SUPERVISED LEARNING OF CLASSIFIERS VIA LEVEL SET SEGMENTATION

Kush R. Varshney and Alan S. Willsky

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology, Cambridge, MA, USA

ABSTRACT

A variational approach based on level set methods popular in image segmentation is presented for learning discriminative classifiers in general feature spaces. Nonlinear, nonparametric decision boundaries are obtained by minimizing an energy functional that incorporates a margin-based loss function. The class of level set contour decision boundaries is discussed in terms of the structural risk minimization principle. A variation on ℓ_1 feature subset selection is developed. Use of level set classifiers as base learners for boosting is discussed.

Index Terms— pattern classification, supervised learning, level set methods, feature selection

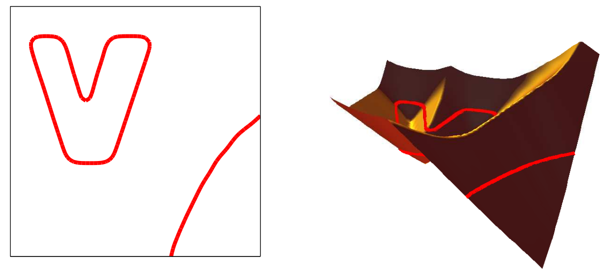
1. INTRODUCTION

One of the central problems of machine learning is supervised discriminative learning of binary classifiers. It is a fundamental problem and the foundation for many other learning tasks as well as being important in numerous applications. Given training examples in some feature space labeled with one of two values, the goal is to find boundaries that divide that feature space into two partitions which classify unseen examples well.

Segmentation is a key problem in image processing and computer vision. Given an image, the goal is to find boundaries that partition that image into regions. Regions should be individually homogeneous and different regions distinct. One can note that partitioning images and partitioning general feature spaces are similar problems. In this paper, we develop a methodology for supervised learning using the level set approach, a technique which has been successfully applied to image segmentation [1].

Several well-developed techniques for supervised discriminative learning exist in the literature, including the perceptron algorithm [2], logistic regression [3], and support vector machines (SVMs) [4]. All of these approaches, in their basic form, produce linear decision boundaries. Nonlinear boundary contours in the given feature space can be obtained using the following technique: mapping the original feature space to a feature space of higher dimension by taking nonlinear functions of the original features. Learning algorithms are applied to the new higher dimensional feature space by treating each dimension linearly. They retain the efficiency of the original lower dimensional space for particular sets of nonlinear functions through the use of kernels [5]. Such contours are parametric in the original feature space.

Many approaches to image segmentation also represent boundary contours parametrically. The level set approach, however, breaks away from this paradigm by using an implicit representation. A scalar-valued function φ, known as the level set function, is used to represent the contour. The function is zero on the contour and only on the contour. The contour is the zero level set of φ. The idea is illustrated in Fig. 1. Any shape can be represented, including topologies with disconnected pieces. The level set representation is more flexible than parametric representations.



(a) (b)

Fig. 1. An illustration of the level set representation of a contour. The contour is shown in (a) and the level set function marked with its zero level set is shown in (b) as a surface plot.

The problem of image segmentation is approached by constructing an energy functional with the level set function as its argument that is minimized when the contour divides the space into a good segmentation. One example is mean squared error of image intensity with two different 'true' image intensities inside the contour and outside the contour. In this work for general supervised classification, we develop an energy functional based on classifier margin-based loss functions such as zero-one loss, hinge loss, logistic loss, and exponential loss [6]. The formulation presented in Sec. 2 can use any margin-based loss function, even discontinuous loss functions, which is not the case with many learning algorithms. (SVMs use the hinge loss by definition and logistic regression uses the logistic loss by definition.)

By finding nonlinear contours directly in the original feature space rather than some higher dimensional space, we can straightforwardly encode constraints or additional objectives about the decision boundaries. One such example, a variation of ℓ_1 feature selection [7], is presented in Sec. 4. Additionally, if one is interested in interpreting the shape or properties of the decision boundary, doing so is more direct with nonlinear contours defined in the original space. Also, with the level set representation, one does not have to worry about parameterization selection or kernel selection.

