Chapter 1

Rise of the Machines

Larry Wasserman

On the 50th anniversary of the Committee of Presidents of Statistical Societies I reflect on the rise of the field of Machine Learning and what it means for Statistics. Machine Learning offers a plethora of new research areas, new applications areas and new colleagues to work with. Our students now compete with Machine Learning students for jobs. I am optimistic that visionary Statistics departments will embrace this emerging field; those that ignore or eschew Machine Learning do so at their own risk and may find themselves in the rubble of an outdated, antiquated field.

1.1 Introduction

Statistics is the science of learning from data. Machine Learning (ML) is the science of learning from data. These fields are identical in intent although they differ in their history, conventions, emphasis and culture.

There is no denying the success and importance of the field of Statistics for science and, more generally, for society. I'm proud to be a part of the field. The focus of this essay is on one challenge (and opportunity) to our field: the rise of Machine Learning.

During my twenty-five year career I have seen Machine Learning evolve from being a collection of rather primitive (yet clever) set of methods to do classification, to a sophisticated science that is rich in theory and applications.

A quick glance at the *The Journal of Machine Learning Research* (`mlr.csail.mit.edu`) and NIPS (`books.nips.cc`) reveals papers on a variety of topics that will be familiar to Statisticians such as:

> conditional likelihood, sequential design, reproducing kernel Hilbert spaces, clustering, bioinformatics, minimax theory, sparse regression, estimating large covariance matrices, model selection, density estimation, graphical models, wavelets, nonparametric regression.

These could just as well be papers in our flagship statistics journals.

This sampling of topics should make it clear that researchers in Machine Learning — who were at one time somewhat unaware of mainstream statistical methods and theory — are now not only aware of, but actively engaged in, cutting edge research on these topics.

On the other hand, there are statistical topics that are active areas of research in Machine Learning but are virtually ignored in Statistics. To avoid becoming irrelevant, we Statisticians need to (i) stay current on research areas in ML and (ii) change our outdated model for disseminating knowledge and (iii) revamp our graduate programs.

1.2 The Conference Culture

ML moves at a much faster pace than Statistics. At first, ML researchers developed expert systems that eschewed probability. But very quickly they adopted advanced statistical concepts like empirical process theory and concentration of measure. This transition happened in a matter of a few years. Part of the reason for this fast pace is the conference culture. The main venue for research in ML is refereed conference proceedings rather than journals.

Graduate students produce a stream of research papers and graduate with hefty CV's. One of the reasons for the blistering pace is, again, the conference culture.

The process of writing a typical statistics paper goes like this: you have an idea for a method, you stew over it, you develop it, you prove some results about

it, and eventually you write it up and submit it. Then the refereeing process starts. One paper can take years.

In ML, the intellectual currency is conference publications. There are a number of deadlines for the main conference (NIPS, AISTAT, ICML, COLT). The threat of a deadline forces one to quit ruminating and start writing. Most importantly, all faculty members and students are facing the same deadline so there is a synergy in the field that has mutual benefits. No one minds if you cancel a class right before the NIPS deadline. And then, after the deadline, everyone is facing another deadline: refereeing each others papers and doing so in a timely manner. If you have an idea and don't submit a paper on it, then you may be out of luck because someone may scoop you.

This pressure is good; it keeps the field moving at a fast pace. If you think this leads to poorly written papers or poorly thought out ideas, I suggest you look at `nips.cc` and read some of the papers. There are some substantial, deep papers. There are also a few bad papers. Just like in our journals. The papers are refereed and the acceptance rate is comparable to our main journals. And if an idea requires more detailed followup, then one can always write a longer journal version of the paper.

Absent this stream of constant deadline, a field moves slowly. This is a problem for Statistics not only for its own sake but also because it now competes with ML.

Of course, there are disadvantages to the conference culture. Work is done in a rush, and ideas are often not fleshed out in detail. But I think that the advantages outweigh the disadvantages.

1.3 Neglected Research Areas

There are many statistical topics that are dominated by ML and mostly ignored by Statistics. This is a shame because Statistics has much to offer in all these areas. Examples include semisupervised inference, computational topology, online learning, sequential game theory, hashing, active learning, deep learning, differential privacy, random projections and reproducing kernel Hilbert spaces. Ironically, some of these — like sequential game theory an reproducing kernel Hilbert spaces — started in Statistics.

