Machine Learning techniques for predicting molecular properties

Viviana Petrescu
I&C, EPFL

.

***Abstract*—The high computational cost of quantum chemistry calculations have prompted the use of less expensive machine learning methods for predicting molecular properties in chemical compound space. Finding good feature representations for molecules is hard, in part because of the graph-like structure geometry of the molecules that need to be represented as high dimensional vectors. Add what this paper is about**

***Index Terms*—thesis proposal, candidacy exam write-up, EDIC, EPFL**

I. Introduction

The discovery of new molecular materials in chemistry has the potential of solving many of the problems we face today. Having a system which predicts both accurately and at a small computational cost the properties of new materials is highly desirable and has applications ranging from novel drugs discovery, water purification to efficient materials design for high energy transmission and storage [2].

II. Background Work

This write-up serves two purposes. First, it forms the basis for your candidacy exam. As such you should summarize the three papers selected by your advisor and yourself, and analyze as well as discuss them critically. Second, the write-up is also your thesis proposal. Therefore, the last one or two pages should be dedicated to your own preliminary work. A road-map of how you plan to advance the state of the art in your chosen area should also be given. For further details please consult the document "PhD Candidacy Exam Overview." You can find the latest version at `http://phd.epfl.ch/page57746-en.html`. Describe briefly the context, the problem, shortcomings in prior approaches, and your proposed approach and solution. Forecast results. Background — Describe the three papers in detail, the problem they tackle, the solutions and results, and their shortcomings, and how they relate to your work. This part builds the basis for the oral candidacy exam.

A. Learning Invariant Representations of Molecules for Atomization Energy prediction

Representing Molecules

While domain specific descriptors for molecules exist [8], recent work [5] has proposed to predict properties of a molecule of size N only from the 3D positions of the atoms R_i $i \in 1..N$ and their nuclear charge Z_i $i \in 1..N$. This has prompted the introduction of the Coloumb Matrix descriptor, whose individual entries appear in the Schroedinger equation. It is defined by a NxN matrix with entries given by:

$$M(i,j) = \begin{cases} 0.5 * Z_i^{2.4} & if i = j \\ Z_i * Z_j / |R_i - R_j| & otherwise \end{cases} \quad (1)$$

where $Ri - Rj$ represents the distance between the atoms i and j.

The dimensionality of the Coloumb Matrix is given by the number of atoms in a molecule. Since that varies across molecules in a dataset, one common trick is to pad with 0's the matrices corresponding to small molecules until they reach the maximum molecule size in a dataset. This limits the size of the molecules that can be used, since the descriptor has complexity $O(N^2)$, where N is the number of atoms.

Desired properties of descriptor Due to the graph-like structure of the molecules, finding a fixed size representation is difficult. The desired properties of a molecular descriptor are invariance to translation and rotation of the molecule and invariance to the indexing of the atoms.

Solved using sorted or Random Coloumb While the Coloumb Matrix representations solves the rotation and translation invariance through the use of distances between atoms

Proposal submitted to committee: June 13th, 2009; Candidacy exam date: June 20th, 2009; Candidacy exam committee: Exam president, thesis director, co-examiner.

This research plan has been approved:

Date: _____

Doctoral candidate: _____
(name and signature)

Thesis director: _____
(name and signature)

Thesis co-director: _____
(if applicable) (name and signature)

Doct. prog. director: _____
(B. Falsafi) (signature)

$R_i - R_j$, invariance in atom indexing still needs to be tackled since any permutation of atom indexes results in a valid Coloumb descriptor. Two variations of the descriptor are proposed in [4]. The first one, called Sorted Coloumb, uses the permutation of the atoms given by the sorting of the row norms of a valid Coloumb matrix. Any molecule has a unique representation given by the Sorted Coloumb matrix. The second representations is based on the idea that the norms of the rows can have very similar values in practice and the sorting, therefore also the atom indexing, is subject to small noise. The new descriptor, Random Coloumb Matrices is a collection of Sorted Coloumb matrices. For every molecule, approximatively 10 Randomly Sorted Coloumb matrices are drawn. One Random Sorted Coloumb Marix is obtained by sorting the set of rows was sorted according to their norm at which a small Gaussian noise was added. New predictions can be made by averging the prediction results.