We are not the first to notice the connection between level set image segmentation and classification, but to the best of our knowledge,

This work was supported in part by an NSF Graduate Research Fellowship, by Shell International Exploration and Production, Inc., and by a MURI funded through AFOSR Grant FA9550-06-1-0324.

there has been very little prior work in this area. In [8], fully general feature spaces are not considered. In particular, examples must be pixels in an image with the feature vector containing the spatial index of the pixel along with other features. The work of [9] does consider general feature spaces, but has a very different energy functional than our margin-based loss functional. It is based on counts of training examples in grid cells and is similar to the mean squared error functional for image segmentation described above. The learning is also based on one-class classification rather than standard discriminative classification. In [10], using level set methods for density-based clustering in general feature spaces is considered.

The paper is organized as follows. In Sec. 2, the basic formulation for using level set methods to train a binary classifier is presented. Sec. 3 discusses the issue of overfitting and the structural risk minimization principle with respect to the level set classifier. Sec. 4 describes an approach for feature selection using ℓ_1 minimization that can be integrated into the formulation of Sec. 2. In Sec. 5, we discuss how level set classifiers can be used as base classifiers in boosting. Sec. 6 provides examples on both generated and real data, whereas Sec. 7 gives a summary of the work.

2. LEVEL SET SUPERVISED CLASSIFICATION

In this section, we give a brief primer on level set methods and then cast supervised classification in the same framework. The literature on level set methods is vast; we only present the ideas necessary to develop the novel classifier. One excellent source of information on level set methods is [11].

2.1. Level Set Method for a Variational Problem

Consider the domain with $\mathbf{x} \in \Omega \subset \mathbb{R}^D$, which is usually the pixel or voxel domain in image segmentation. A contour C partitions Ω into two regions R and R^c, which can be of any topology. A variational problem is to be solved: find the contour C to minimize the energy functional

$$E(C) = \int_R f(\mathbf{x}) d\mathbf{x}. \tag{1}$$

By the calculus of variations, it may be shown that the first variation $\frac{\delta E}{\delta C} = f\mathbf{n}$, where $\mathbf{n}$ is the unit normal vector to C. The energy functional is minimized when $\frac{\delta E}{\delta C} = 0$.[1] Starting from some initial contour, the minimum can be approached by moving in the negative gradient direction. This is known as contour evolution. Defining a time parameter t, the change in the contour is $\frac{\partial C}{\partial t} = -f\mathbf{n}$.[2]

We represent the evolving contour as the zero level set of a function $\varphi(\mathbf{x}; t)$. The level set function satisfies the following properties: $\varphi(\mathbf{x}; t) < 0$ for $\mathbf{x} \in R(t)$, $\varphi(\mathbf{x}; t) > 0$ for $\mathbf{x} \in R^c(t)$, and of course $\varphi(\mathbf{x}; t) = 0$ for $\mathbf{x}$ on the contour $C(t)$. Evolving the contour is equivalent to updating the level set function. The level set update to minimize (1) is:

$$\varphi_t(\mathbf{x}) = f(\mathbf{x})\mathbf{n}(\mathbf{x}). \tag{2}$$

The energy functional can be written with φ instead of C as the argument.

The properties of the level set function given above are quite unconstrained. The level set function is often specialized to be the signed distance function, satisfying the additional constraint that $|\nabla\varphi(\mathbf{x})| = 1$. The magnitude of the signed distance function at a point $\mathbf{x}$ equals the distance from $\mathbf{x}$ to C, and its sign indicates whether it is in R or R^c. The signed distance function is used because it is well-behaved with respect to calculating the normal $\mathbf{n}$ and other geometric quantities. As discussed in Sec. 2.2, the signed distance function is also intimately related to classification margin.

It should be noted that level set update (2) does not take the constraint $|\nabla\varphi| = 1$ into account. The result of updating a signed distance function using (2) is not a signed distance function in general. A level set function may be reinitialized as a signed distance function iteratively through the Eikonal partial differential equation $|\nabla\varphi| = 1$.

