Non-negative matrix factorization for the analysis of genotype and phenotype data and its relation to patient prognosis and outcome

Thomas Christie and Joshua Lynch, with guidance from TaeHyun Hwang

December 14, 2012

Part I
Introduction

1 Motivation

The availability of detailed bioinformatic data is growing quickly, but reliable methods of relating this wealth of information to patient prognosis and diagnosis are often lacking. In the medical domain, biometric data often consists of an array of features for each patient. These features may be simple measures such as height and weight, or the results of complex analyses such as the presence or absence of certain genetic sequences in a patient's DNA.

With the recent explosive growth in computing power, it is becoming more common to collect all available data and determine relationships between data values and patient prognosis and outcomes after the fact. For example, recent studies have shown that collected gene expression data ([2]) and quantified features from breast cancer images ([1]) are predictive of cancer metastasis. Though this "data-driven" analysis can lead to the production of unnecessarily large amounts of data, there are many benefits to this approach. First, data is not deemed useless a priori due to the verdict of domain experts, and this can lead to the discovery of previously unexplored relationships. In [1], image features were discovered that predicted patient outcome better than features traditionally used by physicians. Moreover, an active research is being conducted to develop methods of combining data from different sources (see e.g. [10] for an integration of gene expression and micro-RNA data).

In this investigation, we use a linear algebra approach to uncover structure in biometric data and its relationship to patient prognosis and outcome. In what follows, we introduce a method based on non-negative matrix factorization for selecting features relevant to patient labels and thereby reducing the dimensionality of patient data. We then use non-negative matrix bifactorization to uncover structure underlying patient data that can successfully separate patients into groups of high- and low-survival outcome. Lastly, we use spectral clustering to investigate the ability of features to successfully separate patients into meaningful groups and demonstrate the utility of feature selection.

2 Mathematical Approach

In the case of biometric data, it can be useful to assume that an underlying "cause" is expressed in several of the collected features. For example, the presence of a specific gene can give rise to the synthesis of many proteins. Similarly, a single disease may give rise to a collection of phenotypes. One way to model this relationship is to posit a latent causal or explanatory structure underlying the measurable data (see Figure 1). Discovering hidden explanatory features can potentially lead to a simple explanation for various features,

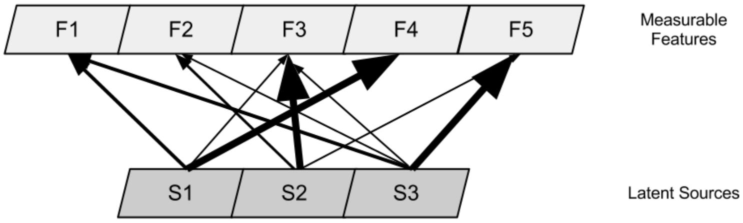

Figure 1: Matrix factorization attempts to find a collection of vectors that can be linearly combined to create measured feature vectors. The discovered vectors can be considered “latent source features” that combine to explain feature combinations manifest in the data. An optional orthogonality constraint can be placed on source vectors. Additionally, a lower number of source vectors can be used to produce an approximation of the feature matrix, resulting in an approximation of the data matrix with reduced dimensionality.

as well as drastically reduce the dimensionality of the data. Though the domain is drastically different, a similar approach is often used to uncover semantic content underlying language data.

2.1 Matrix Decomposition

To model the relationship between latent variables and measurable features, we assume that each feature is a linear combination of some number of latent variables. Mathematically, the features measured for each patient can be written as a vector $\mathbf{x}$, where $\mathbf{X} = [x_1 x_2 ... x_n]^T$ is an $n \times m$ matrix with n patients and m features. In order to uncover the latent factors, we decompose X into two factors W and H such that $X \approx WH$. In this factorization, W is $n \times k$ matrix whose component vectors span the column space of X, where entries of H provide vector weights. In other words, X consists of a linear combination of the column vectors of W with weights from H. W provides the k “sources” or latent variables and H describes how they linearly combine to produce the features in X. WH is an approximation of X, and the quality of the approximation is determined partially by the choice of k. The dimension k is often chosen heuristically, and different values of k prove useful in different problem domains.

Matrix factorization is not inherently unique, so an important question is how to decompose X into H and W. One principled way to perform such a factorization is to find component vectors that are mutually orthogonal. This is called Singular Value Decomposition (SVD), and constructs a unique factorization $X = USV^T$, where U corresponds to W and the latter product corresponds to H. This approach is mathematically robust and well-understood. A lower-rank approximation of X can be constructed using the first k columns of U, S and V^T.

From the perspective of uncovering latent sources, however, SVD presents several problems. First, it may well be the case that latent sources are not actually orthogonal. In this case, the U matrix would not correspond to any domain-specific functional process but would be a mere mathematical convenience. In addition, SVD does not impose a constraint on the sign of values of H. A combination of positive and negative coefficients create both additive and subtractive components of each feature, making it very difficult to interpret how latent sources combine to create features. Lastly, SVD produces results which are dense (few 0s in the components of W and H). This implies that every latent source has an effect on every feature present in X, an implication which is likely not true in many applications. Approaches for addressing mixed signs, orthogonality and density are addressed below.

2.2 Non-Negative Matrix Factorization (NMF)

One approach to creating easily interpretable latent sources is to introduce the additional requirement that the data have values greater than or equal to zero and perform matrix decomposition in such a way that values of H and W are exclusively nonnegative. This means that the underlying components of X combine in a purely additive way to produce X, which gives a more intuitive decomposition of the data matrix. Moreover, in situations where data is naturally nonnegative (such as image data), negative components may not be readily interpretable and non-negative decomposition is preferred.

Biometric data often originates with mixed signs, either inherently or due to common data transforms. When faced with a $n \times m$ matrix X with mixed-sign entries, a common technique for imposing non-negativity is to construct a new $n \times 2m$ matrix $X' = [\max(X, 0) \max(-X, 0)]$. In other words, the left half of X' consists of the originally positive values in X with the negative values replaces by 0. The right half of X consists of the absolute value of originally negative values of$X$, with 0's replacing corresponding positive values. For example:

$$\begin{bmatrix} 1 & -2 & -1 \\ 2 & 5 & 3 \\ 2 & -4 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 1 \\ 2 & 5 & 3 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 4 & 0 \end{bmatrix}$$

This is the approached used in the methods below.

Placing an orthogonality constraint on the vectors of the basis matrix W forces components to be uncorrelated. While this is useful from a data compression standpoint and has other convenient mathematical properties, it may well be the case that distinct underlying sources have overlapping manifestations in feature space. In the approaches used in this paper, the orthogonality constraint is dropped.

2.3 Sparse Decomposition

A matrix is called sparse when the majority of its elements are 0. In cases where a sparse H can be found, the product WH can be seen as an approximation of X in which each feature of X is composed of a small number of columns of W. This is useful, as it highlights which latent components are most responsible for each feature. Note that with sparse data, even through the orthogonality constraint is relaxed, columns have a high probability of being orthogonal (for a more detailed explanation see [5]).

2.4 Matrix Decomposition as Optimization

Factoring a nxm matrix X into a product WH can be formulated as an optimization problem:

$$\min_{W,H} f(W, H) = \frac{1}{2} ||X - WH||^2 \tag{1}$$

where $|| \cdot ||$ is a convenient matrix norm, $W \epsilon \mathbb{R}^{n \times k}$ and $H \epsilon \mathbb{R}^{k \times m}$. If X is non-negative, non-negativity can be imposed as an additional constraint on W and H. Sparsity can be controlled by adding a term penalizing the size of the L^0 norm of W and/or H. In practice the L^1 norm is often used as an approximation of the L^0 norm, as using the L^1 norm gives a convex optimization problem while still enforcing sparsity. There are many well-studied algorithms for efficiently solving convex optimization problems.

