A new perspective on machine learning

H. N. Mhaskar

Claremont Graduate University

ICERM Scientific Machine Learning
January 28, 2019
My understanding of machine learning problem and its traditional solution.

What bothers me about this.

My own efforts to remedy the problems
  ▶ Diffusion geometry based approach
  ▶ Application to diabetic sugar level prediction
  ▶ Problems
  ▶ Hermite polynomial based approach

Applications
Problem of machine learning

Given data (training data) of the form \( \{(x_j, y_j)\}_{j=1}^{M} \), where \( y_j \in \mathbb{R} \), and \( x_j \)'s are in some Euclidean space \( \mathbb{R}^q \), find a function \( P \) on a suitable domain

- that models the data well;
- in particular, \( P(x_j) \approx y_j \).
1. Traditional paradigm
Basic set up

\{(x_j, y_j)\} are i.i.d. samples from an unknown probability distribution \(\mu\).

\(f(x) = \mathbb{E}_\mu(y|x)\), target function

\(\mu^* =\) marginal distribution of \(x\).

\(X =\) support of \(\mu^*\).

\(V_n \subset V_{n+1} \subset \cdots =\) classes of models, \(V_n\) with complexity \(n\) (typically, number of parameters).
Traditional methodology

- Assume \( f \in W_\gamma \) (smoothness class, prior, RKHS).
- Estimate \( E_n(f) = \inf_{P \in V_n} \| f - P \|_{L^2(\mu^*)} = \| f - P^* \|_{L^2(\mu^*)} \).
  Decide upon the right value of \( n \).
- Find

\[
P# = \arg \min_{P \in V_n} \left\{ \text{Loss}\{y_\ell - P(x_\ell)\} + \lambda \| P \|_{W_\gamma} \right\}.
\]
Generalization error

\[
\int_{X \times \mathbb{R}} |y - P^\#(x)|^2 d\mu(y, x) = \int_{X \times \mathbb{R}} |y - f(x)|^2 d\mu(y, x)
\]

**generalization error**

\[= \int_{X \times \mathbb{R}} |y - f(x)|^2 d\mu(y, x)\]

\[+ \| f - P^* \|_{L^2(\mu^*)}^2 \]

**variance**

\[+ \| f - P^\# \|_{L^2(\mu^*)}^2 - \| f - P^* \|_{L^2(\mu^*)}^2 \]

**Approximation error**

**Sampling error**

Only the approximation error and sampling error can be controlled.
Observations on the paradigm

- Too complicated.
  - Bounds of approximation error are often obtained by explicit constructions. The approach makes no use of these constructions.
  - Measuring errors in $L^2$ with function values makes sense only if $f$ is in some RKHS. So, the method is not universal.
Observations on the paradigm

Good is better than best
On the left, the log–plot of the absolute error between the function $x \mapsto |\cos x|^{1/4}$, and its Fourier projection. On the right the corresponding plot with the trigonometric polynomial obtained by our summability operator. This is based on 128 equidistant samples. The order of the trigonometric polynomials is 31 in each case. The numbers on the $x$ axis are in multiples of $\pi$, the actual absolute errors are $10^y$. 
Observations on the paradigm

- The choice of penalty functional/loss functional, kernels, etc. are often ad hoc, and assume a prior on the target function.
- Performance guarantees on new and unseen data are often not easy to obtain, sometimes impossible.
- The optimization algorithms might not converge or converge too slowly.
- The paradigm does not work in the context of deep learning.
The number of parameters required to get a generalization error of \( \epsilon \) is at least constant times

\[ \epsilon^{-\gamma/q}, \]

\( \gamma = \text{smoothness of } f, \ q = \text{number of input variables}. \)

\( ^{1} \text{Donoho 2000, DeVore Howard, Micchelli, 1989} \)
Approximate

\[ F(x_1, \cdots, x_4) = f(f_1(x_1, x_2), f_2(x_3, x_4)) \]

by

\[ Q(x_1, \cdots, x_4) = P(P_1(x_1, x_2), P_2(x_3, x_4)). \]

Only functions of 2 variables are involved at each stage.