1.4 Case Studies

I'm lucky. I am at an institution which has a Machine Learning Department (within the School of Computer Science) and, more importantly, the ML department welcomes involvement by Statisticians. So I've been fortunate to work with colleagues in ML, attend their seminars, work with ML students and teach courses in the ML department.

There are a number of topics I've worked on at least partly due to my association with ML. These include, statistical topology, graphical models, semisu-

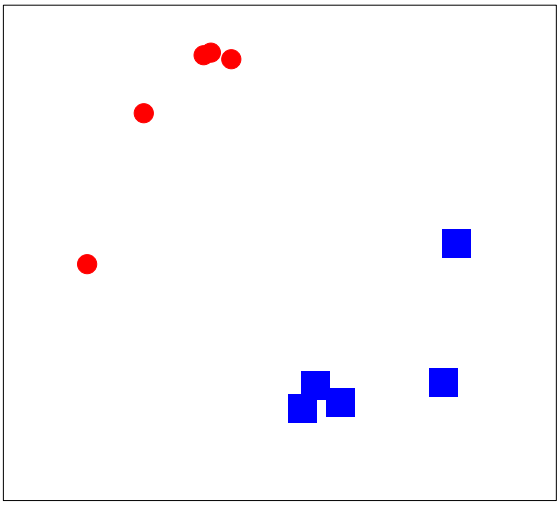

Figure 1.1: Labeled data.

pervised inference, conformal prediction, and differential privacy.

Since this paper is supposed to be a personal reflection, let me now briefly discuss two of these ML problems that I have had the good fortune to work on. The point of these examples is to show how statistical thinking can be useful for Machine Learning.

1.4.1 Case Study I: Semisupervised Inference

Suppose we observe data $(X_1, Y_1), \ldots, (X_n, Y_n)$ and we want to predict Y from X. If Y is discrete, this is a classification problem. If Y is real-valued, this is a regression problem. Further, suppose we observe more data $X_{n+1}, \ldots, X_N$ without the corresponding Y values. We thus have labeled data $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ and unlabeled data $\mathcal{U} = \{X_{n+1}, \ldots, X_N\}$. How do we use the unlabeled data in addition to the labeled data to improve prediction? This is the problem of *semisupervised inference*.

Consider Figure 1.1. The covariate is $x = (x_1, x_2) \in \mathbb{R}^2$. The outcome in this case is binary as indicated by the circles and squares. Finding the decision boundary using only the labeled data is difficult. Figure 1.2 shows the labeled data together with some unlabeled data. We clearly see two clusters. If we make the additional assumption that $P(Y = 1|X = x)$ is smooth relative to the clusters, then we can use the unlabeled data to nail down the decision boundary accurately.

There are copious papers with heuristic methods for taking advantage of unlabeled data. To see how useful these methods might be, consider the following example. We download one-million webpages with images of cats and dogs. We randomly select 100 pages and classify them by hand. Semisupervised methods

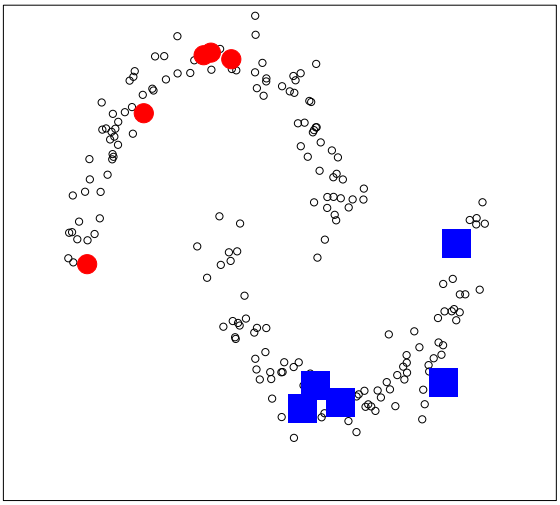

Figure 1.2: Labeled and unlabeled data.

allow us to use the other 999,900 webpages to construct a good classifier.

But does semisupervised inference work? Or, to put it another way, under what conditions does it work? In [1], we showed the following (which I state informally here).

