Machine Learning for Many-Particle Systems – An Introduction

Klaus-Robert Müller !!et al.!!

Today‘s Talk

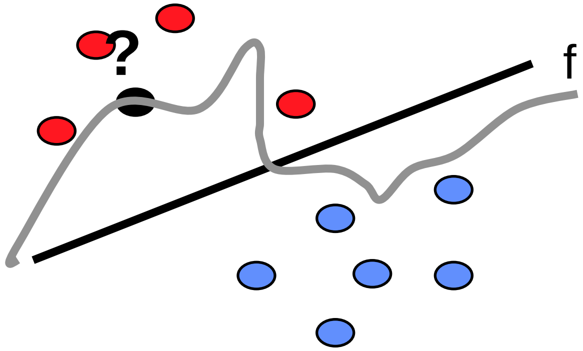
Machine Learning

- introduction: ingredients for ML
- Kernel Methods and Deep networks & remarks

Applications ML to Physics & Materials

- representation
- models
- remarks

Machine Learning in a nutshell



Typical scenario: learning from data

- given data set **X** and labels **Y** (generated by some joint probabilty distribution p(x,y))
- **LEARN/INFER** underlying **unknown** mapping

$$Y = f(X)$$

Example: understand chemical compound space, distinguish brain states …

BUT: how to do this optimally with good performance on **unseen** data?

Basic ideas in learning theory

Three scenarios: regression, classification & density estimation.
Learn f from examples

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \in \mathbb{R}^n \times \mathbb{R}^m \text{ or } \{\pm 1\}, \quad \text{generated from } P(\mathbf{x}, y),$$

such that expected number of errors on test set (drawn from $P(\mathbf{x}, y)$),

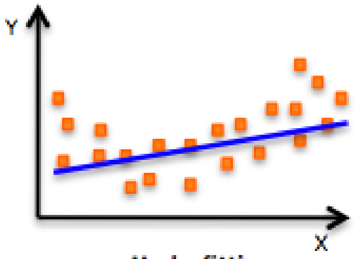
$$R[f] = \int \frac{1}{2} |f(\mathbf{x}) - y)|^2 \, dP(\mathbf{x}, y),$$

is minimal *(Risk Minimization (RM))*.

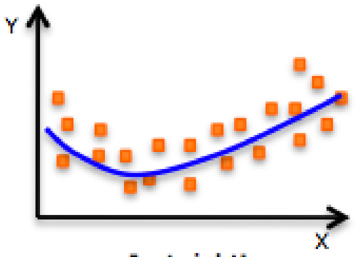
Problem: P is unknown. $\longrightarrow$ need an *induction principle*.

Empirical risk minimization (ERM): replace the average over $P(\mathbf{x}, y)$ by an average over the training sample, i.e. minimize the training error

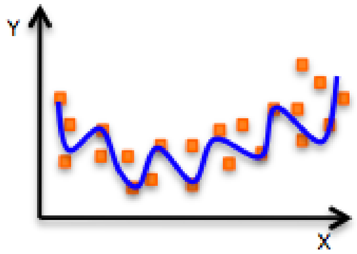
$$R_{emp}[f] = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} |f(\mathbf{x}_i) - y_i|^2$$



Underfitting



Just right!



overfitting

ML tool & models zoo

- **supervised, semi-supervised, unsupervised methods**
- **kernel methods**: support vector machines, kPCA...
- **Boosting**: adaboost bumpboost etc.
- **sparse methods**: compressed sensing, sparse kernel methods, l_1 trick
- **neural networks**: deep or shallow, recursive
- **clustering**: hierarchical, mincut etc.

- **feature selection**: greedy, sparse, l_1 trick, dimensionality reduction
- **relevant dimensionality estimate**: RDE, local RDE
- **explaining nonlinear methods**: relevance propagation, explanation vector fields..
- **projection methods**: dimensionality reduction, PCA, ICA, SSA, LLE, tSNE etc.

