Dimensionality Reduction and Statistical Model Building for Information Retrieval

Miles Efron
efrom@ils.unc.edu
School of Information and Library Science
UNC, Chapel Hill, NC 27599-3360

24th January 2002

Abstract

Latent Semantic Indexing (LSI) attempts to improve information retrieval by projecting documents and queries into an orthogonal space of relatively low dimensionality. Proponents of LSI argue that such dimensionality reduction improves IR by identifying semantic patterns among data while eliding random, noisy variation. Crucial to the identification of a useful semantic subspace is the choice of system dimensionality. This study addresses the matter of dimensionality selection in the vocabulary of statistical model building. We argue that selecting the dimensionality of an LSI system is analagous to selecting predictors in a generalized linear model. In particular we suggest that selecting a relatively low-dimensional model is justified in efforts to avoid overfitting the training data when estimating inter-document relationships.

1 Introduction

Information retrieval (IR) systems attempt to discover documents in a database that are relevant to a user's stated information need. Insofar as this process involves predicting the value of a variable (relevance) given a set of input variables (terms in the queries and documents), IR may be understood as a statistical modeling problem. Here the goal of the IR system is to estimate the functional relation between query terms and relevance, such that an appropriate classification rule may be applied to the data. If we make the simplifying assumption that relevance is a binary variable, IR reduces to a two-class classification problem, where the decision rule comprises a hyperplane that divides the document space into two regions. Documents falling on one side of the hyperplane are classified as relevant, with the remainder classified as non-relevant.

Understanding retrieval in these terms is especially appealing in the context of latent semantic indexing (LSI). LSI systems employ the singular value decomposition (SVD) to reduce the dimensionality of the system. Assuming that evidence in IR tends to be noisy, proponents of LSI argue that reducing an n-dimensional model to k dimensions, where $k \ll n$ and the k remaining dimensions are fit by the method of least squares, leads to more semantically accurate predictions of relevance. While the improvement offered by dimensionality reduction is fairly clear [2, 6] the matter of choosing an appropriate value for k is not. Framing retrieval in terms of statistical modeling can shed some light on this problem.

We demonstrate an analogy between retrieval and logistic regression. By taking this perspective, selecting an appropriate k-value naturally becomes a problem of statistical model building. When fitting statistical models to data, a variety of metrics are available for assessing model fit given a certain set of predictor variables. The process of fitting a statistical model demands that we negotiate underfitting and overfitting. An underfitted model lacks adequate expressiveness to represent important variation in the training data. On the other hand, a model suffers from overfitting when too many predictor variables find their way into the function. This leads the model to be swayed by variation in the data that is in fact random, pertaining to no "interesting" patterns of variance. We argue that finding an appropriate value for k in an LSI system is closely analogous to this problem.

2 Dimensionality in the LSI Model

Proponents of LSI argue that IR systems operate on noisy data, and that we may improve retrieval by applying noise reduction techniques to documentary evidence. For instance a document containing the word *car* may be relevant to a query about automobiles. But documents containing *truck, auto,* and *vehicle* may also be relevant to such a query. Moreover, a document containing *car* may not be relevant to the *automobile* query (after all, *dining cars, cable cars,* and *side cars* aren't automobiles). The goal of LSI is thus to relax the literalness inherent in traditional keyword retrieval.

Under the standard vector space model, documents are are ranked against a query x according to Equation 1.

$$r = \mathbf{A}x \tag{1}$$

where $\mathbf{A}$ is the m by n term-document matrix for a collection containing m documents in n terms. Thus each document is represented as a vector in $\Re^n$ and inter-document similarity is defined as the inner product between document vectors in this space. To overcome the literalness inherent in this model, LSI reduces the dimensionality of the problem. We begin by using the singular value decomposition to factor the term-document matrix $\mathbf{A}$ into the product of three special matrices (Equation 2).

$$\mathbf{A} = \mathbf{T}\Sigma\mathbf{D}^T \tag{2}$$

where Σ is a diagonal matrix containing the singular values of $\mathbf{A}$. The singular values are the positive square roots of the eigenvalues of the matrix $\mathbf{A}\mathbf{A}^T$. The values of Σ are listed in descending, order. Thus $\sigma_{11} \geq \sigma_{22} \geq \sigma_{nn}$. Matrices $\mathbf{T}$ and $\mathbf{D}$ are comprised of the left and right singular vectors of $\mathbf{A}$. They are orthogonal matrices. $\mathbf{T}$ represents each row (term) in $\mathbf{A}$, while $\mathbf{D}$ represents each column (document).

