Landmarking Manifolds with Gaussian Processes

Dawen Liang DLIANG@EE.COLUMBIA.EDU
John Paisley JPAISLEY@COLUMBIA.EDU
Department of Electrical Engineering, Columbia University, New York, NY, USA

Abstract

We present an algorithm for finding landmarks along a manifold. These landmarks provide a small set of locations spaced out along the manifold such that they capture the low-dimensional nonlinear structure of the data embedded in the high-dimensional space. The approach does not select points directly from the dataset, but instead we optimize each landmark by moving along the continuous manifold space (as approximated by the data) according to the gradient of an objective function. We borrow ideas from active learning with Gaussian processes to define the objective, which has the property that a new landmark is "repelled" by those currently selected, allowing for exploration of the manifold. We derive a stochastic algorithm for learning with large datasets and show results on several datasets, including the Million Song Dataset and articles from the New York Times.

1. Introduction

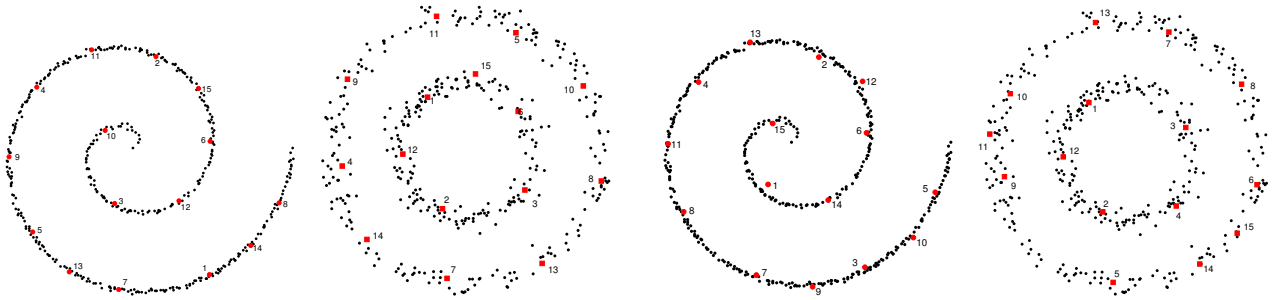
In data analysis problems, a typical goal is to learn the underlying structure of a dataset, whether it be its statistical properties, latent patterns or the shape of the data itself. For example, Bayesian methods hypothesize a generative model for the data in which latent variables capture information about certain structural properties assumed to exist *a priori*. One common property is that the data lies on or near a manifold, which is a nonlinear, low-dimensional space embedded in a high-dimensional ambient space. For example, images of faces or normalized word histograms of documents may have several thousand dimensions, but be restricted in the way they vary within that high-dimensional space based on the intrinsic properties of the data-generating processes.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

In this paper we consider the problem of landmarking manifolds; that is, finding a subset of locations evenly spaced along a manifold that captures its low-dimensional, nonlinear characteristics. Learning the overall structure of a manifold by focusing on local information has many uses, whether it is learning a low-dimensional embedding of the data (Roweis & Saul, 2000; Tenenbaum et al., 2000; Ng et al., 2001), finding relevant observations from the dataset for supervised learning (Cortes & Vapnik, 1995; Tipping, 2001) or unsupervised learning (Silva et al., 2005; Li & Hao, 2009; Cai & He, 2012; Vladymyrov & Carreira-Perpinán, 2013), or for active learning problems (Kapoor et al., 2007; Paisley et al., 2010; Li et al., 2014).

Previous manifold landmarking approaches focus on selecting a subset of points from within the dataset that characterizes the manifold (Silva et al., 2005; Li & Hao, 2009; Vladymyrov & Carreira-Perpinán, 2013). Related supervised approaches such as sparse regression models (Tipping, 2001) implicitly do so as well. Such approaches assume that the dataset provides a densely sampled representation of the manifold, which may not always be the case. Furthermore, these methods typically require either a full kernel constructed from pairwise distances, or the evaluation of a function using all data points, both of which can be computationally prohibitive as the size of the dataset grows.

We present an unsupervised method for finding points along the space of a manifold that does not encounter these issues. Our approach learns landmarks that can fall anywhere in the continuous ambient space, but will lie along the manifold as approximated by the noisy data. The approach greedily learns these locations by optimizing a sequence of objective functions that naturally "repels" each new location from the previously selected ones. The objective function is motivated by a simple active learning method using Gaussian processes (Cohn et al., 1996; Rasmussen, 2006; Kapoor et al., 2007). This supervised method selects the next location to measure according to the level of uncertainty in the predicted response. Interestingly, previous measurements are not used to determine this and so the selection process itself is unsupervised. We modify this algorithm to efficiently explore manifolds.



(a) Active learning (b) Manifold landmarking

Figure 1. Landmarking and active learning with two toy manifolds. (a) The first 15 points selected by the active learning procedure in Section 2. (b) The first 15 points selected using the continuous ambient space landmarking approach of Section 3. The landmarks in (b) do not correspond exactly to any observation, but instead converged to these locations using gradient methods.

In the remainder of the paper, in Section 2 we review active learning with Gaussian processes for supervised learning problems and motivate this as a good approach for unsupervised manifold learning as well. In Section 3 we use this as a starting point for developing our landmarking algorithm. We present a stochastic inference algorithm for finding these landmarks with large datasets in Section 4 using a specific kernel mapping. In Section 5 we evaluate our method on image datasets, the Million Song Dataset, and 1.8 million articles from the New York Times.