\(^2\text{Mh., Poggio, 2016}\)
How to measure generalization error

\[ \int |f(f_1(x_1, x_2), f_2(x_3, x_4)) - P(P_1(x_1, x_2), P_2(x_3, x_4))|^2 d\mu(x_1, x_2, x_3, x_4) \]

\( \mu \) ignores compositionality

\[ \int |f(f_1, f_2) - P(P_1, P_2)|^2 d\nu(?) \]

The distributions of \((f_1, f_2)\) and \((P_1, P_2)\) are different. Must have a different notion of generalization error.\(^3\)

\(^3\)Mh., Poggio, 2016
A new look

Given data (training data) of the form \( \{(x_j, y_j)\}_{j=1}^{M} \), where \( y_j \in \mathbb{R} \), and \( x_j \)'s are in some Euclidean space \( \mathbb{R}^{q+1} \).

▶ Assume that there is an underlying target function \( f : \mathbb{X} \to \mathbb{R} \), such that

\[
y_j = f(x_j) + \epsilon_j.
\]

▶ No priors, just continuity.

▶ Use approximation theory to construct the approximation \( P \).
Objectives

- Universal approximation with no assumptions on prior.
- Generalization error defined pointwise, and adjusts itself per local smoothness.
- Optimization is substantially easier.
- Can be adapted to deep learning easily.
Problem Classical approximation theory results are not adequate.

- Data distributed densely on a known domain, cube, sphere, etc.
- The points $x_j$ need to be chosen judiciously; e.g., Driscoll-Healy points on the sphere or quadrature nodes on the cube, etc.
2. Diffusion geometry based construction
Data \{x_j\} i.i.d. sample from a distribution \(\mu^*\) from a smooth compact manifold \(\mathbb{X}\) (unknown).
\(\{\phi_k\}\) a system of eigenfunctions of a suitable PDE with eigenvalues \(\{\lambda_k\}\).
\(\phi_k, \lambda_k\)'s are computed approximately from a “graph Laplacian” \(^4\)

Set up

Data \{x_j\} i.i.d. sample from a distribution \( \mu^* \) from a smooth compact manifold \( \mathbb{X} \) (unknown).

\( \{\phi_k\} \) a system of eigenfunctions of a suitable PDE with eigenvalues \( \{\lambda_k\} \).

\( \Pi_n = \text{span}\{\phi_k : \lambda_k < n\} \).

\( \|f\| = \sup_{x \in \mathbb{X}} |f(x)| \),

\[
\|f\|_\gamma = \|f\| + \sup_{n \geq 1} n^\gamma \text{dist}(f, \Pi_n).
\]

\( \gamma \) is the smoothness of \( f \).
Construction

\( h: \) a smooth low pass filter (even, \( = 1 \) on \([0, 1/2]\), \( = 0 \) on \([1, \infty)\)).

\[
\Phi_n(x, y) = \sum_{0 \leq k < n} h\left(\frac{\lambda_k}{n}\right) \phi_k(x)\phi_k(y).
\]

Fact: If \( x_j \)'s are sufficiently dense, then there exist \( w_j \) such that for all \( P \in \Pi_n \),

\[
\sum_j w_j P(x_j) = \int_X P(x) d\mu^*(x), \quad \sum_j |w_j P(x_j)| \leq c \int_X |P(x)| d\mu^*(x).
\]

(Marcinkiewicz-Zygmund (MZ) quadrature)

---

Algorithm

- Find $w_j$'s depending only on $x_j$'s and construct\(^5\)

\[
P(x) = \sum_{j=1}^{M} w_j y_j \Phi_n(x, x_j) \\
= \sum_{j=1}^{M} w_j f(x_j) \Phi_n(x, x_j) + \sum_{j=1}^{M} w_j \epsilon_j \Phi_n(x, x_j),
\]