Reach state of the art result at the time. The current new value is 1.5 using Bag of Bonds. The performane of the descriptors was evaluated on predicting the atomization energy on a dataset of 7165 molecules with at most 23 atoms per atom. The predicted values range from -800 to -2000 kcal/mol. The splitting of the data into 5 folds for cross validation is done using stratified sampling. The molecules were clustered into 5 sets with similar atomization energy levels and the folds were created by randomly selecting one molecule from each bucket.

While this is a good strategy for obtaining good generalization performance, in normal machine learning practice we should not touch the test dataset. If we test the performance on a new molecule whose atomization energy is not within the bounds present in the training dataset, it is likely that the prediction will not be very accurate. Moreover, if we have a molecule with more atoms than present in the dataset, we would need to retrain our model with a descriptor of diferent size. The same problem appears across chemical compound space if we try to predict the atomization energy of a molecule with the same number of atoms but with different atom types composition. In general, we do not have guarantees that the a molecule with similar atom composition and similar atom type have target prediction in the same range as our training set.

For experimental evaluation of the performance of descriptors, both kernel ridge regression and multy layer feed forwards networks were tried. The best test performance of 3.1 kcal/mol MAE was obtained using a neural network with Randomly Sorted Coloumb matrix. The improvement was three fold with respect to the previous state of the art, which gave a MAE of 9kcal/mol. A performance level is 1kcal/mol to reach chemical accuracy. Maybe put this as footnote Later work [3], improved upon this result by reducing the current state-of-the-art to only 1.5kcal/mol using a bag of bonds descriptor with Laplacian kernel.

Difficulty of training a neural network, maybe cite tricks of the trade While recently Neural Nets (especially deep networks) have achieved state-of-the-art in many fields ranging from computer vision, natural language processing and speech recogition, one of the main challenges they pose is the difficulty in training for people without a lot of experiences with nnets in particular.

Cite here Neural nets tricks fo the trade In order to obtain the competitive performance of 3.1kcal/mol using neural nets, multiple tricks have been used, inspired from cite here NN. First of all, taking the real entries of the Coloumb matrix as input to the neural net proved to perform poorly. For this, a binarization step was performed which added an extra dimensionality to the dataset. Specifically, if $x \in R^D$ represents a flatten Coloumb matrix, it is mapped to a new binarized descriptor $y \in R^{DxM}$ such that $y_i \in R^{1xM}$ is computed by taking shifted versions of x_i entry that are passed through a sigmoidal function

$$y_i = [..., tanh(\frac{x_i - \theta}{\theta}, tanh(\frac{x_i}{\theta}), tanh(\frac{x_i + \theta}{\theta})] \quad (2)$$

Depending on the range of every x_i, the size of y_i can vary acrosss dimensions if we ignore saturated values. Other parameters that are selected using cross validation are the number of hidden units per layer, the number of layers, the learning rate.

B. Self-Taught Learning: Transfer Learning from Unlabeled Data

Some introduction

One challenge in supervised learning tasks is to obtained labeled data. The higher the dimensionality of the input representations, the more desired labels. Since it can require tedious manual labor work, often solved using Amazon Mechanical turk. This paper introduces another machine learning framework called self-thaught learning, which bases its idea on the fact that unlabeled dataset ismuch easier to obtain. Therefore, in the new setup, both labeled and unlabeled data is used for training a classifier. In Self-Thaught learning, labeled data is transformed in a high level representation learned through the use of a large number of unlabeled data. The new representation is used for training a classifier on the labeled data. The setup is envisioned to have a bigger advantage when the available labeled data is scarce. The more available labeled data, the use of the unlabeled data becomes less important.