LSI's dimensionality reduction is accomplished by truncating the matrices obtained through SVD to their first k rows (columns) and then recombining them. We may thus approximate matrix $\mathbf{A}$ with $\widehat{\mathbf{A}}$, calculated by Equation 3:

$$\widehat{\mathbf{A}} = \mathbf{T}_k \Sigma_k \mathbf{D}_k^T \tag{3}$$

where matrices $\mathbf{T}_k$, Σ_k, and $\mathbf{D}_k$ contain the first k elements of $\mathbf{T}$, Σ, and $\mathbf{D}$. Because the singular values are listed in decreasing order, truncating Σ to the first k rows and columns has the effect of excluding the $n-k$ smallest factors from the representation. It can be shown that $\widehat{\mathbf{A}}$ is the best rank-k approximation of $\mathbf{A}$ in the least-squares sense.

During LSI a document ranking r against a query x is obtained first by projecting x into the termspace defined by $\mathbf{T}_k$ via Equation 4. This derives $\mathbf{q}$, a k-dimensional vector in factor space:

$$\mathbf{q} = \mathbf{T}_k \Sigma_k^{-1} x \tag{4}$$

. Having obtained our projection we rank documents according to their inner product with $\mathbf{q}$ (Equation 5):

$$\mathbf{r} = \mathbf{D}_k \Sigma_k^{-1} \mathbf{q} \tag{5}$$

. The rationale behind LSI has it that this k-dimensional representation will be superior to the untransformed ranking obtained by Equation 1. The benefit is often expressed in terms of noise reduction. That is, the original n-dimensional representation contains random variance which will be elided in the model of reduced dimension.

Thus the matter of choosing an appropriate value for k is crucial [2, 6]. However, this choice retains an air of mystery. According to Manning and Schutze [7] typical choices for k range from 50-150. Landauer and Dumais found that a particularly large corpus demanded approximately 300 dimensions to achieve good representation [6]. What is clear is that for realistic data sets, some $k \ll n$ tends to yield good retrieval.

Several authors have helped to put the choice of an appropriate k-value on more solid theoretical ground. Ding [3] takes a probabilistic approach. He proposes a model based on the dual probabilities of observing the term-document co-occurrences evidenced in $\mathbf{A}$. The model computes $p(x|c_1...c_k)$, the probability of observing document x given the matrix $\mathbf{A}$, where x is a document and c_i is the *ith* "latent semantic structure" of the corpus represented by $\mathbf{A}$ (an analogous representation of terms is also proposed). Ding shows that the singular vectors of $\mathbf{A}$ provide the maximum likelihood estimators of c, and that singular vectors corresponding to small singular values improve the likelihood of the model very little.

Roger E. Story [11] offers an interpretation of LSI based on Bayesian Regression. According to Story, traditional keyword retrieval can be interpreted as a linear regression problem, as in Equation 6. For a given query, we obtain:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon \tag{6}$$

where the response variable $\mathbf{Y}$ is the vector of similarity scores derived from Equation 1. Here $\mathbf{X}$ is the term-document matrix $\mathbf{A}$ and β is the vector of coefficients specific to the query, and ε is a vector of noise. Linear regression derives the vector β by the method of least squares (Equation 7).

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{\dagger}(\mathbf{X}^T\mathbf{Y}) \tag{7}$$

where $(\mathbf{X}^T\mathbf{X})^{\dagger}$ is the Moore-Penrose pseudoinverse of the matrix $(\mathbf{X}^T\mathbf{X})$. Story argues that LSI augments this model by incorporating prior information into the calculation of $\widehat{\beta}$. Specifically, we assume that the values of β will tend to be small (since the influence of any single term on a document's relevance tends to be small, with a few important exceptions). However, Story shows that small singular values lead to large elements of $\widehat{\beta}$. Thus eliminating those singular values removes evidence that contradicts our prior beliefs about the model. According to Story, limiting the dimensionality to the first k factors has the effect of excluding statistically dubious information from the regression.

