Progress Report
Model Setup
April 25, 2018

Contents

Model Overview

Input: data $\{\mathbf{x}_i, y_i\}_{i=1}^n$, and prediction from base models $\{\hat{\mathbf{y}}_k\}_{k=1}^K$.

Step 1: Take prediction as it is, don't learn model-specific GP:

$$\{\hat{y}_k(\mathbf{x})\}_{k=1}^K$$

Step 2: Learn ensemble weights $\{w_k(\mathbf{x})\}_{k=1}^K$

$$w_k = \frac{exp(w'_k)}{\sum_{k=1}^K exp(w'_k)} \qquad \text{where}$$
$$w'_k(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0},\ k_{w,k}(\mathbf{x}, \mathbf{x}') + \sigma_k^2)$$

Under above construction, w_k follows logistic normal distribution [Aitchison and Shen, 1980], which is more flexible (slightly heavier tail) and slightly better behaved (in terms of sampling) compared to Dirichlet . From the view of Bayesian nonparametrics, this construction corresponds to a tail-free dependent process prior for w_k [Jara and Hanson, 2011].
Another option is to assume $w_k \sim Dirichlet(exp(w'_k))$. Initial experiments show that this formulation has less stability, in that slight perturbation in training set can lead to drastic variation in prediction. However it does have the benefit of being a conjugate prior for multinomial distribution, which we can potentially deploy at the third stage. We will later revisit this formulation for more detailed analysis.

Step 3: Generate ensemble predictive distribution:

$$y \sim N\Big(\sum_{k=1}^K w_k(\mathbf{x})\hat{y}_k(\mathbf{x}) + \boldsymbol{\epsilon}(\mathbf{x}),\ \mathbf{I}\Big)$$
$$\boldsymbol{\epsilon}(\mathbf{x}) = \sum_{l=1}^L \epsilon_l(\mathbf{x}) \qquad \epsilon_l(\mathbf{x}) \overset{iid}{\sim} CART(\mathbf{x})$$

where the residual process $\boldsymbol{\epsilon}(\mathbf{x})$ is modelled using an ensemble of CART trees, where each tree is modelled in a fashion similar to that used by gradient boosting [Chen and Guestrin, 2016]. So that it serves as additionally "pasting clay" to complement the prediction of the base models.

Comments

1. For model-specific GP $w'_k(\mathbf{x})$ in **Step 2**, flexibility of these GPs will impact on the ensemble model's predictive behavior in **Step 3**. For example, a very restrictive kernel (say, constant kernel) forces $w_k(\mathbf{x})$ to be a constant for all $\mathbf{x}$, making Step 3 essentially Bayesian linear regression (which, depends on the sample size, may not be a bad choice). On the other hand, a very flexible kernel may lead to model overfit when sample size is small. See Figure 3

2. For Step 2, both models for w_k consistutes valid construction of normalized random measure over the space of candidate models. However, compare to Dirichlet, Logistic Normal is empirically much easier to sample, in terms of (1) computation speed (×30 faster under NUTS) and (2) sample quality (higher ESS). We shall evaluate the relative merit of two options by evaluating their cross validation performance.

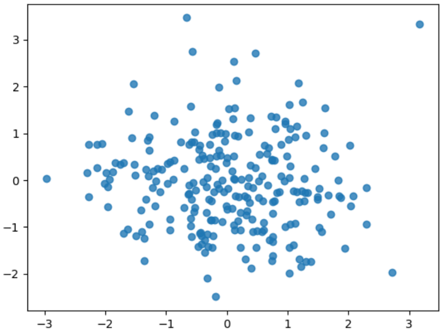
Experiment

Simulated Data

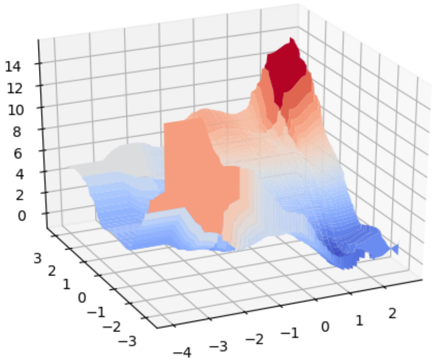
First generate data from a spatial gaussian process with 250 locations, we sample spatial locations (x, y) iid from standard normal, and assume the pollutant z follow below Gaussian Process:

$$z(x,y) \overset{iid}{\sim} N(f(x,y), \sigma^2 = 0.1)$$

$$f(x,y) = 0.2x + 0.5y + \sqrt{x^2 + y^2 + 5 * cos(x * y)} + sin(x) + cos(x) + log\Big(exp(x * y) + exp(x)\Big)$$