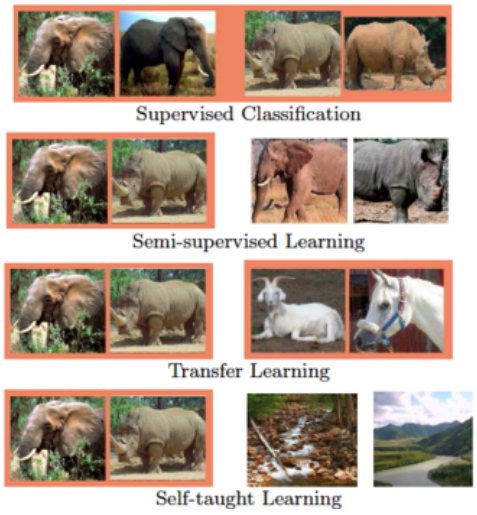
1) Machine Learning framewroks that exists, with picture.: According to the type of data available (labeled and/or unlabeled) there are a number of frameworks in machine learning

- Supervised Classification - training labels are provided
- Semi-Supervised Classification - training labels are not provided, but test data comes from the same distribution as the training data
- Transfer Learning - training labels are provided but the data is assumed to be from a different distribution
- Self-Thaught Learning - training data does not have labels and it is also not assumed to come from the same distribution as the test data

An intuitive example is shown in Fig., where the orange bounded boxes represent labeled data.

2) Problem formulation: Self thaught learning can be expressed as a two part optimization problem.

In the first step, the unlabeled data $x_u^{(i)} \in R^n, i = 1..k$ is used to find a set of s *bases* $b_i \in R^n, i = 1..s$ and a set of



Morre bla

Fig. 1. Simulation Results

sparse activations $a^j_{(i)}$ with $a^{(i)} \in R^s$ such that x_i can be reconstructed from the s bases vectors b and its corresponding activation weights $a^{(i)} \in R^s$ given by the formula:

$$\begin{matrix} \min_{a,b} \sum_{i=1}^{k} ||x_u^{(i)} - \sum_j a_j^{(i)} b_j||_2^2 + \beta||a^{(i)}||_1 \\ s.t. ||b_j||_2 \leq 1, j = 1..s \end{matrix} \quad (3)$$

The equation above can be easily optimized using the algorithm from citehere by alternating the optimizating a and b, e.g. fix one variable and optimize the other one at every step. The problem is equivalent with a least square L1 regularized problem in variable a and a least square constrained problem in variable b. The L1 regularization enforces sparse representations, which were experimentally shown to perform better across different datasets. We note that PCA can also be used for finding a new representation of the input, but it leads to different solutions were the activations are linear combination of the input. Moreover, PCA can not be used for generating more bases than the input dimension.

In the second step, the bases b learned in the previous step are used for finding a higher level representation for labeled input data $x_l^{(i)}, i = 1..m$ with corresponding target values $y_l^{(i)}, i = 1..m$. The new representation is the set of activation vectors $a^l(x_l^{(i)}), i = 1..m$ obtained by solving another L1-regularized optimization problem

$$a^l(x_l^{(i)}) = \arg\min_{a^{(i)}} ||x_l^{(i)} - \sum_j a_j^{(i)} b_j||_2^2 + \beta||a^{(i)}||_1. \quad (4)$$

As a last step a classifier is trained on the pairs $a(x_l^{(i)}), y^{(i)} i = 1..m$ and compared with a classifier (same type e.g. SVM) trained on the initial pairs $x_l^{(i)}, y^{(i)} i = 1..m$.

3) Experimental Results: The

4) Conclusions:

C. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting

Optimizing an unknown function is an important problem. Often done using Gaussian processes in a Bayesian setting. Many heuristic have been proposed for defining an acquisition function. Few is known for its conergence. PUT ALSO PICTURE WITH EIGENSPECTRUM DECAY Optimizing a black box function $f : D- > R$ that is expensive to evaluate is a common problem with applications ranging from active user modelling, hierarchical reinforcement learning[1] and more recently, hyper parameter optimization of machine learning models [7]. This problem can be posed as a multi-armed bandit problem where the function to be estimated is either sampled from a Gaussian Process or has low norm in reproducing Kernel Hilbert space.

The optimization is done in a sequential manner, where at round t a point x_t in the domain is chosen according to an *acquisition function* and the value $f(x_t)$ is evaluated. The goal is to maximize f with as few samples as possible. Among the common used acquistion functions we note the probabilty of improvement, maximum expected improvement or upper confidence bound functions, with the latter two experimentally shown to perform better in practice.

