Fast Nonlinear Regression via Eigenimages Applied to Galactic Morphology

Brigham Anderson
Robotics Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
brigham@cmu.edu

Andrew Moore
Robotics Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
awm@cs.cmu.edu

Andrew Connolly
Astrophysics Dept.
University of Pittsburgh
Pittsburgh, PA 15213
ajc@phyast.pitt.edu

Robert Nichol
Inst. of Cosmology and Gravitation
University of Portsmouth
Portsmouth, UK
bob.nichol@port.ac.uk

ABSTRACT

Astronomy increasingly faces the issue of massive, unwieldly data sets. The Sloan Digital Sky Survey (SDSS) [11] has so far generated tens of millions of images of distant galaxies, of which only a tiny fraction have been morphologically classified. Morphological classification in this context is achieved by fitting a parametric model of galaxy shape to a galaxy image. This is a nonlinear regression problem, whose challenges are threefold, 1) blurring of the image caused by atmosphere and mirror imperfections, 2) large numbers of local minima, and 3) massive data sets.

Our strategy is to use the eigenimages of the parametric model to form a new feature space, and then to map both target image and the model parameters into this feature space. In this low-dimensional space we search for the best image-to-parameter match. To search the space, we sample it by creating a database of many random parameter vectors (prototypes) and mapping them into the feature space. The search problem then becomes one of finding the best prototype match, so the fitting process a nearest-neighbor search.

In addition to the savings realized by decomposing the original space into an eigenspace, we can use the fact that the model is a linear sum of functions to reduce the prototypes further: the only prototypes stored are the components of the model function. A modified form of nearest neighbor is used to search among them.

Additional complications arise in the form of missing data and heteroscedasticity, both of which are addressed with weighted linear regression. Compared to existing techniques, speed-ups achieved are between 2 and 3 orders of magnitude. This should enable the analysis of the entire SDSS dataset.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

Categories and Subject Descriptors

I.5 [**Computing Methodologies Pattern Recognition**]: Implementation

General Terms

Algorithms

Keywords

Astronomy,Morphology,Nearest Neighbor,Regression,Pincipal Component Analysis

1. INTRODUCTION

In order to understand the formation of large scale structures in the universe, it is necessary to understand the varied galaxy morphologies. This is still an open area of research in astronomy; it is not precisely known how galaxy shapes arise. The distribution of shapes and their correlation with other measured properties of galaxies is important to generating and testing hypotheses about the nature of the universe. This requires extracting various types of information from large numbers of faint and noisy images of galaxies, e.,g., whether the galaxy is spherical, elliptical, or disk-shaped, the size of the central bulge relative to the size of the disk, etc. Example images are given in Figure 1.

Figure 2 illustrates a common classification system still in use today: the Hubble "tuning fork" [3], which roughly divides all galaxies into ellipticals (bulge-dominated) and spirals (disk-dominated). The model parameter that corresponds to this is the bulge-to-disk ratio, which describes the relative amounts of light emanating from the two components. Hence a coarse galaxy classification of can be made directly from a single structural parameter, which can be obtained by fitting a mathematical model to a galaxy image. In this case, the model is an additive combination of a disk image, a bulge image, and a background (sky) image. The form of this model will be given in Section 2.1.

There are significant obstacles, however, to fitting these models. The most challenging is the Point Spread Function (PSF). Images of galaxies from ground-based telescopes are smeared by a turbulent atmosphere, and distorted by lens imperfections (telescope, fil-

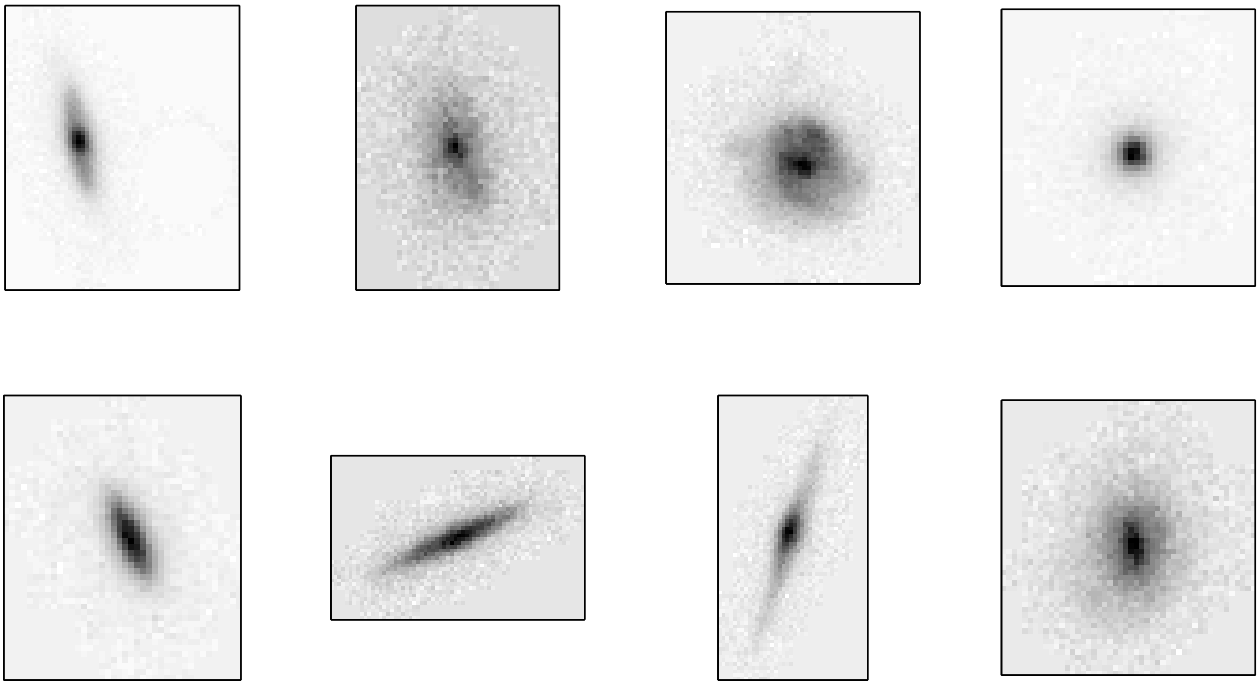

Figure 1: Galaxy images taken from the Sloan Digital Sky Survey

ters, mirrors, lenses, etc.) Figure 4 illustrates the effect of the PSF on a disk image. To understand the mechanism of the PSF, imagine a single ray of light coming from the direction of the galaxy and aimed at the center pixel/detector of the telescope. Without distortion, the resulting image would be a single point. However, as the ray travels through the atmosphere and the telescope's optics, it spreads out and is distorted before it hits the center pixel/detector. The resulting image is a PSF.[1] The PSF can be viewed as the probability mass function for the arrival of a single photon at a given pixel, given that the photon was initially aimed at the center pixel. The problem is that every incoming photon is subjected to the influences summarized in the PSF, so the resulting image is smeared. Section 2.1.2 describes the action of the PSF.