2. Active learning with Gaussian processes

Our unsupervised method uses ideas from active learning with Gaussian processes as a starting point and so we briefly review this approach. Assume we have a dataset $(x_1, y_1), \ldots, (x_n, y_n)$, where y is a response associated with location $x \in \mathbb{R}^d$. We also have a set $\mathcal{D} = \{x\}$ of locations without the corresponding responses y. Active learning seeks to pick the next location $x_{n+1} \in \mathcal{D}$ for which to query y_{n+1} such that a large amount of information is gained according to some measure. In the Gaussian process regression setting, where y is a real-valued number (possibly latent), this can be done by selecting x_{n+1} for which the uncertainty of y_{n+1} is greatest as measured by the variance of y_{n+1}.

We recall that $y(x)$ is a Gaussian process (GP) (Rasmussen, 2006) if all marginals evaluated at a finite set of locations are multivariate Gaussian distributed. A GP is defined by a mean function $m(x) = \mathbb{E}[y(x)]$ and covariance function $k(x, x') = \mathbb{E}[(y(x) - m(x))(y(x') - m(x'))]$, also called a kernel. Assuming a Gaussian process, the vector $y = [y(x_1), \ldots, y(x_n)]^T$ discussed above is therefore distributed as

$$y \sim \mathcal{N}(m, K), \tag{1}$$

with $m = [m(x_1), \ldots, m(x_n)]^T$ and $K_{ij} = k(x_i, x_j)$. We will assume that $m(x) = 0$ in this paper.

Let the set $\mathcal{D}_n$ contain the first n measured locations, $x_1, \ldots, x_n$, and K_n be the kernel matrix constructed from points in $\mathcal{D}_n$. Given $(x_1, y_1), \ldots, (x_n, y_n)$, the value $y(x)$ at a new x is distributed as

$$\begin{aligned} y(x) \,|\, y &\sim \mathcal{N}\left(\xi(x), \Sigma(x)\right), \\ \xi(x) &= k(x, \mathcal{D}_n) K_n^{-1} y, \\ \Sigma(x) &= k(x, x) - k(x, \mathcal{D}_n) K_n^{-1} k(x, \mathcal{D}_n)^T, \end{aligned} \tag{2}$$

where $k(x, \mathcal{D}_n) = [k(x, x_1), \ldots, k(x, x_n)]$. To pick the next location $x \in \mathcal{D}$ for measuring y, one can simply select the point with the greatest uncertainty,

$$x_{n+1} = \arg\max_{x \in \mathcal{D}} \; k(x, x) - k(x, \mathcal{D}_n) K_n^{-1} k(x, \mathcal{D}_n)^T. \tag{3}$$

When $k(x, x_i) = c \cdot \exp(-\|x - x_i\|^2 / \eta)$, the term $k(x, x)$ is a constant. In this case, the selected point will have the smallest second term. We observe two properties of this objective function:

1. It does not depend on the observed values of y.
2. The sequence $x_1, x_2, x_3, \ldots$ is selected such that the space in which x resides is efficiently explored.

The second property is because $k(x, \mathcal{D}_n)$ penalizes closeness to previously selected locations. Since K_n is a PSD matrix, when x is not close to any point $x' \in \mathcal{D}_n$ the value of $k(x, x')$ is nearly all zero and the second term becomes less negative. We illustrate this on two toy manifolds in Figure 1(a), where we show the first 15 points selected.

3. Landmarking with Gaussian processes

As shown in Figure 1(a), the active learning method described in Section 2 also provides a good approach to landmarking a manifold. However, this requires the landmarks to correspond exactly to observed locations, and also the evaluation of a kernel at (ideally) every point in the dataset.

For small, densely sampled and low-dimensional data this may be reasonable, but for bigger problems it has drawbacks. For example, in high dimensions data usually isn't densely sampled, even if the manifold dimension is low relative to the ambient space. We may also believe *a priori* that a landmark shouldn't correspond exactly to an observation, for example in face or document datasets, in which case we might want the landmark to be a local average of related faces, or the underlying topics of a corpus of documents. In these cases, we might wish to avoid selecting from the raw data regardless of the data size.

We build on the ideas of Section 2 to derive an algorithm for finding relevant points along a manifold as defined by the observed noisy data. We demonstrate the output of the algorithm we will present on the same toy manifolds from Section 2 in Figure 1(b). We see that we again learn points evenly spaced along the manifolds, but this time those points are not required to correspond exactly to an observed location. Instead, we extend the objective function in Equation (3) to allow for a gradient method that converges to a local optimal location in the continuous manifold space as approximated by the data.

Let $\mathcal{M}$ be a manifold in some ambient space $\mathbb{S}$. ($\mathbb{S}$ is not necessarily $\mathbb{R}^d$, for example, it can be the intersection of the unit sphere with the positive orthant.) Often $\mathcal{M}$ has a low-dimensional nonlinear structure. Let both μ and $\mathcal{N}$ be probability distributions on $\mathbb{S}$, with the support of μ being constrained to the manifold $\mathcal{M}$ and $\mathcal{N}$ a zero mean noise process. We assume the observed data point $x = \hat{x} + \epsilon \in \mathbb{S}$, where $\hat{x} \sim_{\text{i.i.d}} \mu$ and $\epsilon \sim_{\text{i.i.d.}} \mathcal{N}$; that is, the data is a randomly selected point from the manifold corrupted by noise, which we assume to be small (Little et al., 2012).