2.4.1 Convex Optimization

The general problem of factoring a matrix into a product of two or more matrices is under-specified even with additional constraints such as non-negativity. In fact, requiring a non-negative factorization results in a more difficult problem than a simple unconstrained decomposition such as SVD because in such cases no closed-form solution exists and results are not guaranteed to be unique. For the most part these problems are formulated as optimization problems and solved iteratively. Solving an optimization problem such as 1 efficiently with guarantees of convergence and with known error bounds is in general difficult. If the problem can be converted to a convex optimization it may become tractable. Several techniques may be used including 'relaxation' to a convex norm (substituting a convex norm for a non-convex norm), and alternately optimizing one matrix while holding the others constant. The factorization methods used in this paper utilize both of these techniques.

2.4.2 Lagrange Multipliers

A common approach to incorporating equality and inequality constraints into an optimization problem such as 1 is to reformulate the problem with additional variables, one for each constraint. This new formulation is the Lagrangian dual problem and the new variables are analogous to Lagrange multipliers. For convex optimization problems the solution to the Lagrange dual problem may be the exact solution to the original, or primal, problem. Even when the primal and dual solutions are not identical the dual solution is at least a lower bound on the solution to the primal problem.

2.5 Data Sources and Preparation

Two datasets were used in the investigations below. The first, which we term the Gene-Image dataset, consists of both gene expression information and cell image feature information from cancerous cells. The gene and image components were acquired from [2] and [1], respectively.

The complete gene expression dataset contains 24,481 features for 248 subjects. This data was preprocessed to remove expression information related to genes for which no name was provided. Also, gene expression features with absolute values lower than the 60th percentile were removed, as were gene expression values with variance lower than the 30th percentile. After preprocessing 3,550 expression features remained, which were then standardized to have zero mean and unit variance.

The complete image feature data set contains 6,642 features for the same 248 subjects. All 6,642 features were used. As with the gene expression data, the cell image features were standardized to zero mean and unit variance.

The gene expression and cell image data both contained negative feature values. In order to use multiplicative updates, every entry in the matrix to be factored must be positive. After column-wise standardization, each data matrix was doubled along the columns to impose non-negativity as described above.

Each patient was labeled as having cancer which had metastasized or not, and duration of patient survival following treatment onset was also included.

The second dataset is called the Wisconsin Diagnostic Breast Cancer dataset ([7], the shorter 'Wisconsin Dataset' is used below), collected in the lab of Dr. William H. Wolberg at the University of Wisconsin. The data consists of 10 image features extracted from each of three cells from 569 subjects. Only two cells (20 total features) were used, and the feature matrix was normalized and made non-negative. Each patient was labeled as having benign or malignant tumors.

Part II

Experiments

The goal of this project was to evaluate the use of non-negative matrix factorization as a tool for uncovering structure in large datasets. With this in mind, we had three goals:

1. Given a dataset with patients, corresponding features, and labels, to use non-negative matrix factorization to uncover features, latent and actual, most relevant to patient labels.
2. Given the same dataset and two different types of features, to determine whether a common underlying structure (or common basis, in mathematical terms) in those features relates to patient survival.
3. To perform spectral clustering on the dataset both in a naive fashion (using all features) and using the matrix factorization-based feature selection developed below.

3 Supervised Approach

In this section, we used non-negative matrix factorization to uncover features and feature combinations related to patient tumor category.

3.1 Non-negative Matrix Trifactorization

In traditional SVD, a matrix X is factored into three matrices $X = USV^T$ in such a way that the columns of U and V are orthonormal and are typically composed of mixed-sign entries. Columns of U are a basis for the column space of X, and are combinations of values that can be thought of as 'latent features' composed of combinations of the original columns such that they point in the direction of greatest variance, next greatest, and so on for each column of U.

In an analogous way, we perform a non-negative matrix trifactorization that uncovers latent features that account for the greatest variation in our data. If X is composed of two concatenated feature sets, the latent features are linear combinations of the original features, and thereby provide insight into which features combine to predict variability in the patient labels.

In order to maintain consistency with other non-negative matrix factorization literature, we use the matrix labels F, S and G to represent the three factors of X. Each row of X corresponds to a subject, and each column corresponds to a feature. Since we are attempting to uncover features and feature combinations as they are relevant to patient labels (and not to overall variability among patients), we will fix the matrix F and use it to encode patient labels. Unlike SVD, non-negative matrix factorization is non-unique, and because we are drastically constraining the values of F we are actually computing S and G^T in such a way that FSG^T most closely approximates X rather than factoring it exactly. From this perspective, the factorization is turned into a minimization problem, namely computing non-negative S and G in a way that minimizes $||X - FSG^T||_F$, where $||\cdot||_F$ indicates the Frobenius norm, or rooted sum of squared elements, of a matrix.

Let X_1 and X_2 be the two non-negative subject-by-feature matrices with common subjects and different feature sets, and let L be a vector of binary labels for subjects. We then create an matrix $X = [X_1 X_2]$ as a concatenated subject-by-feature matrix. Let F be a 2-column matrix such that entries in the first column of F in rows where L is 1, and entries in the second column of F are 1 where L is 0. S is a matrix with 2 rows and k columns, where k is the number of latent feature vectors to be discovered. Finally, m is the total number of concatenated non-negative feature vectors, and G is a $m \times k$ matrix such that each row corresponds to a non-negative feature and each column corresponds to a latent feature to be discovered. As described below, k is chosen so that the rank of G corresponds to the number of singular values that explain roughly 90% of the variance in X. A visual representation of the matrices can be seen in Figure 2.

3.2 Problem Formulation

Keeping in mind that we desire sparse and non-negative S and G, the problem described above can be written as the following:

$$\min_{S\geq 0, G\geq 0} \frac{1}{2}(||X - FSG^T||_F^2 + \lambda_S \sum_j ||s_j||_1^2 + \lambda_S \sum_{j'} ||g_{j'}||_1^2) \tag{2}$$

where both X and F are fixed. The two additional 1-norm terms serve to encourage sparsity in S and G, and represent a sum of the absolute values of each entry of each column, all summed and then the result squared. Since entries in S and G are non-negative, the terms represent the sum of all elements of S and G, respectively. Sparsity is ideally computed using the so-called 0-norm, which is the count of non-zero elements of a matrix. The 1-norm is a commonly used relaxation of the 0-norm, which is the closest norm to the 0-norm that is also convex, and can therefore be optimized using convex optimization methods.

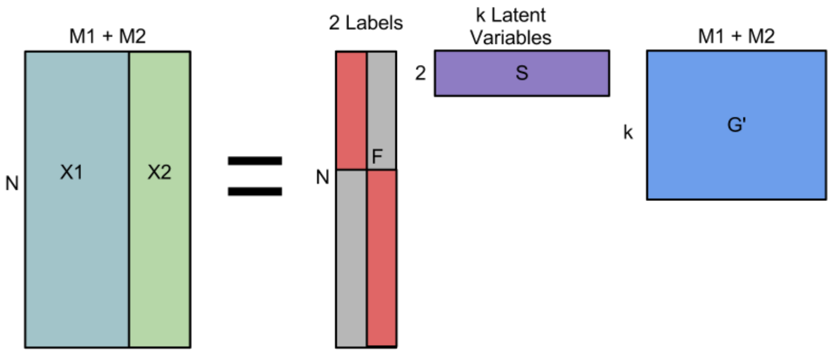

Figure 2: Non-negative trifactorization schematic using patient labels. X corresponds to the data matrix, F to enforced labels, and S and G are learned matrices. $M1$ and $M2$ correspond to the data from different sources that are concatenated to create X. N is the number of subjects, and k is the number of latent features in G.

Following a similar formulation in [10], the objective function corresponding to 2 can be written

$$\mathcal{F}(S,G) = \frac{1}{2}(||X - FSG^T||_F^2 + \lambda_S \sum e_{1\times 2}SS^T e_{2\times 1}^T + \lambda_G \sum e_{1\times k}GG^T e_{k\times 1}^T) \tag{3}$$

with the constraints $G \geq 0$ and $S \geq 0$. The latter two terms again represent the sum of entries S and G squared, and e is a vector of 1s.