Large numbers of local minima are another problem. The noise of the images and a sometimes too-flexible model make finding the correct fit difficult. For example, some disks can be well approximated by a combination of bulge and sky. In the presence of noise, the two possibilities can be impossible to distinguish. In order to enumerate these local minima, some form of global search is generally necessary, and this is quite time-consuming. The most trusted of the current 2-d morphology techniques is a simulated annealing algorithm [9], which is robust to local minima, but is slower due to its caution. The algorithm described in this paper is able to sample the parameter space with on order of 50,000 samples, and is therefore probably more robust to local minima.

Currently standard nonlinear regression techniques are used to fit these images, such as simulated annealing [9] and Levenberg-Marquardt [7]. These approaches are all effective, but time consuming, e.g., roughly 1-3 minutes per 64×64 image on a 1.4 GHz pentium desktop. Assuming this, 100 million galaxies would re-

[1]The Point Spread Function is more generally a smooth two-dimensional function. We discretize it into a finite-resolution image.

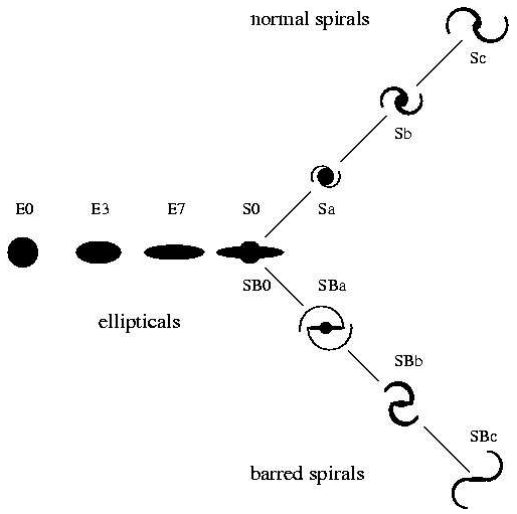

Figure 2: Hubble tuning fork diagram. The fundamental division of galaxies is into ellipticals (E) and spirals (S). The number after the ellipticals is the ratio of their major axis to their minor axis, called the ellipticity. The total light from the central bulge relative to that from the disk (the bulge-to-disk ratio) diminishes from left to right.

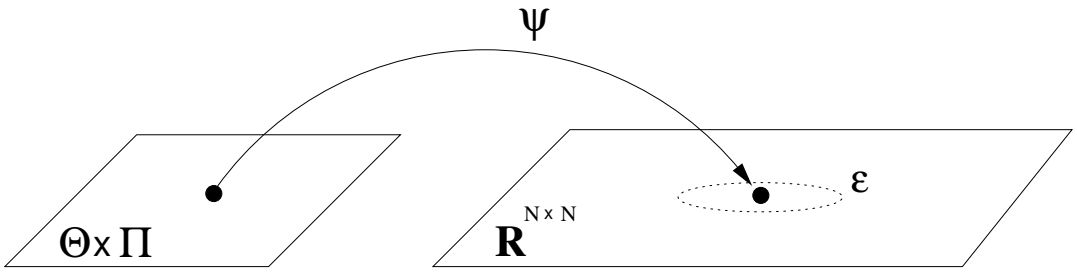

Figure 3: The generative model. Parameters from the space $\Theta \times \Pi$ are mapped into image space, $\mathbb{R}^{N \times N}$ by ψ. The error model ε describes the variance in each of the N^2 pixels/dimensions of the resulting image.

quire about 200 years of CPU time. For higher resolution images, performance rapidly degrades. The code introduced in this paper performs the same fits in less than a second, and is robust to changes in resolution of the target image.

Other machine learning approaches to similar problems have included the use of EM in classifying certain "bent-double" galaxies [5], the application of the Information Bottleneck method to classification of galaxy spectra [10], and the use of artificial neural networks in classifying galaxies along a single "galaxy-type" dimension [1].

2. PROBLEM

The general task is to fit a parametric model to data. In this case the data is an image of a galaxy. In the galactic morphology (GM) task, the model is a function whose 12 free parameters are the morphological characteristics of the galaxy, e.g., shape, size, and location (see Appendix for complete list.) This task is described in detail in [9]. We assume all images are square with N pixels per side, giving a total of N^2 pixels. Occasionally images must be represented as vectors, so they are denoted with an overhead arrow ($\vec{\cdot}$). Images can be turned into vectors by vertically concatenating the columns.

2.1 Generative Model

The most basic assumption that we make is that the target image can be modeled. Here we use a model consisting of a set of four main elements: $\{\psi, \Theta, \Pi, \varepsilon\}$, which describes the expected number of photons to arrive at a particular pixel ij in a matrix of pixels. The ultimate objective is to invert this model for each target image.

- ψ is a function that maps parameters onto matrices as in Figure 3. The matrix's elements correspond to the pixels in an image and/or to the detectors in a telescope. ψ is the sum of c component functions:

$$\psi(\theta, \pi) = \sum_{k=1}^{c} \psi^k(\theta^k, \pi) \tag{1}$$

  In the galactic morphology task, ψ has three components: a disk, a bulge, and constant background. This function is described in greater detail in Section 2.1.1 and in the Appendix.