2.2. Supervised Discriminative Classification

Having given a brief description of using level set methods to minimize energy functionals, we now come to the problem of supervised discriminative training of classifiers. Consider the training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_M, y_M)\}$, with feature vectors $\mathbf{x}_m \in \Omega \subset \mathbb{R}^D$ and labels $y_m \in \{-1, +1\}$. A classifier is a mapping from feature vectors to labels, $h : \Omega \to \{-1, +1\}$. Also consider the loss function $\mathrm{L}(z)$, where z is referred to as the *margin*.[3]

The margin of an example in the training set is its distance to the classifier decision boundary. The sign of the margin is positive if the example is on the 'right' side of the boundary and negative if the example is on the 'wrong' side. That is, z_m, the margin of example m, is positive if $h(\mathbf{x}_m) = y_m$ and negative if $h(\mathbf{x}_m) \neq y_m$. The classifier can thus be written as $h(\mathbf{x}_m) = \operatorname{sign}(y_m z_m)$. From this description of margin and the description of signed distance function in Sec. 2.1, it is apparent that the following equalities hold: $z_m = y_m\varphi(\mathbf{x}_m)$; $\mathrm{L}(z_m) = \mathrm{L}(y_m\varphi(\mathbf{x}_m))$; $\varphi(\mathbf{x}_m) = y_m z_m$; and $h(\mathbf{x}_m) = \operatorname{sign}(\varphi(\mathbf{x}_m))$.

The objective of discriminative training is to find the decision boundary that minimizes the sum of the loss in the training set. Based on the observation relating the signed distance function to classifier margin, the objective can be written as an energy functional like (1) with a particular $f(\mathbf{x})$. The energy functional is:

$$E(C) = \int_{\Omega = R + R^c} f(\mathbf{x}) d\mathbf{x} = \sum_{m=1}^{M} \mathrm{L}(y_m\varphi(\mathbf{x}_m)). \tag{3}$$

The level set update equation is:

$$\varphi_t(\mathbf{x}_m) = -\operatorname{sign}(\varphi(\mathbf{x}_m))\,\mathrm{L}(y_m\varphi(\mathbf{x}_m))\mathbf{n}(\mathbf{x}_m). \tag{4}$$

To learn the classifier, we start with some initial contour or equivalently signed distance function and evolve it according to the level set update equation. Frequently reinitializing the level set function to a signed distance function is more important here than with other energy functionals because the energy functional depends on the actual value of the margin. The procedure is implemented on a D-dimensional grid with values of $\varphi(\mathbf{x}_m)$ obtained by interpolation.

---

[1] The equation $\frac{\delta E}{\delta C} = 0$ is an Euler-Lagrange partial differential equation.

[2] Note that for energy functionals over the region R^c, i.e. $E(C) = \int_{R^c} f(\mathbf{x}) d\mathbf{x}$, the first variation $\frac{\delta E}{\delta C}$ is $-f\mathbf{n}$ and the change in the contour is $\frac{\partial C}{\partial t} = f\mathbf{n}$.

[3] Examples of loss functions include:

$$\mathrm{L}_{\text{zero-one}}(z) = \begin{cases} 1, & z < 0 \\ 0, & z \geq 0 \end{cases}$$

$$\mathrm{L}_{\text{hinge}}(z) = \begin{cases} 1 - z, & z < 1 \\ 0, & z \geq 1 \end{cases}$$

$$\mathrm{L}_{\text{logistic}}(z) = \log(1 + e^{-z})$$

$$\mathrm{L}_{\text{exponential}}(z) = e^{-z}.$$

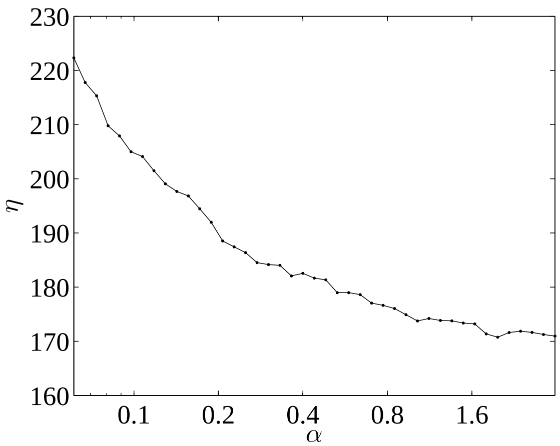


Fig. 2. Controlling classifier complexity using a contour area penalty. The plot shows the estimated VC dimension η as a function of the weight given to the contour area penalty α on a logarithmic scale.