Equation 3 contains two variables and is therefore not simultaneously convex in both, but only in each individually. To achieve convexity and minimize the objective function, we fix either S or G, update the other, and iterate until a convergence criteria is met. In order to construct a method of calculating S and G independently, we first construct the Lagrange $\mathcal{L}$. Let ϕ_{ij} and ψ_{ij} be the Lagrange multipliers for the constraints $S \geq 0$ and $G \geq 0$, respectively. Then

$$\mathcal{L}(S,G) = \mathcal{F} + Tr(\Phi S^T) + Tr(\Psi G^T)$$

where $\Phi = [\phi_{ij}]$ and $\Psi = [\psi_{ij}]$. The partial derivatives of $\mathcal{L}$ with respect to S and G are:

$$\frac{\partial \mathcal{L}}{\partial S} = -F^T XG + F^T FSG^T G + \lambda_S e_{2\times 2} S + \Phi$$

$$\frac{\partial \mathcal{L}}{\partial G} = -X^T FS + GS^T F^T FS + \lambda_G e_{k\times k} G + \Psi$$

Based on the KKT conditions $\phi_{ij} S_{ij} = 0$ and $\psi_{ij} G_{ij} = 0$, we get the following equations for S_{ij} and G_{ij}:

$$(-F^T XG)_{ij} S_{ij} + (F^T SG^T G + \lambda_S e_{2\times 2} S)_{ij} S_{ij} = 0$$

$$(-X^T FS)_{ij} G_{ij} + (GS^T F^T FS + \lambda_G e_{k\times k} G)_{ij} G_{ij} = 0$$

Moving the first term of each to the right side of the equal sign and dividing by the first part of the second term gives the following update rules, which are used iteratively:

$$S_{ij} \leftarrow S_{ij} \frac{(F^T XG)_{ij}}{(F^T SG^T G + \lambda_S e_{2\times 2} S)_{ij}}$$

$$G_{ij} \leftarrow G_{ij} \frac{X^T FS}{(GS^T F^T FS + \lambda_G e_{k\times k} G)_{ij}}$$

If the values are initialized as non-negative, the update rules are guaranteed to be non-increasing and are only stationary if S and G are at a stationary point (see [10] supplementary material for a proof). In practice, a small epsilon is added to the denominator of each update rule to preclude devision by zero. Additionally, F is column-normalized and G is row-normalized at each iteration so that S absorbs the scaling from both matrices.

As the update rules are multiplicative and each new value of S and G is a function of past values, S and G must first be initialized. The matrix$F$ contains patient labels and can be thought of as an indicator matrix for clustering patients. Correspondingly, the matrix G can be thought of as clustering features, which correspond to columns of X. With this in mind, we initialize G using k-means clustering on columns of X, where the number of clusters used is the number of columns (latent features) in G. The resulting matrix is an $n \times k$ indicator matrix of feature clusters in which each column has a single 1 representing cluster membership and all other entries are 0. The initial G is set to this matrix. S is initialized to random numbers between 0 and 1.

3.3 Feature Selection

As mentioned above, rows of F indicate an enforced clustering on subjects corresponding to clinical patient labels, in which each row contains a zero and nonzero entry (not 1 because of column-wise scaling). Group membership can be determined by assigning a subject to the column-label with corresponding highest value in the subject's row in F. Since F is predetermined and fixed, this merely represents a change in interpretation from label to cluster membership.

Once S and G are learned, we can consider the $2 \times m$ product SG^T, where m is the total number of non-negative features. This can be thought of as an indicator matrix similar to F, except that each feature is assigned to one of 2 feature clusters. Unlike the random initialization of G, in which features were clustered without taking into account patient labels, the learned SG^T represents feature cluster membership as they relate to the patient labels. In other words, each (normalized) column ofSG^T represents a posterior probability of that feature being relevant to classifying a patient in one label or the other.

When considered as probabilities, it is reasonable to state that the higher a probability corresponding to one label is, the stronger the corresponding feature will be in predicting class membership. On the other hand, for features in which both probabilities are similar (both roughly 0.5), the predictive value of that feature is relatively small.

We tested this method using the Wisconsin dataset. As indicated above, F was created by normalizing a binary indicator matrix for feature labels. S was initialized randomly, and G was initialized by choosing the k-means clustering result with the lowest SSE over 10 runs. k was chosen to be 29, which corresponds to the number of singular values of the data matrix corresponding to 90% of the variance of the data matrix.

The results of the trifactorization are shown in Figure 3.

Class labels in F are readily apparent in Figure 3. As desired, G is sparse and S approximately so, with only a few values being relatively very high. It is not immediately apparent how the features of W correspond to the labels in F, however. Multiplying S and G^T gives a much clearer picture, shown in Figure 4. It is clear that values corresponding to actual cells typically show much stronger contrast between one class and another than the random data. It appears that values corresponding to positive feature values (or, the high-end of values once features are standardized) are are predictive of the first class, whereas lower/negative values tend to be predictive of the second class. Of course, since there are only two classes, any extreme value is by default predictive of membership in both classes. The expected ratio between class membership values for non-predictive features is equivalent to the proportion of subjects in each group. Since there are roughly 1.8 times as many subjects in the second column of F in Figure 3 as the first, we expect the ratio

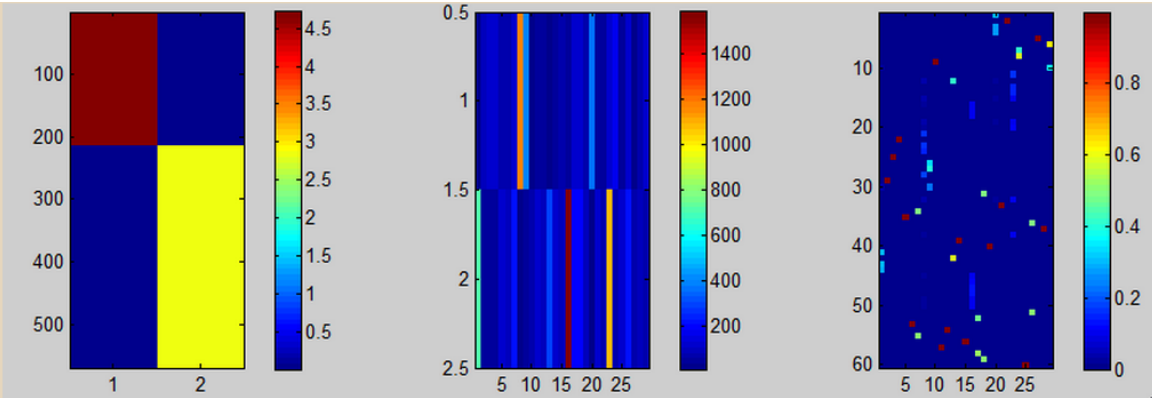

Figure 3: Trifactorization of modified Wisconsin Diagnostic Breast Cancer dataset X into F,S and G, respectively.

for non-predictive noise attributes in the second row of SG^T to be about 1.8, which is indeed what we see.

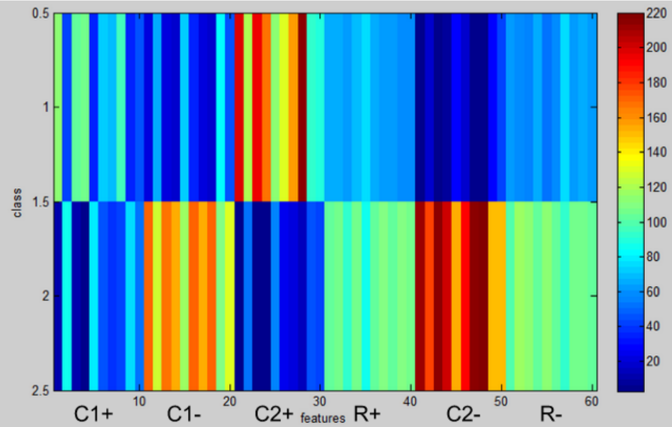

Figure 4: $SG^T$from the factorization of the modified Wisconsin Diagnostic Breast Cancer dataset. Columns 1-20 correspond to 10 positive and 10 negative features from Cell 1. Features 21-30 and 41-50 are positive and negative features from Cell 2. Features 31-40 and 51-60 are positive and negative random features drawn from a normal distribution with mean 1 and standard deviation 0. Note that columns are not normalized in this figure.

