MIT CBCL preprint — 1996

A variational principle for model-based interpolation

Lawrence K. Saul and Michael I. Jordan
{lksaul,jordan}@psyche.mit.edu

Center for Biological and Computational Learning
Massachusetts Institute of Technology
79 Amherst Street, E10-243
Cambridge, MA 02139

Abstract

Given a multidimensional data set and a model of its density, we consider how to define the optimal interpolation between two points. This is done by assigning a cost to each path through space, based on two competing goals—one to interpolate through regions of high density, the other to minimize arc length. From this path functional, we derive the Euler-Lagrange equations for extremal motion; given two points, the desired interpolation is found by solving a boundary value problem. We show that this interpolation can be done efficiently, in high dimensions, for Gaussian, Dirichlet, and mixture models.

1 Introduction

The problem of non-linear interpolation arises frequently in image, speech, and signal processing. Consider the following two examples: (i) given two profiles of the same face, connect them by a smooth animation of intermediate poses[1]; (ii) given a telephone signal masked by intermittent noise, fill in the missing speech. Both these examples may be viewed as instances of the same abstract problem. In qualitative terms, we can state the problem as follows[2]: given a multidimensional data set, and two points from this set, find a smooth adjoining path that is consistent with available models of the data. We will refer to this as the problem of *model-based interpolation*.

In this paper, we examine the problem of non-linear interpolation as it arises from statistical models of multidimensional data. Specifically, our focus is on models that have been derived from some form of density estimation. Though there exists a large body of work on the use of statistical models for regression and classification, there has been comparatively little work on the other types of operations that these models support. Non-linear interpolation is an example of such an operation, one that has important applications to video email[3], low-bandwidth teleconferencing[4], and audiovisual speech recognition[5].

Our approach to this problem is based on a variational principle that assigns a value to each path through space. The modeled density plays a fundamental role in inducing the metric and geometry of this space. We propose a path functional based on two competing goals—one to interpolate through regions of high density, the other to minimize arc length. The optimal interpolation between two points is then defined as the path which minimizes this functional. This path (or an approximation to it) is found by deriving the Euler-Lagrange equations[6] and solving the associated boundary value problem.

We show that this interpolation can be done efficiently, in high dimensions, for Gaussian, Dirichlet, and mixture models[7]. In fact, our framework is especially geared to mixture models, which represent the data as a collection of two or more clusters. These models, quite popular in practice, are well-suited to handling complicated (multimodal) data sets. Roughly speaking, for mixture models the problem of interpolation can be divided into two tasks—how to interpolate between points in the same cluster, and how to interpolate between points in different clusters. Thus our paper will be organized along these lines: in section 2, we consider the problem of *intracluster* interpolation; in section 3, the problem of *intercluster* interpolation.

Previous approaches to non-linear interpolation have exploited the properties of radial basis function (RBF) networks[1] and locally linear models[2]. We have been influenced by both these works, especially in the abstract formulation of the problem. Nevertheless, our approach has a number of distinguishing features—namely, the fundamental role played by the density, the treatment of non-Gaussian models, the use of a continuous variational principle, and the description of the interpolant by a differential equation.

2 Intracluster interpolation

The goal of this section is to motivate, in its simplest form, the variational principle for model-based interpolation. To this end, let $Q = \{\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \ldots, \mathbf{q}^{|Q|}\}$ denote a set of multidimensional data points, and let $P(\mathbf{q})$ denote a model of the distribution from which these points were generated. Given two points, our problem is to find a smooth adjoining path that respects the statistical model of the data. In particular, the desired interpolation should not pass through regions of space that the modeled density $P(\mathbf{q})$ assigns low probability.

2.1 Clusters and metrics

To develop these ideas further, we begin by considering a special class of models—namely, those that represent clusters. We say that $P(\mathbf{q})$ models a *data cluster* if $P(\mathbf{q})$ has a unique (global) maximum; in turn, we identify the location of this maximum, $\mathbf{q}^*$, as the *prototype*.

Let us now consider the geometry of the space inhabited by the data. To endow this space with a geometric structure, we must define a metric, $g_{\alpha\beta}(\mathbf{q})$, that provides a measure of the distance between two nearby points:

$$\mathcal{D}[\mathbf{q}, \mathbf{q}+\mathbf{dq}] = \left[\sum_{\alpha\beta} g_{\alpha\beta}(\mathbf{q})\ dq_\alpha dq_\beta\right]^{\frac{1}{2}} + \mathrm{O}\left(|\mathbf{dq}|^2\right). \qquad (1)$$

Intuitively speaking, the metric should reflect the fact that as one moves away from the center of the cluster, the density of the data dies off more quickly in some directions than in others. A natural choice for the metric, one that meets the above criteria, is the negative Hessian of the log-likelihood:

$$g_{\alpha\beta}(\mathbf{q}) = -\frac{\partial^2}{\partial q_\alpha \partial q_\beta}\left[\ln P(\mathbf{q})\right]. \qquad (2)$$

This metric is positive-definite if $\ln P(\mathbf{q})$ is concave; this will be true for all the examples we discuss. Figure 1a shows a schematic plot of the function $\ln P(\mathbf{q})$; it looks like an upside-down bowl, centered about the maximum at $\mathbf{q}^*$.