We define the kernel function between points $t, t' \in \mathbb{S}$ as

$$k(t, t') = \int_{\hat{x} \in \mathbb{S}} \phi_{\hat{x}}(t)\phi_{\hat{x}}(t')d\mu(\hat{x}). \tag{4}$$

In this paper we use $\phi_{\hat{x}}(t) = \exp(-\|t - \hat{x}\|^2/\eta)$. Notice that μ has support $\mathcal{M}$, and so the integral is effectively over the manifold. This kernel function is closely related to the Gaussian kernel in Section 2, but will only consider t and t' to be "close" (i.e., $k(t, t')$ will be "large") according to the path between them along the manifold.

This representation is problematic since we do not have the distribution μ, or even the samples $\hat{x} \sim_{\text{i.i.d.}} \mu$. We therefore approximate Equation (4) with the observed noisy data using a plug-in estimator,

$$k(t, t') \approx \frac{1}{N}\sum_{i=1}^{N} \phi_{x_i}(t)\phi_{x_i}(t') := \frac{1}{N}\vec{\phi}(t)^T\vec{\phi}(t'), \tag{5}$$

In this case, the data serves a different purpose from Section 2. With this approach we are constructing $k(t, t')$ with any two points t, t' from the continuous ambient space $\mathbb{S}$, but restricting the kernel integral to the manifold. In Equation (5) the data $x \in \mathcal{D}$ allows us to approximate this manifold and helps define what is being integrated out, whereas using a Gaussian kernel in the framework of Section 2, the kernel is evaluated at the data points and the implied integral is over $\mathbb{R}^d$ with $d\mu(x) \to dx$.

Returning briefly to the ideal setting, given n selected landmarks $\mathcal{T}_n = \{t_1, \ldots, t_n\}$ from $\mathbb{S}$, let K_n be the pairwise kernel matrix of points in $\mathcal{T}_n$ using Equation (4). As with active learning in Section 2, the goal is to select a new t that is informative according to the objective function,

$$t_{n+1} = \arg\max_{t \in \mathbb{S}} \; k(t, t) - k(t, \mathcal{T}_n)K_n^{-1}k(t, \mathcal{T}_n)^T. \tag{6}$$

However, since we cannot calculate k exactly, we estimate each of these values with the plug-in approximation of Equation (5). Defining the matrix $\Phi = [\vec{\phi}(t_1), \ldots, \vec{\phi}(t_n)]$, we can approximate t_{n+1} as

$$t_{n+1} \approx \arg\max_{t \in \mathbb{S}} \; \vec{\phi}(t)^T\vec{\phi}(t) - \vec{\phi}(t)^T\Phi(\Phi^T\Phi)^{-1}\Phi^T\vec{\phi}(t). \tag{7}$$

This objective function can be interpreted as selecting the value of t such that the high-dimensional mapping $\vec{\phi}(t)$ extends the greatest Euclidean distance into the null space defined by Φ. This can be seen by rewriting the objective in Equation (7) as $\vec{\phi}(t)^T(I - \Phi(\Phi^T\Phi)^{-1}\Phi^T)\vec{\phi}(t)$: the product $(I - \Phi(\Phi^T\Phi)^{-1}\Phi^T)\vec{\phi}(t)$ projects $\vec{\phi}(t)$ into the null space of Φ, and so the objective is equal to the squared magnitude of this projection. Since $\vec{\phi}(t)$ is calculated using the data, we can see that t_{n+1} should be close to many $x \in \mathcal{D}$, but this set of points ideally will be disjoint from those used to define the span of $\vec{\phi}(t_1), \ldots, \vec{\phi}(t_n)$.

4. A stochastic algorithm for landmarking

We next present an algorithm for finding the sequence of landmarks $t_1, t_2, \ldots$, near the manifold $\mathcal{M}$. Since t can be any point in the continuous ambient space $\mathbb{S}$, we cannot simply evaluate over all possible values as done with the active learning approach in Section 2. Furthermore, the objective function in Equation (7) does not have a simple closed form solution, and the number of observations N may be too large to construct $\vec{\phi}(t)$ at each observed point in practice. We therefore derive a stochastic gradient algorithm for learning each value of t (Bottou, 1998).

For point t_{n+1}, we rewrite the objective in Equation (7) as

$$\begin{aligned} f_n(t, \mathcal{D}) &= \sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}\phi_{x_i}(t)\phi_{x_j}(t), \\ M_{ij} &= \delta_{ij} - \left(\Phi(\Phi^T\Phi)^{-1}\Phi^T\right)_{ij}, \end{aligned} \tag{8}$$

with δ_{ij} a delta function indicating whether $i = j$. A simple

Algorithm 1 Manifold landmarking with GPs

To find landmark t_{n+1} given $t_1, \ldots, t_n$, initialize $t_{n+1}^{(1)}$ and do the following:
for $s = 1, \ldots S$ **do**
  Randomly subsample a set B_s of observations $x \in \mathcal{D}$.
  For each t_k, construct $\vec{\phi}_s(t_k)$ using $x \in B_s$ and the function $\phi_x(t_k) = \exp(-\|x - t_k\|^2/\eta)$.
  Define the matrix $\Phi = [\vec{\phi}_s(t_1), \ldots, \vec{\phi}_s(t_n)]$ and set $M = I - \Phi(\Phi^T\Phi)^{-1}\Phi^T$.
  Let $f_n(t, B_s) = \sum_{x_i, x_j \in B_s} M_{ij}\phi_{x_i}(t)\phi_{x_j}(t)$.
  Calculate $\gamma = t_{n+1}^{(s)} + \rho_s \nabla_t f_n(t, B_s)|_{t_{n+1}^{(s)}}$ using Equation (10) and step size ρ_s.
  Project γ onto $\mathbb{S} \subseteq \mathbb{R}^d$ to obtain $t_{n+1}^{(s+1)}$.
end for