\(\sigma_n(f)(x)\) noise part

---

\(^5\) Ehler, Filbir, Mhaskar, 2012
Theorem

\[ f \in W_\gamma \text{ if and only if } \| f - \sigma_n(f) \| = O(n^{-\gamma}). \]

If \( f \in W_\gamma \), \( P \) the noisy version of \( \sigma_n(f) \), then with high probability, and \( n \sim (M/\log M)^{1/(2q+2\gamma)} \),

\[ \| f - P \| \leq cn^{-\gamma}. \]

\footnote{Maggioni, Mhaskar, 2008}
3. An application\textsuperscript{7}
Continuous blood glucose monitoring

Source: http://www.dexcom.com/seven-plus

**Problem:** Estimate the future blood glucose level based on the past few readings, and the direction in which it is going – up or down.
Numerical accuracy is not as critical as classification errors. Depending upon low, normal, high blood sugar, the results are classified as accurate, or wrong but with no serious consequences (benign) or outright errors.
Deep diffusion network

Given sugar levels $s(t_0), s(t_1), \cdots$ at times $t_0, t_1, \cdots$ or different patients, we form a data set $\mathcal{P} = \{(x_j, y_j)\}, x_j = (s(t_0), \cdots, s(t_6)), y_j = s(t_{12})$ (30 minute prediction), and a training data $\mathcal{C} \subset \mathcal{P}$. 
Deep diffusion network

- Divide $C$ into 3 clusters $C_o$, $C_e$, $C_r$ depending whether 5 minute prediction indicates hypo-glycemic, eu-glycemic, or hyper-glycemic condition.  

- *(First layer training)* Use approximation theory based on $\{\lambda_k, \phi_k\}$ with 30% training data in each cluster to get predictions $f_o, f_e, f_r$ respectively:

\[
  f_s(x) = \sum_{z \in C_s} w_{z,s} f(z) \Phi_n(x, z), \quad s = o, e, r, \ x \in P
\]

- *(Second layer training)* Using the same ideas, train a *judge* to decide based on the training data which prediction gives the best PRED-EGA grid result.

---

8 Mhaskar, Naumova, Pereverzyev, 2013
9 Ehler, Filbir, Mhaskar, 2012
Results

Used clinical readings for 26 patients from DirecNet data. Average percentages in each PRED-EGD category:

<table>
<thead>
<tr>
<th>Ho-A</th>
<th>Ho-B</th>
<th>Ho-E</th>
<th>Eu-A</th>
<th>Eu-B</th>
<th>Eu-E</th>
<th>Hr-A</th>
<th>Hr-B</th>
<th>Hr-E</th>
</tr>
</thead>
<tbody>
<tr>
<td>95.81</td>
<td>2.80</td>
<td>1.40</td>
<td>82.96</td>
<td>15.26</td>
<td>1.79</td>
<td>65.65</td>
<td>21.56</td>
<td>12.79</td>
</tr>
</tbody>
</table>

Deep diffusion network (30% training data)

<table>
<thead>
<tr>
<th>Ho-A</th>
<th>Ho-B</th>
<th>Ho-E</th>
<th>Eu-A</th>
<th>Eu-B</th>
<th>Eu-E</th>
<th>Hr-A</th>
<th>Hr-B</th>
<th>Hr-E</th>
</tr>
</thead>
<tbody>
<tr>
<td>48.33</td>
<td>4.51</td>
<td>47.16</td>
<td>80.26</td>
<td>14.94</td>
<td>4.80</td>
<td>65.38</td>
<td>17.41</td>
<td>17.21</td>
</tr>
</tbody>
</table>

Deep neural network (65% training data)

<table>
<thead>
<tr>
<th>Ho-A</th>
<th>Ho-B</th>
<th>Ho-E</th>
<th>Eu-A</th>
<th>Eu-B</th>
<th>Eu-E</th>
<th>Hr-A</th>
<th>Hr-B</th>
<th>Hr-E</th>
</tr>
</thead>
<tbody>
<tr>
<td>54.09</td>
<td>5.43</td>
<td>40.48</td>
<td>77.39</td>
<td>17.33</td>
<td>5.28</td>
<td>57.13</td>
<td>23.97</td>
<td>18.91</td>
</tr>
</tbody>
</table>