- Θ is the space of possible values for the θ parameter of ψ. In the galactic morphology task, these parameters are disk flux, disk angle, disk inclination, bulge flux, etc. (see Appendix.) These are the parameters sought by the regression.

- Π is the space of possible values for the *fixed* π parameter of ψ. Unlike Θ, these parameters are not fitted. In the galactic morphology problem, π is the PSF. The space Π contains all the PSFs that could occur.

- ε is a noise model. For the GM task, we assume additive, zero-mean Gaussian noise.[2] The noise can also be heteroscedastic, i.e., variance can differ between pixels. For clarity, the ε component may sometimes be omitted in the text.

The elements of the model described in this section are illustrated in Figure 3. Each pixel has an independent Gaussian distribution, so the target image y is assumed to have the distribution

$$y_{ij} \sim N\left(\psi_{ij}(\theta, \pi), \varepsilon_{ij}\right) \tag{2}$$

where the distribution for each pixel value is Gaussian with a mean of $\psi_{ij}(\theta, \pi)$ and a variance of ε_{ij}.

2.1.1 The Function ψ

The model f is a function on a 2-dimensional plane which indicates the density of flux at a given point on the plane. The general shape of the function is a sharp peak at the center of the galaxy, tapering off with distance. The disk tapers off exponentially with respect to distance from the center of the galaxy, and the bulge tapers off exponentially w.r.t. the cube root of the distance. The appendix contains the details of the function and its motivations. Importantly, the model is not smooth and has no derivatives at $(0,0)$. There is a single sharp spike at this location, which creates difficulties later on when trying to deconvolve ψ and π.

2.1.2 The Fixed Parameter(s) π

In the GM task, the relationship between the fixed parameter π (the PSF) and the model ψ is one of convolution.

$$\psi(\theta, \pi) = \pi \star \psi(\theta, \Delta) \tag{3}$$

Where Δ is an image with a single delta function, which indicates no blurring; convolution with Δ results in a perfect replication of the original image. The effect of convolution is illustrated in Figure 4.

Since π is an image of the PSF, convolving an image with a particular π is equivalent to blurring with a particular atmospheric condition and/or mirror imperfection. Fortunately, convolution is a linear operation, since each pixel becomes a linear combination of all other pixels. Hence, Equation 3 can be rewritten as

$$\psi(\theta, \pi) = \pi \star \psi(\theta, \Delta) = \sum_{k=1}^{c} \pi \star \psi^k(\theta^k, \Delta) \tag{4}$$

Due to the physical interpretation of π, all the pixels of the PSF must be nonnegative. Also, conservation of energy requires that the pixels sum to one.

[2] As an approximation to the true noise distribution, which is Poisson.

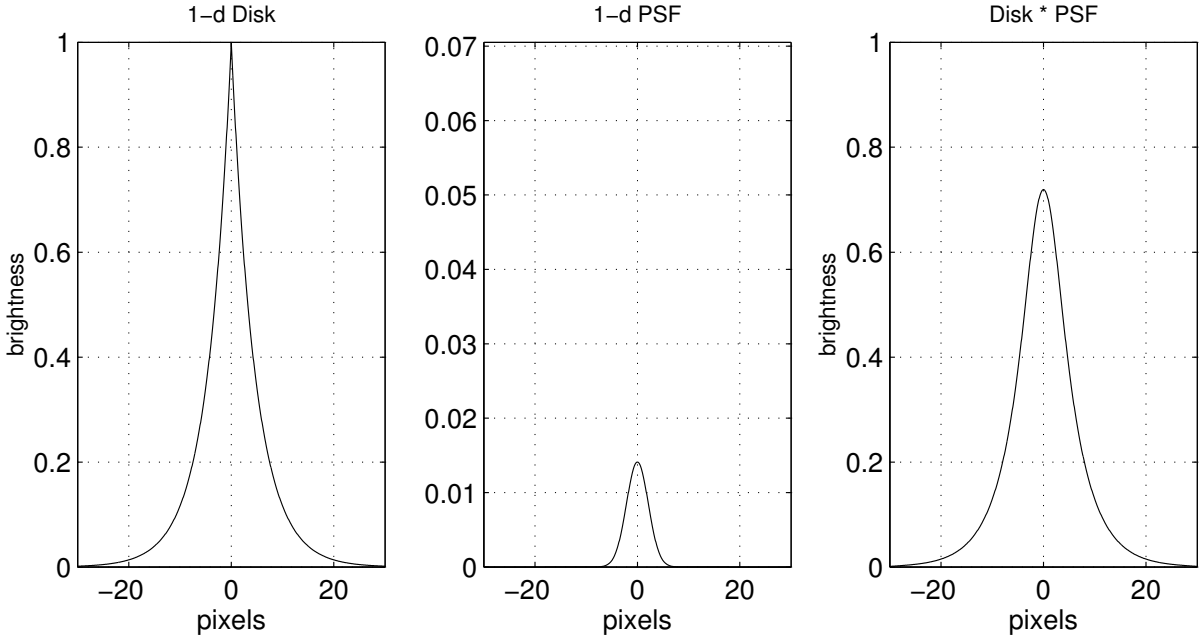

Figure 4: Example of effect of convolution with PSF. The far left figure is a one-dimensional brightness profile of a slice from a disk galaxy image. The middle image is an example of a one-dimensional PSF, in this case a Gaussian. The far right plot is the convolution of the disk profile by the PSF profile. Note that this operation results in a "blurring" of the original image.

2.1.3 The Noise Model ε

Although ψ is the expected number of photons to arrive during an exposure, the *realized* number will be a discrete counting process. So the value will follow a Poisson distribution with a mean and variance of $\psi(\theta,\pi)$. Because of this, error variance at a pixel is proportional to the amount of signal at the pixel. This heteroscedasticity must be accounted for in the regression.

We consider the Poisson to be well approximated in this case with a Gaussian distribution with a mean and variance of $\psi(\theta,\pi)$, since the number of photons is usually greater than 30 in the more influential central pixels.

The noise model can also be used to "mask" bad pixels in the input image. If the galaxy of interest is close to another galaxy, or artifacts are present in the image, then a mask is used to specify which pixels to ignore. Bad pixels can be masked out entirely by setting their ε values to infinity.