3. STRUCTURAL RISK MINIMIZATION PRINCIPLE

The goal of learning is not good classification performance on training examples, but good performance on unseen examples. The real criterion is not training error, but generalization error. The structural risk minimization principle specifies a tradeoff between training error and classifier complexity measured using the Vapnik-Chervonenkis (VC) dimension [4]. The principle begs the question: how can the complexity of a level set classifier be controlled?

Since any shape (up to the discretization in the grid used in implementation) can be represented using the level set function, overfitting can be an issue. One way to prevent overfitting is to regularize the energy functional (3) with a contour area penalty as follows:

$$E_{\text{reg}}(C) = \sum_{m=1}^{M} \mathrm{L}(y_m \varphi(\mathbf{x}_m)) + \alpha \oint_C d\mathbf{s}, \qquad (5)$$

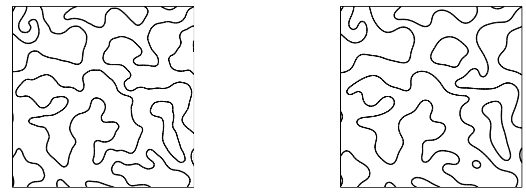
where $d\mathbf{s}$ is an infinitesimal area element on the contour C and α is the weight given to the regularization term. Working through the first variation of the functional, it can be shown that the level set update equation is:

$$\varphi_t(\mathbf{x}_m) = (-\operatorname{sign}(\varphi(\mathbf{x}_m))\, \mathrm{L}(y_m \varphi(\mathbf{x}_m)) + \alpha\kappa(\mathbf{x}_m))\, \mathbf{n}(\mathbf{x}_m), \qquad (6)$$

where κ is the mean curvature of the contour.

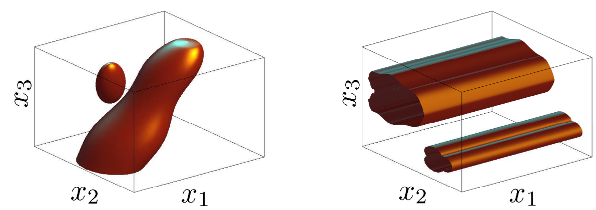
We can empirically measure the VC dimension of classifiers with the contour area penalty using the procedure outlined in [12]. Using $\mathrm{L}_{\text{zero-one}}(\cdot)$ and random initializations, we perform contour evolution to minimize (5) on ten sets of randomly generated training sets with 3000 positive and 3000 negative examples and $D = 2$. Carrying this out for several values of the regularization weight α, estimating the VC dimension using the calculation of [12], and averaging over the ten trials gives a plot of estimated VC dimension η as a function of α. The grid is eighty by eighty, and similar results are obtained for other loss functions.

The relationship between η and α, shown in Fig. 2 is nearly monotonic. Fig. 3 shows the classifiers for smaller and larger values of α corresponding to one instance of the random training set and one initialization. The smoother contour corresponding to the larger value of α can shatter fewer points. In this section, it has been shown empirically that the complexity of the classifier measured by VC dimension can be directly controlled using the weight on a contour area regularization term.



(a) (b)

Fig. 3. Classifiers learned from one instance of a random training set for (a) smaller and (b) larger values of α used to estimate VC-dimension by the procedure of [12].



(a) (b)

Fig. 4. Decision boundary that (a) uses all features, and (b) selects the two features x_2 and x_3 for classification. Note that the figures show the decision boundary contours, not the level set function.

4. FEATURE SELECTION

In feature spaces where some of the dimensions are irrelevant for classification, feature subset selection is important to prevent overfitting [7]. The idea is to learn classifiers which only make use of the relevant dimensions. Linear decision boundaries in a D-dimensional space can be specified with a length D vector of coefficients; feature selection can be formulated through the preference that this coefficient vector be sparse, i.e. have few nonzero elements. An ℓ_1-norm penalty is well known for producing sparse solutions as well as being tractable. In this section, we extend the idea of ℓ_1-based feature selection for linear decision boundaries to decision boundaries represented by level set functions.