Shallow network (65% training data)

Ho=Hypo-glycemic, Eu=Eu-glycemic, Hr=Hyper-glycemic
A=Accurate, B=Benign, E=Erroneous.
Problems

- Out-of-sample extension
  - Since $\phi_k$'s are computed in an entirely data dependent manner, a new computation is needed if a new datum appears.
  - Nyström extension does not always have good approximation bounds
- The measure $\mu^*$ is not known.
4. A more direct construction
Hermite functions

\[
\psi_k(x) = \frac{(-1)^k}{\sqrt{\pi^{1/2} k! 2^k}} \exp(x^2/2) \left(\frac{d}{dx}\right)^k \exp(-x^2).
\]

If \( k = (k_1, \cdots, k_q) \), \( x = (x_1, \cdots, x_q) \),

\[
\psi_k(x) = \prod_{j=1}^{q} \psi_{k_j}(x_j).
\]

\[
\int_{\mathbb{R}^q} \psi_k(x) \psi_m(x) \, dx = \delta_{k,m}.
\]

\[
\text{Proj}_{m,q}(x, y) = \sum_{k : |k|_1 = m} \psi_k(x) \psi_k(y)
\]
Hermite functions

Mehler formula

\[ \sum_{m=0}^{\infty} w^m \text{Proj}_{m,Q}(x, y) = \frac{1}{(\pi(1-w^2))^{Q/2}} \exp \left( \frac{4w x \cdot y - (1 + w^2)(|x|^2 + |y|^2)}{2(1-w^2)} \right). \]

With \((1 - w^2)^{(Q-q)/2} = \sum_{k=0}^{\infty} d_k w^k,

\[ \tilde{P}_{m,Q,q}(x, y) = \sum_{k=0}^{m} d_{m-k} \text{Proj}_{k,Q}(x, y). \]

\[ \tilde{\Phi}_{n,Q,q}(x, y) = \sum_{m<n^2} h \left( \frac{\sqrt{m}}{n} \right) \tilde{P}_{m,Q,q}(x, y) \]
Let $\mathbb{X}$ be a smooth, compact, $q$-dimensional sub-manifold of $\mathbb{R}^Q$, $\mu^*$ be its Riemannian volume measure, $0 < \gamma < 1$. For sufficiently smooth $f \in C(\mathbb{X})$ and $x \in \mathbb{X}$,

$$\left| \int_{\mathbb{X}} \tilde{\Phi}_{n,Q,q}(x,y)f(y)d\mu^*(y) - f(x) \right| \leq cn^{-\gamma}.$$
5. An application\textsuperscript{10}

\textsuperscript{10}Mhaskar, Cloninger, Cheng, 2019
Discriminative model

Based on a data \( \{(x_j, y_j)\}_{j=1}^M \), \( x_j \in \mathbb{R}^q \), with \( y_j \) taking one of finitely many values, estimate the probability \( p(y = k | x) \) for any \( x \) in the domain space.

**One-hot classification** For any label \( k \), do a binary classification: output 1 if \( x \) has the label \( k \), \(-1\) otherwise.

**Problems**

- Not every \( x \) has a label.
- There may be more than one label with any \( x \) with different probabilities.
Marginal distribution of the $x_j$’s is $\mu^*$.

Wanted:
A function $G$ such that $G(x) = 1$ if $x$ has label 1, $-1$ if $x$ has label $-1$, and 0 if $x$ is not in the support of $\mu^*$ or has an uncertain label.

A generative network: A network $G$ with this property.
Assume $Q = q$, write $\Phi_n$ for $\tilde{\Phi}_{n,q,q}$.

- $\mu^*$ is absolutely continuous: $d\mu^*(x) = f(x)dx$ (Lebesgue measure on $\mathbb{R}^q$)

- There are smooth functions $F_1(x) = 1$ if $x$ is reliably in class 1, $F_2(x) = 1$ if $x$ is reliably in class $-1$.