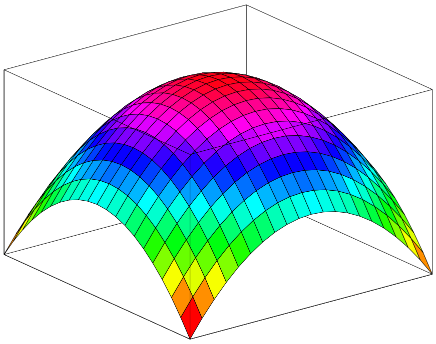
2.2 From densities to paths

The problem of model-based interpolation is to balance two competing goals—one to interpolate through regions of high density, the other to avoid excessive deformations. Using the metric in eq. (1), we can now assign a cost (or penalty) to each path based on these competing goals. In this section, we will motivate a path functional for model-based interpolation and derive the differential equation satisfied by the optimal path.

Consider the path from $\mathbf{a}$ to $\mathbf{b}$ shown in Figure 1b, as well as a particular parametrization, $\mathbf{q}(t)$. We begin by dividing the path into segments, each of which is traversed in some small time interval, dt. Note that the length of each segment, $\mathcal{D}[\mathbf{q}(t), \mathbf{q}(t+dt)]$, can vary with time, depending on the parametrization of the path. We assign a value to each segment by

$$\phi(t) = \left\{\left[\frac{P(\mathbf{q}(t))}{P(\mathbf{q}^*)}\right] e^{-\ell}\right\}^{\mathcal{D}[\mathbf{q}(t), \mathbf{q}(t+dt)]}, \qquad (3)$$

where $\ell \geq 0$. For reasons that will become clear shortly, we refer to ℓ as the *line tension*. The value assigned to each segment depends on two terms: a ratio of



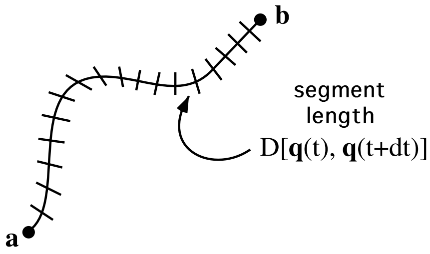


Figure 1: (a) Schematic plot of the concave function, $\ln P(\mathbf{q})$; (b) Interpolating path, broken down into differential elements.

probabilities, $P(\mathbf{q}(t))/P(\mathbf{q}^*)$, which favors points near the prototype, and the constant multiplier, $e^{-\ell}$. Both these terms are upper bounded by unity, and hence so is their product. The value of the segment also decays with its length, as a result of the exponent, $\mathcal{D}[\mathbf{q}(t), \mathbf{q}(t+dt)]$.

We derive a path functional by piecing these segments together, multiplying their individual contributions, and taking the continuum limit. A value for the entire path is obtained from the product:

$$e^{-\mathcal{S}} = \prod_t \phi(t). \tag{4}$$

Taking the logarithm of both sides, and considering the limit $dt \to 0$, we obtain the path functional

$$\mathcal{S}[\mathbf{q}(t)] = \int \left\{ -\ln\left[\frac{P(\mathbf{q}(t))}{P(\mathbf{q}^*)}\right] + \ell \right\} \left[\sum_{\alpha\beta} g_{\alpha\beta}(\mathbf{q})\ \dot{q}_\alpha \dot{q}_\beta\right]^{\frac{1}{2}} dt, \tag{5}$$

where $\dot{\mathbf{q}} \equiv \frac{d}{dt}[\mathbf{q}]$ is the tangent vector to the path at time t. Here, the sign of $\mathcal{S}$ has been chosen so that the resulting functional represents a quantity to be minimized. The terms in this functional balance the two competing goals for non-linear interpolation. The first favors paths that interpolate through regions of high density, while the second favors paths with small arc lengths; both are computed under the metric induced by the modeled density. The line tension ℓ determines the cost per unit arc length and modulates the competition between the two terms.

Note that the value of the functional does not depend on the rate at which the path is traversed; it is invariant to reparametrizations, $t \to f(t)$, where $f(\cdot)$ is a monotonically increasing function. We have purposefully designed the functional with this property, as the quality of an interpolation does not depend on the rate at which it is performed.

To minimize this functional, we use the following result (derived in the appendix) from the calculus of variations. Let $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ denote the integrand of eq. (5), such that $\mathcal{S}[\mathbf{q}(t)] = \int dt\ \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$. Then the path which minimizes this functional obeys the Euler-Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) = \frac{\partial \mathcal{L}}{\partial \mathbf{q}}. \tag{6}$$

The function $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ is known as the *Lagrangian*. Note, however, that the form of the Lagrangian in eq. (5) is quite different than its counterpart in classical mechanics[6]. In our case, the terms in the Euler-Lagrange equations depend on the modeled density, $P(\mathbf{q})$, and the line tension, ℓ. We define the model-based interpolant between two points as the path which minimizes this functional; it is found by solving the associated boundary value problem.

In the next sections, we present eq. (5) for two distributions of interest—the multivariate Gaussian and the Dirichlet. For these distributions, one can efficiently obtain an optimal or near-optimal interpolation, even in high dimensions. We emphasize that efficiency is a paramount consideration, since real-world data sets (e.g., images of faces) tend to inhabit very high dimensional spaces.