Figure 5 displays ratios of the first to the second row as well as the converse. Large values in both cases correspond to the highest power of features to differentiate between patient classes. Several aspects of this figure are worth noting. First, entries corresponding to noise features show consistently low ratios (close to 1.8). Ratios between corresponding positive and negative features from Cell 2 are both high (in entries 21-30 and 41-50), which suggests that higher-than-average values predict one class and lower-than-average of the same feature predict the other class. Finally, features from Cell 2 (entries 21-30 and 41-50) appear to be generally stronger predictors than features from Cell 1 (entries 1-20).

In order to validate the applicability of features with high ratios in Figure 5, we compare the top values from the top and bottom ratios to the features with highest information gain with respect to class labels. The top features from Row 1/Row 2 are features 23, 24, 21, 28 and 4, in that order. These correspond to features 14, 11, 13, 4 and 18 in the mixed-sign $X' = [X1X2]$ matrix of cell features. The top features in Row 2 / Row 1 are numbers 43, 41, 48, 47 and 44, which correspond to features 13, 11, 18, 17 and 14 in X'. We used WEKA ([4]) to calculate the information gain with respect to patient class. According to this measure, the most informative six features are 13, 14, 11, 18, 4 and 17, all of which are in the top feature ratios from Figure 5. Thus, non-negative matrix factorization with a class-label matrix F produces a SG^T matrix that can readily detect which features are relevant to class labels.

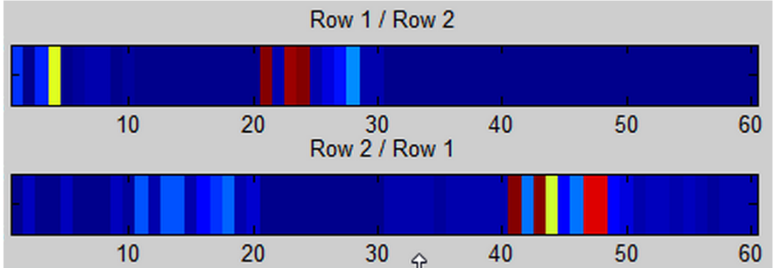

Figure 5: Ratios between rows 1 and 2 of SG^T resulting from matrix trifactorization.

.

3.4 Feature Interaction

The matrix G^T in Figure 3 does not readily reveal insights into the relationships between features and classes. Each entry corresponds to a feature's belonging to a latent variable which itself points in a direction of high variation. Pairwise correlations between rows of G^T reveal which pairs of features have similar presence across latent features, and therefore represent an association between features as they relate to patient labels. Taking correlations between features across subjects in the data matrix X reveals unrestricted relationships between features, while taking correlations between features in the G^T matrix acts to filter the correlations, revealing those feature correlations that most strongly reflect patient labels. These two calculations are compared in Figure 6.

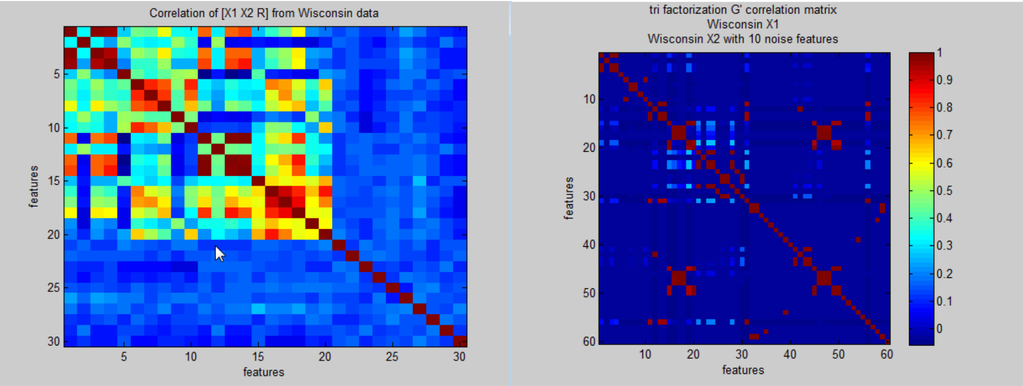

Figure 6: The left graphic shows correlations between features in the data matrix X. The right graphic shows correlations in the doubled matrix G^T. Rows and columns of both correspond to image features. Recall that features 1-20 are image features from two cells, and features 21-30 are noise features. The right correlation matrix is much sparser than the left. This is a direct effect of the sparse matrix decomposition, and serves to highlight only those correlations that relate to the subject labels. However, some correlations in noise features can also be seen in the right matrix which are not present in the left.

4 Unsupervised Approach

Accurate patient prognosis is a vital part of medical practice. When a patient is diagnosed with a disease, the expected course of the disease largely determines the choice of treatment. Moreover, studies are now finding

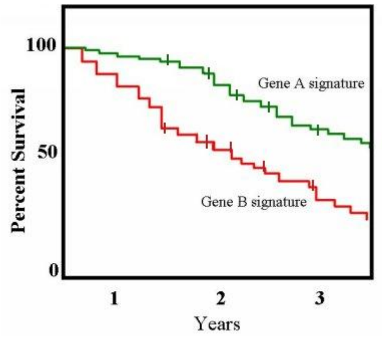

Figure 7: Example Kaplan-Meier survival curve. Image is public domain and was downloaded from the Wikipedia.org site on Kaplan-Meier survival analysis.

that patient genetic information is can be highly predictive of a patient's response to particular medications. Treatments for serious illnesses are often costly and can have permanent and debilitating side-effects. The ability to accurately predict the utility of various treatments on an individual patient is therefore critical for both patient well-being and the appropriate allocation of medical funds.

In what follows, we use non-negative matrix factorization to uncover structure in patient biometric data that is predictive of patient survival.

4.1 Survival analysis

Kaplan-Meier survival analysis is a commonly used technique for determining the ability of a measured feature to significantly predict the lifespan of a group of patients. To perform the analysis, matched patients are given a treatment and the number of years the patient survives during treatment is recorded. The goal is to find biometric data that is predictive of survival duration. This discovered metric can then be used to determine whether future patients should be given the treatment in question. In a typical Kaplan-Meier plot, subjects are split into two groups according to the value of a biometric marker. The fraction of each group surviving at every time point is then plotted. The log-rank statistical test is used to determine whether the difference between survival curves is statistically significant.

The Gene-Image dataset described in the introduction includes survival times for each patient. Our goal is to use non-negative matrix factorization to uncover linear combinations of features that separate patients into groups with significantly different survival rates. The Gene-Image dataset includes both gene expression and cancer cell image feature data. Gene expression data is presumably linked in an unknown way to phenotype data as expressed in image features. Our underlying assumption is that there exists a common set of basis vectors underlying these two datasets that contains information regarding patient response to cancer and therefore their survival outcome.

Finally, optimally combining data from multiple sources is an active area of research in the data mining and bioinformatics communities. As part of our approach, we wish to determine whether including a term representing potential gene-image interactions has an effect on the ability of this method to predict patient outcome.

4.2 Bifactorization with additional constraints

Let X_G be the non-negative $n \times g$ matrix of gene expression data, prepared as explained in the introduction, and let X_I be the $n \times i$ matrix of image features. Our goal is to discover an approximate common basis for X_G and X_I, and use the basis vectors to separate patients into groups. Once separated, we perform Kaplan-Meier survival analysis to determine whether the groups have significantly different survival outcome, and therefore whether the common basis vectors relate to patient outcome. Patient labels are not used in the factorization, so this represents an unsupervised approach to uncovering structure in patient biometric data.