The effectiveness of a sampling scheme in finding $x*$ (may not be unique) s.t $x* = \arg\max_{x \in D} f(x)$ is evaluated using the *instantaneous* and the *cumulative regret*. The instantenous regret $r_t = f(x*) - f(x_t)$ is the error that we make at round t for choosing point x_t instead of $x*$. The cumulative regret is the error accumulated up to thie current round $R_T = \sum_{t=1..T} r_t$.

Besides their practical success, few was known in practice about their convergence properties for finite and infinte input dimensional space. The main contribution of the paper is to show sublinear convergence rates in T for GP-UCB, the number of rounds of the cumulative regret. In other words, as T goes to infinity, we expect the cumulative regret to be 0 $lim \frac{R_T}{T} = 0$.

The bound on the cumulative regret R_T is done in two steps. First bound R_T on the information gain, then bound the information gain using the empirical spectrum operator For the inifinite domain case, this is done by further bounding on the kernel operator spectrum.

For simplicity and since RKHS is only relevent in a non-Bayesian setting, it will be skiped from the previous explanation.

1) Gaussian Processes basics: A Gaussian Process (GP) defines a distribution over functions f, $f\ GP(u(x), k(x, x'))$, in a similar way in which probability densities define distributions over random variables. It is fully specified by its mean $u(x)$ and its covariance matrix K. In a Bayesian setting, we assume our function f is sampled from a Gaussian with prior $GP(0, k(x, x'))$, where k is the kernel or covariance matrix. The most common kernel types are the linear, squared exponential and Matern kernel.

For T sampled points in the presence of noise, we have $y_t = f(x_t) + \epsilon_t$, where $\epsilon_t \in N(0, \sigma^2)$. The posterior over f, conditioned on the $y_{1..T}$ and $x_{1..T}$ is given by $P(f_T|y_{1:T}, x_{1..T}) = N(u_T(x), \sigma_T^2(x))$, where:

- K_T is positive definite with entries $k(x, x'), x, x' \in A_T$
- $k_T(x) = [k(x_1, x), ...k(x_T, x)]^T$
- $k_T(x, x') = k(x, x') - k_T(x)^T (K_T + \sigma^2 I)^- 1 k_T(x')$
- $u_T(x) = k_T(x)^T (K_T + \sigma^2 I)^{-1} y_T$

- $\sigma_T^2(x) = k_T(x,x)$

2) Information Gain and Experimental Design: Unlike function optimization where the goal is to find the maximum of a function, in experimental design the goal is to find a good approximation of the function globally with few samples as possible. We note that the same algorithm can not be employed for both tasks, since in function optimization we might want to avoid sampling in the regions where we are confident that the function has small values.

For a sampled set $x \in A$, we define $y_A = f_A + \epsilon_A$, where $f_A = [f(x)]_{x \in A}$ and the noise $\epsilon_A \sim N(0, \sigma^2)$. The mutual information gain is

$$I(y_A; f) = H(y_A) - H(y_A|f) \tag{5}$$

where $H(y_A)$ and $H(y_A|f)$ represent the entropy or uncertainy in y_A and the uncertainty in y_A if we know f. In other words, information gain expresses the reduction in uncertainy about f given y_A.Shouldn t be the other way around?

Using the entropy formula for a Gaussian $H(N(u,\sigma)) = |\frac{1}{2} log|2\pi e\Sigma|$ we can rewrite $F(A) = I(y_A; f) = I(y_A; f_A) = |\frac{1}{2} log|I + \sigma^{-2} K_A|$ Finding the subset A in the domain D with $|A| \leq |T|$ that maximizes $I(y_A; f)$ is NP-hard. In practice, greedy approximation solutions are used that find a near-optimal solution by choosing a sequence of points in A with maximum variance. Thus, the following equivalence relation holds at round t:

$$x_t = \arg\max_{x \in D} F(A_{t-1} \cup x) \iff x_t = \arg\max_{x \in D} \sigma_{t-1}(x) \tag{6}$$

with $t \in 1...T$

In cite it is shown that $F(A)$ is a submodular function and if A_T is obtained using our greedy aproach from above, it holds that

$$F(A_T) \geq (1 - \frac{1}{e}) \max_{|A| \leq T} (F(A)) \tag{7}$$

Thus, a greedy set creation of A_T leads to a near optimal solution.