(a) Sampled saptial location for monitoring sites (standardized)



(b) Simulated pollution concentration surface over space

We then generate prediction for $z(x, y)$ from 10 base GP models [Duvenaud et al., 2013]:

- 4 Polynomial Kernels, degree 1, 2, 3, 4
- Gaussian RBF Kernel, with ARD (i.e. automated relevance determination [Neal, 2012])
- 3 Matérn Kernels: $\nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}$ with ARD
  (Corresponding to non-, one- and twice-differentiable functions from Sobolev space)
- MLP, with ARD. [Williams, 1998]
  (a 1-layer BNN with Gaussian CDF activation & infinite hidden units)
- Spectral Mixture [Wilson and Adams, 2013]
  (Inverse fourier transformation of a Gaussian mixture. Theoretically can model the spectral density of arbitrary stationary functions)

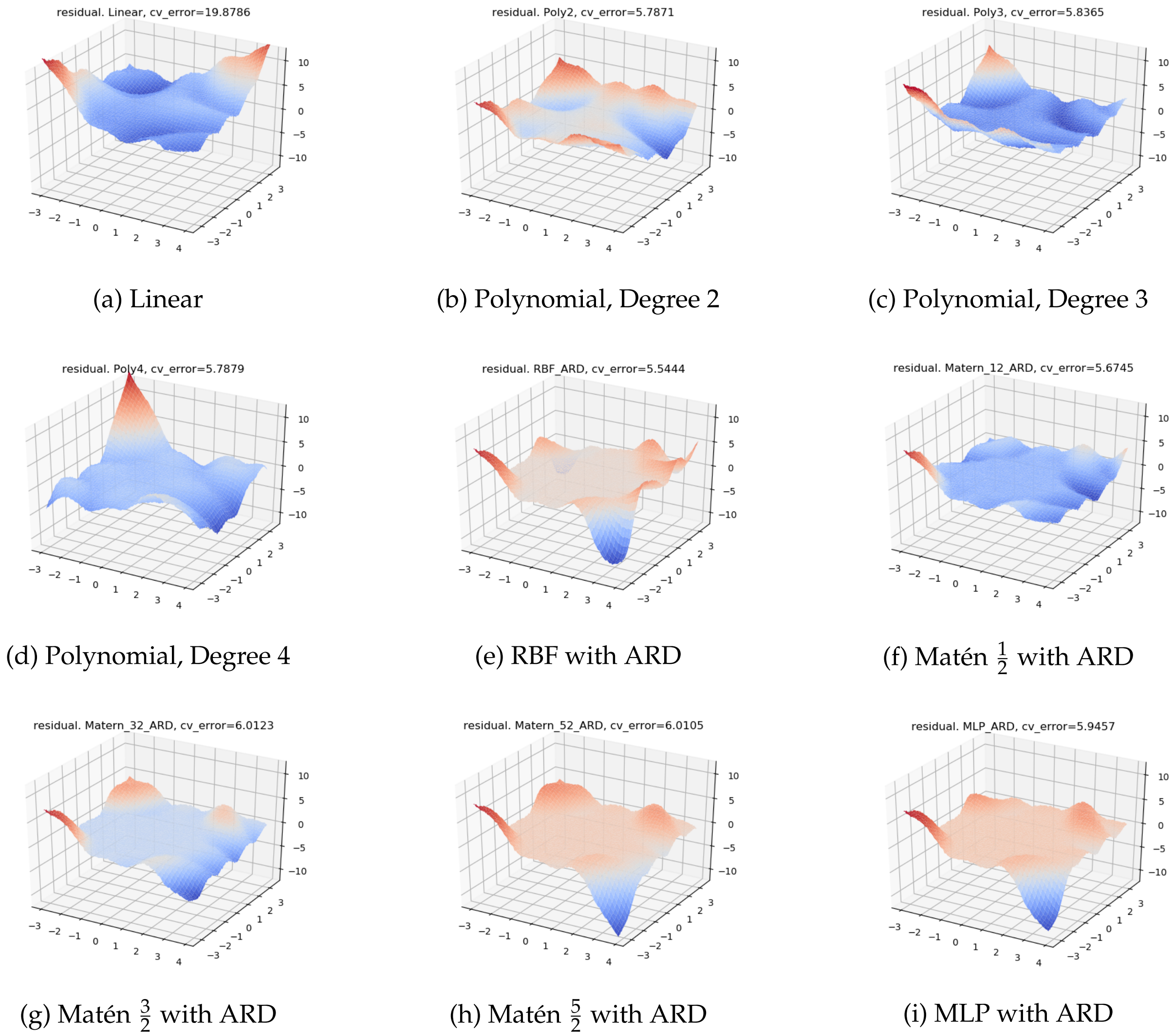
Result

Individual Model

First visualize residual process from individual model:

Linear	Poly 2	Poly 3	Poly 4	RBF	Matern 1/2	Matern 3/2	Matern 5/2	MLP	SpecMix
19.8766	5.7871	5.8365	5.7879	5.5445	5.6745	6.0123	6.0106	5.9457	5.5445

Table 1: CV Performance for base models



(a) Linear

(b) Polynomial, Degree 2