2.3 Gaussian cloud

The simplest model of multidimensional data is the multivariate Gaussian. In this case, the data is modeled by

$$P(\mathbf{x}) = \frac{|\mathbf{M}|^{1/2}}{(2\pi)^{N/2}} \exp\left\{-\frac{1}{2}\left[\mathbf{x}^T\mathbf{M}\mathbf{x}\right]\right\}, \tag{7}$$

where $\mathbf{M}$ is the inverse covariance matrix and N is the dimensionality. Without loss of generality, we have chosen the coordinate system so that the mean of the data coincides with the origin. For the Gaussian, the mean also defines the location of the prototype; moreover, from eq. (2), the metric induced by this model is just the inverse covariance matrix. From eq. (5), we obtain the path functional:

$$\mathcal{S}[\mathbf{x}(t)] \ = \ \int \left\{\frac{1}{2}\left[\mathbf{x}^T\mathbf{M}\mathbf{x}\right] + \ell\right\} \left[\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}}\right]^{\frac{1}{2}} dt, \tag{8}$$

Given two endpoints, this functional assigns a cost to each possible interpolation, based on its adherence to the statistical model of the data.

To find the model-based interpolant, we seek the path that minimizes this functional. Because the functional is parameterization-invariant, it suffices to consider paths that are traversed at a constant (unit) rate: $\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}} = 1$. From eq. (6), we find that the optimal path with this parameterization satisfies:

$$\left\{\frac{1}{2}\left[\mathbf{x}^T\mathbf{M}\mathbf{x}\right] + \ell\right\}\ddot{\mathbf{x}} + \left[\mathbf{x}^T\mathbf{M}\dot{\mathbf{x}}\right]\dot{\mathbf{x}} = \mathbf{x}. \tag{9}$$

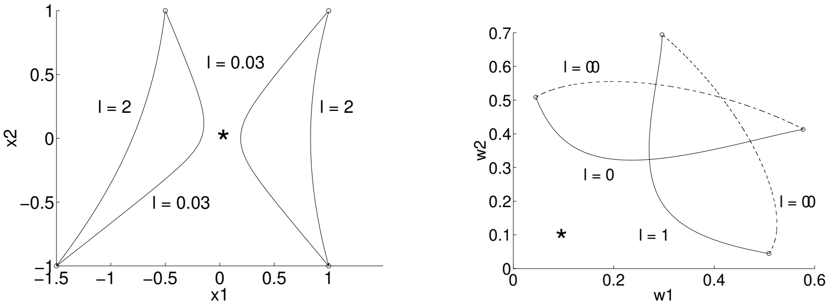


Figure 2: Model-based interpolations for the (a) Gaussian distribution; (b) Dirichlet distribution. The prototypes are shown as asterisks; ℓ denotes the line tension.

This is a set of coupled non-linear equations for the components of $\mathbf{x}(t)$. However, note that at any moment in time, the acceleration, $\ddot{\mathbf{x}}$, can be expressed as a linear combination of the position, $\mathbf{x}$, and the velocity, $\dot{\mathbf{x}}$. It follows that the motion of $\mathbf{x}$ lies in a plane; in particular, it lies in the plane[1] spanned by the initial conditions, $\mathbf{x}$ and $\dot{\mathbf{x}}$, at time $t = 0$.

Suppose we are given two endpoints, $\mathbf{a}$ and $\mathbf{b}$, and asked to provide the non-linear interpolation based on eq. (9). The crux of the boundary value problem[8] is to find the initial "shooting" direction at $\mathbf{x} = \mathbf{a}$ such that the path passes through $\mathbf{x} = \mathbf{b}$ at some later time. The search for this direction is greatly simplified by the fact that $\mathbf{x}(t)$ is confined to a plane; in effect, this collapses the boundary value problem from N dimensions to one. In particular, the initial velocity is constrained to lie in the plane defined by $\mathbf{a}$, $\mathbf{b}$, and the origin; moreover, its magnitude is set by our parameterization convention, $\dot{\mathbf{x}}^T\mathbf{M}\dot{\mathbf{x}} = 1$. Thus, even in high dimensions, it is straightforward to solve this boundary value problem; it does not suffer from the curse of dimensionality.

Figure 2a shows some solutions to this boundary value problem for different values of the line tension, ℓ. Here, we have calculated interpolations for the two dimensional Gaussian model with $M_{11} = M_{22} = 1$, $M_{12} = M_{12} = 0$. The paths were obtained by numerical integration of eq. (9); this was coupled with a shooting-like procedure to determine the correct initial velocity. Note how the paths bend toward the cluster center, with the degree of curvature determined by the line tension, ℓ.

[1] In fact, the trajectories of motion can be expressed in terms of elliptic integrals, but the result is hardly enlightening.

2.4 Dirichlet simplex

For many types of data, the multivariate Gaussian distribution is not the most appropriate model. Suppose[2] that the data points are vectors of positive numbers whose elements sum to one. In particular, we say that $\mathbf{w}$ is a *probability vector* if:

$$\mathbf{w} = (w_1, w_2, \ldots, w_N) \in \mathcal{R}^N, \quad w_\alpha > 0 \text{ for all } \alpha, \text{ and } \sum_\alpha w_\alpha = 1. \tag{10}$$

The elements of these vectors have many interpretations, as (i) probability "masses" for an event with N possible outcomes, (ii) hidden unit activations in a normalized radial basis function (RBF) network[1], or (iii) coordinates over the interior of an N-dimensional simplex. Clearly, the multivariate Gaussian is not suited to data of this form, since no matter what the mean and covariance matrix, it cannot assign zero probability to vectors outside the simplex. Instead, a more natural model is the Dirichlet distribution:

$$P(\mathbf{w}) = \Gamma(\theta) \prod_\alpha \frac{w_\alpha^{\theta_\alpha - 1}}{\Gamma(\theta_\alpha)}, \tag{11}$$

where $\theta_\alpha > 0$ for all α, and $\theta \equiv \sum_\alpha \theta_\alpha$. Here, $\Gamma(\cdot)$ is the gamma function, and θ_α are parameters that determine the statistics of $P(\mathbf{w})$. Note that $P(\mathbf{w}) = 0$ for vectors that are not probability vectors; in particular, the simplex constraints on $\mathbf{w}$ are implicit assumptions of the model.

