LEVEL SET ESTIMATION ON THE SPHERE

Alexander Lorbert and Peter J. Ramadge
Dept. of Electrical Engineering, Princeton University, Princeton NJ

ABSTRACT
We investigate a technique for estimating level sets of functionals on the 2-sphere. The surface of the sphere is finely partitioned using a tree decomposition and a candidate level set is obtained by minimizing a regularized cost on the tree. A cycle spinning scheme, implemented as an ensemble classification method, is developed to decrease the variance of the tree-based estimate. Both constructions are compatible with many existing hierarchical discretizations of the 2-sphere, e.g. HEALPix and HTM. We present simulation results of a synthetic data set and an fMRI data set.

Index Terms— Pattern classification, Quadrrees, Mesh generation, Image edge analysis.

1. INTRODUCTION
We examine the estimation of a level set of a functional defined on the unit sphere S^2 in R^3. This problem arises in many contexts in which noisy data is measured or mapped on the sphere, e.g. in astronomy, cosmology, geophysics and neuroscience [1–5].

Our investigation is motivated by a problem in the analysis of fMRI data. In a typical fMRI experiment, anatomical (MRI) and functional (fMRI) data are collected in the volume. The anatomical data can be used to extract a two-dimensional model of brain cortex, a highly folded and convoluted two-dimensional anatomical surface in the brain volume. Cortex is posited to be the main locus of cognitive processing. After extraction, each cortical hemisphere is inflated to obtain a smooth surface, which is then projected to the sphere S^2 in R^3. The associated fMRI data is aligned with the anatomical scan and mapped onto the inflated hemisphere. The functional time series is then a (sampled) vector valued function on S^2 and various statistics derived from the functional data are functionals on S^2. Regions of activation under the stimulus are determined by finding a level set of an appropriate statistic on S^2. This can be done using a univariate threshold test on the statistic. However, recently multivariate methods for activation region selection and detection have been emerging (e.g. [6–8]). A multivariate approach is capable of exploiting spatial and functional relationships among voxels and hence of detecting subtle relationships and spatial patterns in the data.

Recently, a novel multivariate method has been developed for level-set estimation of functionals on Euclidean space [9]. This level set method, hereafter denoted LSM, uses decision trees in R^d to spatially regularize the selection of the level set. The method has similarities with the well-known approach of Classification and Regression Trees (CART) [10]. Whereas CART penalizes tree complexity by assigning equal weight to all terminal nodes, the LSM weights terminal nodes based on their depth in the tree. This is well matched to the goal of coding the boundary of a set indicator function. A second feature of the LSM is the use of cycle spinning to aggregate classification across many trees. Motivated by LSM, our objective is the development of an extension of this approach to estimate level sets of functionals defined on S^2. To do so requires consideration of tree partitions of S^2 and cycle spinning over such tree partitions.

Hierarchical discretizations of S^2 are routinely used to represent and analyze large data sets, e.g. full-sky data sets in astrophysics and cosmology. Examples include the HEALPix discretization [1] and the Hierarchical Triangular Mesh (HTM) discretization [2]. Each of these tree decompositions of S^2 possess different characteristics, which are leveraged by extensive data sets. Hence we desire a level set estimation method that is not tied to any particular decomposition model. This will allow the method to be applied across various existing discretizations of S^2. For the same reason, we seek a surrogate for cycle spinning that is not tied to the attributes of a particular discretization, e.g. equal area at each level.

Our main contribution is to show that a level set estimator in the spirit of [9] can be implemented over a variety of existing tree pixelations of S^2. Hence the approach is applicable to a variety of existing data sets and the data structures used to support them, including HEALPix and HTM. Furthermore, we formulate a probabilistic notion of cycle spinning for adaptation to S^2. As shown experimentally, this improves classification accuracy significantly.