(c) Polynomial, Degree 3

(d) Polynomial, Degree 4

(e) RBF with ARD

(f) Matén $\frac{1}{2}$ with ARD

(g) Matén $\frac{3}{2}$ with ARD

(h) Matén $\frac{5}{2}$ with ARD

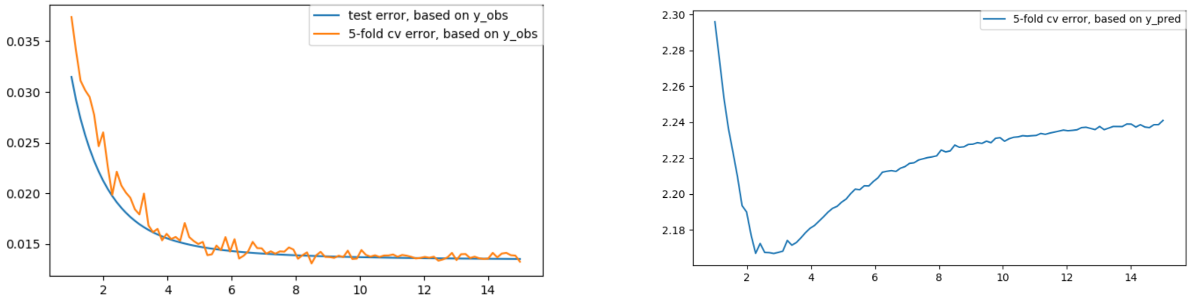
(i) MLP with ARD

Figure 2: residual process from individual base models

<u>Overall Performance: Cross validation Error</u>

Given observed data $\mathbf{y}_{obs}$, and K model predictions $\widehat{\mathbf{Y}}_{obs}(\mathbf{x}) = \{\hat{\mathbf{y}}(\mathbf{x})_k\}_{k=1}^K$, we estimate $\mathbf{w} = \{w_k(\mathbf{x})\}_{k=1}^K$, and evaluate resulting ensemble $\hat{\mathbf{y}}_{ens,obs}(\mathbf{x}) = \widehat{\mathbf{Y}}_{obs}(\mathbf{x})\mathbf{w}(\mathbf{x})$.
In this sets of experiments, we model ensemble weights using the logistic normal formulation $w_k = softmax(w'_k)$, where w'_k follows a Gaussian process with RBF kernel $exp(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{\sigma})$. We vary the hyper-parameter σ to examine its impact on model's 5-fold cross-validation error $||\mathbf{y}_{cv,obs} - \hat{\mathbf{y}}_{ens,obs}(\mathbf{x}_{cv})||_2^2$.
(In order to save time, I followed John's advice and used MAP estimates).