2.4.1 Cluster in $\ln \mathbf{w}$

We can rewrite the Dirichlet distribution in a more revealing form as follows. First, let $\mathbf{w}^*$ denote the probability vector with elements $w_\alpha^* = \theta_\alpha/\theta$. Then, making a change of variables[3] from $\mathbf{w}$ to $\ln \mathbf{w}$, we have:

$$P(\ln \mathbf{w}) = \frac{1}{Z_\theta} \exp\left\{ -\, \theta \left[\mathrm{KL}\left(\mathbf{w}^* || \mathbf{w}\right)\right] \right\}, \tag{12}$$

where Z_θ is a normalization factor that depends on θ_α (but not $\mathbf{w}$), and the quantity in the exponent is θ times the Kullback-Leibler (KL) divergence[9],

$$\mathrm{KL}\left(\mathbf{w}^* || \mathbf{w}\right) = \sum_\alpha w_\alpha^* \ln \left[\frac{w_\alpha^*}{w_\alpha} \right]. \tag{13}$$

The KL divergence is a non-negative measure of the mismatch between $\mathbf{w}$ and $\mathbf{w}^*$, with $\mathrm{KL}(\mathbf{w}^*||\mathbf{w}) = 0$ if and only if $\mathbf{w} = \mathbf{w}^*$. Hence, we see that the most probable value of $\ln \mathbf{w}$ occurs when $\mathbf{w}$ matches the proportions of the parameter vector, θ_α. Since $\mathrm{KL}(\mathbf{w}^*||\mathbf{w})$ has no other minima besides the one at $\mathbf{w}^*$, we shall say that $P(\ln \mathbf{w})$ models a data cluster *in the variable* $\ln \mathbf{w}$.

[2] Section 2.4 on the Dirichlet model may be omitted on a first reading.
[3] We use $\ln \mathbf{w}$ as shorthand for the vector $(\ln w_1, \ln w_2, \ldots, \ln w_N)$.

2.4.2 Metric

Let us consider the metric induced by this modeled density, following the prescription of eq. (2). We can derive this metric by performing a Taylor expansion of $\mathrm{KL}(\mathbf{w}^*||\mathbf{w})$ and examining the second-order terms in the expansion. A subtlety arises here, however, due to the simplex constraints on $\mathbf{w}$. Recall that the metric is used to reflect the fact that not all directions are equal—i.e., the density dies off more quickly in some directions than in others. In deriving the Dirichlet metric, it is important to consider only those directions that point *inside* the simplex. This is most easily done by rewriting the KL divergence as:

$$\mathrm{KL}\left(\mathbf{w}^*||\mathbf{w}\right) = \sum_\alpha w^*_\alpha \ln\left[\frac{w^*_\alpha}{w_\alpha/\sum_\beta w_\beta}\right]. \tag{14}$$

Clearly, this definition reduces to the previous one when the components of $\mathbf{w}$ sum to unity. However, it has the additional property that the right hand side is invariant to scaling transformations ($\mathbf{w} \rightarrow \xi\mathbf{w}$ with $\xi > 0$) that take vectors outside the simplex. Hence, the gradient in this (unallowed) direction vanishes, and the calculation of the Hessian can proceed in a straightforward fashion.

Following the prescription of eq. (2), we now identify the metric with the negative Hessian of the log-likelihood. The Hessian of eq. (14), taken with respect to $\ln\mathbf{w}$, is given by:

$$\left(w_\alpha\frac{\partial}{\partial w_\alpha}\right)\left(w_\beta\frac{\partial}{\partial w_\beta}\right)\left[\mathrm{KL}\left(\mathbf{w}^*||\mathbf{w}\right)\right] = \frac{w_\alpha\delta_{\alpha\beta}\sum_\gamma w_\gamma - w_\alpha w_\beta}{\left(\sum_\gamma w_\gamma\right)^2}. \tag{15}$$

Evaluating eq. (15) at points inside the simplex, we find: $g_{\alpha\beta}(\ln\mathbf{w}) = \theta(w_\alpha\delta_{\alpha\beta} - w_\alpha w_\beta)$. In fact, this expression for $g_{\alpha\beta}(\ln\mathbf{w})$ can be further simplified. Recall that we need the metric to compute distances between nearby probability vectors. Let $\mathbf{w}$ and $\mathbf{w}+\mathbf{dw}$ represent two such vectors. The squared distance between them is given by:

$$\begin{aligned} ds^2 &= \sum_{\alpha\beta} g_{\alpha\beta}(\ln\mathbf{w})\;[d\ln w_\alpha]\;[d\ln w_\beta], && (16)\\ &= \theta\sum_{\alpha\beta}\left[w_\alpha\delta_{\alpha\beta} - w_\alpha w_\beta\right]\left(\frac{dw_\alpha}{w_\alpha}\right)\left(\frac{dw_\beta}{w_\beta}\right), && (17)\\ &= \theta\sum_\alpha \frac{dw_\alpha^2}{w_\alpha}, && (18)\end{aligned}$$

where we have used the fact that $\sum_\alpha dw_\alpha = 0$ inside the simplex. As the outer-product terms in the metric make no contribution to ds^2, we can adopt the simpler expression, eq. (18), to compute local distances[4] between probability