Given X_G and X_I, we wish to find a common matrix of basis vectors W and two coefficient matrices H_G and H_I such that $||X_G - WH_G||_F^2 + ||X_I - WH_I||_F^2$ is minimized. It is convenient to impose further constraints on W, H_G and H_I. Namely, we wish to constrain the size of W and encourage sparse H_G and H_I. Sparse coefficient matrices ensure that each data matrix is composed of small linear combination of basis vectors, such that the most important component vectors are emphasized and less significant components are forced to 0. The nature of our data also provides an additional constraint: known gene-gene interactions are included in a matrix A (provided by TaeHyun Hwang) as prior knowledge that serve to force covariance between gene expression values that are known to be interrelated. Finally, hypothetical and unknown gene expression / image feature relationships can be represented by a matrix B. The complete objective function to be minimized is given in Equation 4

$$\begin{aligned}\mathcal{F}(W, H_G, H_I) = &||X_G - WH_G||_F^2 + ||X_I - WH_I||_F^2 \\ &- \lambda_1 Tr(H_G A H_G^T) - \lambda_2 Tr(H_I B H_G^T) \\ &+ \gamma_1 ||W||_F^2 + \gamma_2 (\sum_j ||h_j||_1^2 + \sum_{j'} ||h_{j'}||_1^2) \\ &+ \lambda_3 ||H_I^T H_G - B_0||_F^2\end{aligned} \tag{4}$$

The first line indicates the factor approximation of X_G and X_I to be computed. Terms in the second line represent gene-gene interaction and gene-image relationships, respectively. They are subtracted as we wish to compute W, H_G and H_I in such a way that these interactions are taken into account. The third line contains a term penalizing the growth of W, as well as a term encouraging sparse H_G and H_I. As in the trifactorization above, L^1-norm approximations are used for what is essentially a L^0-norm optimization, and the sparsity term merely indicates the sum of all elements of each H, taken together and squared. The last line is an optional term used for "learning" the gene-image interaction matrix B, and its use is described below.

In order to determine the effect of including the gene-image interaction matrix B in the optimization, we calculated W, H_I and H_G under three conditions:

1. Omit the interaction matrix B altogether. That is, set $\lambda_2 = \lambda_3 = 0$.

2. Set B equal to the correlation matrix between columns of X_I and X_G, namely $B = B_0$. Do not update B and set $\lambda_3 = 0$.

3. Initialize B_0 as the correlation matrix between columns of X_I and X_G. During each iteration, set $B = H_I^T H_G$, and set $\lambda_3 > 0$ in order to keep B somewhat constrained by its initial value. This adds an additional constraint on H_I and H_G, encouraging them to develop coefficients that incorporate correlated gene expression and image features.

We also wish to determine whether finding a common basis of X_G and X_I is a better predictor than simply computing a basis for X_G or X_I separately. In order to do this, we also compute a simple bi-factorization to minimize $\mathcal{F}(W, H) = ||X - WH||_F^2$ for $X = X_I$ and $X = X_G$ separately.

As in the trifactorization above, $\mathcal{F}$ is not a convex function. In order to force convexity we fix two of W, H_I and H_G, compute the remaining variable, and repeat, updating each in turn. Determining the appropriate multiplicative update rules for W,H_I and H_G proceeds as follows, following the supplementary material in [10]. First, the objective function $\mathcal{F}$ can be rewritten as the following:

$$
\begin{aligned}
\mathcal{F} = & Tr(X_G X_G^T) - 2Tr(X_G H_G^T W^T) + Tr(W H_G H_G^T W^T) \\
& + Tr(X_I X_I^T) - 2Tr(X_I H_I^T W^T) + Tr(W H_I H_I^T W^T) \\
& - \lambda_1 Tr(H_G A H_G^T) - \lambda_2 Tr(H_I B H_G^T) + \gamma_1 Tr(W W^T) \\
& + \gamma_2 (e_{1\times k} H_I H_I^T e_{k\times 1}^T + e_{1\times k} H_G H_G^T e_{k\times 1}^T)
\end{aligned}
$$

To minimize $\mathcal{F}$ according to the additional constraints $W \geq 0$, $H_I \geq 0$ and $H_G \geq 0$ we create the Lagrangian

$$\mathcal{L}(W, H_I, H_G) = \mathcal{F} + Tr(\Phi W^T) + Tr(\Psi_I H_I^T) + Tr(\Psi_G H_G^T)$$

where $\Phi = [\phi_{ij}]$, $\Psi_I = [(\psi_I)_{ij}]$ and $\Psi_G = [(\psi_G)_{ij}]$ are the Lagrange multipliers for the constraints $W_{ij} \geq 0$,$(H_I)_{ij} \geq 0$ and $(H_G)_{ij} \geq 0$ respectively. The partial derivatives of $\mathcal{L}$ with respect to W, H_I and H_G are:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W} &= -2X_I H_I^T + 2W H_I H_I^T - 2X_G X_G^T + 2W H_G H_G^T + 2\gamma_1 W + \Phi \\
\frac{\partial \mathcal{L}}{\partial H_I} &= -2W^T X_I + 2W^T W H_I - \lambda_2 H_G B^T + \gamma_2 2 e_{k\times k} H_I - \lambda_3 2 H_G B^T + \lambda_3 2 H_G H_G^T H_I + \Psi_I \\
\frac{\partial \mathcal{L}}{\partial H_G} &= -2W^T X_G + 2W^T W H_G - \lambda_1 2 H_G A - \lambda 2 H_I B + \gamma_2 2 e_{k\times k} H_G - \lambda_3 2 H_i B_0 + 2 H_I H_I^T H_G + \Psi_G
\end{aligned}
$$

Using the KKT condition $\phi_{ij} W_{ij} = 0$, $(\psi_I)_{ij}(H_I)_{ij} = 0$ and $(\psi_G)_{ij}(H_G)_{ij} = 0$, we can set each partial derivative to 0 and write

$$
\begin{aligned}
0 =& (-2X_I H_I^T - 2X_G X_G^T)_{ij} W_{ij} + (2W H_I H_I^T + 2W H_G H_G^T + 2\gamma_1 W)_{ij} W_{ij} \\
0 =& (-2W^T X_I - \lambda_2 H_G B^T - \lambda_3 2 H_G B^T)_{ij} (H_I)_{ij} + (2W^T W H_I + \gamma_2 2 e_{k\times k} H_I + \lambda_3 2 H_G H_G^T H_I)_{ij} (H_I)_{ij} \\
0 =& (-2W^T X_G - \lambda_1 2 H_G A - \lambda 2 H_I B - \lambda_3 2 H_i B_0)_{ij} (H_G)_{ij} + (2W^T W H_G + \gamma_2 2 e_{k\times k} H_G + 2 H_I H_I^T H_G)_{ij} (H_G)_{ij}
\end{aligned}
$$

Moving the first term of each to the right side of the equation and dividing gives the following update rules:

$$W_{ij} \leftarrow \frac{(X_I H_I^T + X_G X_G^T)_{ij}}{(W H_I H_I^T + W H_G H_G^T + \gamma_1 W)_{ij}}$$

$$(H_I)_{ij} \leftarrow \frac{(W^T X_I + \frac{\lambda_2}{2} H_G B^T + \lambda_3 H_G B_0^T)_{ij}}{(W^T W H_I + \gamma_2 e_{k\times k} H_I + \lambda_3 H_G H_G^T H_I)_{ij}}$$

$$(H_G)_{ij} \leftarrow \frac{(W^T X_G + \lambda_1 H_G A + \frac{\lambda_2}{2} H_I B + \lambda_3 H_I B_0)_{ij}}{(W^T W H_G + \gamma_2 e_{k\times k} H_G + \lambda_3 H_I H_I^T H_G)_{ij}}$$

As with the trifactorization multiplicative updates, a small epsilon is added to the denominator of each update to prevent devision by zero.

Following [10], parameters λ_1, λ_2, λ_3, γ_1 and γ_2 were set to 0.0001, 0.01, 0.001, 20 and 10, respectively. A parameter sweep confirmed that these values give consistently reasonable results. 20 factorizations were performed for each of the 5 conditions detailed above (3 complete bi-factorizations, one image-only and one gene-only). For each optimization, W, H_I and H_G were initialized to random values between 0 and 1, and the same W, H_I and H_G were used for each of the 5 methods. Multiplicative updates were repeated until

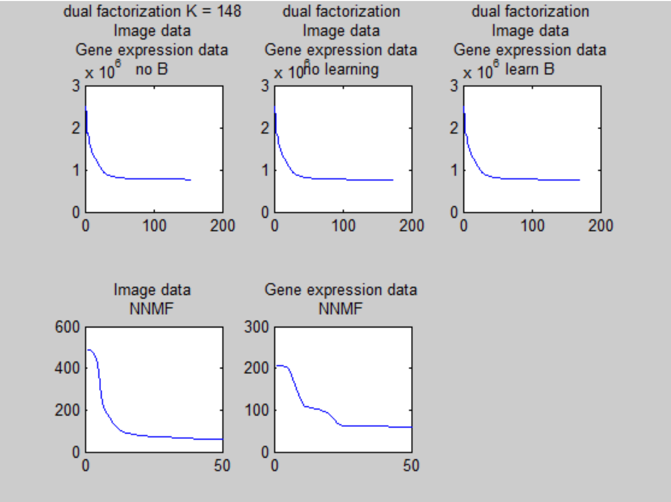

Figure 8: Typical convergence trajectories of the objective function for five bi-factorization methods.