Suppose that $X_i \in \mathbb{R}^d$. Let $\mathcal{S}_n$ denote the set of supervised estimators; these estimators use only the labeled data. Let $\mathcal{SS}_N$ denote the set of semisupervised estimators; these estimators use the labeled data and unlabeled data. Let m be the number of unlabeled data points and suppose that $m \geq n^{2/(2+\xi)}$ for some $0 < \xi < d-3$. Let $f(x) = \mathbb{E}(Y|X=x)$. There is a large, nonparametric class of distributions $\mathcal{P}_n$ such that the following is true:

1. There is a semisupervised estimator $\widehat{f}$ such that

$$\sup_{P \in \mathcal{P}_n} R_P(\widehat{f}) \leq \left(\frac{C}{n}\right)^{\frac{2}{2+\xi}} \tag{1.1}$$

   where $R_P(\widehat{f}) = \mathbb{E}(\widehat{f}(X) - f(X))^2$ is the risk of the estimator $\widehat{f}$ under distribution P.

2. For supervised estimators $\mathcal{S}_n$ we have

$$\inf_{\widehat{f} \in \mathcal{S}_n} \sup_{P \in \mathcal{P}_n} R_P(\widehat{f}) \geq \left(\frac{C}{n}\right)^{\frac{2}{d-1}}. \tag{1.2}$$

3. Combining these two results we conclude that

$$\frac{\inf_{\widehat{f} \in \mathcal{SS}_N} \sup_{P \in \mathcal{P}_n} R_P(\widehat{f})}{\inf_{\widehat{f} \in \mathcal{S}_n} \sup_{P \in \mathcal{P}_n} R_P(\widehat{f})} \leq \left(\frac{C}{n}\right)^{\frac{2(d-3-\xi)}{(2+\xi)(d-1)}} \to 0 \tag{1.3}$$

and hence, semisupervised estimation dominates supervised estimation.

The class $\mathcal{P}_n$ consists of distributions such that the marginal for X is highly concentrated near some lower dimensional set and such that the regression function is smooth on this set. We have not proved that the class must be of this form for semisupervised inference to improve on supervised inference but we suspect that is indeed the case. Our framework includes a parameter α that characterizes the strength of the semisupervised assumption. We showed that, in fact, one can use the data to adapt to the correct value of α.

1.4.2 Case Study II: Statistical Topology

Computational topologists and researchers in Machine Learning have developed methods for analyzing the shape of functions and data. Here I'll briefly review some of our work on estimating manifolds ([6, 7, 8]).

Suppose that M is a manifold of dimension d embedded in $\mathbb{R}^D$. Let $X_1, \ldots, X_n$ be a sample from a distribution in P supported on M. We observe

$$Y_i = X_i + \epsilon_i, \quad i = 1, \ldots, n \tag{1.4}$$

where $\epsilon_1, \ldots, \epsilon_n \sim \Phi$ are noise variables.

Machine Learning researchers have derived many methods for estimating the manifold M. But this leaves open an important statistical question: how well do these estimators work? One approach to answering this question is to find the minimax risk under some loss function. Let $\widehat{M}$ b an estimator of M. A natural loss function for this problem is Hausdorff loss:

$$H(M, \widehat{M}) = \inf\Big\{\epsilon : \ M \subset \widehat{M} \oplus \epsilon \ \text{ and } \ \widehat{M} \subset M \oplus \epsilon\Big\}. \tag{1.5}$$

Let $\mathcal{P}$ be a set of distributions. The parameter of interest is $M = \text{support}(P)$ which we assume is a d-dimensional manifold. The minimax risk is

$$R_n = \inf_{\widehat{M}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[H(\widehat{M}, M)]. \tag{1.6}$$

Of course, the risk depends on what conditions we assume on M and one the noise Φ.

Our main findings are as follows. When there is no noise — so the data fall on the manifold — we get $R_n \asymp n^{-2/d}$. When the noise is perpendicular to M, the risk is $R_n \asymp n^{-2/(2+d)}$. When the noise is Gaussian the rate is $R_n \asymp 1/\log n$. The latter is not surprising when one considers the similar problem of estimating a function when there are errors in variables.

The implications for Machine Learning are that, the best their algorithms can do is highly dependent on the particulars of the type of noise.