2.2 Objective

The objective is to invert ψ. Specifically, to take a given target image y, PSF π, and error model ε, and to find a parameter vector $\theta^* \in \Theta$ that, when fed to the generative model of Equation 1, produces an image $\hat{y}$ close to y. By 'close' we mean to minimize the distance between the two images. This distance function will be denoted by χ^2.

$$\chi^2(y,\hat{y},\varepsilon) = \sum_i^N \frac{(y_i-\hat{y}_i)^2}{\varepsilon_i^2} \tag{5}$$

This distance function is the least squares criterion with heteroscedastic noise. Assuming that the noise model ε is correct, its minimum will occur at the most likely fit.

3. NONPARAMETRIC FITTING

The algorithm must be able to quickly invert the galaxy image model and to deconvolve images that have been blurred by a PSF. The most successful algorithm type we have found has been a variant of the nearest neighbor algorithm. This approach creates a mapping from image space $\mathbb{R}^{N\times N}$ to parameter space Θ by remembering and generalizing from many previous Θ-to-$\mathbb{R}^{N\times N}$ mappings (via prototypes).

We are thus using a nonparametric technique to perform a parametric regression. Two reasons primarily motivate this choice: 1) the model is expensive to evaluate because the PSF convolution requires $O(N\log N)$ operations each time a model image is generated. Iterative techniques, e.g., Levenberg-Marquardt, need to evaluate the model at every step, so search becomes costly, and 2) the number of local minima is large, so most descent-based methods are inappropriate due to their vulnerability to local minima.

3.1 Brute Force Approach

For purposes of exposition, we will start with a naive, infeasible approach. The prototypes that will be used to map from images to parameters are the members of the set of prototypes $X = \{(x_i,\theta_i,\pi_i)\}_{i=1}^p$, where $x_i = \psi(\theta_i,\pi_i)$ and $|X|$ is the number of prototypes. We generate this set by sampling uniformly from Θ and Π.

The regression task in this context is to find θ^* by finding the smallest distance between the target image and each of the prototypes,

$$x^* \quad = \quad \mathsf{argmin}_{x\in X}\left[\chi^2(x,y,\varepsilon)\right] \tag{6}$$

where θ^* is the parameter vector corresponding to the prototype x^*. This is the core strategy of this regression algorithm. There are, however, clear barriers to overcome. First, the dimensionality of the prototype space is very high; one dimension per pixel for a 32×32 image gives 1024 dimensions. Comparing prototypes to incoming queries will thus be expensive. Second, the presence of the uncontrollable fixed parameters π means that a large number of prototypes will be required to adequately sample the parameter space

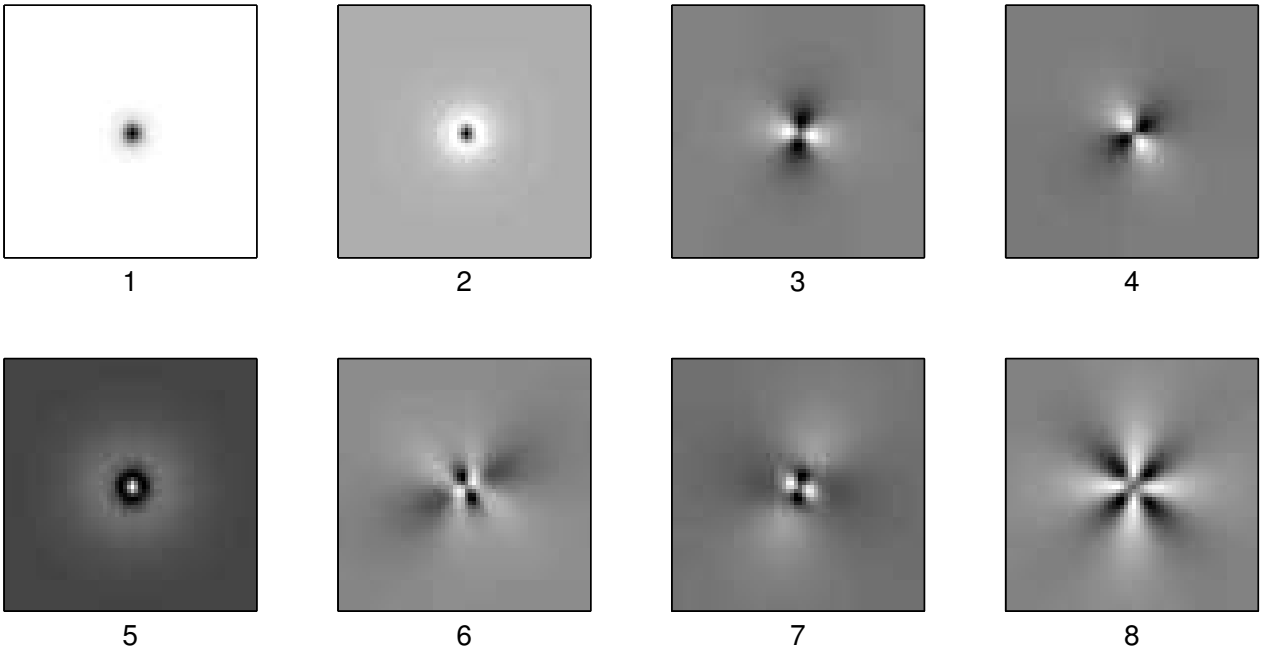

Figure 5: The first 8 eigengalaxies obtained from galaxies which have been convolved with a Gaussian PSF with 2-pixel standard deviation.

$\Theta \times \Pi$. However, nearest neighbor search can exploit three strategies, which account for most of the speed gains made, 1) Principal Components Analysis, 2) PSF-local Principal Component Analysis, and 3) prototype decomposition.

4. FEATURE SPACE CREATION

One problem with working in image space $\mathbb{R}^{N\times N}$ is the computational load of so many dimensions, i.e., one per pixel. Without noise, however, the model $\{\psi,\Theta,\Pi\}$ creates images that can *at most* occupy a manifold whose dimension is equal to the number of model parameters. Ideally, one should focus one's efforts in the most relevant subspace and ignore the rest, especially since nearest neighbor algorithms are sensitive to excessive dimensionality. Principal Components Analysis (PCA) achieves this by determining the linear subspace in which the most variance resides.