3) GP-UCB: For the function optimization problem, a good sampling sequence x_t is one that makes a tradeoff between exploration and exploitation. By choosing points with high variance (exploitation) we try to reduce the overall uncertainty. By choosing points with high mean (exploitation) we try to concentrate samples in a region were we are more confident that is close to the optimal. This idea is expressed by choosing in every round t, a point x_t according to an acquistion function $x_t = \arg\max_{x \in D} u_{t-1} + \beta_t^{1/2} \sigma_{t-1}(x)$ with β_t defined in cite. The function that is being maximized is also called the acquistion function or utility function and corresponds to Upper Confidence Bound criterion.

To sum up, at every round, the GP-UCB algorithm selects a point according to the above equation, evaluates $y_t = f(x_t) + \epsilon$ and updates u_t and σ_t using Bayes rule from equation 1.

An important property of the UCB defined as above is that it does not choose points x for which the upper confidence value is smaller than the maximum lower confidence value found so far. This can prune large subareas of the search space.

Kernel	Linear	RBF	Matern
Regret R_T	$d\sqrt{T}$	$\sqrt{T(log(T)^{d+1})}$	$v + d(d+1)/2v + d(d+1)$

TABLE I
REGRET BOUNDS.

4) Regret Bounds: This paper is the first to prove sublinear convergence rates for the cumulative regret R_T with both finite and infinite input domain D with smoothness assumptions on the kernel. They show that in the limit GP-UCB has no regret $lim \frac{R_T}{T} = 0$ and the convergence results for the most popular kernels can be found in table II-C4, up to polylog factors. We notice that the dependance on the dimensionality d of the input is weak. namely, for the linear kernel the dependency is linear and for the squared exponential kernel d appears only as a power of $log(T)$.

5) Bounding the Regret using Information Gain: The first step of the proof is to bound R_T by the maximum information gain $\gamma_T = \max_{A \subseteq D, |A| = T} I(y_A; f_A)$ after T rounds. In particular, it is shown that with high probability

- $Pr(R_T \leq \sqrt{C_1 T \beta_T \gamma_T}) \leq 1 - \delta$ for finite D
- $Pr(R_T \leq \sqrt{C_2 T \beta_T \gamma_T} + 2) \leq 1 - \delta$ for D compact and convex

where $\delta \in (0,1)$ and appropiate constants C_1, C_2, β_t . The second inequality holds by making further assumptions that the function is smooth, by enforcing that the probability of the partial derivatives of the function to have a large value is small.

6) Bounding the Information Gain using the empiricial spectrum: The case of infinite input dimensionionality are treated similarly with the finite case by assuming the existence of a discretized set $D_T \subset D, T \in N$ with nearest neighbor distance O(1), dense in the limit. As stated previously, the value of $\gamma_T = \max_{A \subset D_T, |A| = T} I(y_A; f_A)$ can be bounded by the greedy maximization value $\gamma_T \leq \frac{1}{1 - e^(-1)} F(A_T)$ since it is a submodular function. Selecting a x_t is equivalent with selecting a vector $v_t \in R|D|$ with values 0's except the entry at position t, thus $f \ N(v_t f, \sigma^2)$. By using another relaxation procedure, we let $||v_t|| = 1$, it is shown that the $v_t's$ are selected among the eigenvectors of the kernel matrix $K_{D_T} = k(x, x')x, x' \in D_T$. The maximum information gain is further bounded by an expression involving the eigenvalues $\lambda_t^\sim$ of K_{D_T} as follows:

$$\gamma_T \leq \frac{0.5}{1 - e^{-1}} \max_{m_1, m_2, ..., m_D} \sum_{t=1}^{|D|} log(1 + \sigma^{-2} m_t \lambda_t^\sim) \tag{8}$$

where $m_t \in N$, $\sum_t m_t = T$ and $\lambda_1^\sim \geq \lambda_2^\sim \geq \lambda_3^\sim \geq ...$ Here we have considered a worst case scenario of assigning eingevectors of K_T to v_t. In the next step, the information gain is bounded using the tail spectrum of K_{D_T}, $B(T_*) = \sum_{t=1}^{T_*} \lambda_t^\sim$ and $n_T = \sum_{t=1}^{T_*} \lambda_t^\sim$ for any $T_* = 1..T$:

$$\gamma_T \leq O(\sigma^{-2}[B(T_* T + T_* log n_T T)]) \tag{9}$$

From equation above, we conclude that if the tail spectrum $B(T_*)$ is small, or in others words, the eingenspetrum of K_{D_T} decays fast, the bound on γ_T is more tight.