First, let us consider what it means for a classifier to use or not use a feature. As seen in Fig. 4, a classifier that does not use a particular feature is a cylinder with any cross-section whose axis is parallel to that feature dimension. In other words, the decision boundary is constant and does not change as a function of the unused feature x_i. The partial derivative of the level set function with respect to the unused feature $\varphi_{x_i}(\mathbf{x})$ is zero for all $\mathbf{x} \in \Omega$.

If $\varphi_{x_i}(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Omega$, then the scalar value:

$$\int_\Omega |\varphi_{x_i}(\mathbf{x})| d\mathbf{x}$$

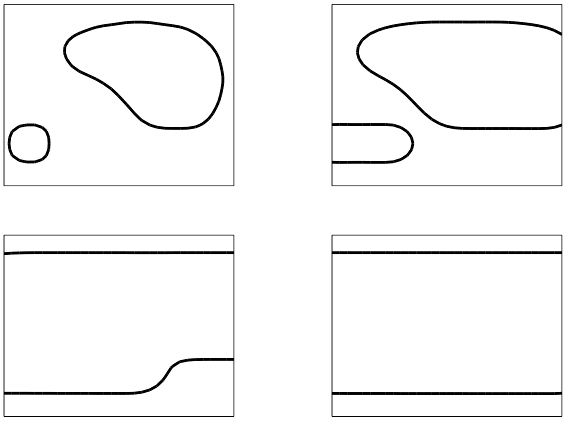


Fig. 5. Contour evolution with the contour area penalty and one of the partial derivative terms for feature selection. The evolution from the initial contour to the final contour is shown in raster scan order. For this illustration, the energy functional contains no training loss term. The final contour is a cylinder.

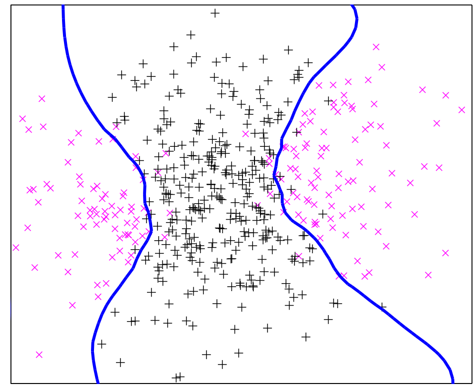


Fig. 6. Classifier learned from 500 examples using zero-one loss. The magenta $\times$ markers are training examples with label $+1$ and the black $+$ markers are training examples with label -1 plotted on the x_1–x_2 plane. The blue lines are the classifier decision boundaries.

equals zero. Consequently, a length D vector:

$$\begin{bmatrix} \int_\Omega |\varphi_{x_1}(\mathbf{x})| d\mathbf{x} \\ \vdots \\ \int_\Omega |\varphi_{x_D}(\mathbf{x})| d\mathbf{x} \end{bmatrix}$$

may be constructed, which should be sparse for feature subset selection.

Applying the ℓ_1-norm to this vector and appending it to (5), gives the following energy functional:

$$E_{\text{fs}}(C) = E_{\text{reg}}(C) + \beta \sum_{i=1}^{D} \int_\Omega |\varphi_{x_i}(\mathbf{x})| d\mathbf{x}, \tag{7}$$

where β is the weight given to the feature selection term. The level set function update is readily obtained:

$$\varphi_t(\mathbf{x}_m) = -\operatorname{sign}(\varphi(\mathbf{x}_m))\, \mathrm{L}(y_m \varphi(\mathbf{x}_m)) \mathbf{n}(\mathbf{x}_m) + \left(\alpha\kappa(\mathbf{x}_m) + \beta \sum_{i=1}^{D} |\varphi_{x_i}(\mathbf{x}_m)| \right) \mathbf{n}(\mathbf{x}_m). \tag{8}$$

The above contour evolution may be used for feature subset selection integrated with classifier training in the same way as, for example, ℓ_1-regularized logistic regression for linear decision boundaries [7]. In Fig. 5, we show contour evolution from an initial contour with the energy functional containing one of the D partial derivative feature selection terms and containing no training loss term. The final contour is a cylinder, as in Fig. 4b.