The remainder of the paper is organized as follows. §2 develops a method to obtain a level set estimate on S^2 using spherical quadrees. We then address how a surrogate for cycle spinning on S^2 can be incorporated to refine the level set estimate. We provide two examples of the method in §3: one based on synthetic data and a second based on fMRI data.
2. LEVEL SET ESTIMATION ON $S^2$

We first briefly describe the main features of LSM (see [9] for complete details). Given scalar $\gamma$, for bounded, unknown $f : [0, 1]^d \mapsto \mathbb{R}$ we wish to estimate $S^*_\gamma \equiv \{ x \in [0, 1]^d : f(x) \geq \gamma \}$. This is formulated as a binary classification problem. The estimated level set is represented by a tree $T$ on $[0, 1]^d$ (Fig. 1). A regularized loss $L(T)$ is then minimized over admissible trees to estimate the level set. The cost is designed to minimize a weighted symmetric difference between a candidate level set, $S$, and $S^*_\gamma$ while regularizing over tree structure. To do so $L(T)$ uses the node level when assigning a cost to terminal nodes. At each tree level, each node is assigned equal volume. Therefore, a node level penalty is equivalent to a node volume penalty. So deep terminal nodes containing useful boundary information do not add a significant cost. This allows the LSM to select highly-unbalanced trees consistent with representing a level set (deep at boundaries, shallow everywhere else).

It is well known that locally sensitive transforms tend to produce artifacts after denoising [11]. These artifacts are a product of signal discontinuities lying strictly inside locally-binned regions attributed to the transform. To combat these effects, the usual "transform-threshold-inverse transform", is replaced by "shift-transform-threshold-inverse transform-inverse shift", followed by averaging over shifts. This process is called cycle spinning. Cycle spinning serves to mitigate the high variance associated with trees. While trees are able to incorporate local spatial relationships, a property well-suited for our problem, high variance is a serious obstacle. The LSM implements cycle spinning to locally smooth the classifier and better adapt it to the level set boundary.

In the case of $S^2$, we have already noted that there are a variety of existing tree decompositions. One class is obtained by first embedding a convex regular polyhedron with $M$ sides in $S^2$. Only 5 such polyhedra exist ($M = 4, 6, 8, 12,$ and 20). This results in $M$ uniform regions on the sphere, each a projection of a polyhedron face. Each face is then subdivided and projected to the sphere. For example, embedding an icosahedron into $S^2$, partitions $S^2$ into 20 congruent spherical triangles. The icosahedron is the root node (level 0) of a tree. At level 1 there are twenty nodes, each representing a face of the icosahedron. For each face, we project the mid-

points of the three sides (aligned with great circles) onto $S^2$. This yields four child triangles with vertices on $S^2$. This process can be iterated until a mesh of the desired resolution is attained (Fig. 2). The HTM discretization, mentioned earlier, results from an equivalent construction starting from an icosahedron ($M = 8$). The root has 8 child nodes and each of these is further partitioned via a quadtree. The five examples in this class of partitions have the rotational symmetry group of the underlying platonic polyhedron.

The HEALPix discretization is in a different class. The sphere is hierarchically tessellated into curvilinear quadrilateral pixels with each pixel at a given tree level having equal area and pixels distributed on lines of equal latitude. The root node has 12 child nodes. Each of these equal area regions is then subdivided iteratively into 4 equal area children. At each level the nodes represent curvilinear quadrilaterals of equal area but different shape. Thus in the HEALPix tree, the root splits into twelve child nodes and each child node is further refined by a quadtree. At any level, all nodes represent equal area regions. The HEALPix discretization has the rotational symmetry group of the octahedron [1].

We can compute a tree based level set estimate that will encompass many tree-based spherical partition schemes, with the following model: at level 1 the sphere is partitioned into $M$ regions; each level 1 region is then subdivided using congruent t-trees (e.g. quadtrees). For any tree $T$, let $|T|$ denote the number of terminal nodes, and $a_i$, $l(a_i)$ and $A_i$ denote the $i$-th terminal node, its label ($0/1$), and its area, $i = 1, \ldots, |T|$. After growing a deep, full tree, the LSM is used to prune each tree. To do so we use the given data $D = \{(x_j, y_j)\}_{j=1}^n$ with $x_j \in S^2$ and $y_j = f(x_j)$ plus noise. Here $f$ is the unknown functional for which we seek the level set at $\gamma$. Following [9], assuming an equal-area subdivision scheme, we prune to minimize $L(T) = R(D, T) + C(T)$ where

$$R(D, T) = \sum_{i=1}^{|T|} \sum_{j=1}^n \frac{\gamma - y_j}{2\delta} \mathbb{I}_{a_i}(x_j)(\mathbb{I}_{S(a_i)} - \mathbb{I}_{S(\bar{a}_i)}),$$

$$C(T) = \frac{\rho}{\sqrt{n}} \sum_{i=1}^{|T|} \sqrt{8(\log(2/\delta) + |a_i| \log 2)} p_{a_i}(\delta).$$

Here, $S$ is a candidate level set, $C$ is a normalizing constant,
\(\rho\) and \(\delta\) are tuning parameters, \(\hat{p}_{a_i}(\delta)\) is an empirical approximation to \(\mathbb{P}(x \in a_i)\), and \(|a_i|\) is the number of bits required to encode the position of \(a_i\). For a non-equal-area subdivision scheme, knowledge of the \(A_i\)'s, i.e., the area of a spherical triangle, is required for estimating the empirical probability.