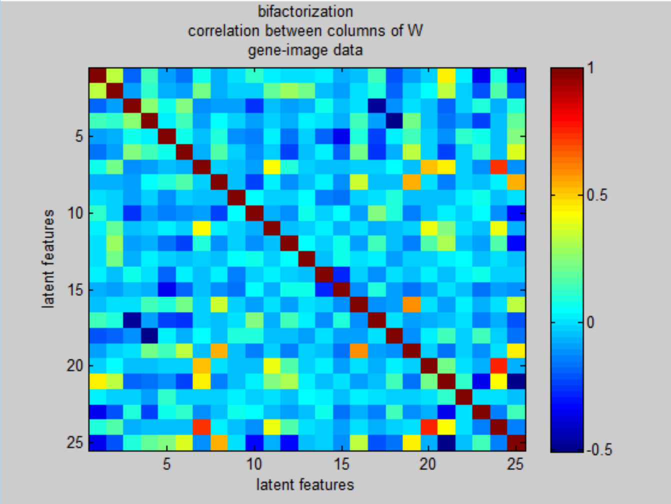

Figure 9: Example correlations between columns of W.

the objective function was improved by less than 0.01%. Figure 8 shows typical trajectories of the objective functions for each method.

For each initialization of W, H_I and H_G, each of the optimization algorithms was run. Once W was computed, subjects were ranked according to corresponding values of each column of W in turn. The top 50 subjects were labeled Group 1 and the bottom 50 subjects were labeled Group 2. The log-rank test was then performed to determine whether the survival outcome of the two groups was significantly different. The number of columns of W providing a significant separation of group outcomes was then counted.

Though this method is used in the literature for uncovering latent explanatory variables (see [6]), it is worth asking whether the count of columns leading to significant group difference has meaning. Non-negative matrix factorization does not guarantee orthogonal columns in its result. Columns of W may be correlated, in which case counting 2 columns of W that produce significant group differences may amount to tallying two vectors in a similar direction. As can be seen in 9, the majority of columns of W are nearly orthogonal. There are notable exceptions, however. In this example, column 24 is highly positively correlated with columns 7 and 20, and there are a number of high negative correlations.

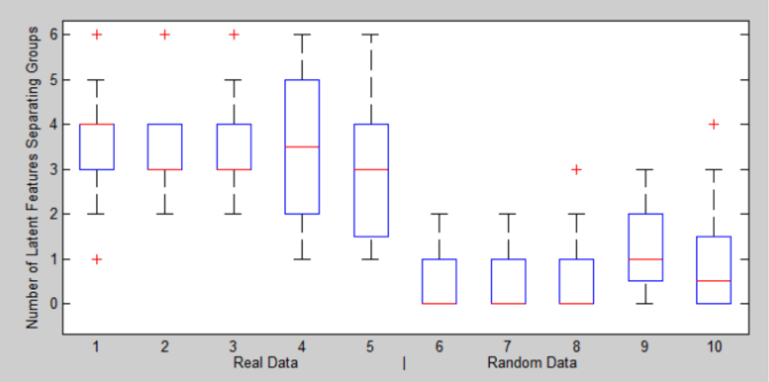

Figure 10: Number of columns of W that significantly differentiated patients into low- and high-survival groups at an $p = 0.05$ significance level. Entries 1-3 are the 3 variations of the bifactorizaiton method. Entry 4 uses only gene data, and entry 5 uses only image data. Entries 6-10 are results from the same methods using shuffled data.

Due to the lack of an orthogonality constraint, columns of W are not independent and column count cannot be used as a standalone measure of the ability of bi-factorization to uncover features that significantly differentiate patient groups. In order to provide a better basis of comparison, we created a randomized dataset and performed the same five factorization routines. We shuffled feature values across subjects within each feature (i.e. reordered each value in columns of X_G and X_I) so that distributions of each feature were maintained. Results for both real and randomized data are shown in Figure 4.2.

The results we found were somewhat surprising. There was virtually no difference between variations of the bifactorization method, that is, including a gene-image interaction term did not impact the number of columns of W significantly differentiating patients. However, including both types of data constrained the column count considerably, as entries 1-3 have much less variation than entries 4 and 5 (gene-only and image-only, respectively) in Figure 4.2. Another interesting result is that factorizations using real data consistently uncovered several columns predictive of patient survival, whereas the median number of columns from randomly permuted data predicting patient survival was much lower (0 in the case of the 3 methods using both data types).

5 Clustering

Clustering is a general family of unsupervised learning methods for partitioning a data set into groups of similar observations. There are many varieties of clustering algorithms, and each is most usefully applied to data with particular characteristics. Classical clustering methods include hierarchical clustering and k-means clustering. Hierarchical clustering assigns group labels based on some distance measure between observations, grouping nearby observations together. The k-means method is similar in that it uses a distance measure between observations, but rather than using all pairwise distances between observations this method tries to optimally locate some number, k, of cluster centers and assign each observation to the nearest cluster center.

Because these methods are based on a distance measurement they may not be good choices for data with irregularly shaped clusters. High-dimensional data is also challenging for the classical clustering methods since variance of distance between observations approaches zero as dimension increases and as the number of observations decreases. We use a relatively modern, generally robust method called spectral clustering to evaluate the feature selection results provided by non-negative matrix trifactorization. Spectral clustering methods are able to operate on a more general notion of similarity between observations than k-means and hierarchical clustering, although choosing an appropriate similarity measure is important to arriving at reasonable results. This allows for selection of similarity measures that may be more appropriate than Euclidean

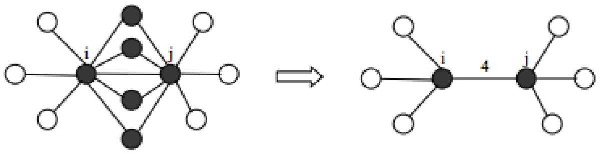

Figure 11: Creation of a Shared-Nearest-Neighbor (SNN) similarity matrix ([8])

distance or some related distance. Additionally, spectral clustering methods have simple implementations that take advantage of general-purpose linear algebraic operations.

Classification by spectral clustering on all features of the Wisconsin and Gene-Image data establishes a useful point of comparison for the feature-selection performance of the trifactorization, although we certainly expect trifactorization to perform better because it is a supervised method.

5.1 Shared Nearest Neighbor Similarity

Some clustering algorithms operate in feature-space, whereas others require a graph-based representation of the data in the form of an adjacency matrix. Consider each observation as a node in a graph. An adjacency matrix A consists of entries A_{ij} that correspond to some measure of "similarity" between observations/nodes i and j in the graph. In order to construct A, we must use a method that creates a similarity measure between observations using their location in feature-space. In low-dimensional space, Euclidean distance is often used as a similarity measure. As the number of dimensions grows, however, all pairwise distances become more alike and correspondingly less information-bearing regarding pairwise observation similarity.

One method for evaluating the similarity of points is to look at the collection of points surrounding each in feature space. The idea is that if two points have a similar feature "neighborhood," they will then be similar to each other. Using several nearest neighbors makes the Euclidean distance metric more robust to high dimensionality. The adjacency matrix is then comprised of the number of "nearest neighbors" two features have in common. The benefit of the SNN method is that it is relative to the local geometry. That is, SNN constructs a definition of similarity that is relatively robust in the face of high dimensionality and variations in density throughout the feature space.