Each prototype is a point in $\mathbb{R}^{N\times N}$. To determine the best subspace of $\mathbb{R}^{N\times N}$, we would like to know the shape of the subspace that these prototypes occupy, which is the manifold $\{\psi(\theta,\pi)|\theta \in \Theta,\pi \in \Pi\}$. One measure of shape is the covariance between the values of each dimension/pixel of the model manifold.

$$UU^T = \int_\Theta \int_\Pi [\psi(\theta,\pi)-\mu]\,[\psi(\theta,\pi)-\mu]^T\, d\pi d\theta \qquad (7)$$

where μ is the mean model image over all $\theta \in \Theta$ and $\pi \in \Pi$. This is the pixel covariance matrix, UU^T, whose ij-th element is the covariance between pixel i and pixel j in the model space. Once the pixel covariance is known, the first K eigenvectors of UU^T form an "optimal" subspace of $\mathbb{R}^{N\times N}$. This subspace, out of all possible linear K-dimensional subspaces of $\mathbb{R}^{N\times N}$, explains the maximum variance possible. These eigenvectors are the first K principal components.

We must first estimate UU^T. This can be done efficiently by sampling the model manifold, i.e., by randomly choosing θs and πs from the parameter space $\Theta \times \Pi$ and generating images from them using the model $\{\psi,\Theta,\Pi\}$. These sample images are then mean-normalized[3], lined up as column vectors as the matrix U, and multiplied to produce UU^T.

In the PCA paradigm, the first $K \ll N^2$ eigenvectors of UU^T form the basis for a new space, which we will denote Φ. This basis Φ is an $N \times K$ orthonormal matrix. The span of the columns of Φ will be referred to as an eigenspace and the individual columns as eigenimages. See Figure 5 for eight sample eigengalaxies. Note that Φ^T is a projection matrix such that $\tilde{y} = \Phi^T y$. Vectors projected into eigenspace will be denoted by $\widetilde{\bullet}$ and are also vectors.

4.1 Projection into Feature Space

The entire nearest neighbor search should now take place within the eigenspace Φ. This requires projecting all of the prototype images and all target images into Φ. Since the eigenvectors are orthogonal, projection into the eigenspace is straightforward:

$$\tilde{y} = \Phi^T y \qquad (8)$$

However, if the noise is heteroscedastic, the projection requires more care. Different pixels/dimensions will be weighted differently by the distance function χ^2, so distances between images will behave as if the image space has been warped. The stretching will be axis-aligned in image space, because each pixel's noise is independent of the others'.

If noise is heteroscedastic, the optimal projection is a weighted linear regression. The diagonal matrix Σ_y is the covariance of y in $\mathbb{R}^{N\times N}$. The matrix Σ_y is just a reorganization of exactly the same information contained in ε.

$$\mathsf{diag}(\Sigma_y) = \overrightarrow{\varepsilon} \qquad (9)$$

[3]The mean of all sample images is subtracted from each image, as in Equation 7.

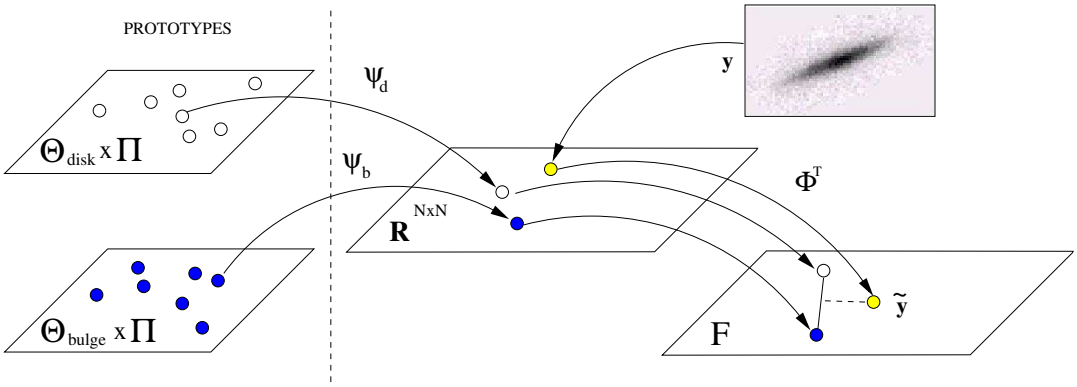

Figure 6: Fitting image y. The empty circle is a disk component and the filled circle is a bulge component. The fitting is done in feature space F. The intersection of the dashed line and the solid line in F is the best-fitting linear combination of the bulge and disk to $\tilde{y}$. This process is repeated for multiple combinations of bulge and disk.

The projection of y into Φ is

$$\tilde{y} = (\Phi^T \Sigma_y^{-1} \Phi)^{-1} \Sigma_y^{-1} \Phi^T y \tag{10}$$

and the resulting error covariance of $\tilde{y}$ is

$$\Sigma_{\tilde{y}} = (\Phi^T \Sigma_y^{-1} \Phi)^{-1} \tag{11}$$

The matrix $\Sigma_{\tilde{y}}$ is the covariance matrix of $\tilde{y}$, reflecting any uncertainty in the eigenspace projection of y. In contrast to Σ_y, the covariance $\Sigma_{\tilde{y}}$ usually has off-diagonal terms because the noise is not axis-aligned with respect to the basis Φ. Thus $\tilde{y}$ will most likely have correlated errors due to the projection.

4.2 Nearest Neighbor in Feature Space

Now we can attack the regression problem while inside the eigenspace, where we enjoy a much reduced dimensionality. The algorithm uses only the eigencoordinates of the prototypes, denoted $\tilde{X}$. The algorithm becomes slightly more complicated since the χ^2 distance function must now account for any correlated errors in $\tilde{y}$ introduced by projection of y onto Φ. The new algorithm:

$$\tilde{x}^* \quad = \quad \text{argmin}_{\tilde{x} \in \tilde{X}} \left[(\tilde{x} - \tilde{y})^T \Sigma_{\tilde{y}}^{-1} (\tilde{x} - \tilde{y}) \right] \tag{12}$$

where $\tilde{x}$ and $\tilde{y}$ are the eigencoordinates of x and y respectively. The fitted parameter vector θ^* is the θ corresponding to $\tilde{x}^*$.