5. BASE CLASSIFIERS FOR BOOSTING

Boosting and in particular the AdaBoost algorithm is a way to sequentially design ensemble classifiers from a set of base classifiers [13]. We propose the use of level set classifiers as base classifiers for boosting. Boosting is quite general with few requirements on the base classifiers.

The first requirement is the ability to weight the examples in the training set. The energy functional (3) is easily modified with non-negative weights w_m that sum to one:

$$\sum_{m=1}^{M} M w_m \, \mathrm{L}(y_m \varphi(\mathbf{x}_m)). \tag{9}$$

The second requirement is that there should be an ordering of the base classifiers.

In most instances of boosting, the base classifiers are so-called 'weak learners' and counting their votes is a way to make a strong classifier. Here, the motivation is not so much due to the base classifiers being weak, but that implementation of level set methods for large feature space dimension D becomes cumbersome due to the need to store and update a D-dimensional grid. Consequently, the base classifiers that we employ are level set classifiers which use $d < D$ dimensions of the feature space. We randomly sample the dimensions without replacement to obtain an ordering. Proceeding through this order in sets of cardinality d gives the base classifiers for the sequential learning of the ensemble classifier by boosting.

6. EXAMPLES

In this section, we give a few examples of classifiers learned from both generated and real data, including with the integrated feature selection. The real data comes from geology and geophysics; the two classes are sandstone and shale, and the features are rock properties. We also compare the ten-fold cross-validation error of the level set classifier described in this paper with other classifiers on two standard datasets from the UCI machine learning repository [14].

The first training set consists of 500 examples of generated two-dimensional data, as shown in Fig. 6. A linear decision boundary would clearly be unable to well-classify this type of data. The classifier that is learned using the zero-one loss in (5), also shown in

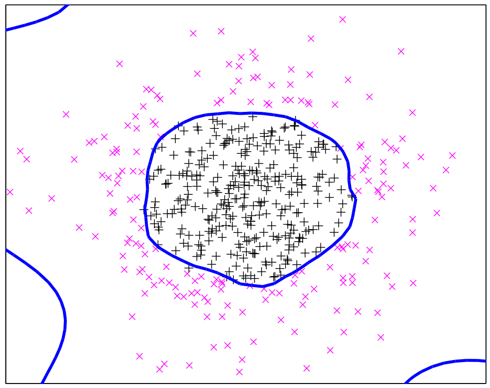


Fig. 7. Classifier learned from 500 examples using logistic loss. The magenta $\times$ markers are training examples with label $+1$ and the black $+$ markers are training examples with label -1 plotted on the x_1–x_2 plane. The blue lines are the classifier decision boundaries.

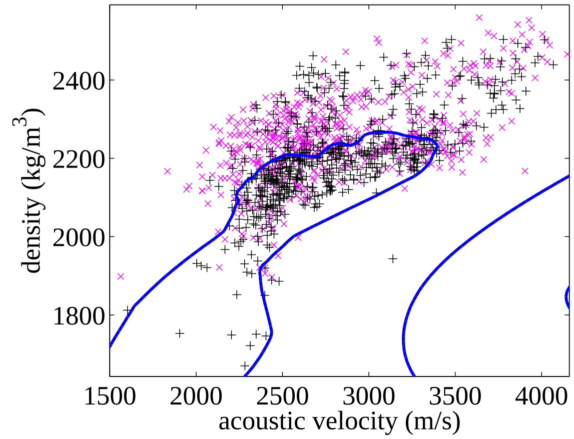


Fig. 8. Classifier learned from 1210 examples using logistic loss. The magenta $\times$ markers are training examples with label *shale* and the black $+$ markers are training examples with label *sandstone* plotted on the acoustic velocity–density plane. The blue lines are the classifier decision boundaries.

Fig. 6, partitions the domain into three regions, the outer two corresponding to label $+1$ and the inner one corresponding to label -1.