ML ingredients

- **Representation** X, i.e. **what** we put into learning not only whether we use vectors, matrices, graphs, strings, tensors etc.
- **Optimization**: how to set up training of the learning machine, what is error measure

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \quad \tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\textstyle\sum_{i=1}^{\ell}\xi_i$$

$$y_i((\boldsymbol{w}\cdot\Phi(\boldsymbol{x}_i))+b)\geq 1-\xi_i, \quad i=1,\ldots,\ell, \quad \text{with} \quad \xi_i>0,$$

  Note: error/cost measures exist beyond mean squared error, e.g. divergences, information theoretic measures, ranking errors, true cost etc

- **Regularization**: avoid overfitting by enforcing smoothness, simplicity, sparseness, include prior knowledge …

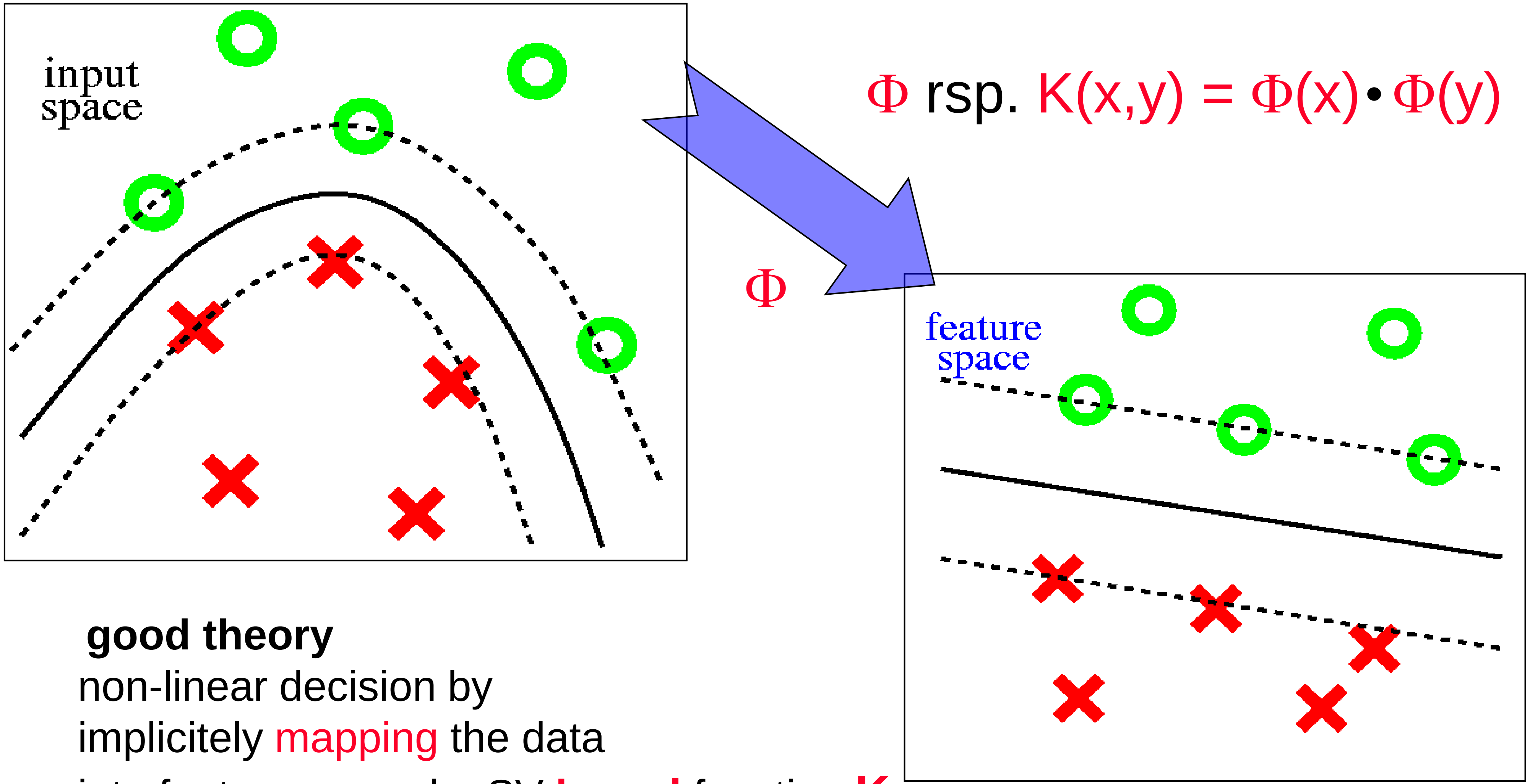
$$\text{error(f)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}|f(\mathbf{x}_i)-y_i|^2 + \lambda\,\mathbf{|Pf|}^2$$