projected gradient method (Bertsekas, 1999) for maximizing f_n is to iterate between the following two steps:

$$\gamma = t_{n+1}^{(s)} + \rho_s \nabla_t f_n(t, \mathcal{D})|_{t_{n+1}}, \quad t_{n+1}^{(s+1)} = \text{Proj}_{\mathbb{S}}(\gamma), \quad (9)$$

where ρ_s is a step size and $\text{Proj}_{\mathbb{S}}(\cdot)$ is the projection onto the feasible set $\mathbb{S} \subset \mathbb{R}^d$. (When $\mathbb{S} = \mathbb{R}^d$, this step is unnecessary.) For the non-convex objective in Equation (8), this procedure will converge to a local optimal solution.

When $\phi_x(t) = \exp(-\|t - x\|^2/\eta)$, the gradient of f_n is

$$\nabla_t f_n = -\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{4M_{ij}}{\eta}\left[t - \frac{x_i + x_j}{2}\right]\phi_{x_i}(t)\phi_{x_j}(t). \quad (10)$$

We observe that symmetry can be exploited to efficiently calculate this vector in practice.

The more data that is available, the better defined the sampled manifold $\mathcal{M}$ will be, which will help learn better landmarks. However, when the number of observations is very large, calculating the vectors $\vec{\phi}(t)$ can be prohibitively slow. The final step of our algorithm is to perform stochastic gradient optimization of $f_n(t, \mathcal{D})$ by randomly subsampling a subset of points $B_s \subset \mathcal{D}$ at step s and approximating the gradient of f_n.[1] To ensure convergence, we use step sizes such that $\sum_s |\rho_s| = \infty$, $\sum_s \rho_s^2 < \infty$ (Robbins & Monro, 1951). We summarize the final algorithm for manifold landmarking in Algorithm 1.

5. Experiments

We evaluate our manifold landmarking algorithm on images, text and music data. For images, we consider the data as lying near a manifold in the ambient space $\mathbb{S} = \mathbb{R}_+^d$.

[1] We observe that the original gradient is stochastic as well by approximating $\mathcal{M}$ with a noise-corrupted $\hat{x} \sim_{\text{i.i.d.}} \mu$.

Figure 2. The first eight landmarks from the Yale faces dataset.

For the music and text problems, the data consists of vectors that lie on the intersection of the unit sphere with the positive orthant, which is a result of the data processing discussed later. For both of these problems our projection onto $\mathbb{S}$ is made accordingly. For all problems we use a step size of $\rho_s = (s_0 + s)^{-\tau}$ with $s_0 = 10$ and $\tau = 0.51$. The algorithm was robust to changes in these values. We take 1000 steps for each landmark and use batch size $|B_s| = 1000$ unless noted otherwise. We set the kernel width $\eta = \sum_i \hat{\sigma}_i^2$, where $\hat{\sigma}_i^2$ is an empirical approximation of the variance of the ith dimension of the data. To initialize each landmark, we draw from a Gaussian with the empirical mean and diagonal covariance of the data.

5.1. Qualitative evaluation

We evaluate our method qualitatively on two face datasets. In Figure 2 we show the first eight landmarks using 2,475 images of size 42×48 from the Yale faces database[2]. The dataset contains 165 images of various illuminations for 15 people. We see that the first eight landmarks capture various illuminations of an average face that doesn't correspond to any single person in the dataset.

We also consider the larger PIE faces dataset, consisting of 11,554 images of size 64×64 across 68 people with various illuminations and frontal poses. In Figure 3 we show a 2D embedding of 1,000 randomly selected images from the dataset, along with the first twenty landmarks learned from the full dataset, using the t-SNE algorithm (Van der Maaten & Hinton, 2008). It is evident that the landmarks effectively explore the space where the data resides. We also show the five closest faces to some of the landmarks. We again see averages of various genders and ethnicities for different poses and illuminations.

We show running times for the PIE data in Figure 4(a) for 32×32 and 64×64 images. We see that the time to learn a new landmark increases as the number of existing landmarks increases, due to the larger size of the matrix inversions and products in Equation (10). We also observe that