[4]Note that as in the Gaussian case, the metric does not depend on the location of the prototype.

vectors. In fact, up to a multiplicative factor of 2θ, eq. (18) measures the infinitessimal KL divergence between $\mathbf{w}$ and $\mathbf{w}+\mathbf{dw}$. This is a natural metric for vectors whose elements can be interpreted as probabilities.

2.4.3 Equation of motion

The functional for non-linear interpolation is found by substituting the modeled density and the induced metric into eq. (5). For the Dirichlet distribution, this gives:

$$\mathcal{S}_0[\mathbf{w}(t)] = \int \left\{ \theta \left[\mathrm{KL}\left(\mathbf{w}^* || \mathbf{w}\right)\right] + \ell \right\} \left[\theta \sum_\alpha \frac{\dot{w}_\alpha^2}{w_\alpha} \right]^{\frac{1}{2}} dt. \tag{19}$$

Our problem is to find the path that minimizes this functional. Because the functional is parameterization-invariant, it again suffices to consider paths that are traversed at a constant rate, or $\sum_\alpha \dot{w}_\alpha^2/w_\alpha = 1$. In addition to this, however, we must also enforce the constraint that $\mathbf{w}$ remains inside the simplex. This is done by including an extra term in the functional, in the form of a Lagrange multiplier:

$$\mathcal{S}[\mathbf{w}(t)] = \mathcal{S}_0[\mathbf{w}(t)] + \int \lambda(\mathbf{w}) \left[1 - \sum_\alpha w_\alpha \right] dt. \tag{20}$$

To derive the optimal path, we can now perform an unconstrained minimization with respect to $\mathbf{w}$ and then choose the Lagrange multipler $\lambda(\mathbf{w})$ to enforce the constraint. Following this procedure, we find that the optimal path is described by:

$$\left[\theta\, \mathrm{KL}(\mathbf{w}^*||\mathbf{w}) + \ell\right] \left\{ \ddot{w}_\alpha - \frac{\dot{w}_\alpha^2}{2w_\alpha} + \frac{w_\alpha}{2} \right\} - \theta \left[\sum_\beta \frac{w_\beta^*}{w_\beta} \dot{w}_\beta \right] \dot{w}_\alpha = \theta(w_\alpha - w_\alpha^*). \tag{21}$$

Given two endpoints, this differential equation defines a boundary value problem for the optimal path. Unlike before, however, in this case the motion of $\mathbf{w}$ is not confined to a plane. Hence, the boundary value problem for eq. (21) does not collapse to one dimension, as does its Gaussian counterpart, eq. (9).

To remedy this situation, we consider an approximation that finds a near-optimal interpolation, in lieu of the optimal one. This is done in two steps: first, by solving eq. (21) in the limit $\ell \to \infty$, where the optimal path is simply the "shortest" one; second, by using this limiting solution to parametrize the desired interpolation. As we shall see, the advantage of this approximation is that it collapses the boundary value problem to one dimension, thus allowing cheap computation of the Dirichlet interpolants.

2.4.4 Geodesics

When $\ell \gg \theta$, minimizing arc length is much more important than traversing regions of high density. In the limit $\ell \to \infty$, the optimal interpolation between two points is simply the shortest adjoining path, with arc length computed under the metric, eq. (18). Such paths are the *geodesics* associated with this metric. Taking the limit $\ell \to \infty$ in eq. (21), we obtain the geodesic equation of motion:

$$\ddot{w}_\alpha - \frac{\dot{w}_\alpha^2}{2w_\alpha} + \frac{w_\alpha}{2} = 0. \tag{22}$$

It is straightforward to show that solutions to eq. (22) take the form:

$$w_\alpha = K_\alpha \left[1 + \sin(t + \psi_\alpha)\right], \tag{23}$$

where K_α and ψ_α are determined either by the initial conditions or the boundary values. Note that these paths do not define straight lines in Euclidean space. Rather, they lie in the three-dimensional subspace spanned by the vectors K_α, $K_\alpha \cos\psi_\alpha$, and $K_\alpha \sin\psi_\alpha$. Interestingly, these vectors have a simple geometric structure: the first defines a point in the simplex (i.e., $\sum_\alpha K_\alpha = 1$), while the second and third define allowed displacements.