Additionally, since in addition to $\mathbf{y}_{obs}$, we also have base model's cross-validation prediction in spatial regions that is not in training data $\mathbf{y}_{obs}$ (call this $\mathbf{x}_{pred}$), we also compute the "true" cross-validation error $||\mathbf{y}_{cv,pred} - \hat{\mathbf{y}}_{ens,obs}(\mathbf{x}_{pred})||_2^2$.



(a) Cross-validation Error, based on $\mathbf{y}_{obs}$. x-axis: σ^2 of the RBF kernel for weight GP

(b) Cross-validation Error, based on $\mathbf{y}_{pred}$. x-axis: σ^2 of the RBF kernel for weight GP

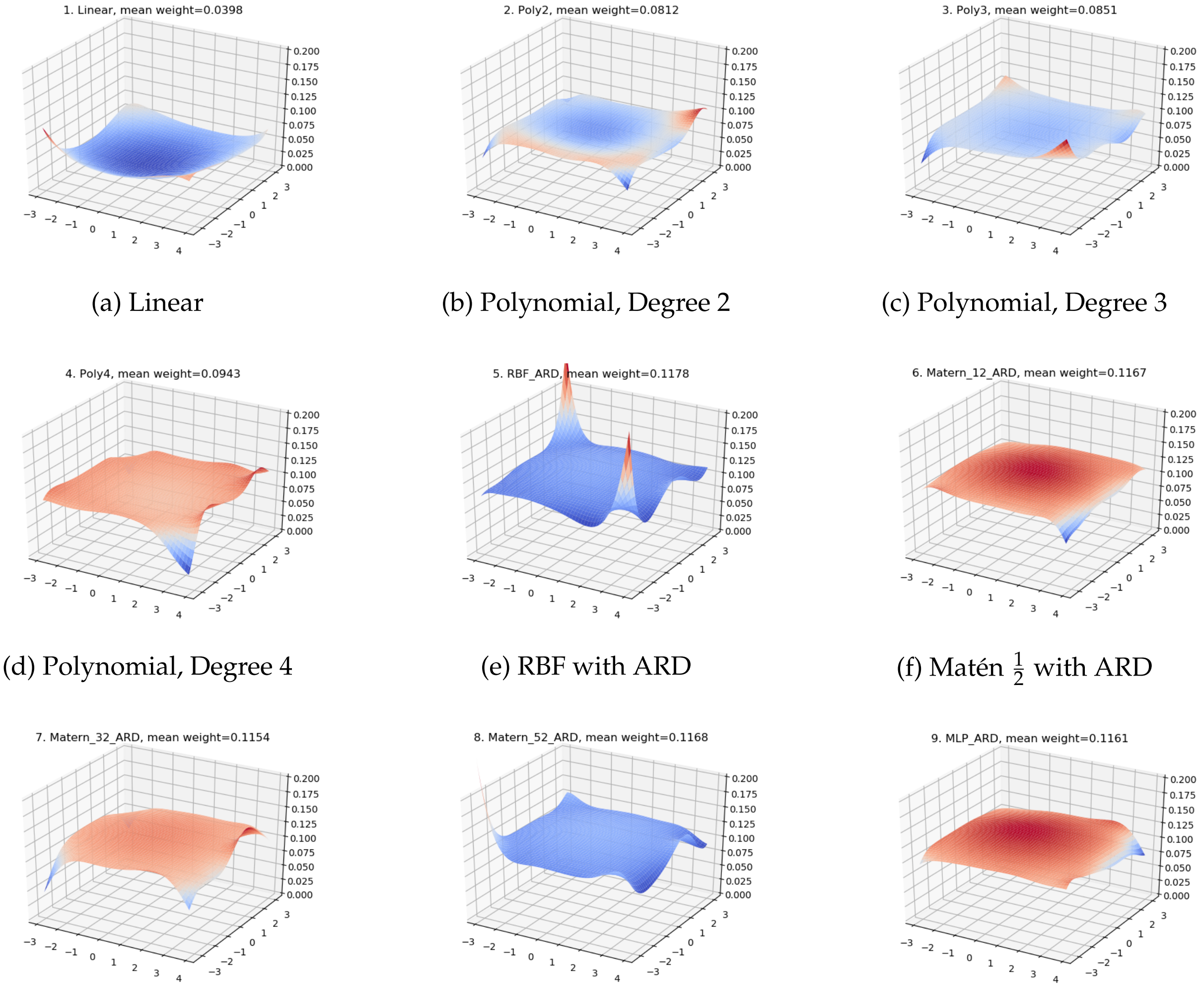
Figure 3: 5-fold Cross-validation Error in training set and hold-out set