In order to construct a shared nearest neighbor adjacency matrix for subjects, pairwise Euclidean distance is first computed using standardized features. For each subject, distances to other subjects are ranked and the closest p subjects to that subject are identified. Once each subject has a set of p "nearest neighbors," the SNN similarity measure between two subjects i and j, A_{ij}, is calculated as the cardinality of the pairwise intersections between two subjects' nearest neighbor sets. The parameter p is user-specified and controls the sparsity of the adjacency matrix and was chosen to be 30 in our calculations.

5.2 Spectral Clustering

Clustering is a very popular unsupervised learning technique for finding related groups in large data sets. Due to the requirement to reduce the dimensionality of the data, we found it convenient to convert the data from feature space into a graph-based representation. Spectral clustering is a robust and well-understood method for clustering nodes in a graph. Accordingly, we use it to partition the adjacency matrix in order to uncover patient groups.

The basic spectral clustering method is to treat the data set as a graph with observations as vertices and similarity between observations as edge weights, calculate the corresponding adjacency matrix and the related graph Laplacian matrix L and cluster the elements of a subset of eigenvectors of L. Once a similarity

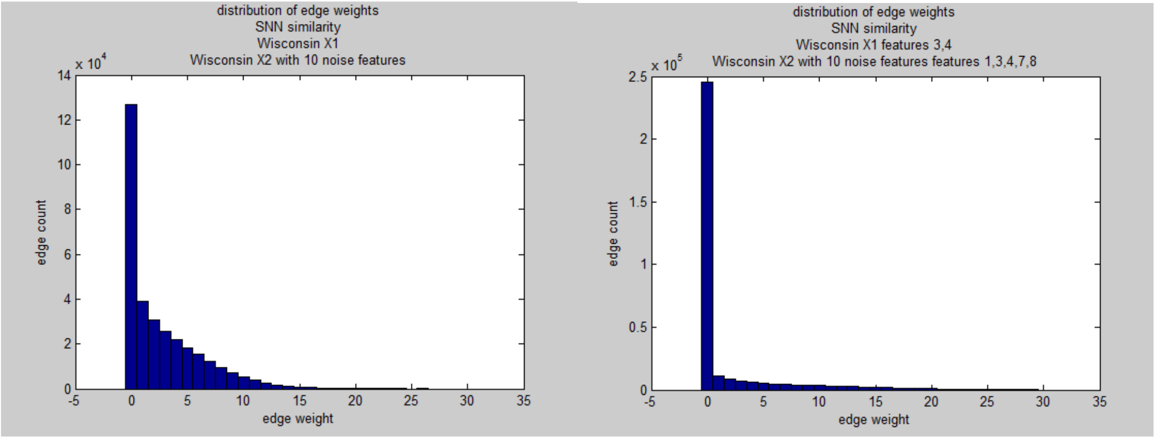

Figure 12: Edge weight distribution of the subject adjacency matrix A with SNN similarity for Wisconsin data with and without feature selection. Note that feature selection creates a much sparser adjacency matrix.

measure ϕ is chosen the corresponding adjacency matrix is calculated where each element A_{ij} is the similarity between observations i and j:

$$A_{ij} = \phi(i, j) \tag{5}$$

Next a graph Laplacian is calculated using the adjacency matrix A and the diagonal matrix of row sums of A, called D. At this point another choice must be made between three graph Laplacians:

$$L_U = D - A \tag{6}$$

$$L_{sym} = D^{-1/2} L_U D^{-1/2} \tag{7}$$

$$L_{rw} = D^{-1} L_U \tag{8}$$

These graph Laplacian matrices are called, respectively, 'unnormalized', 'normalized symmetric', and 'normalized random walk'. The tutorial [9] suggests using the distribution of degrees in the adjacency matrix as a deciding factor. If the distribution of degrees is narrow then the three graph Laplacians are likely to give similar clustering results. If the distribution of degrees is broad the normalized graph Laplacians may be better choices. Furthermore the normalized random walk graph Laplacian may be the better choice. [9] gives several supporting arguments for this choice.

Finally, compute the k smallest eigenvalues v_n and corresponding eigenvectors u_n of the chosen graph Laplacian, construct a matrix C with columns u_n, and cluster the rows of C (using a method such as k-means) to assign cluster labels. Since each row corresponds to an observation the row clusters of C assign labels to the observations.

The distributions of SNN edge weights was generally broad for both the Wisconsin data and the gene-image data. For this reason the random walk graph Laplacian was used for spectral clustering in all cases.

Sparse graphs are typically easier to cluster. As shown in Figures 12 and 13, selecting a subset of features leads to the construction of a much sparser graph, with the vast majority of pairwise edge weights equalling 0.

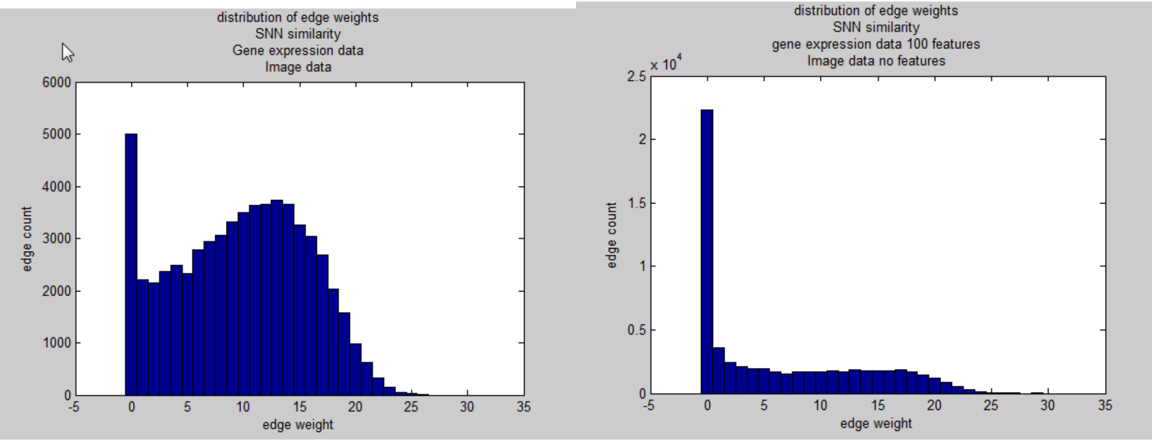

Figure 13: Edge weight distribution of the adjacency matrix A with SNN similarity for the Gene-Image dataset with and without feature selection. Note that once again the adjacency matrix is much sparser after using feature selection.

5.3 Feature Selection

Spectral clustering is used here as a method for evaluating feature selection derived from matrix bifactorization, where subjects were labeled according to individual columns of F. The benefit of feature selection is evaluated with total cluster entropy, defined as

$$H_{total} = \sum_i \frac{|C_i|}{N} H_{C_i}$$

$$H_C = -\sum \frac{|C_i^j|}{|C_i|} \log(\frac{|C_i^j|}{|C_i|})$$

where N is the number of observations, i indexes the clusters, and j indexes the known subject labels. This formulation represents a sum of the entropy of individual clusters, weighted by cluster size.

The Wisconsin breast cancer data set has relatively small dimensionality and is well-separated in the Cell 2 data, as can be seen in a plot of the first two singular vectors 14. Spectral clustering classifies this data with total cluster entropy 0.4430 (error rate of 0.11) without feature selection. Selecting the best 6 features as determined by matrix trifactorization, spectral clustering classifies this data with total cluster entropy 0.2553 (error rate 0.09).

The Gene-Image dataset is high-dimensional and has a relatively small number of subjects. Spectral clustering with all features gave poor classification results, with total cluster entropy 0.9259 (error rate 0.35). Selecting features based on the corresponding trifactorization improved the results, giving total cluster entropy 0.8426 (error rate 0.32).

In order to provide a 2D visualization with and without feature selection, Figure 14 contains plots of the Wisconsin dataset projected onto the first two singular vectors with all features and only the top features, respectively. Interestingly, despite an improvement in spectral clustering using the SNN-derived adjacency matrix, there is negligible difference between the data in top-2 singular vector space using all and selected features.