Let us show that the boundary value problem for these geodesics can be solved efficiently, even in high dimensions. In particular, suppose we are given two endpoints, $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. Then, from eq. (22), the boundary value problem is to obtain the parameters τ, K_α, and ψ_α, such that:

$$w_\alpha^{(1)} = K_\alpha \left[1 + \sin\psi_\alpha\right], \tag{24}$$
$$w_\alpha^{(2)} = K_\alpha \left[1 + \sin(\tau + \psi_\alpha)\right]. \tag{25}$$

Here, the parameter τ determines the path length. Note that in addition to the boundary conditions, we must also enforce the constraint that the trajectory remains inside the simplex, or $\frac{d}{dt}\sum_\alpha w_\alpha = 0$. Enforcing this at time $t = 0$ gives:

$$0 = \sum_\alpha K_\alpha \cos\psi_\alpha. \tag{26}$$

Together, eqs. (24–26) give us $2N + 1$ equations for $2N + 1$ unknowns. Though non-linear, these equations are easily solved in the following manner: (i) from eqs. (24) and (25), express the unknowns ψ_α and K_α in terms of τ, $w_\alpha^{(1)}$ and $w_\alpha^{(2)}$; (ii) substitute these expressions into eq. (26) to obtain a one-dimensional equation for τ; (iii) locate (numerically) the root of this equation by searching the interval $\tau \in [0, 2\pi]$; (iv) compute numerical values for ϕ_α and K_α from the value obtained for τ. Note that this procedure scales efficiently to high-dimensional spaces, requiring on the order of N computations to find the parameters τ, ψ_α, and K_α.

2.4.5 Mixed paths

In this section, we show how the geodesic from $\mathbf{w}^{(1)}$ to $\mathbf{w}^{(2)}$ can be used to construct a near-optimal interpolation, in the sense of eq. (19). In particular, suppose we have obtained the parameters τ, ψ_α, and K_α that determine the geodesic from $\mathbf{w}^{(1)}$ to $\mathbf{w}^{(2)}$. Now consider the family of paths that can be expressed as a convex combination (or a *mixture*) of the geodesic and the prototype at $\mathbf{w}^*$:

$$w_\alpha(t) = m(t)w^*_\alpha + [1 - m(t)]\, K_\alpha[1 + \sin(t + \psi_\alpha)]. \tag{27}$$

Here, $m(t)$ is any continuous function, bounded between zero and one, that satisifies the boundary conditions, $m(0) = m(\tau) = 0$. The constraints on $m(t)$ ensure that the right hand side of eq. (27) lies in the simplex, with endpoints at $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$.

The motivation for eq. (27) is that the optimal interpolation can be approximately expressed in this "mixed" form. In particular, for finite values of ℓ, we expect the optimal interpolation to deviate from the geodesic path and bend toward the prototype at $\mathbf{w}^*$. The paths parameterized by eq. (27) are able to assume this general shape, provided that $m(t)$ starts at $m(0) = 0$, rises to some maximum value, then decreases back to $m(\tau) = 0$ at its endpoint.

A near-optimal interpolation can therefore be found by substituting eq. (27) into the path functional, eq. (19). The resulting functional, $\mathcal{S}[m(t)]$, is optimized by solving its Euler-Lagrange equation, $\frac{d}{dt}(\partial\mathcal{L}/\partial\dot{m}) = \partial\mathcal{L}/\partial m$. In particular, given two endpoints, we find the best interpolation of the form eq. (27) by (numerically) solving the one-dimensional boundary value problem for $m(t)$. Figure 2b shows some model-based interpolations obtained in this manner for the Dirichlet distribution with $\theta_\alpha = 1$ and $N = 12$. For ease of visualization, we have projected the twelve-dimensional trajectories onto the w_1w_2–plane.

How accurate is the approximation, eq. (27)? Clearly, this depends on the value of the line tension, ℓ, the dimensionality, N, and the distribution, $P(\mathbf{w})$. The approximation is exact in two special cases: the limit $\ell \to \infty$, where $m(t) = 0$, and the case $N = 3$, where the simplex is a plane. For intermediate regimes, we tested the approximation in a series of numerical experiments. This was done over a wide range of values for ℓ, N, and θ.

Each experiment consisted of generating three points inside the simplex—the prototype $\mathbf{w}^*$ and two endpoints—then computing the optimal, geodesic, and mixed paths based on eqs. (21), (23), and (27). We also computed the costs of each of these paths by numerically integrating eq. (19). Let $\mathcal{S}_O$, $\mathcal{S}_G$, and $\mathcal{S}_M$ denote these costs for the optimal, geodesic, and mixed paths. The ratios $r_G = (\mathcal{S}_G - \mathcal{S}_O)/\mathcal{S}_O$ and $r_M = (\mathcal{S}_M - \mathcal{S}_O)/\mathcal{S}_O$ measure the relative losses incurred by the geodesic and mixed paths with respect to the optimal one. Figure 3 shows how these losses depend on the line tension, ℓ, and the dimensionality, N; here, each data point represents an average over one hundred experiments. As expected, the smaller the line tension, the more that the optimal path deviates

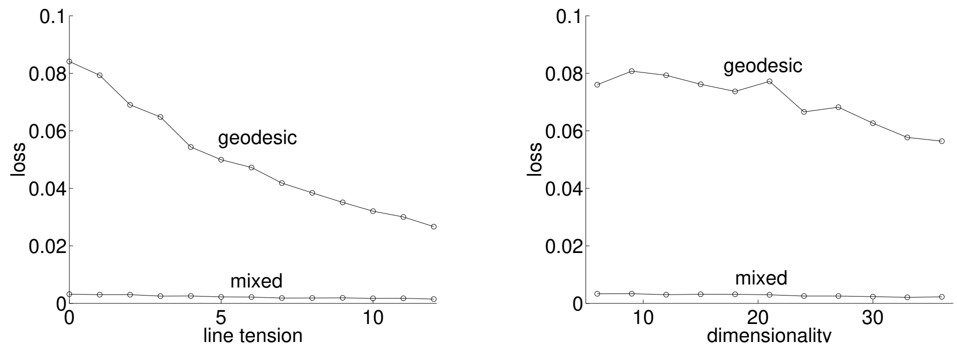


Figure 3: Relative losses incurred by the geodesic and mixed paths. Plots shown: (a) r_G and r_M vs. ℓ for fixed $\theta = N = 12$; (b) r_G and r_M vs. $N = \theta$ for fixed $\ell = 1$.

from the geodesic. Note, however, that even for small ℓ and large N, the mixed path finds a close approximation to the optimal one.