The second training set is also 500 examples of generated two-dimensional data, shown in Fig. 7. A classifier is learned using the logistic loss function. The circular boundary matches well with the training examples. There are also some decision boundaries at the corners of the domain where there are no training examples. These boundaries have little effect on classifier performance.

The third training set consists of real geologic data. The two features are density and acoustic velocity, i.e. the speed at which sound waves propagate through the material. The data was acquired with wireline logs in an offshore oilfield, along with various other measurements. On the basis of the other features, the examples have been assigned a rock type: sandstone and shale. The training set is shown in Fig. 8. A classifier is learned on this set using the logistic loss function as well. The decision boundary that is learned is fairly complex, but not overly so. The large curved boundary on the bottom right does not affect training error, but does affect the logistic loss; its presence reduces the loss incurred by misclassified examples.

We now show the effect of training with the ℓ_1 feature selection term of Sec. 4 included. A classifier, shown in Fig. 9, is learned on the geologic data set, again with logistic loss. It can be noted that the classifier is not fully a cylinder like the final contour of Fig. 5, but is much more cylindrical than the classifier without feature selection given in Fig. 8. Density is a more relevant feature than acoustic velocity in distinguishing shale from sandstone, but acoustic velocity is not completely irrelevant either.

The final example looks at using the feature selection term when all of the features are completely relevant. We train a classifier on the same generated training set as above that produced the circular decision boundary in Fig. 7, but this time with the ℓ_1 feature selection. The learned classifier, shown in Fig. 10, is nearly identical to the classifier learned without feature selection. This example shows that training with feature selection does not eliminate features that are important for classification performance. One difference between the classifier learned with feature selection and without feature selection is that the artifactual decision boundaries on the corners are not present with feature selection. By removing these corner boundaries, the value of the ℓ_1 term is decreased with very little change to the loss term and contour area term.

We now look at the testing performance of classifiers designed by contour evolution to minimize (5). Two standard datasets are considered, the Pima Indians diabetes dataset and the Wisconsin diagnostic breast cancer dataset [14]. Test performance is measured using ten-fold cross-validation. AdaBoost with $d = 2$, as described in Sec. 5, is used with logistic loss. The regularization parameter α is manually set to 0.4 beforehand for both datasets. A principled method of selecting the parameter such as by cross-validation may be used in future work; hence there is room for improvement in the results.

We perform four rounds of boosting on the eight-dimensional feature space of the Pima Indians diabetes dataset, obtaining 25.95% ten-fold cross-validation error. With fifteen boosting rounds on the thirty-dimensional feature space of the Wisconsin diagnostic breast cancer dataset, the ten-fold cross-validation error is 6.68%. These preliminary results are competitive with other classifiers. The error percentages reported in [9] for other classifiers on the two datasets are: (23.69%, 7.02%) for the naïve Bayes classifier, (25.64%, 4.92%) for the Bayes net classifier, (27.86%, 3.68%) for the k-nearest neighbor classifier with inverse distance weighting, (27.33%, 7.20%) for the C4.4 decision tree, (26.17%, 6.85%) for the C4.5 decision tree, (25.64%, 7.21%) for the naïve Bayes tree classifier, (22.66%, 2.28%) for the SVM classifier with polynomial kernel, (24.60%, 5.79%) for the radial basis function network classifier, and (29.94%, 6.50%) for the level learning set classifier of [9].

7. CONCLUSION

In this paper, a novel, nonlinear, nonparametric classifier has been developed that minimizes margin-based training loss. The approach, a type of level set segmentation of the feature space, can incorporate any margin-based loss function. Any shape decision boundary can

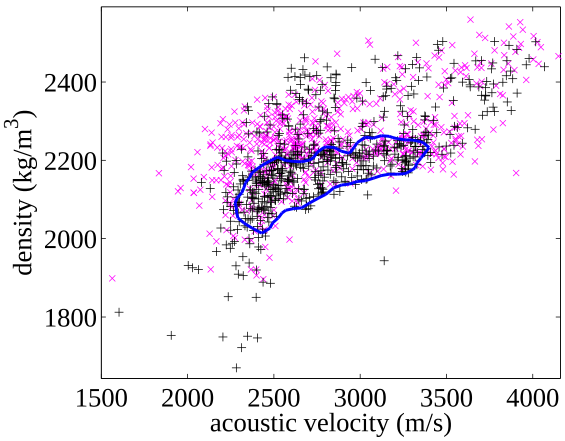


Fig. 9. Classifier learned from 1210 examples using logistic loss and integrated feature selection. The magenta $\times$ markers are training examples with label *shale* and the black $+$ markers are training examples with label *sandstone* plotted on the acoustic velocity–density plane. The blue lines are the classifier decision boundaries.