The approaches of Ding and Story both describe LSI in the vocabulary of statistical model building. As outlined below, our study takes a similar, but new strategy. We argue that retrieval can be understood as a logistic regression, and that such a perspective allows us to understand the matter of dimensionality reduction in terms of statistical model building, where the fundamental problem involves balancing trade-offs between descriptiveness of the model with respect to the training data, and the ability of the model to generalize to new data.

3 Methodology

To demonstrate the utility of statistical model building for the study of LSI systems, we analyzed documents and queries from the Cycstic Fibrosis (CF) database [10]. This collection contains all documents entered into the US Government's National Library of Medicine Medlars database between 1974 and 1979 with *cystic fibrosis* as a subject heading. For this study document representations consisted of the titles and abstracts of all 1239 documents. Of these, half were allocated to a training set, leaving the other half as a test set. All results below describe performance averaged across twenty queries (chosen at random from the collection's 100 total queries).

3.1 Logistic Regression

Like linear regression, logistic regression [5, 8] defines a functional relationship between one or more predictor variables (represented by the vector $\mathbf{x}$) and a response variable, $\mathbf{y}$. Under the linear regression model, it is assumed that $y = E(y|x) + \varepsilon$, where elements of ε are normally distributed with mean equal to zero and variance equal to σ^2. In the case of IR, however, our response variable (relevance to a given query) is binary, and thus non-Gaussian. Logistic regression alters the model by applying the binomial distribution to the response variables. In this case we take ε to be binomially distributed with mean zero and variance equal to $E(y|x)[1 - E(y|x)]$.

Assuming that its values follow the logistic distribution, we estimate the conditional probability of seeing the binary vector $\mathbf{y}$ given the input $\mathbf{x}$ via Equation 8.

$$E(y|x) = \pi(x) = \frac{e^{g(x)}}{1 + e^{g(x)}} \tag{8}$$

where $g(x)$ is the logit transformation defined by Equation 9

$$g(x) = \ln\left[\frac{\pi(x)}{1 - \pi(x)}\right] \tag{9}$$

$$= \beta_0 + \beta_1 x_1 + \beta_2 x_x + ... + \beta_p x_p$$

for a p-dimensional model. Fitting the coefficient vector β involves maximizing the log-likelihood of the model. In other words, we choose the vector β that is most likely to be accurate, given the training data. More detailed descriptions of this method may be found in [5, 8].

3.2 Statistical Model Building

Given a matrix of predictor variables, $\mathbf{x}$ and a response vector, $\mathbf{y}$, the goal of statistical modeling is to discover a functional relationship between $\mathbf{x}$ and $\mathbf{y}$ [1, 4, 8, 9]. Statistical model building is concerned specifically with discovering appropriate predictor variables for a certain model [9]. A well-fitted model balances competing imperatives. The model must fit the training data adequately. However, By adding polynomial covariates of sufficient degree, we may fit any data set perfectly. Such a model appears to describe the input data, but it will fail to generalize to new data. That is, due to the *ad hoc* nature of its hypothesized functional relation this model reduces residual error by modeling random variation in the training data. Such a model is said to be over-fitted. Building a good model thus involves a negotiation between goodness of fit and predictive power. In general, we desire a model that achieves a proper balance of description and parsimony. It is our contention that selecting a k-value for LSI is in fact a model building problem. In particular, we argue that the decline in retrieval performance observed when k becomes too large is due to model over-fitting.

To motivate our analysis of the over-fitting problem in LSI, we turn to the Akaike Information Criterion (AIC), a popular metric for assessing goodness of fit in a statistical model. AIC is based on the log-likelihood of a given model, with a bias-correcting measure added to offset an overfitting bias [1]. AIC is described in Equation 10:

$$AIC = -2log(L(\widehat{\theta}|y)) + 2K \tag{10}$$

where $L(\widehat{\theta}|y)$ is the likelihood of the model, and K is the number of estimated parameters. A low AIC score implies a well-fit model. Models with high likelihood (i.e. with low residual error) drive AIC down. However, Akaike showed that simply minimizing the log-likelihood consistently leads to overfitting. Thus AIC penalizes the addition of parameters to the model. It is important to stress that the computed AIC values are of little interest. Instead, what is important is the difference in AIC across a range of models. Typically, one selects among candidate models by choosing the one leading to the lowest AIC.