Interpretation of the plot:
The x-axis is the value for σ^2, the hyperparameter for the RBF kernel for weight GP, larger value of σ^2 indicates a more restrictive $w_k(\mathbf{x})$ (in terms of how $w_k(\mathbf{x})$ can vary as a function of $\mathbf{x}$). In the extreme case of $\sigma^2 > 10$, $w_k(\mathbf{x})$ is nearly constant across all spatial locations.

Left plot is based on model's in-sample prediction $\widehat{\mathbf{Y}}_{obs}(\mathbf{x}) = \{\hat{\mathbf{y}}_{obs}(\mathbf{x})_k\}_{k=1}^K$. As we can see, as $w_k(\mathbf{x})$ becomes more restrictive, the train/cv error based on in-sample data becomes smaller and are close to each other. We can understand this from the view-point of bias-variance trade-off: since all in-sample predictions $\hat{\mathbf{y}}_{obs}(\mathbf{x})_k$ are already very close to data $\mathbf{y}_{obs}$, there is little to no bias in the ensemble model regardless of how restrictive the weights are (that is to say, since the base model prediction are already perfect even across space, using constant weights is sufficient to guarantee good prediction). But at the same time, since more flexible model of $w_k(\mathbf{x})$ has higher variance, more flexible models with smaller σ^2 will (somehow counterintuitively) give higher overall error.
On the other hand, in the right plot, the CV error based on unobserved data $\mathbf{y}_{pred}$ (i.e. data that base models have no access to during training) shows classical "dipping valley" pattern. The model has best performance at $\sigma^2 \in (2, 2.5)$. From the view of bias-variance trade-off, this tells us that there exists bias among $\hat{\mathbf{y}}_{obs}(\mathbf{x}_{pred})$ from the base models, therefore a somewhat flexible model can help striking a better balance in terms of mitigating bias while controlling estimator variance.
In conclusion, allowing $w_k(\mathbf{x})$ to be a function of $\mathbf{x}$ is useful in improving prediction of model ensemble. Our approach is promising

Posterior Estimate: Posterior Mean for Ensemble Weights $w_k(\mathbf{x})$

Using the logistic normal formulation for weights $w_k = softmax(exp(w'_k + \epsilon))$, where w'_k follows a Gaussian process with RBF kernel with $\sigma^2 = 2$.



(a) Linear

(b) Polynomial, Degree 2

(c) Polynomial, Degree 3

(d) Polynomial, Degree 4

(e) RBF with ARD

(f) Matén $\frac{1}{2}$ with ARD

(g) Matén $\frac{3}{2}$ with ARD

(h) Matén $\frac{5}{2}$ with ARD

(i) MLP with ARD

Figure 4: posterior mean for ensemble weight $w_k(\mathbf{x})$ from individual models

Interpretation of the plot:
Recall that the ensemble weights are trained only on 250 spatial locations as shown in Figure 1a. There exists high uncertainty in four corners of the spatial region due to lack of data, which explains the unstable behavior of individual weight surfaces on the corners.

Direction of Improvement

Two idea for improvement:

1. **Step 1, recover probability distribution using a simpler model**.
   I think it should still be ok to use Gaussian process (but will a simpler kernel). The rationale being: under a zero-mean GP, for training data $\mathbf{x}_{obs}$ and testing data $\mathbf{x}^*$, the marginal distribution of $[\epsilon(\mathbf{x}_{obs}), \epsilon(\mathbf{x}^*)] \sim N(\mathbf{0}, \mathbf{K})$ is still zero-mean. It is just that the mean for conditional posterior $\epsilon(\mathbf{x}^*)|\epsilon(\mathbf{x}_{obs})$ is non-zero. Since our focus in this step is just to recover a probability distribution, the focus should be learn hyperparameter of the kernel functions rather than looking at conditional distribution between observations. (However, still a good idea to make the kernel restrictive)