How do we actually estimate these manifolds in practice? In ([8]) we take the following point of view: if the noise is not too large, then the manifold should be close to a d-dimensional hyper-ridge in the density $p(y)$ for Y. Ridge finding is an extension of mode finding, which is a common task in computer vision.

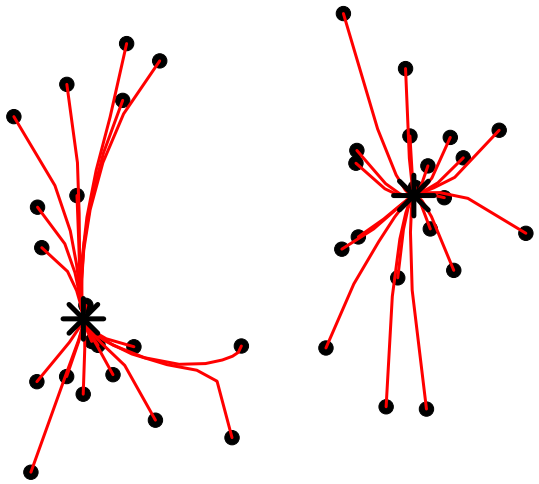

Figure 1.3: The Mean Shift Algorithm. The data points move along trajectories during iterations until they reach the two modes marked by the two large asterisks.

Let p be a density on $\mathbb{R}^D$. Suppose that p has k modes $m_1, \ldots, m_k$. An integral curve, or path of steepest ascent, is a path $\pi : \mathbb{R} \to \mathbb{R}^D$ such that

$$\pi'(t) = \frac{d}{dt}\pi(t) = \nabla p(\pi(t)). \tag{1.7}$$

Under weak conditions, the paths π partition the space and are disjoint except at the modes [9, 2].

The *mean shift algorithm* ([5, 3]) is a method for finding the modes of a density by following the steepest ascent paths. The algorithm starts with a mesh of points and then moves the points along gradient ascent trajectories towards local maxima. A simple example is shown in Figure 1.3.

Given a function $p : \mathbb{R}^D \to \mathbb{R}$, let $g(x) = \nabla p(x)$ denote the gradient at x and let $H(x)$ denote the Hessian matrix. Let

$$\lambda_1(x) \geq \lambda_2(x) \geq \cdots \geq \lambda_D(x) \tag{1.8}$$

denote the eigenvalues of $H(x)$ and let $\Lambda(x)$ be the diagonal matrix whose diagonal elements are the eigenvalues. Write the spectral decomposition of $H(x)$ as $H(x) = U(x)\Lambda(x)U(x)^T$. Fix $0 \leq d < D$ and let $V(x)$ be the last $D-d$ columns of $U(x)$ (that is, the columns corresponding to the $D-d$ smallest eigenvalues). If we write $U(x) = [V_\diamond(x) : V(x)]$ then we can write $H(x) = [V_\diamond(x) : V(x)]\Lambda(x)[V_\diamond(x) : V(x)]^T$. Let $L(x) = V(x)V(x)^T$ be the projector on the linear space defined by the columns of $V(x)$. Define the *projected gradient*

$$G(x) = L(x)g(x). \tag{1.9}$$

If the vector field $G(x)$ is Lipschitz then by Theorem 3.39 of [9], G defines a global flow as follows. The flow is a family of functions $\phi(x,t)$ such that $\phi(x,0) = x$ and $\phi'(x,0) = G(x)$ and $\phi(s,\phi(t,x)) = \phi(s+t,x)$. The flow lines, or integral curves, partition the space and at each x where $G(x)$ is non-null, there is a unique integral curve passing through x. The intuition is that the flow passing through x is a gradient ascent path moving towards higher values of p. Unlike the paths defined by the gradient g which move towards modes, the paths defined by G move towards ridges.

The paths can be parameterized in many ways. One commonly used parameterization is to use $t \in [-\infty, \infty]$ where large values of t correspond to higher values of p. In this case $t = \infty$ will correspond to a point on the ridge. In this parameterization we can express each integral curve in the flow as follows. A map $\pi : \mathbb{R} \to \mathbb{R}^D$ is an integral curve with respect to the flow of G if

$$\pi'(t) = G(\pi(t)) = L(\pi(t))g(\pi(t)). \tag{1.10}$$

Definition: The *ridge* R consists of the destinations of the integral curves: $y \in R$ if $\lim_{t\to\infty} \pi(t) = y$ for some π satisfying (1.10).