For illustration, consider the icosahedron embedding in the sphere and quadtree subdivision of each face. In the default orientation, the base icosahedron has two vertices lying on the z-axis and another vertex in the positive yz-plane. From here, we grow 20 full quadtrees to the desired precision followed by pruning of all level-1 quadtrees. This yields a level set estimate for each projected face of the icosahedron without consideration of neighboring face information. Clearly, the independent pruning of subtrees is a serious limitation. For example, when a portion of the set boundary straddles the projected edge of the base icosahedron, the estimate produced will generally prune away the fine detail. Similarly, for a boundary segment lying well-inside an icosahedral face, the quality of the estimate depends on how the segment aligns with the subdivided region. Not surprisingly, very different tree classifiers are obtained when the base icosahedron is rotated, i.e., the coordinate system is changed.

Let \(\theta \in \Theta\) be a rotation parameter. In cycle spinning, we form estimators \(\hat{h}_{D,\theta}(x), \theta \in \Theta\). For example, in denoising an image, \(\theta\) is a two dimensional vector of circular-shift offsets. Cycle spinning then averages over \(\theta\) to yield the estimator \(\hat{h}_{D}(x) = (1/|\Theta|) \sum_{\theta} h_{D,\theta}(x)\) in the discrete case. When \(|\Theta|\) is large, averaging over all shifts may be costly. In [9] the authors propose taking a sample average over random shifts. More generally, we can treat \(\theta\) as a random variable and set \(h_{D}(x) = \mathbb{E}_{\theta}[\hat{h}_{D,\theta}(x)]\).

For level set estimation on \(\mathbb{S}^2\), let \(\Theta = \{\theta^v, \theta^\phi\} \) where \(\theta^v \sim \mathcal{U}\mathbb{S}^2\) is random vector and \(\theta^\phi \sim \mathcal{U}[-\pi, \pi]\) is a random angle \((\mathcal{U}\Theta Denotes the uniform distribution over the set \(\Theta\)). To generate instances of \(\theta\), let \(u_i\) for \(i = 1, 2, 3\), denote independent samples from \(\mathcal{U}(0, 1)\) and set \(\theta^v(1) = 2\sqrt{u_1 - u_2^2} \cos(2\pi u_2), \theta^v(2) = 2\sqrt{u_1 - u_2^2} \sin(2\pi u_2),\) and \(\theta^v(3) = 2u_1 - 1\), and \(\theta^\phi = 2\pi u_3\) [12]. Then from a default orientation, rotate \(\mathbb{S}^2\) through angle \(\theta^\phi\) about the vector \(\theta^v\). Because \(\theta\) consists solely of continuous variables, we need not be concerned about the symmetry group of the tree decomposition being used. Let \(h_{D,\theta}: \mathbb{S}^2 \mapsto \{0, 1\}\) denote the tree classifier obtained from training set \(D\) and base rotation \(\theta\). In general, the sample average does not yield a binary function. With \(x\) fixed, we assume that \(\{h_{D,\theta_i}(x)\}_{i=1}^K\) can be modeled as a sequence of Bernoulli trials. Rather than the sample average, we use the sample median to produce the final binary output. Using the sample median is equivalent to a voting classifier. Generating an odd number of classifiers of equal weight guarantees that a final ‘0/1’ decision results. This ensemble method can correct for visual artifacts of a single classifier, a product of the high variance associated with trees. Whereas various methods address the problem of high variance, e.g. [13–16], a voting classifier is arguably the simplest ensemble method, and, for our application, is directly linked to cycle spinning.

As an immediate extension, we can also generate estimates derived from varying tree constructions and include these classifiers in the voting process. This has the potential to decrease variance across multiple pixelations. In general, level set estimation is regarded as an unsupervised learning problem. However, if given a small subset of labeled examples, we can reweight the votes to arrive at a final estimate by measuring accuracy over this small subset.