Class boundary: $F(x) = F_1(x) - F_2(x)$ is small.

This may mean that the label for $x$ is uncertain or that $x$ is not in the support of $\mu^*$, and hence, does not have a label.

Witness function for $F_1, F_2$

$$G_n(x) = \frac{1}{M} \sum_{j=1}^{M} (F_1(x_j) - F_2(x_j))\Phi_n(x,x_j).$$
Algorithm

- **Input:** Data sets $X$ and $Y$, points $Z = \{z_1, ..., z_K\}$ at which to inspect significance, level of confidence $A$.

- **Tunable parameters:** $p$, $N$, $A$

- $y \leftarrow X \cup Y \quad M \leftarrow |X| + |Y| \quad c_j \leftarrow \begin{cases} 1, & \text{if } y_j \in X \\ -1, & \text{if } y_j \in Y \end{cases}$

- $\hat{F}(z_j) \leftarrow \frac{1}{M} \sum_{j=1}^{M} c_j \Phi_n(z_j, y_j)$ Main estimator
Permutation test for significance

For $k = 1$ to $N$

\[
\pi \leftarrow \text{Permutation}(M)
\]

\[
F_k(z_j) \leftarrow \frac{1}{M} \sum_{j=1}^{M} c_{\pi(j)} \Phi_n(z_j, y_j)
\]

end for

\[
T(z_j) \leftarrow \text{Percentile}(\{ F_k(z_j) \}^N_{k=1}, p)
\]

\[
D(z_j) \leftarrow 1 \left( |\hat{F}(z_j)| > A \cdot T(z_j) \right)
\]

$D(z_j) = 1$ means $|\hat{F}(z_j)|$ is significant.

return $\hat{F}(z_j), D(z_j)$
6. Examples
(Left) Embedding of training data into 2D VAE latent space. 
(Right) Reconstructed images from grid sampling in 2D VAE latent space.
Reconstructed images only of grid points that are deemed “significant” by the kernel.

(Left) Witness function with the Hermite kernel.
(Right) Witness function with the Gaussian kernel.
Gaussian mixture model based generation of class centroids

Prototypical points from each class of MNIST digits, computed from 2D VAE embedding.
(Left) All point GMM centroids reconstructions
(Right) Witness function region GMM centroids reconstructions
Science news data set

- 1046 articles in 8 categories, using 1153 popular words (binary vector rather than count of words)
- Used hierarchical topic modeling
- Generated “fake” documents from a grid deemed significant using our algorithm
(Left) Hierarchical topic embedding of documents.  
(Right) Embedding highlighted by grid points deemed significantly within one class.
Science news data set

Improving nearest neighbor search

<table>
<thead>
<tr>
<th></th>
<th>Neighbor in given set of docs</th>
<th>Centroid computed from given set of docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>All documents</td>
<td>51.43%</td>
<td>53.44%</td>
</tr>
<tr>
<td>Sig. documents</td>
<td>71.56%</td>
<td>76.35%</td>
</tr>
</tbody>
</table>

Accuracy with nearest neighbor classification of Science News documents across all documents and across only significant documents.
CIFAR10

- Color images in 10 classes, 50K training, 10K test
- Features selected from last hidden layer of VGG-16 (512 dimensions)
- Error rate: 0.6%.
- **Goal:** Based on the features, detect and remove the “bad apples” in the test data without a priori knowledge of the ground truth in test data.
(Left) Classification error on points deemed “significant” as a function of level of confidence $A$. (Middle) Classification error as a function of the number of points removed for being “uncertain” (Right) The relationship between the number of points dropped and parameter $A$. 
Conclusions

- Generalization error is better measured pointwise (in a probabilistic sense).
- Our new approximation theory techniques
  - Are simple to implement
  - Obtain universal approximation with no priors required
  - Training requires only minimal optimization

Further work:
- Approximation of measures on manifolds,
- Precise estimates on out-of-sample extensions
- Feature selection
Thank you.