3.3 Data Simulations

To demonstrate the application of model building techniques to the IR problem, we constructed a set of artificial data consisting of 200 documents represented in 50 terms. We labeled half of these documents relevant and the other half as non-relevant. The terms in the collection were modeled using normally distributed, random data. Of the 50 terms, 20 were considered "real ." We constructed these with different means for relevant and non-relevant documents. The remaining 30 terms were "noise." These terms had identical distributions across relevant and non-relevant documents.

As an initial baseline we modeled standard keyword retrieval by deriving a logistic regression in which the response variable y_i for the *ith* document is relevance, and the predictor variables $\mathbf{X}$ are the term scores for that document. We call this the full model. The full model had $AIC = 100.04$, with classification accuracy 0.88. We then applied an automatic stepwise model building approach. Under this approach, the algorithm begins with a very simple model. In our case, we begin by fitting only an intercept. The step-wise algorithm then iteratively adds one variable at a time to the model. At each iteration, the process determines and adds the term whose inclusion will improve the model AIC by the greatest margin. The process ends when no terms improve the model AIC. We call this the final model.

In the case of our simulation, the final model included 19 terms, very close to the "true" 20-dimensional model that we constructed. The AIC of the final model was 38.95.

We then fitted a second model using our simulated data. While the response variable remained the same for the second model, we changed the predictors. Instead of using the initial term counts, we predict the response, $\mathbf{y}$ based on the matrix $\mathbf{P}$ defined in Equation .

$$\mathbf{P} = \mathbf{D}\Sigma^{-1} \tag{11}$$

where matrices $\mathbf{D}$ and Σ are derived by the SVD of the initial term-document matrix. Thus the predictor variables in this case are the scaled right singular vectors of the input matrix.

Using this input, the full model (i.e. using $k = k_{max}$) had $AIC = 100.04$. Applying the step-wise model building algorithm derived a final model with 15 predictors. This model showed an AIC of 34.21. Since low AIC scores imply a well-fitted model, we appear to have improved our representation by reducing model dimensionality to 15. An important result from these simulations is that models of dimensionality lower than the "true" dimensionality of the data are optimal for the logistic regression. This is due to the fact that we are modeling the functional relationship of a specific query. As such, not all information in the data plays a role in a given classification problem. Thus we do not make the claim that the k-values we derive here are appropriate for general IR applications, where many queries must be fielded by a single system. Instead, our aim is to show that the process of adding singular vectors to an LSI system involves the same problem faced during statistical model building. That is, as we add terms to the model, we increase the expressiveness of the function at the expense of descriptive abstraction.

4 Results

Initial experiments suggest that models of low dimensionality adequately characterize the CF data for the two-class discrimination problem. However, ad hoc retrieval demands a more versatile representation. Thus in addition to our findings for the two-class problem, we present results applicable to wider retrieval tasks.

4.1 Logistic Regression Models for Two-Class Discrimination

To cast a typical information retrieval problem into the language of statistical model building, we fit a logistic regression model as described above. After training the model on half of the CF-database documents, we test it by predicting the relevance for each of the remaining documents in the collection. Figure 1 shows the classification accuracy achieved by logistic regression models of varying dimensionality. The x-axis of Figure 1 shows k, the dimensionality of each model. The y-axis shows the accuracy of each model, averaged across 20 queries. Consistent with other empirical studies, Figure 1 shows system accuracy increasing as we add dimensions to the initial model. Once we have added approximately 30 dimensions to the model, we achieve very good classification accuracy. However, as the model dimensionality increases to $k > 150$ performance begins to decline.