3 Intercluster interpolation

The Gaussian and Dirichlet distributions of the previous section are clearly inadequate for modeling for multimodal data sets. In this section, we extend the variational principle to mixture models, which describe the data as a collection of $k \geq 2$ clusters. In particular, suppose the data is modeled by

$$P(\mathbf{q}) = \sum_{z=1}^{k} \pi_z P(\mathbf{q}|z). \tag{28}$$

Here, we have assumed that the conditional densities $P(\mathbf{q}|z)$ model data clusters as defined in section 2.1, and the coefficients $\pi_z = P(z)$ define prior probabilities for the latent variable, $z \in \{1, 2, \ldots, k\}$.

The crucial step for mixture models is to develop the appropriate generalization of eq. (5). To this end, let $\mathcal{L}_z(\mathbf{q}, \dot{\mathbf{q}})$ denote the Lagrangian derived from the conditional density, $P(\mathbf{q}|z)$, and ℓ_z the line tension that appears in this Lagrangian. To respect the weighting of the mixture components in eq. (28), we set these line tensions according to $\ell_z = \ell - \ln \pi_z$. Thus, components with higher weights have lower line tensions; also, as in the unimodal case, there is only one free parameter $\ell \geq 0$ that specifies the overall line tension.

For model-based interpolation, we now combine these Lagrangians into a single functional:

$$\mathcal{S}[\mathbf{q}(t), z(t)] = \int dt \; \mathcal{L}_{z(t)}(\mathbf{q}, \dot{\mathbf{q}}). \tag{29}$$

Note that eq. (29) is a functional of two arguments, not one. For mixture models, which define a joint density $P(\mathbf{q}, z) = \pi_z P(\mathbf{q}|z)$, our goal is to find the optimal path in the joint space $\mathbf{q} \otimes z$. Here, $z(t)$ is a piecewise-constant function of time that assigns a discrete label (between 1 and k) to each point along the path; in other words, it provides a temporal segmentation of the path, $\mathbf{q}(t)$. Accordingly, we view the path $\mathbf{q}(t)$ as a string of intracluster interpolants, stitched together at the points where $z(t)$ changes value. The purpose of $z(t)$ in eq. (29) is to select which Lagrangian is used to compute the contribution from the interval $[t, t+dt]$.

As before, we define the model-based interpolant as the path $\mathbf{q}(t)$ that minimizes eq. (29). In this case, however, both $\mathbf{q}(t)$ and $z(t)$ must be simultaneously optimized to recover this path. We have implemented an iterative scheme to perform this optimization, one that alternately (i) estimates the segmentation $z(t)$, (ii) computes the model-based interpolation within each cluster based on this segmentation, and (iii) reestimates the points (along the cluster boundaries) where $z(t)$ changes value. In short, the strategy is to optimize $z(t)$ for fixed $\mathbf{q}(t)$, then optimize $\mathbf{q}(t)$ for fixed $z(t)$. Since each of these steps decreases the value of $\mathcal{S}$ (and since the path functional is bounded below), it follows that this strategy will converge to a local, if not global, minimum.

Let us review the three phases of the algorithm—*segmentation*, *interpolation*, and *restitching*—in more detail. In the segmentation phase, we optimize $z(t)$ for fixed $\mathbf{q}(t)$, assigning each point along the path according to:

$$z(t) = \arg\max_z \left[L_z(\mathbf{q}, \dot{\mathbf{q}})\right]. \tag{30}$$

Eq. (30) is a hard assignment of the type used in standard clustering algorithms. In the interpolation phase of the algorithm, we optimize $\mathbf{q}(t)$ for fixed $z(t)$, computing the intracluster interpolants from their Euler-Lagrange equations. Note that the segmentation from eq. (30) provides the boundary conditions for these interpolants; in particular, the endpoints occur at the boundary crossings where $z(t)$ changes value. Finally, in the restitching phase of the algorithm, we adjust these boundary crossings, adapting their locations by gradient descent in the functional, eq. (29). In particular, if such a crossing occurs at $\mathbf{q}(t_c)$, then we adapt its location by:

$$\Delta\mathbf{q}(t_c) = -\eta \left[\frac{\delta\mathcal{S}}{\delta\mathbf{q}(t_c)}\right] = \eta \left[\frac{\partial\mathcal{L}_{z(t)}}{\partial\dot{\mathbf{q}}}\right]\Bigg|_{t_c^-}^{t_c^+}, \tag{31}$$

where $\eta > 0$ is the learning rate. Eq. (31) for $\delta\mathcal{S}/\delta\mathbf{q}(t_c)$ is derived in the appendix. Note that even though $\mathcal{S}[\mathbf{q}(t), z(t)]$ defines a functional over the entire path, the gradient in eq. (31) requires only local information—namely, the values of $\mathbf{q}$ and $\dot{\mathbf{q}}$ at $t = t_c$.