3. EXPERIMENTS

For both examples presented in this section, we used an icosahedron for the base polyhedron and the subdivision discussed earlier. We first consider a synthetic example based on the function: \(f(x) = \sum_{i=1}^K \exp\{-\alpha_i^2(w_i^T x - 1)^2\}\). Here \(K = 3, \alpha_{1,2,3} = \{6, 9, 20\}\), and \(x, w_i \in \mathbb{S}^2\) with \(w_1 = \)
[0.9623 0.1925 0.1925]^T, w_2 = [0.1925 0.9623 0.1925]^T, and w_3 = [0.0990 0.0990 0.9901]^T (Fig. 3(a)). The level set for γ = 0.45 is shown in Fig. 3(b). Adding 0.4-power iid Gaussian noise produces an SNR of −6.8dB (Fig. 3(c)). Trees were grown to a depth of 8 for a total of 20 × 4^T = 327,680 samples on the sphere and a set of 31 random rotation tree classifiers were obtained. The level set estimate of the voting classifier shown in Fig. 3(d). We also calculated the classification error with respect to the full tree of depth 8 for the nonrotated base icosahedron. When averaged over 20 runs, simple thresholding at 0.45 resulted in 26.4% classification error with a standard deviation of 0.07%, whereas the voting classifier error was 1.1% with a standard deviation of 0.03%. Without voting, a single classifier yielded an error of 4.1% with a standard deviation of 0.14%. The mean run time to generate a single tree classifier (growing to 327,680 terminal nodes and pruning) was 0.97 seconds on an Intel® Core™ i7-920 with 64-bit Linux®.

Table 1 summarizes results for varying tree depth. These results show that voting greatly improves classification accuracy when dealing with a small number of samples. Roughly, run times increase four-fold when tree depth increases by a level, i.e. a four-fold increase in terminal nodes. For this example, our method yields a noticeable improvement over thresholding, even for a small number of samples.

<table>
<thead>
<tr>
<th>Tree Depth</th>
<th>Thresholding</th>
<th>Single Classifier</th>
<th>Voting</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>26.8 (2.28)</td>
<td>24.7 (1.80)</td>
<td>2.6 (0.88)</td>
</tr>
<tr>
<td>4</td>
<td>26.2 (1.17)</td>
<td>22.3 (1.31)</td>
<td>2.1 (0.31)</td>
</tr>
<tr>
<td>5</td>
<td>26.4 (0.57)</td>
<td>19.1 (0.99)</td>
<td>1.9 (0.19)</td>
</tr>
<tr>
<td>6</td>
<td>26.4 (0.26)</td>
<td>14.1 (0.99)</td>
<td>1.6 (0.09)</td>
</tr>
<tr>
<td>7</td>
<td>26.4 (0.16)</td>
<td>6.0 (0.47)</td>
<td>1.4 (0.04)</td>
</tr>
</tbody>
</table>

Table 1. Mean % classification error and (standard deviation)

Our second example uses the data set from [17]. The F-statistics, obtained from the statistical parameter map of a “Face vs. Object” experiment, are shown on cortex in a close-up view centered around the Fusiform Face Area in Fig. 3(e). For a significance level of 10^-3, thresholding at 10.88 yields the estimated activation region shown in Fig. 3(f). Assume the F-statistics are produced by an unknown underlying function corrupted by noise. We wish to estimate the level set at γ = 10.88. The raw dataset consists of 144,002 sample points on a 1-millimeter spherical mesh. To accommodate the data, a tree of depth 8 (327,680 terminal nodes) was constructed with node values being assigned the value of the nearest data point. The result of one tree classifier is shown in Fig. 3(g). Visual artifacts are easily perceived, a product of high tree variance. The result of a 15 tree voting classifier, shown in Fig. 3(h), exhibits “edge aware” smoothing; subregions containing intricate portions of the level set boundary are exposed via the voting process. If this test was repeated for the same subject, we can then begin to cross validate.

4. CONCLUSION

We have investigated a method for level set estimation on the unit sphere. By exploiting existing tree pixelations of the sphere, and implementing a voting classifier as an instance of a simple ensemble method, we are able to transfer recent ideas for level set estimation in Euclidean space to the 2-sphere. By adapting cycle spinning to the sphere, we were able to decrease the variance of our classifier. The performance of the technique was tested on both synthetic and real data, where both numerical and visual improvements were observed.

5. REFERENCES