7) Bound the empirical spectrum by the operator spectrum: For a kernel k with $k(x,x)u(x) = 1$ where u(x) is uniformly distributed on D, Mercer's theorem shows the existence of a discrete kernel eigenspectrum $\lambda_s(x), \phi_s(x)$ with $k(x,x') = \sum \lambda_s \phi_s(x)\phi_s(x')$

Another challenge of this paper is to bound the empirical tails spectrum $B(T_*)$ by the kernel operator tails eigenspectrum $B_k(T_*) == \sum_{t=1}^{T_*} \lambda_t$. Analytical expression bounds on $B_k(T_*)$ for the most common kernels are given by Seeger [6].

8) Conclusions: The contribution of the paper are two fold. First, a novel connection is being made between experimental design and GP optimization, by bounding the cumulative regret by the maximum information gain. Secondly, for the first time sublinear regret bounds for GP-UCB optimization are shown with weak dependence on the dimensionality of the input. The bound is derived in two steps. First the cumulative regret is bounded by the maximum information gain. In a second step, the max information gain is bounded by the tails of the empirical eigenspectrum which in turn is bound by the tails of the kernel operator spectrum. The bound is tighter, the more rapidly the kernel spectrum decays.

Although convergence rates are proven for infinite input dimensional spaces as well, the complexity of the GP step that incorporates the inversion of the covariance matrix makesthe complexity of the algorithm cubic in the number of samples T, which makes it often inpractical for high dimensional input.

III. Research Proposal

A.

Proposing a new descriptor which is invariant to permutations of the atom. 2.6+/ or 2.1 with noise. (still gives comparable accuracy to same DFT models)

B.

Propose of augumententing the data sets more easily and make the cross validation less sensitive to splitting -at the moment : take non H atoms, then sort then do CV.

Pose the problem as one of the semi supervised, self tought learning etc problems and try to make it generalize across compound space, or learn new embeddings of the atoms.

Although the use of Gaussian processes in material design is not new, it s major drawback is the computational bottleneck. In our scenario, this can be used for tuning the hyper-parameters of the model trained. Here the input dimensionality is given by the nbr of the hyper parameters (learning rate, activation fct, nbr hidden layers) used and the fct to be minimized is the cross validation error. The result presented in the previous subsection were obtained like that. Talk if we have time about bayesian neural nets, were simpler models outperform more easy models just be using hyper parameter optimization instead of grid search.

Write how you propose to advance the state of the art given the background. What is new technically? How does it improve over prior work? Summarize, suggest an approximate timeline, and list references.

References

[1] Eric Brochu, Vlad M. Cora, and O De Freitas. A tutorial on bayesian optimization of expensive cost functions, withapplicationtoactiveusermodeling andhierarchical reinforcement learning. 2009.

[2] Johannes Hachmann, Roberto Olivares-Amaya, Sule Atahan-Evrenk, Carlos Amador-Bedolla, Roel S. Snchez-Carrera, Aryeh Gold-Parker, Leslie Vogt, Anna M. Brockway, and Aln Aspuru-Guzik. The harvard clean energy project: Large-scale computational screening and design of organic photovoltaics on the world community grid. *The Journal of Physical Chemistry Letters*, 2(17):2241–2251, 2011.

[3] O.A. von Lilienfeld K.-R. Mller K. Hansen, F. Biegler and A. Tkatchenko. *Interaction Potentials in Molecules and Non-Local Information in Chemical Space.* Phys. Rev. Lett. (March 10, 2014)., 2014.

[4] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole V Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In *Advances in Neural Information Processing Systems*, pages 440–448, 2012.

[5] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108:058301, Jan 2012.

[6] MW. Seeger, SM. Kakade, and DP. Foster. Information consistency of nonparametric gaussian process methods. *IEEE Transactions on Information Theory*, 54(5):2376–2382, May 2008.

[7] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. pages 2951–2959, 2012.

[8] Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*, volume 11. Wiley-VCH, 2000.