As mentioned above, the integral curves partition the space and for each $x \notin R$, there is a unique path π_x passing through x. The ridge points are zeros of the projected gradient: $y \in R$ implies that $G(y) = (0,\ldots,0)^T$. [10] derived an extension of the mean-shift algorithm, called the *subspace constrained mean shift* algorithm that finds ridges which can be applied to the kernel density estimator. Our results can be summarized as follows:

1. Stability. We showed that if two functions are sufficiently close together then their ridges are also close together (in Hausdorff distance).

2. We constructed an estimator $\widehat{R}$ such that

$$H(R,\widehat{R}) = O_P\left(\left(\frac{\log n}{n}\right)^{\frac{2}{D+8}}\right) \tag{1.11}$$

   where H is the Hausdorff distance. Further, we showed that $\widehat{R}$ is topologically similar to R. We also construct an estimator $\widehat{R}_h$ for $h > 0$ that satisfies

$$H(R_h,\widehat{R}_h) = O_P\left(\left(\frac{\log n}{n}\right)^{\frac{1}{2}}\right) \tag{1.12}$$

   where R_h is a smoothed version of R.

3. Suppose the data are obtained by sampling points on a manifold and adding noise with small variance σ^2. We showed that the resulting density p has a ridge R_σ such that

$$H(M,R_\sigma) = O\left(\sigma^2 \log^3(1/\sigma)\right) \tag{1.13}$$

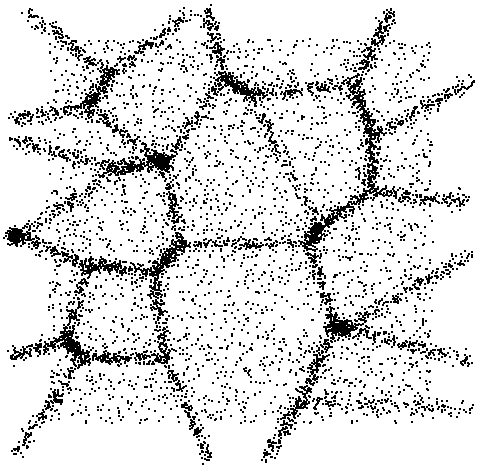

Figure 1.4: Simulated cosmic web data.

and R_σ is topologically similar to M. Hence when the noise σ is small, the ridge is close to M. It then follows that

$$H(M, \widehat{R}) = O_P\left(\left(\frac{\log n}{n}\right)^{\frac{2}{D+8}}\right) + O\left(\sigma^2 \log^3(1/\sigma)\right). \tag{1.14}$$

An example can be found in Figures 1.4 and 1.5. I believe that Statistics has much to offer to this area especially in terms of making the assumptions precise and clarifying how accurate the inferences can be.

1.5 Computational Thinking

There is another interesting difference that is worth pondering. Consider the problem of estimating a mixture of Gaussians. In Statistics we think of this as a solved problem. You use, for example, maximum likelihood which is implemented by the EM algorithm. But the EM algorithm does not solve the problem. There is no guarantee that the EM algorithm will actually find the MLE; it's a shot in the dark. The same comment applies to MCMC methods.

In ML, when you say you've solved the problem, you mean that there is a polynomial time algorithm with provable guarantees. There is, in fact, a rich literature in ML on estimating mixtures that do provide polynomial time algorithms. Furthermore, they come with theorems telling you how many observations you need if you want the estimator to be a certain distance from the truth, with probability at least $1 - \delta$. This is typical for what is expected of an estimator in ML. You need to provide a provable polynomial time algorithm and a finite sample (non-asymptotic) guarantee on the estimator.