- **Modelselection**: choose model hyperparameters, e.g. C, λ: Bayes, CV

Support Vector Machines in a nutshell

$$f(\mathrm{x}) = \mathrm{sgn}\left(\mathrm{w} \cdot \Phi(\mathrm{x}) + b\right)$$

Φ rsp. $K(x,y) = \Phi(x) \bullet \Phi(y)$



good theory

non-linear decision by implicitely mapping the data into feature space by SV **kernel** function **K**

[e.g. Vapnik 95, Muller et al 2001, Schölkopf & Smola 2002, Montavon et al 2013]

SVM: more details

- Compute hyperplane $(\mathbf{w} \cdot \Phi(\mathbf{x}) + b)$ with maximum margin in feature space. Introduce slack variables ξ_i to allow for training errors. This amounts to the following QP:

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \quad \tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i$$

$$y_i((\boldsymbol{w}\cdot\Phi(\boldsymbol{x}_i)) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, \ell, \quad \text{with} \quad \xi_i > 0,$$

- Forming the dual problem one finds: $\boldsymbol{w} = \sum_i y_i \alpha_i \Phi(\boldsymbol{x}_i)$.

- To find the coefficient α_i solve the dual problem:

$$\max_{\boldsymbol{\alpha}} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell}\alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j \,\mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C,\ i = 1, \ldots, \ell,\ \text{and } \sum_{i=1}^{\ell}\alpha_i y_i = 0,$$

- Sparse, unique (!) solutions (i.e. many α_i are zero).

[cf. Vapnik 95, Schölkopf et al 99, Müller et al. 2001,
Schölkopf and Smola 2002, Laskov et al. 2005]

Digestion: Use of kernels

- Question: What makes kernel methods (e.g. SVM) perform well?

- Answer:
  - In the first place: a good idea/theory. $R[f] \leq R_{emp}[f] + \sqrt{\frac{d\left(\log\frac{2N}{d}+1\right)-\log(\eta/4)}{N}}.$
  - But also: The kernel

- Using kernels, we work explicitly in extremely high dimensional spaces (RKHS) with interesting features for themselves (depending on the kernel) [SSM et al. 98]

- Common choices: Gaussian kernel $\exp(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/c)$ or polynomial kernel $(\boldsymbol{x}\cdot\boldsymbol{y})^d$.

- Almost any linear algorithm can be transformed to feature space.[SSM et al. 98]