It seems clear from our results that non-negative matrix trifactorization is a useful tool for supervised feature selection. Spectral clustering performance improved in both the Wisconsin and Gene-Image dataset when feature selection was applied. This is not surprising, but serves as an additional validation step of the non-negative matrix trifactorization feature selection method.

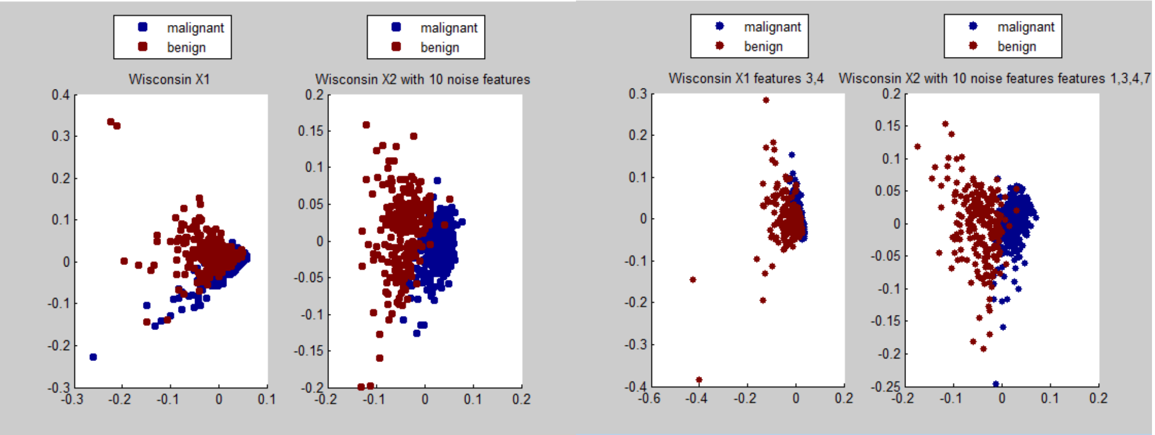

Figure 14: Wisconsin data projected onto the top-2 singular vector space, with and without feature selection.

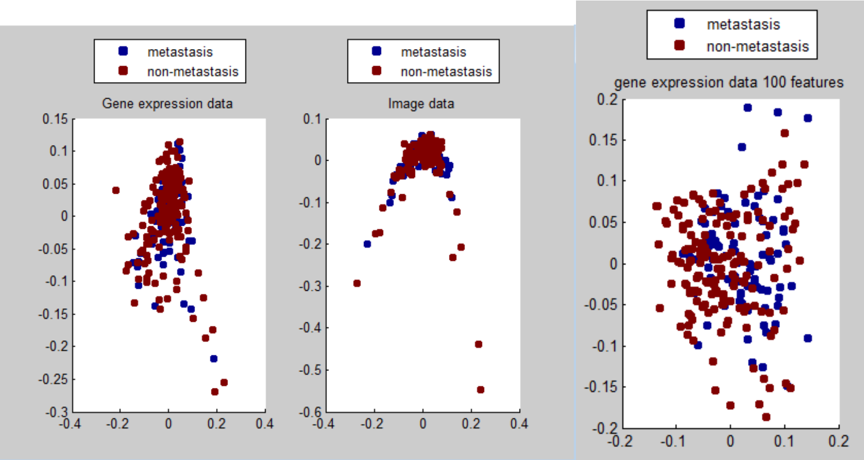

Figure 15: Gene-Image data points projected onto the first 2 singular vectors of the individual gene expression and image feature data. The left plot shows the projection using all features, and the right plot shows the same projection using only the top 100 features. None of the image features were among the top 100 features as determined by non-negative trifactorization feature selection.

It is interesting that in the feature selection of the Gene-Image dataset, all top 100 features are from genetic data and not image data. To look at it another way, gene expression data appears to be more predictive of whether cancer will metastasize than visual inspection of the cancer cells themselves. This suggests that genetic tests are a critical biomarker in determining patient prognosis.

Part III

Discussion and Future Work

Non-negative trifactorization appeared to be a useful method for supervised feature selection. Given that predictive features are identified by the ratio of rows of SG^T, bifactorization would be an equivalent, simpler method. The trifactorization method used here (and feature information gain, which gives similar results) only select features as they independently contribute to patient labels. Including a term with known feature interactions into the trifactorization, much as we did in the later bifactorization method, might provide better results. It is well known that information does not always optimally select features for use in classification, for example in situations where only a combination of features is predictive of class, but individual features are not. It would be worthwhile to test whether trifactorization with an additional interaction was useful in such cases. It is possible that rather than providing comparable results to information gain, feature selection might be considerably improved.

The findings presented in the bifactorization section are an interesting combination of negative and positive results. It is notable that including a gene-image interaction term did not appear to change the quality of the results at all. This may be a result of the parameter size used, and a parameter sweep of λ_3 would help determine this. It appears encouraging that using real data separates patient groups better than randomized data. Upon further reflection, however, we believe that the count of features separating patients into groups with different survival outcomes is not important without an orthogonality constraint on W. The presence of a single predictive column, combined with several additional correlated columns, would be enough to artificially raise the count. In addition, the entry-shuffled matrix used as a control removed the predictiveness of the columns of W. However, the result could also be explained by the shuffling simply removing correlations between columns of X and accordingly lowering the count of predictive columns. Performing multiplicative updates with an orthogonality constraint is not difficult (see [3] for examples), so we would include this requirement were we to perform this analysis in the future.

It would be convenient to have a way to uncover predictive basis vectors that determine survival outcome without knowing the survival rate beforehand. Perhaps using a cross-validation approach, combined with working backwards from basis vectors to combinations of important features, would be one way to build a model of feature predictiveness.

Finally, the initial (unrealized) goal of this project was to uncover relationships between data from multiple sources, and specifically between the gene expression and image feature data in the Gene-Image dataset. further work should involve a careful analysis of the contents of the learned B matrix in third bifactorization method used along with comparison between it and straight correlation data.

6 Acknowledgements

We thank TaeHyun Hwang for providing inital guidance and data for this project. The multiplicative code updates used in our tri- and bifactorization experiments were modified versions of those by [6] and [10], respectively. The implementation of the log-rank test used was written by G. Cardillo and is available at http://www.mathworks.com/matlabcentral/fileexchange/22317.

References

[1] Andrew H. Beck, Ankur R. Sangoi, Samuel Leung, Robert J. Marinelli, Torsten O. Nielsen, Marc J. van de Vijver, Robert B. West, Matt van de Rijn, and Daphne Koller. Systematic analysis of breast cancer morphology uncovers stromal features associated with survival. *Sci Transl Med*, 3(108):108ra113, 11 2011.

[2] van Vijver de MJ, He YD, van't Veer LJ, Dai H, Hart AA, Voskuil DW, Schreiber GJ, Peterse JL, Roberts C, Marton MJ, Parrish M, Atsma D, Witteveen A, Glas A, Delahaye L, van der Velde T, Bartelink H, Rodenhuis S, Rutgers ET, Friend SH, and Bernards R. A gene-expression signature as a predictor of survival in breast cancer. *The New England journal of medicine*, 347(25):1999–2009, 2002.

[3] Chris Ding, Tao Li, Wei Peng, and Haesun Park. *Orthogonal nonnegative matrix t-factorizations for clustering*. ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, 2006.

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[5] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–502, 6 2007.

[6] Seung-Jun . J. Kim, TaeHyun Hwang, and Georgios B. Giannakis. Sparse robust matrix tri-factorization with application to cancer genomics. In *2012 3rd International Workshop on Cognitive Information Processing (CIP)*, pages 1–6. IEEE, 5 2012.

[7] Nick Street, William H. Wolberg, and Olvi L. Mangasarian. *Nuclear feature extraction for breast tumor diagnosis*. University of Wisconsin-Madison, Computer Sciences Dept., Madison, Wis., 1992.

[8] Pang-Ning . N. Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Addison Wesley, Boston, 2006.

[9] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[10] Shihua Zhang, Qingjiao Li, Juan Liu, and Xianghong Jasmine Zhou. A novel computational framework for simultaneous integration of multiple types of genomic data to identify microrna-gene regulatory modules. *Bioinformatics*, 27(13):i401–9, 7 2011.