To understand why performance does not continue to increase (or plateau) as the number of dimensions approaches $rank(\mathbf{A})$, we suggest considering the problem of overfitting. That is, we may consider the loss of test accuracy at high k values to result from an overfitted model. As we add predictor variables to the model, we will improve the model's description of the training data. In terms of LSI we may understand this in matrix terms as follows. We approximate

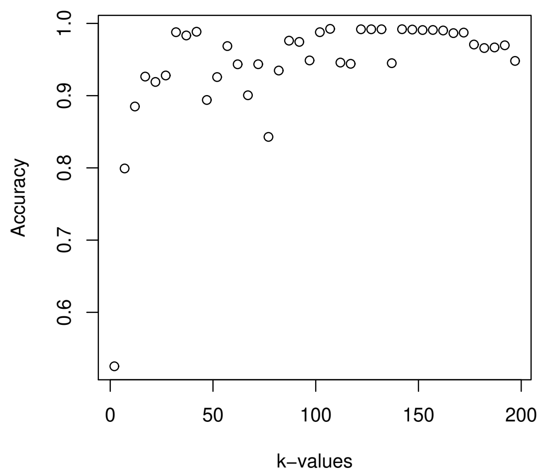

Figure 1: Classification Accuracy versus Model Dimensionality

the input matrix $\mathbf{A}$ as $\widehat{\mathbf{A}} = \mathbf{T_k}\Sigma_k\mathbf{D}'_k$. This always yields the best rank-k approximation of the original matrix, in the least squares sense. But at low values of k our approximation will not be very close to $\mathbf{A}$. By increasing k we improve the approximation. However, by doing so, a model based on $\widehat{\mathbf{A}}$ (i.e. a document classification model) loses generality.

Figure 2 plots AIC versus k, the dimensionality of each model. The linearity of the rightmost points suggests that adding additional predictors after $k \approx 75$ is gratuitous, yielding little benefit and potentially overfitting the model. At $k \approx 75$ the rate at which AIC is increasing steepens. In fact, at this point the plot becomes very linear. From $k \approx 75$, AIC increases at the rate of $2k$. At $k \approx 75$, then, the log-likelihood has been maximized. Adding further variables to the model cannot improve its fit. Perhaps more important in Figure 2, however, is the obvious optimization of the model at $k = 2$. This clear conflict with Figure 1 demonstrates an important result: AIC's standard penalty for adding parameters is inadequate for modeling LSI. This merits further research, but initially suggests that adding a singular vector to the model results in a loss of several degrees of freedom, whereas the AIC method assumes that adding a predictor loses only one degree of freedom.

Our suspicion that the dimensions corresponding to the first ≈ 75 singular values are the most informative is borne out by Figures 3 and 4. Figure 3 plots the reduction in residual deviance achieved by adding the next k factors to the model. The variance of the deviance residuals across queries was quite high, as evidenced by the extreme peak near $k = 60$. However, the data do suggest a strong trend. Adding factors corresponding to low (but not the lowest) singular