Figure 4 shows how this algorithm operates on a simple mixture of Gaussians. In this example, the covariance matrices of the Gaussians were set equal to the identity matrix, and the means were distributed along a circle in the x_1x_2–plane. Note that with each iteration, the interpolant converges more closely

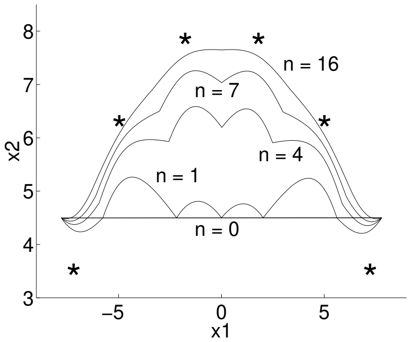


Figure 4: Convergence of the iterative algorithm for a mixture of Gaussians; here, n denotes the number of iterations. At $n = 0$, the path is initialized as a straight line; subsequently, it converges to the circle of means.

to the path that traverses the circle of means. The effect is similar, but not identical, to the manifold-snake algorithm of Bregler and Omohundro[2]. The main difference lies in the nature of the *intermediate* solutions. In our case, the interpolant starts out with a pronounced segmentation, then becomes smoother as the boundary crossings adapt according to eq. (31). The pronounced segmentation is a by-product of the intra/intercluster decomposition; effectively, the intracluster interpolation, given by eq. (6), proceeds at a much faster rate then the intercluster interpolation, which relies on gradient descent. The advantage of the decomposition is that it fully exploits the structure of the cost function, including those parts that are analytically tractable. On the other hand, the intermediate solutions seem somewhat less robust than the manifold-snake, which relaxes (more or less) uniformly to the optimal path.

4 Discussion

In this paper we have presented a coherent framework for model-based interpolation, one rooted in ideas from statistics, differential geometry, and the calculus of variations. For theorists, our framework provides a rich mathematical structure; for practitioners, a suite of efficient algorithms. We have applied our ideas to Gaussian, Dirichlet, and mixture models, and shown that the resulting algorithms scale well to high dimensions. Though not explored in this work, the description of the interpolant by a differential equation has obvious applications to data compression. For example, one can imagine encoding a sequence of images by the parameters of this equation and a set of initial conditions. Future work will concentrate along these lines.

A Calculus of variations

In this appendix, we summarize some key results from the calculus of variations[6]. Let $\mathcal{S}[q(t)]$ denote the path functional

$$\mathcal{S}[\mathbf{q}(t)] = \int dt\ \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}). \tag{32}$$

To find the path that minimizes this functional (assuming it is bounded below), we consider the effect of small variations $\delta\mathbf{q}(t)$ on the value of $\mathcal{S}$:

$$\delta\mathcal{S} = \int dt\ \left\{ \frac{\partial\mathcal{L}}{\partial\mathbf{q}}\ \delta\mathbf{q}(t)\ + \frac{\partial\mathcal{L}}{\partial\dot{\mathbf{q}}}\ \delta\dot{\mathbf{q}}(t) \right\}. \tag{33}$$

Integrating the second term in eq. (33) by parts gives:

$$\delta\mathcal{S} = \int dt\ \left\{ \frac{\partial\mathcal{L}}{\partial\mathbf{q}} - \frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\mathbf{q}}}\right) \right\}\ \delta\mathbf{q}(t)\ + \frac{\partial\mathcal{L}}{\partial\dot{\mathbf{q}}}\ \delta\mathbf{q}(t)\ \Big|_{t_1}^{t_2}, \tag{34}$$

where $\mathbf{q}(t_1)$ and $\mathbf{q}(t_2)$ are the endpoints of the path. If these endpoints are fixed, then the last term in eq. (34) vanishes, and the conditions for stationarity are seen to be the Euler-Lagrange equations, eq. (6). On the other hand, for paths that already satisfy eq. (6), it is only this last term that survives. For our purposes, the first and second terms in eq. (34) are exactly the functional derivatives required for intracluster and intercluster interpolation. In particular, the second term gives rise to the rule for adapting the boundary crossings, eq. (31).

Acknowledgements

The authors thank Tommi Jaakkola, Peter Dayan, and Philip Sabes for useful comments. This work was supported by NSF grant CDA-9404932, ONR grant N00014-94-1-0777, and Daimler-Benz Research.

References

[1] T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. of IEEE*, vol 78:9 (1990).

[2] C. Bregler and S. Omohundro. Nonlinear image interpolation using manifold learning. In G. Tesauro, D. Touretzky, and T. Leen (eds.). *Advances in Neural Information Processing Systems 7*, 973–980. MIT Press, Cambridge, MA (1995).

[3] T. Ezzat. Example based analysis and synthesis for images of faces. *MIT EECS M.S. thesis* (1996).

[4] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. *AI Memo 1161*, MIT (1993).

[5] C. Bregler and S. Omohundro. Nonlinear manifold learning for visual speech recognition. In E. Grimson (ed.), *Proc. of the 5th Intl. Conf. on Computer Vision*, 494–499. IEEE Computer Society Press, Los Alamitos, CA (1995).

[6] H. Goldstein. *Classical Mechanics.* Addison-Wesley, London (1980).

[7] G. McLachlan and K. Basford. *Mixture models: inference and applications to clustering.* Marcel Dekker (1988).

[8] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes.* Cambridge University Press, Cambridge (1986).

[9] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York (1991).