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

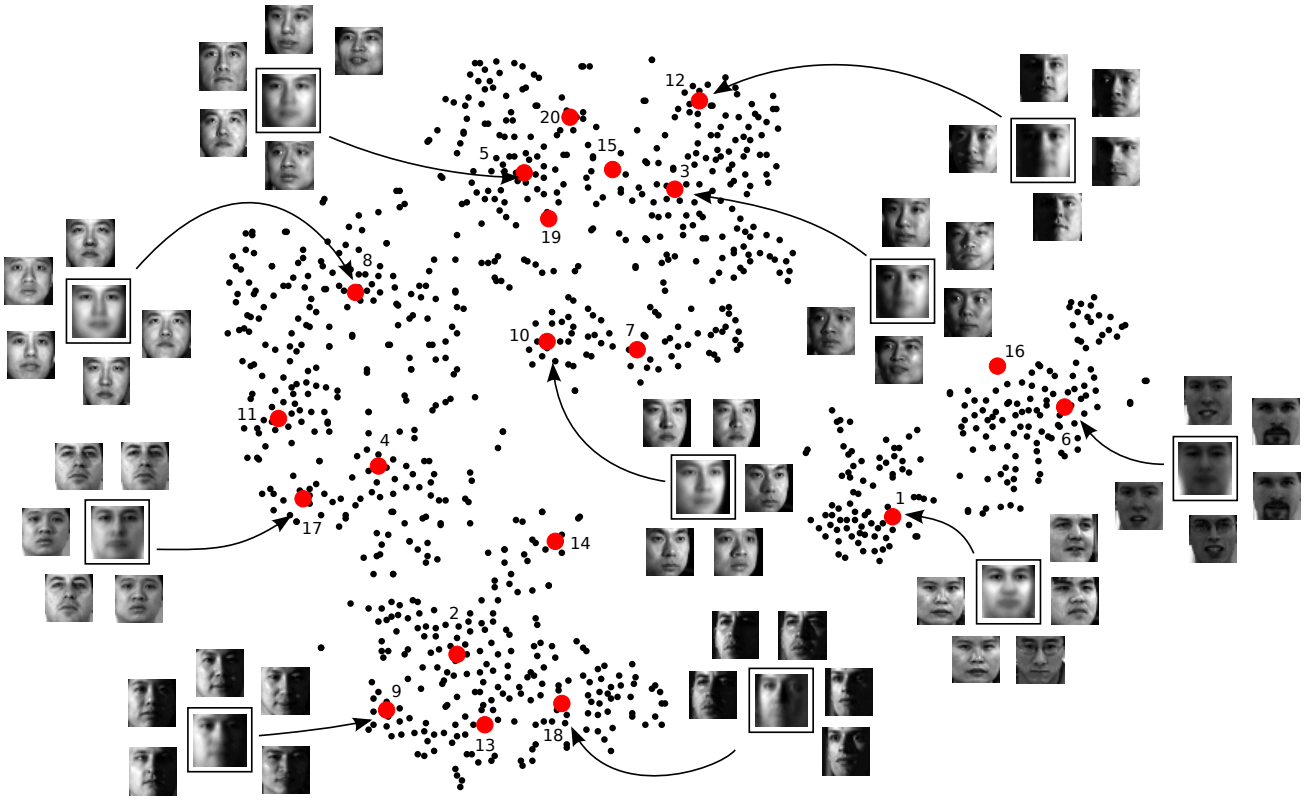
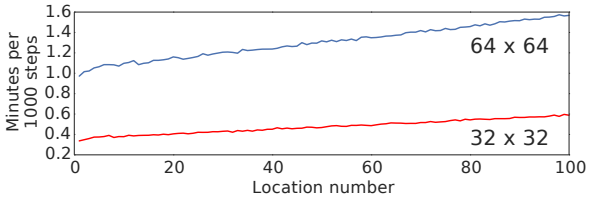
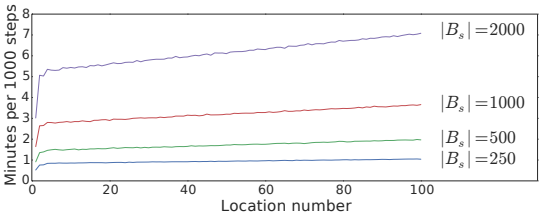


Figure 3. A 2D embedding of 1,000 randomly selected images from the PIE faces dataset (black dots), along with the first twenty landmarks (numbered red dots), using t-SNE algorithm (Van der Maaten & Hinton, 2008). For some of the landmarks, we also show the closest five faces. The landmarks locally average along the manifold (see later quantitative comparison with k-means).

as the dimensionality increases, the running time increases.



(a) Runtime for PIE



(b) Runtime for New York Times

Figure 4. The running time for (a) PIE as a function of landmark number and image size and batch size $|B_s| = 1000$; (b) the New York Times dataset. Learning speed is comparable to scalable topic models such as online LDA (Hoffman et al., 2013).

We also consider a corpus of roughly 1.8 million documents from the New York Times, as well as the 20 Newsgroup data set. For this data, we set each data point x_d near the manifold to be the square root of the normalized word histogram constructed using a vocabulary size of 8000 and 1545, respectively. That is, if w_{dn} is the index of the nth word in the dth document and document d has n_d total words, then we set

$$x_d(j) = \sqrt{\frac{\sum_n \mathbb{1}(w_{dn} = j)}{n_d}}. \tag{11}$$

Each landmark t is also restricted to lie in this same space. The function $\phi_x(t)$ therefore uses the Hellinger distance to measure closeness between a landmark and a document.

We can naturally interpret the square of the elements of t as a topic comparable to those learned by topic models. The meaning of a landmark can then be interpreted by showing the "most probable" words in the standard way. In Table 1 we show the top words for the first 11 landmarks of the New York Times and the first 12 landmarks of 20 Newsgroup. As is clear, these landmarks correspond to thematically meaningful concepts such as "sports", "food", and "politics". In Figure 4(b) we show the running time per

Table 1. (top) The "most probable" words for the first 11 landmarks learned on the 1.8 million document New York Times dataset. (bottom) The first 12 landmarks from the 20 Newsgroup dataset.