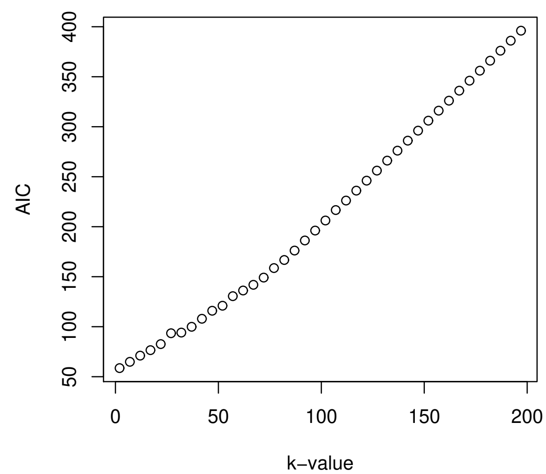

Figure 2: AIC plotted against k, the model dimensionality

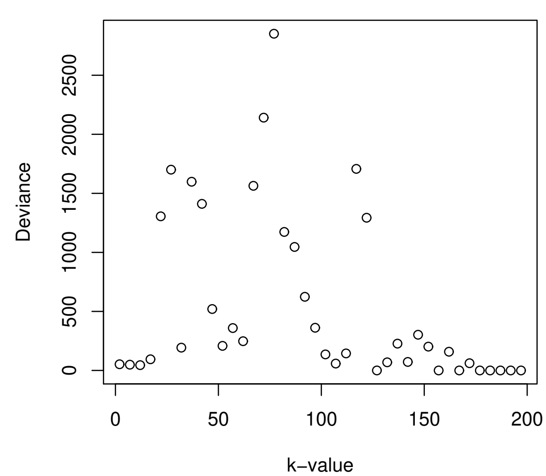

Figure 3: Reduction in Residual Deviance for *kth* Predictor

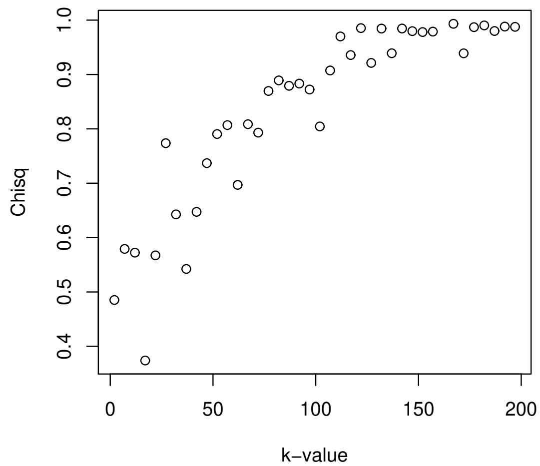

Figure 4: χ^2 Values for *kth* Predictor

values reduced the model deviance most dramatically. That is, including the 50th factor, for instance, reduced the residual deviance by approximately 300 on average. On the other hand, factors much greater than $k = 100$ appear to add little to the model. These dimensions, corresponding to small singular values, reduced residual deviance by trivial amounts, compared to the factors in lower-dimensional models.

Figure 4 provides another glimpse of the information afforded by Figure 3. Figure 4 shows the χ^2 statistic for the model comprised of the first k predictor variables. Deviance residuals are chi-square distributed with $J - p + 1$ degrees of freedom, where J is the number of covariate patterns, and p is the number of parameters in the model. Thus the y-axis of Figure 4 shows the confidence with which we reject the null hypothesis: that a k-dimensional model achieves the observed reduction in residual deviance by chance. Again, low-dimensional factors appear to be crucial. By the time we have added 150 covariates, we appear to be able to reject the null hypothesis (no model effect) with about 95% confidence.

4.2 Linear Models

Because of our surprise at seeing optimal AIC readings at exceedingly low k-values, we augmented our analysis with another round of modeling. Following Roger Story's [11] example, we modeled retrieval as a linear regression, as described above. In this case, the predictor variables were again first k scaled singular vectors. However, for the linear model, the response variable was not a

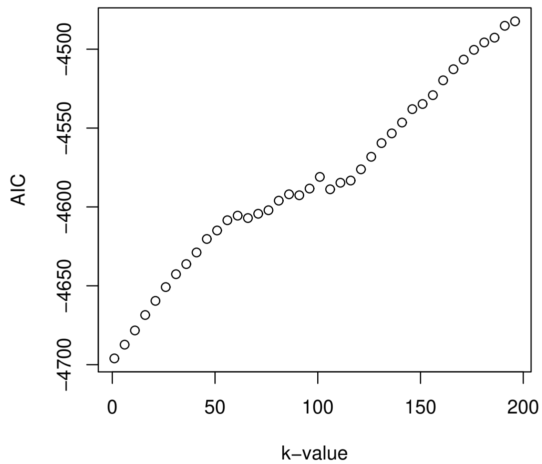

Figure 5: AIC for Linear Regression

binary relevance judgment. Instead, the linear response was the cosine similarity between a given query q_i and each document d_j.

Figure 5 shows the AIC for the linear model as dimensions are added to the analysis. Again, the initial model, with very low k-value appears to be optimal. And again, the graph becomes steeper after $k \approx 75$. Thus the high-dimensional models appear to suffer a definite overfitting problem. However Figure 6 suggests again that AIC's penalty for adding parameters must be adjusted for the case where our predictors are singular vectors. Here we plot the linear model's coefficient of determination, R^2 iteratively, for increasing values of k. Above $k \approx 75$ we see particularly steep increase of R^2 between k-values of approximately 50-100, suggesting that these factors contribute the most information to the model. After approximately 100 dimensions, the increased fit attributable to a new factor appears to taper off.