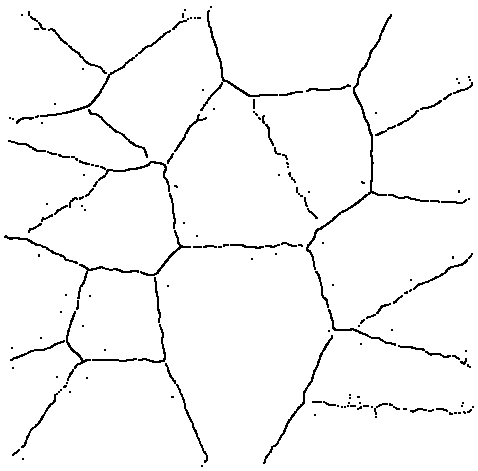

Figure 1.5: Ridge finder applied to simulated cosmic web data.

ML puts heavier emphasis on computational thinking. Consider, for example, the difference between P and NP problems. This is at the heart of theoretical Computer Science and ML. Running an MCMC on an NP hard problem is often meaningless. Instead, it is usually better to approximate the NP problem with a simpler problem. How often do we teach this to our students?

1.6 The Evolving Meaning of Data

For most of us in Statistics, data means numbers. But data now includes images, documents, videos, web pages, twitter feeds and so on. Traditional data — numbers from experiments and observational studies — are still of vital importance but they represents a tiny fraction of the data out there. If we take the union of all the data in the world, what fraction is being analyzed by statisticians? I think it is a small number.

This comes back to education. If our students can't analyze giant datasets like millions of twitter feeds or millions of web pages then other people will analyze those data. We will end up with a small cut of the pie.

1.7 Education and Hiring

The goal of a graduate student in Statistics is to find and advisor and write a thesis. They graduate with a single data point: their thesis work.

The goal of a graduate student in ML is to find a dozen different research problems to work on and publish many papers. They graduate with a rich data set: many papers on many topics with many different people.

Having been on hiring committees for both Statistics and ML I can say that the difference is striking. It is easy to choose candidates to interview in ML. You have a lot of data on each candidate and you know what you are getting. In Statistics, it is a struggle. You have little more than a few papers that bear their advisor's footprint.

The ML conference culture encourages publishing many papers on many topics which is better for both the students and their potential employers. And now, Statistics students are competing with ML students, putting Statistics students at a significant disadvantage.

There are a number of topics that are routinely covered in ML that we rarely teach in Statistics. Examples are: Vapnik-Chervonenkis theory, concentration of measure, random matrices, convex optimization, graphical models, reproducing kernel Hilbert spaces, support vector machines, and sequential game theory. It is time to get rid of antiques like UMVUE, complete statistics and so on and teach modern ideas.

1.8 If You Can't Beat Them, Join Them

I don't want to leave the reader with the impression that we are in some sort of competition with ML. Instead, we should feel blessed that a second group of Statisticians has appeared. Working with ML and adopting some of their ideas enriches both fields.

ML has much to offer Statistics. And Statisticians have a lot to offer ML. For example, we put much emphasis on quantifying uncertainty (standard errors, confidence intervals, potserior distributions), an emphasis that is perhaps lacking in ML. And sometimes, statistical thinking casts new light on existing ML methods. A good example is the statistical view of boosting given in [4]. I hope we will see collaboration and cooperation between the two fields thrive in the years to come.

Acknowledgements: I'd like to thank Kathryn Roeder, Rob, Tibshirnai, Ryan Tibshirani and Isa Verdinelli for reading a draft of this essay and providing helpful suggestions.

Bibliography

[1] M. Azizyan, A. Singh, and L. Wasserman. Density-sensitive semisupervised inference. *The Annals of Statistics*, 2013.

[2] Chacón. Clusters and water flows: a novel approach to modal clustering through morse theory. *arXiv preprint arXiv:1212.1384*, 2012.

[3] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603 –619, may 2002.

[4] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000.

[5] Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21:32–40, 1975.

[6] Christopher R. Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Manifold estimation and singular deconvolution under hausdorff loss. *The Annals of Statistics*, 40:941–963, 2012.

[7] Christopher R. Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Minimax manifold estimation. *Journal of Machine Learning Research*, pages 1263–1291, 2012.

[8] C.R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Nonparametric ridge estimation. *arXiv preprint arXiv:1212.5156*, 2012.

[9] M.C. Irwin. *Smooth dynamical systems*, volume 94. Academic Press, 1980.

[10] Ozertem and Erdogmus. Locally defined principal curves and surfaces. *Journal of Machine Learning Research*, 12:1249–1286, 2011.