5. PSF-LOCAL FEATURES

Up to this point we have included all possible PSFs in our PCA. The model manifold $\{\psi(\theta,\pi) \mid \theta \in \Theta, \pi \in \Pi\}$ has relatively few dimensions, [4] but it is highly nonlinear with respect to π. This means that the number of eigenspace dimensions required to represent it adequately will be larger. There are at least two approaches to this problem: attempting to remove the effect of π and PSF-local PCA.

The most direct approach is to modify y to remove the effect of the PSF π. This can be accomplished via deconvolution. Unfortunately, deconvolution has difficulty with regions of the image with sudden changes in intensity, which is the part of the image with the most information relevent to our model. The central spike (which is the galaxy center) is a discontinuity that is very difficult for deconvolution to reconstruct. Also, deconvolution suffers from instability in the presence of noise.

The next approach is PSF-local PCA. This maintains a different Φ for every PSF. Each eigenspace is obtained by fixing π and repeating the steps of PCA in Section 4. Each eigenspace is optimal for its π. The number of dimensions required is therefore quite small, 20 dimensions captures over 99.999% of the variance.

It is feasible to reuse a Φ generated from a single PSF π because most PSFs in a given run of galaxy images are similar, and because small differences between PSFs generally produce small differences in resulting images.

The algorithm stores a relatively small number n_s of PSF-specific eigenspaces $\{(\pi_i, \Phi_i)\}_{i=1}^{n_s}$. The initial PSF population consists of a few Gaussians of varying standard deviation, to which PSFs are added during on-line operation. The decision to add a PSF to the database is made, somewhat arbitrarily, when an incoming PSF differs by more than 0.02 in variance explained from its closest match in the database. Variance explained here is $1 - \Sigma_i^N (\pi_i - \pi_i')^2 / \Sigma_i \pi_i'^2$. Where π' is a PSF from the existing database.

6. SPLITTING PROTOTYPES

The fact that the model is of the form $\psi(\theta,\pi) = \sum_k^c \psi^k(\theta^k,\pi)$ can be used to good advantage; only prototypes for the component $\psi^k(\theta^k,\pi)$ images need to be created. For example, in the GM task the component functions are the disk, bulge, and background. Instead of generating and storing large numbers of individual combinations of disks and bulges to form the set X, we can store three much smaller sets of prototypes: a disk set X_d, a bulge set X_b, and a sky set X_{sky}. Far fewer prototypes will be needed to represent the same number of images. The size of the representable number of prototypes is now $|X_d| \times |X_b| \times |X_{sky}|$, but only $|X_d| + |X_b| + |X_{sky}|$ images need be created.

The feature vectors of the prototypes are decomposable as well because

$$\tilde{\psi}(\theta,\pi) = \Phi^T \vec{\psi}(\theta,\pi) = \sum_{k=1}^{c} \Phi^T \vec{\psi}^k(\theta^k,\pi) \tag{13}$$

The same is true in the heteroscedastic case; since projection is still achieved with a matrix operation. The matrix in question is the product of the matrices which are multiplied by y in the right hand side of Equation 10.

Nearest neighbor algorithms require some notion of distance,

[4] We assume that the class of possible PSFs is constrained to lie in some manifold with dimension much less than $\mathbb{R}^{N \times N}$. E.g., images of trees, human faces, white noise, etc. are not in Π.

Algorithm	Strategy	Speed
GIM2d	Simulated Annealing	~360 sec
Galfit	Levenberg-Marquadt	~30 sec
GMORPH	Instance-Based	~1 sec
1-d approaches (biased)	Descent	<1 sec

Table 1: Comparison of the different strategies and speeds of existing algorithms for the galaxy morphology task.

and distances are typically defined between two points. However, since we are combining components of prototypes we must define a distance metric between a particular set of components $\{\tilde{x}^0, \tilde{x}^1, \tilde{x}^2, ..., \tilde{x}^c\}$ and a target image $\tilde{y}$. In the GM task, this would be finding the distance between a galaxy image and, for instance, bulge #55 with disk #1244. The c components will define a (hyper)plane of possible images which consists of all non-negative linear combinations of the c components.

Given our definition of χ^2, the error-minimizing metric is the distance to the nearest point on that plane. This distance χ^2 is calculated via weighted linear regression:

$$Z = \left[\tilde{x}^0\ \tilde{x}^1\ \tilde{x}^2\ ...\ \tilde{x}^c\right] \quad (14)$$

$$\beta = (Z^T \Sigma_{\tilde{y}}^{-1} Z)^{-1} \Sigma_{\tilde{y}}^{-1} Z^T \tilde{y} \quad (15)$$

$$\chi^2 = (Z^T \beta - \tilde{y})^T \Sigma_{\tilde{y}}^{-1} (Z^T \beta - \tilde{y}) \quad (16)$$

Importantly, this linear regression also determines the optimal linear combination of the particular components $\{\tilde{x}^0, \tilde{x}^1, \tilde{x}^2, ..., \tilde{x}^c\}$ used in Equation 14 for the particular target $\tilde{y}$. The optimal coefficients are the elements of the vector β. Having thus determined c of the parameters of the model via the relatively cheap operation of a linear regression, the remaining dimension of the search space is reduced by c. In the case of the GM problem, the three coefficients of β are the total fluxes of disk, bulge, and background. Figure 6 illustrates the procedure for just a bulge and a disk component.

At this point, we in principle have only to calculate χ^2 for all combinations of disks and bulges, and select the combination with the smallest χ^2. Unfortunately, speed would then be unacceptably compromised, so instead we search selectively.