In both the logistic and linear models, factors between approximately 50 and 100 appear to be the most informative for purposes of information retrieval. Bringing model building techniques to bear on the matter of dimensionality reduction allows us to estimate the significance of each factor as we iterate through the process of selecting a final LSI system. Our data suggest that for the CF database, while factors in the range of 50-100 are the most significant in general, not all factors in this range are equally informative. An important result from our study is the varied statistical significance of factors in this crucial range. This variety implies that selecting the first k LSI factors may not provide the "best" model for retrieval purposes. Instead, system designers might do well to select the k most significant factors, averaged across a number of training

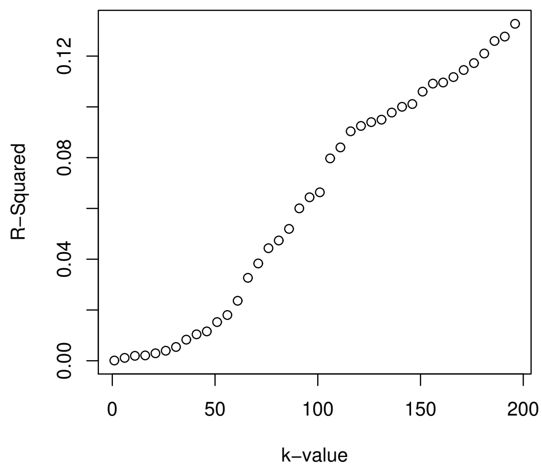

Figure 6: R^2 for Linear Regression

queries. To test the validity of this implication, further research is in order.

5 Conclusion

Casting information retrieval as a regression problem allows us to view feature selection in terms of statistical model building. This is particularly helpful when we confront the choise of an appropriate k-value for an LSI system. This study argues that adding singular vectors to an LSI-based classifier is analagous to building a statistical model. Thus we argue that choosing an appropriate value for k entails balancing the descriptive power and parsimony of competing models. An LSI system will perform well when k is chosen to be high enough to capture the semantics of the data set, but not so large as to overfit the model. We have seen that classification accuracy for the CF database declines as k increases above approximately 150, suggesting that higher-order singular vectors are leading to an over-fitted model. This result is important insofar as it lends rigor to the common assertion in LSI literature that these singular vectors correspond to "noise." In other words, models of high-dimensionality suffer because the first singular vectors lead to a well fit model; adding additional predictors decreases residual error at the expense of prediction.

A second important implication of this study invloves the question of which singular vectors should be added to an LSI system. Figure 3 shows that high-ordered singular vectors add little to classification models. However, the power of lower-ordered singular vectors is not consistently high. For instance, the first three predictors add very little to the model fit, while singular vectors near 75 appear highly significant. This suggests that adding the k largest singular vectors may not lead to the best system performance. Instead, we might do better to choose the k "most significant" singular vectors. Using more advanced model selection criteria will aid in this analysis. Moreover, how to extrapolate this technique to unlabeled data (i.e. collections without relevance judgements) is a topic we hope to address in future research.

Other areas that will require more attention relate to two main areas. First, use of the AIC metric in this study demonstrates that models using singular vectors as predictors lose degrees of freedom more quickly than standard model building algorithms anticipate. As such, we hope to complete an analysis of the degrees of freedom under LSI. Second, we hope to extend our analysis to more realistically sized data sets. The CF database is very small by contemporary research standards. Thus in future work we will use larger, more complex data to analyze dimensionality reduction from the perspective of statistical model building.

References

[1] K. P. Burnham and D. R. Anderson. *Model Selection and Inference: A Practical Information-Theoretic Approach.* Springer, New York, 1998.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. of the American Society for Information Science*, 41(6):391–407, 1990.

[3] C. H. Q. Ding. A similarity-based probability model for latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer, New York, 2001.

[5] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression.* Wiley, New York, 2000.

[6] T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[7] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing.* MIT Press,, Cambridge, 1999.

[8] P. McCullagh and Nelder J. A. *Generalized Linear Models.* Chapman and Hall, Boca Raton, second edition, 1989.

[9] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied Linear Statistical Models*. Irwin, Chicago, 1996.

[10] W. M. Jr. Shaw, J. B. Wood, R. E. Wood, and H. R. Tibbo. The cystic fibrosis database: Content and research opportunities. *Library and Information Science Research*, 13:347–366, 1991.

[11] R. E. Story. An explanation of the effectiveness of latent semantic indexing by means of a bayesian regression model. *Information Processing and Management*, 32(3):329–344, 1996.