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
percent	inc	beloved	street	treasury	republican	minutes	mrs	game	percent	film
going	net	notice	sunday	bills	house	add	daughter	season	market	life
national	share	paid	music	rate	bush	oil	graduated	team	stock	man
public	reports	deaths	avenue	bonds	senate	salt	married	games	billion	story
life	earns	wife	theater	bond	political	cup	son	play	yesterday	book
ago	qtr	loving	art	notes	government	pepper	father	second	prices	movie
house	earnings	mother	museum	municipal	democrats	tblspoon	yesterday	left	quarter	love

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
good	windows	team	turkish	encryption	god	ftp	car	israel	nasa	scsi	gun
make	dos	game	turks	key	jesus	file	good	israeli	gov	drive	guns
ve	card	year	armenia	technology	bible	pub	cars	jews	space	ide	weapons
work	mb	games	soviet	government	christ	mail	price	arab	long	mb	crime
back	system	season	today	chip	christians	program	buy	state	orbit	hard	control

landmark on New York Times for several different batch sizes on a laptop computer. As is to be expected, the time increases as batch size increases, but all experiments can be performed within a few hours on a single computer, which is comparable to scalable topic models such as online LDA (Hoffman et al., 2013).

5.2. MNIST classification with landmarks

One major distinction between our proposed method and active learning with Gaussian processes as described in Section 2 is that we allow the landmarks to move along the continuous ambient space $\mathbb{S}$. From the low-dimensional toy examples in Figure 1, the advantage appears small because the data is dense on the manifold. In the next experiment, we quantitatively evaluate the landmarks learned from high-dimensional image data.

We consider the handwritten digit classification problem on the MNIST dataset (LeCun et al., 1998), and use a low-dimensional representation from different landmark approaches to evaluate their performance.[3] Given n selected landmarks $\mathcal{T}_n = \{t_1, \ldots, t_n\}$, we compute the n-dimensional landmark-based feature for the image x_d as $\vec{w}(x_d) = [\phi_{t_1}(x_d), \ldots, \phi_{t_n}(x_d)]^T$ where again we use $\phi_{t_k}(x_d) = \exp(-\|t_k - x_d\|^2/\eta)$. We perform ℓ_2-regularized logistic regression for classification. We use 50,000 images for training to learn both the landmarks and to train the classifier. We use 10,000 images as a validation set to select the regularization parameter among $\lambda = \{0.001, 0.01, \ldots, 1000\}$, and another 10,000 images for classification testing.

[3]This is intended to quantitatively compare methods in the same "domain", and not argue for our approach as a state-of-the-art dimensionality reduction technique.

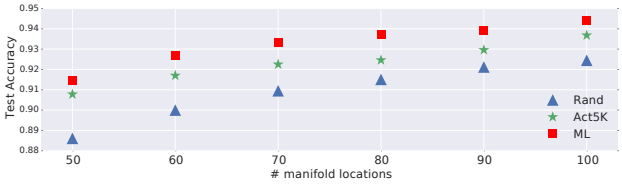


Figure 5. Test accuracy on MNIST with different landmark-derived features.

In addition to our landmarking approach, we consider two other approaches for obtaining landmarks, $\mathcal{T}_n$:

1. *Random selection:* This serves as a baseline. We simply randomly select n data points as the landmarks.

2. *Active learning:* The landmarks are selected using Equation (3). However, since this requires constructing the kernel matrix, which cannot be entirely read into memory even for moderate-sized datasets, we first subsample the digits and select landmarks from within this group using active learning. Here we report results on a 5,000-image subset. (We note the results are similar with other subsample sizes.)

We show the test accuracy as a function of the number of landmarks for our manifold landmarking algorithm (ML), random selection (Rand), and active learning (Act5K) in Figure 5. Not surprisingly, randomly landmarking does the worst. On the other hand, our proposed method is consistently better than active learning, which indicates that in the high-dimensional ambient space, we benefit from allowing landmarks to fall on the continuous manifold between data points rather than correspond to exactly one of them.