7. NEAREST NEIGHBOR SEARCH

After the eigenspace has been selected and the target image has been projected into the space, then the search for a nearest neighbor begins. The search could be accomplished by an exhaustive search of all bulge-disk combinations. However, we save time with the following two-part search algorithm which has global and a local search components:

1. **Global: Random Pair Sampling** starts by extensively randomly sampling a large number of disk/bulge pairs from $\tilde{X}_d$ and $\tilde{X}_b$ (the sky $\tilde{X}_{sky}$ is held fixed as a constant.) Each pair is fit to y via weighted linear regression as in Equations 14-16. We typically are able to sample 50,000 pairs, which is an unusually dense covering of the parameter space for this particular problem.

2. **Local: Iterative search** starts with the best candidate from phase 1. The bulge-related parameters are held fixed while the bulge component is then paired with all disk prototypes from $\tilde{X}_d$ and a χ^2 is calculated for each combination. The best combination becomes the new start point for another 'step'. Now the disk is fixed while $\tilde{X}_b$ is searched for a better bulge. The process continues in this manner until no improvement results. To evaluate each combination, χ^2 is obtained by weighted linear regression as in Equation 16. which also determines the optimal linear combination coefficients, β, of the prototypes for that particular target $\tilde{y}$.

The process is guaranteed to converge because the search space is finite, and the sequence of pairs must always have a decreasing χ^2. Phase 2 is run on the top 10 or 20 candidates from phase 1. We have found the local search as described to generally converge to a better minimum than simple local (e.g., hillclimbing) search. We conjecture that this is due to the large number of local minima inherent in the problem.

8. RESULTS

Table 1 summarizes the strongest difference between this algorithm and its predecessors, which is speed. Implemented in MATLAB [4], GMORPH can analyze a 64×64 image in approximately 1 second. The nearest competitor can do the same image in about 30 seconds, but it is a descent method and vulnerable to local minima. The times were obtained by generating random galaxy parameters from the range $F_d \in [0,1], F_b \in [0,1], \mu_x = 0, \mu_y = 0, r_d \in (0,16], \gamma_{inc} \in [0°, 85°], \gamma_d \in [0°, 180°], r_e \in (0,16], \varepsilon \in [0, 0.7]$, and $\gamma_b \in [0°, 180°]$, and were used to generate 64×64 images of galaxies. The PSFs were Gaussian with a standard deviation of 2 pixels.

Figure 7 contains the results of a comparison between GMORPH and the traditional and currently most-trusted measure of galaxy shape: human classification. We tested the agreement between GMORPH and an already-classified dataset with 300 galaxies. Each image had been classified visually by a panel of four human experts onto a scale which varies from 0 (all bulge) to 5 (all disk), with 6 being 'irregular'. The results show a clear correlation between GMORPH and expert classification.

Figure 8 plots the agreement between GMORPH and GIM2d [9] on the disk radius for low-noise, predominantly disk galaxy images from the Sloan Digital Sky Survey. Both algorithms were run on 100 images, each with a unique PSF. The catalog of prototypes used by GMORPH had $|X_{disk}| = 1000$, $|X_{bulge}| = 1000$, and $|X_{sky}| = 1$. The size of the images varied, but were approximately 50×50. The agreement between the two methods is apparent here, however, in high-noise images the two methods produce different results. Although we are still investigating the source of these occasional discrepencies, there is preliminary evidence that these are cases in which either the galaxy morphology diverges from the assumed bulge/disk model, or noise is too severe to fit the data with confidence.

9. CONCLUSIONS

We report on an ongoing investigation of a nonlinear regression problem from astronomy: given a massive dataset of noisy, distorted images of unknown galaxies, rapidly fit a nonlinear model to each image in the dataset. A instance-based method for accomplishing this task has been described.

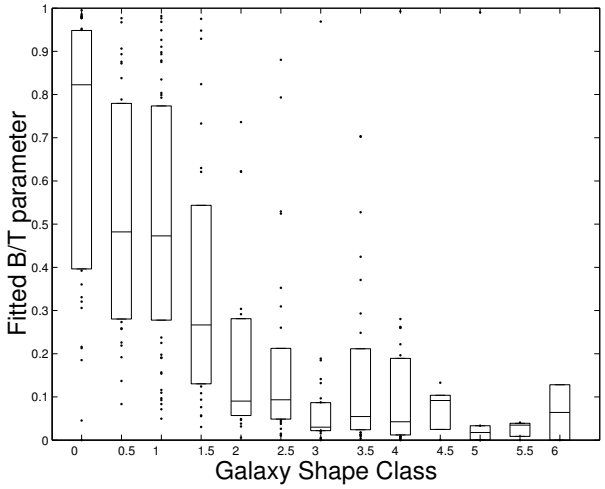

Figure 7: Comparison to human expert classification of 300 galaxies. The horizontal axis is the galaxy classification, which varies from 0 (all bulge) to 5 (all disk), with 6 being 'irregular'. The vertical axis is the bulge-to-total flux ratio returned by GMORPH. Each box indicates the 25th, 50th, and 75th quartiles.

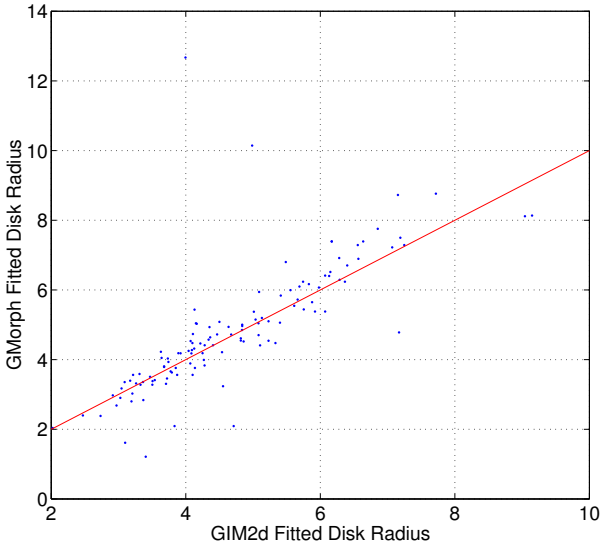

Figure 8: Noise and PSF effect on recovered disk radius error. The agreement between disk radius parameters fitted by GIM2d and those fitted by GMORPH on images from the Sloan Digital Sky Survey.