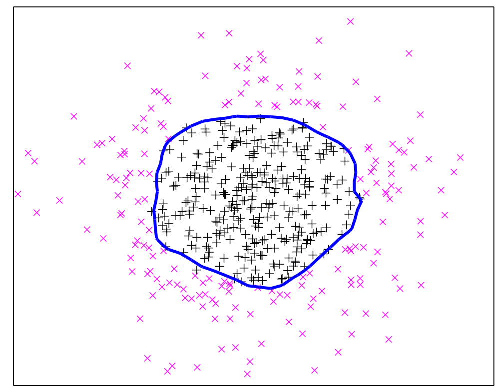


Fig. 10. Classifier learned from 500 examples using logistic loss and integrated feature selection. The magenta $\times$ markers are training examples with label $+1$ and the black $+$ markers are training examples with label -1 plotted on the x_1–x_2 plane. The blue lines are the classifier decision boundaries.

be obtained; the complexity of the classifier can be controlled using contour area regularization. Feature subset selection may be incorporated into the level set method and level set classifiers may be used in boosting.

The preliminary test results indicate that performance on par with various other classifiers can be obtained. Further improvements are possible by choosing parameters in a principled way, choosing the order of the features for the boosting rounds in a principled way, and trying various other loss functions.

It is a known fact that no one type of classifier is best for all types of datasets. Performance of classifiers may vary quite a bit depending on the data characteristics. Some say that choosing classifiers is more of an art than a science; here we have given the artist another color in the palette. The ability to plug in various loss functions within a common framework and incorporate different geometric preferences about the decision boundary may be useful in this regard as well.

8. ACKNOWLEDGMENT

The authors thank Dr. J. W. Fisher III for valuable input and Shell International Exploration and Production for providing the geologic dataset.

9. REFERENCES

[1] S. Osher, "Level set methods," in *Geometric Level Set Methods in Imaging, Vision, and Graphics*, S. Osher and N. Paragios, Eds., pp. 3–20. Springer-Verlag, New York, 2003.

[2] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[3] B. Efron, "The efficiency of logistic regression compared to normal discriminant analysis," *J. Am. Stat. Assoc.*, vol. 70, no. 352, pp. 892–898, Dec. 1975.

[4] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[5] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2002.

[6] Y. Lin, "A note on margin-based loss functions in classification," *Stat. Prob. Letters*, vol. 68, no. 1, pp. 73–82, June 2003.

[7] A. Y. Ng, "Feature selection, L_1 vs. L_2 regularization, and rotational invariance," in *Proc. Int. Conf. Machine Learning*, Banff, Canada, 2004, pp. 79–90.

[8] A. Tomczyk and P. S. Szczepaniak, "On the relationship between active contours and contextual classification," in *Proc. Int. Conf. Computer Recognition Systems*, Rydzyna, Poland, 2005, pp. 303–310.

[9] X. Cai and A. Sowmya, "Level learning set: A novel classifier based on active contour models," in *Proc. European Conf. Machine Learning*, Warsaw, Poland, 2007, pp. 79–90.

[10] A. M. Yip, C. Ding, and T. F. Chan, "Dynamic cluster formation using level set methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 6, pp. 877–889, June 2006.

[11] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, England, 1996.

[12] V. Vapnik, E. Levin, and Y. Le Cun, "Measuring the VC-dimension of a learning machine," *Neural Computation*, vol. 6, pp. 851–876, 1994.

[13] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. System Sci.*, vol. 50, no. 1, pp. 119–139, Aug. 1997.

[14] A. Asuncion and D. J. Newman, "UCI machine learning repository," 2007.