2. **Step 3, Enhance prediction by adding a model for the overall residual**
   Model the residual process of overall ensemble $\epsilon(\mathbf{x})$ using a (infinite) ensemble gradient boosting trees/ random feature [Rahimi and Recht, 2009] to enhance prediction.
   This approach leads to enhanced model prediction, and also opportunity to develope new theory since by adding infinitely many random base models to the ensemble, the space of model index $k \in \mathcal{K}$ becomes countably infinite. In this case, we can model $w_k(\mathbf{x})$ with a, say, Dirichlet process prior (e.g. use the GP-based dependent DP construction [Jara and Hanson, 2011]) following the Bayesian nonparametric formalism , making $\sum_{k=1}^{\infty} w_k(\mathbf{x}) y_k(\mathbf{x})$ a weak-limit approximation of $\int_{k \in \mathcal{K}} w_k(\mathbf{x}) y(\mathbf{x}|k) dk = y_{ensemble}(\mathbf{x})$.
   Our approach is distinct from traditional Bayesian model averaging (BMA) in that:

   (a) requires only model prediction from the base model (instead of the full posterior of base models, as required by BMA)

   (b) allow ensemble weights to vary across space and time (i.e. the feature space). (and also model sparsity).

   (c) opportunity to improve upon the base-model-only ensemble (due to incorporation of random feature for modeling the residual process).

   (d) potential for elegant (Bayesian nonparametrics) interpretation, in case we seek to publish in statistics journal.For marketing purpose, maybe we can call this
   Bayesian nonparametric integration of infinite spatio-termporal ensemble? :).

Timeline

1. (April. Week 2)
   Initial implementation.

2. (April. Week 3-4)
   Initial experiment on simulated data. Investigating:

   (a) Sensitivity to model and kernel specification

   (b) Whether improved performance for overall ensemble, how it is tighed to the property of the base models. More specifically:

   - i. Variance among model prediction
   - ii. Number and bias of models (few strong model with good prediction for different aspect of the data, or large collection of weak models)

   (c) Identifiability of individual weights

3. (May. Week 1-2)
   More realistic experiment, gradually increase complexity of data-generation mechanism toward (using the mean-surface from QD or Itai's prediction)

4. (May. Week 3) Initial result on real data

References

J. Aitchison and S. M. Shen. Logistic-Normal Distributions: Some Properties and Uses. *Biometrika*, 67(2): 261–272, 1980. ISSN 0006-3444. doi: 10.2307/2335470. URL `http://www.jstor.org/stable/2335470`.

A. Jara and T. E. Hanson. A class of mixtures of dependent tail-free processes. *Biometrika*, 98(3):553–566, September 2011. ISSN 0006-3444. doi: 10.1093/biomet/asq082. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3398659/`.

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *arXiv:1603.02754 [cs]*, pages 785–794, 2016. doi: 10.1145/2939672.2939785. URL `http://arxiv.org/abs/1603.02754`. arXiv: 1603.02754.

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Structure Discovery in Nonparametric Regression through Compositional Kernel Search. *arXiv:1302.4922 [cs, stat]*, February 2013. URL `http://arxiv.org/abs/1302.4922`. arXiv: 1302.4922.

Radford M. Neal. MCMC using Hamiltonian dynamics. *arXiv:1206.1901 [physics, stat]*, June 2012. URL `http://arxiv.org/abs/1206.1901`. arXiv: 1206.1901.

Christopher K. I. Williams. Computation with Infinite Neural Networks. *Neural Computation*, 10 (5):1203–1216, July 1998. ISSN 0899-7667. doi: 10.1162/089976698300017412. URL `http://www.mitpressjournals.org.ezp-prod1.hul.harvard.edu/doi/10.1162/089976698300017412`.

Andrew Wilson and Ryan Adams. Gaussian Process Kernels for Pattern Discovery and Extrapolation. In *International Conference on Machine Learning*, pages 1067–1075, February 2013. URL `http://proceedings.mlr.press/v28/wilson13.html`.

Ali Rahimi and Benjamin Recht. Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1313–1320. Curran Associates, Inc., 2009. URL `http://papers.nips.cc/paper/3495-weighted-sums-of-random-kitchen-sinks-replacing-minimization-with-randomization-in-learnipdf`.