Instance-based methods allow for very fast identification of these galaxies through sampling of the parameter space, the use of an eigenspace, and through splitting the prototypes into components. GMORPH can avoid the expense of calculating the PSF during the search process, and can scan through the space of galaxy images rapidly because it restricts search to the much smaller subspace determined by PCA. We have measured the performance via simulation and it should in principle allow for unprecedented analysis of astronomical datasets of galaxy images.

10. REFERENCES

[1] N. Ball, M. Fukugita, O. Nakamura, S. Okamura, J. Brinkmann, and R. J. Brunner. Galaxy Types in the Sloan Digital Sky Survey Using Supervised Artificial Neural Networks. *Mon.Not.Roy.Astron.Soc.*, 348:1038, 2004.

[2] K. C. Freeman. On the Disks of Spiral and s0 Galaxies. *Astrophysical Journal*, 160:811, 1970.

[3] E. Hubble. *The realm of the nebulae*. Yale University Press, 1936.

[4] Mathworks Inc. Matlab, version 6, release 13, 2003.

[5] S. Kirshner, I. Cadez, P. Smyth, and C. Kamath. Learning to classify galaxy shapes using the EM algorithm. In *Advances in Neural Information Processing Systems*, volume 15. Morgan Kaufmann, 2002.

[6] J. Kormendy. Brightness distributions in compact and normal galaxies. III - Decomposition of observed profiles into spheroid and disk components. *Astrophysical Journal*, 217:406–419, 1977.

[7] C. Y. Peng, L. C. Ho, C. D. Impey, and H. Rix. Detailed Structural Decomposition of Galaxy Images. *Astronomical Journal*, 124:266–293, 2002.

[8] K. Ratnatunga, R. Griffiths, and E. Ostrander. Disk And Bulge Morphology Of Wfpc2 Galaxies: The Hst Medium Deep Survey Database. *accepted to Astronomical Journal*, 1999.

[9] Luc Simard, Christopher N. A. Willmer, Nicole P. Vogt, Vicki L. Sarajedini, Andrew C. Phillips, Benjamin J. Weiner, David C. Koo, Myungshin Im, Garth D. Illingworth, and S. M. Faber. The Deep Groth Strip Survey II. Hubble Space Telescope Structural Parameters of Galaxies in the Groth Strip. *Astrophysical Journal Supplements*, 142(1), 2002.

[10] N. Slonim, R. Somerville, N. Tishby, and O. Lahav. Objective Classification of Galaxy Spectra using the Information Bottleneck Method. *Monthly Notices to the Royal Astronomical Society*, 323, 2001.

[11] D. G. York, J. Adelman, J.E. Anderson, and et al. The Sloan Digital Sky Survey: Technical Summary. *Astronomical Journal*, 120:1579–1587, 2000.

APPENDIX

A. SURFACE BRIGHTNESS FUNCTION

Before being blurred by the PSF, the galaxy is created by the surface brightness function, ψ, which takes as an argument a vector from Θ. Here are the 12 model parameters of Θ, a brief description, and their units:

F_b, F_d total integrated flux of bulge and disk components ($erg \cdot cm^2/sec$)

μ_x, μ_y the x and y offset of the galactic center from the center of the image ($pixels$)

r_e, r_d bulge and disk scale lengths ($pixels$)

ε_b apparent bulge ellipticity (*unitless*)

γ_{inc} disk inclination (*degrees*). Rotation toward viewer

γ_b γ_d bulge and disk angle of rotation (*degrees*). Clockwise rotation relative to viewer

sky sky background offset ($flux/cm^2$)

Sersic a bulge shape parameter that is fixed to the value 4 for all experiments

The classic model of galaxies has been additive: a linear combination of a bulge image, a disk image, and a sky (background) image [9, 7, 8]. The sky image is a constant, and will omitted from the formulae for clarity.

Before discretization into pixels, g is a continuous brightness function defined over the 2-dimensional image plane uv.[5] The expected number of photons to hit a pixel/detector is found by integrating g over the uv area of the pixel on the plane.

The function g is the sum of c component functions:

$$g(u,v,\theta,\pi) = \sum_{k=1}^{c} g^k(u,v,\theta^k,\pi) \tag{17}$$

In the galactic morphology task, g has three components: a disk, a bulge, and constant background. We will assume henceforth that the PSF is a delta function, so we omit π from the discussion. We refer to the entire unblurred disk function as $g_{disk}(\theta)$, and the unblurred bulge image as $g_{bulge}(\theta)$.

The surface brightness, g_{disk}, of a pure disk galaxy w.r.t. radius has been found to have an exponential form [2, 6]. A commonly used model consists of an infinitely thin disk with brightness in the plane of the disk tapering off exponentially away from the center. When projected onto the image plane, the brightness g_{disk} has the form

$$g_{disk}(u,v) \propto F_d exp\left(-\frac{\sqrt{x^2+y^2\cos^{-2}\gamma_{inc}}}{r_d}\right) \tag{18}$$

where F_d is the integrated brightness of the disk, γ_{inc} is the degree of inclination of the disk towards the viewer, and r_d is the disk "radius", or scale parameter. Both Equation 18 and Equation 19 are simplified for presentation in that they omit clockwise rotation and fix the center of the galaxy at $(0,0)$.

The bulge is modeled with a classical de Vaucouleurs profile. Also known as the $r^{1/4}$ law, de Vaucouleurs' law is perhaps the most widely used empirical law to describe the surface brightness profile of a pure bulge galaxy. The bulge brightness is

$$g_{bulge}(u,v) \propto F_b exp\left(-b\left[\frac{\sqrt{x^2+y^2(1-\varepsilon_b)^{-2}}}{r_b}\right]^{\frac{1}{4}}\right) \tag{19}$$

where F_d is the integrated brightness of the bulge, ε_b describes the ellipticity of the bulge, and r_b is the bulge "radius".

The actual image recorded by the telescope is digitized into pixels. Pixels are elements of the matrix ψ.

$$\psi_{ij}(\theta,\pi) = \int_i^{i+1}\int_j^{j+1} g(u,v,\theta,\pi)dudv \tag{20}$$

[5]The variables u and v are used only because x and y appear elsewhere in this paper.