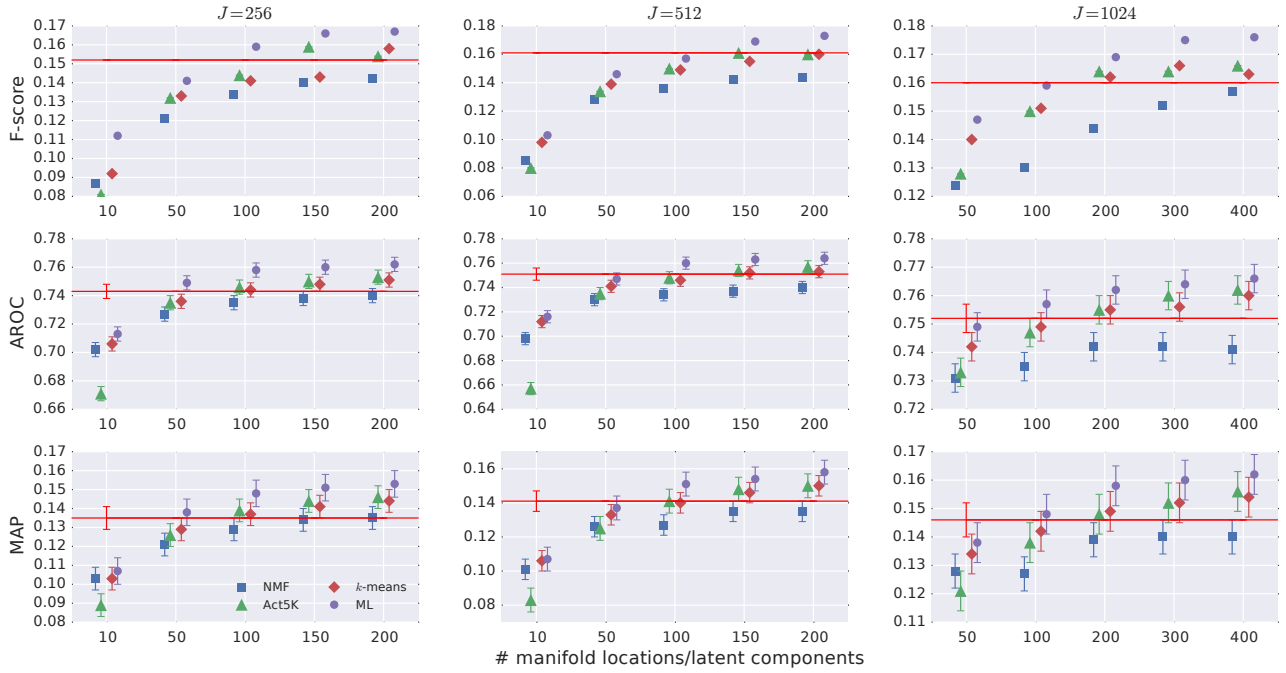


Figure 6. The annotation and retrieval performance of each algorithm with various codebook size, J. (Small amount of jitters is added for visualization.) For the feature derived from manifold landmarks (ML), active-learning-based landmarks (Act5K), k-means, and non-negative matrix factorization (NMF), the metrics are reported with increasing number of landmarks/latent components. The straight line is the baseline performance of logistic regression trained on the raw VQ features. Error bars correspond to one standard error.

5.3. Automatic Music tagging

We use an automatic music tagging problem to further evaluate the performance of our algorithm on a more constrained ambient space.

Automatic music tagging (Eck et al., 2007) is the task of analyzing the audio content (waveform) of a music recording and assigning to it human-relevant semantic tags concerning, e.g., style, genre or instrumentation. We perform experiments on the Million Song Dataset (Bertin-Mahieux et al., 2011) which contains the audio features and metadata (user tagging information from *Last.fm*) for one million songs. After preprocessing the data and removing songs with fewer than 20 tags from the test set, we obtained a dataset with 561 tags, 371,209 songs for training and 2,757 songs for testing.

Instead of directly working with the audio, we vector quantize features extracted from audio using the standard procedure: We run the k-means algorithm on a large subset of randomly selected feature vectors to learn J cluster centroids (codewords). For each song, we assign each feature vector extracted from the song to the cluster with the smallest Euclidean distance to the centroid, normalize a histogram of these quantizations on this codebook and take the square root to obtain a location x_d for song d. This VQ approach (without taking the square root) has been successfully applied to the music tagging problem and achieved state-of-the-art results (Xie et al., 2011; Liang et al., 2014). We use Echo Nest's timbre features provided in the Million Song Dataset to learn the codebook, which is similar to the widely used Mel-frequency cepstral coefficients (MFCCs). Since songs tend to have consistent timbre, the manifold approximated by x_d should be lower dimensional than the ambient space, which in this case is the intersection of the unit sphere with the positive orthant.

We treat music tagging as a binary classification problem: For each tag, we make independent predictions on whether the song is tagged with it or not. To this end, using the manifold landmarks learned from all x_d, we vectorize the dth song as $\vec{w}(x_d) = [\phi_{t_1}(x_d), \ldots, \phi_{t_n}(x_d)]^T$ where again $\phi_{t_k}(x_d) = \exp(-\|t_k - x_d\|^2/\eta)$. Again this uses the Hellinger distance between two probability vectors.

We evaluate the performance on an annotation and a retrieval task: For the annotation task we seek to automatically tag unlabeled songs. To evaluate the model's ability to annotate songs, we compute the average per-tag precision, recall, and F-score on the held-out test set. For the retrieval task, given a query tag we seek to provide a list of songs which are related to that tag. To evaluate retrieval performance, for each tag in the vocabulary we ranked each song

in the test set by the predicted probability. We then calculate the area under the receiver-operator curve (AROC) and mean average precision (MAP) for each ranking.

For both of these tasks, we use ℓ_2-regularized logistic regression on the vectors $\vec{w}$. Logistic regression has been shown to have state-of-the-art performance when applied directly on VQ features (Xie et al., 2011), which we use as our baseline. We also consider the following three approaches for comparison:

Non-negative matrix factorization (NMF): NMF (Lee & Seung, 2001) learns a parts-based representation. If we consider the learned latent components as landmarks, it shares the same property with our algorithm: The landmarks do not have to correspond exactly to a data point. The difference is that NMF can only capture linear structure. We fit the unnormalized VQ histogram using NMF with Kullback-Leibler divergence cost function and use the weights from the learned factorization as features to train the logistic regression.

