluster structures. Which subspace should we pick?

Feature-selection approaches

Following supervised-learning terminology, you can categorize feature subset selection methods for unlabeled data as *filter* or *wrapper* approaches.