Active learning: Similar to the MNIST experiment, we use Equation (3) to select landmarks from 5,000 subsampled songs and derive landmark-based feature as in Section 5.2.

k-means: We also treat the centroids of k-means as landmarks. These centroids can also capture non-linear structure, but the absence of a kernel may result in centroids that fall well off of the manifold. Another key difference is that k-means does not enjoy the sequential property of our method, i.e., we learn landmarks in their order of informativeness, whereas k-means must be restarted if the number of clusters changes. We fit the data with k-means++ (Arthur & Vassilvitskii, 2007) and treat the cluster centroids as landmarks, constructing landmark-based features in the same fashion as active learning and our proposed algorithm.

We show results for both annotation and retrieval in Figure 6 for several codebook size. For each logistic regression model, we use 5-fold cross-validation to search for the best regularization parameter among $\lambda = \{0.001, 0.01, \ldots, 1000\}$. As these plots show, the features derived from the proposed method consistently outperform those from other methods, regardless of the number of landmarks/latent components. Furthermore, the model trained on the landmark-derived features often outperforms the model trained on the raw VQ features. For example, with a codebook size $J = 1024$, we achieve similar results with only 100 locations (less than 10% of the original dimensionality) and significantly better with 200 locations. We note that similar results were observed for larger values of J.

6. Conclusion and Discussion

We have presented a method for finding landmarks on manifolds. Our approach borrows ideas from active learning with Gaussian processes to define an objective function for finding each landmark sequentially. We treat the data as noise-corrupted i.i.d. samples from some underlying distribution on the manifold, which we use to derive a stochastic gradient algorithm for finding landmarks near the manifold as approximately defined by these samples. This has the benefit of not requiring each landmark to correspond exactly to an observation, and allows for a fast stochastic learning algorithm.

Currently, we set the kernel width η from a simple heuristic. As future work, we will investigate a joint optimization over all the landmarks, which could potentially reduce the influence on the choice of η. Also, as presented each location is learned in a greedy fashion and then fixed. However, a simple (but slower) extension for joint optimization would be to modify a previously learned landmark given the subsequent ones. This can have the advantage of further spacing out the landmarks to provide better coverage of the manifold. For example, in the "circles" manifold of Figure 1(b) we see that landmarks 5, 9 and 11 are not spaced as well as might be desired. This is because landmarks 5 and 9 were fixed after being learned, and landmark 11 represents a local optimal solution sensitive to these values. If we returned to landmark 9 and continued to step along the gradient given the subsequent landmarks, this point would move down the circle to be more evenly spaced between the 5th and 11th landmarks.

References

Arthur, David and Vassilvitskii, Sergei. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

Bertin-Mahieux, Thierry, Ellis, Daniel P. W., Whitman, Brian, and Lamere, Paul. The Million Song Dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 591–596, 2011.

Bertsekas, Dimitri P. *Nonlinear programming*. Athena Scientific, 1999.

Bottou, Léon. Online learning and stochastic approximations. *On-line Learning in Neural Networks*, 17(9), 1998.

Cai, Deng and He, Xiaofei. Manifold adaptive experimental design for text categorization. *IEEE Trans. on Knowledge and Data Engineering*, 24(4):707–719, 2012.

Cohn, David A., Ghahramani, Zoubin, and Jordan,

Michael I. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

Cortes, Corinna and Vapnik, Vladimir. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Eck, Douglas, Lamere, Paul, Bertin-Mahieux, Thierry, and Green, Stephen. Automatic generation of social tags for music recommendation. In *Advances in Neural Information Processing Systems*, pp. 385–392, 2007.

Hoffman, Matthew D., Blei, David M., Wang, Chong, and Paisley, John. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Kapoor, Ashish, Grauman, Kristen, Urtasun, Raquel, and Darrell, Trevor. Active learning with Gaussian processes for object categorization. In *International Conference on Computer Vision*, 2007.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, Daniel D. and Seung, H. Sebastian. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pp. 556–562, 2001.

Li, Cheng, Liu, Haifeng, and Cai, Deng. Active learning on manifolds. *Neuroscience*, 123:398–405, 2014.

Li, Jun and Hao, Pengwei. Finding representative landmarks of data on manifolds. *Pattern Recognition*, 42 (11):2335–2352, 2009.

Liang, Dawen, Paisley, John, and Ellis, Daniel P. W. Codebook-based scalable music tagging with Poisson matrix factorization. In *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 167–172, 2014.

Little, Anna V, Maggioni, Mauro, and Rosasco, Lorenzo. Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature. Technical report, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2012.

Ng, Andrew Y., Jordan, Michael I., and Weiss, Yair. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 2001.

Paisley, John, Liao, Xuejun, and Carin, Lawrence. Active learning and basis selection for kernel-based linear models: A Bayesian perspective. *IEEE Trans. on Signal Processing*, 58(5):2686–2700, 2010.

Rasmussen, Carl E. *Gaussian processes for machine learning*. MIT Press, 2006.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Roweis, Sam T. and Saul, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Silva, Jorge, Marques, Jorge S, and Lemos, João Miranda. Selecting landmark points for sparse manifold learning. In *Advances in Neural Information Systems*, 2005.

Tenenbaum, Joshua B., De Silva, Vin, and Langford, John C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Tipping, Michael E. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

Van der Maaten, Laurens and Hinton, Geoffrey. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Vladymyrov, Max and Carreira-Perpinán, Miguel Á. Locally linear landmarks for large-scale manifold learning. In *European Conference on Machine Learning*, 2013.

Xie, Bo, Bian, Wei, Tao, Dacheng, and Chordia, Parag. Music tagging with regularized logistic regression. In *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 711–716, 2011.