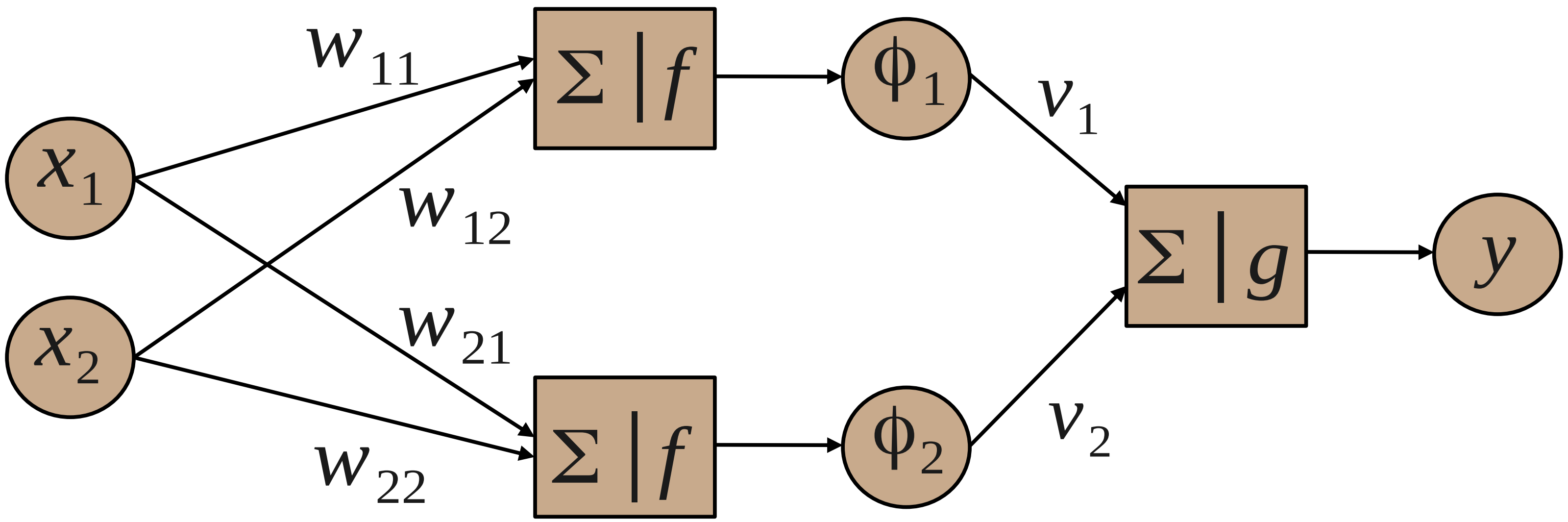
- With suitable regularization it outperforms its linear counterpart. [Mika et al. 02]

- The kernel can be adopted to specific tasks **[Zien et al. 00, Tsuda et al. 02, Sonnenburg et al. 05]**

More recent **insight**: Kernel representation make very efficient use wrt. data per effective dimension!

[Braun, Buhmann, Müller 07, 08, Montavon et al 13]

Multilayer networks



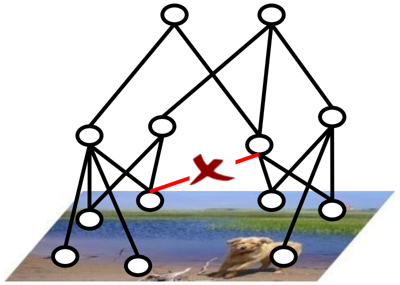
$$\phi_1 = f\left(x_1 w_{11} + x_2 w_{12} + b_1\right)$$
$$\phi_2 = f\left(x_1 w_{21} + x_2 w_{22} + b_2\right)$$
$$y = g\left(\phi_1 v_1 + \phi_2 v_2 + c\right)$$

Matrix form:

$$y = g\left(V \cdot f\left(W \cdot x\right)\right)$$

Deep Neural Networks

- recently the hot ML method: Q: Why?
- A: sociological & faster computers
- Deep net architecture can be structured
- Representation is learned
- Multiscale information



- parallelization is possible and GPU implementation available
- highly successful in practice
- remark: statistical estimators 1/N

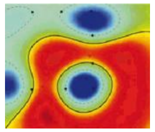
Disgestion

- **kernel methods**: kernel defines representation and regularizer (see also SSM 98)
- **neural networks**: learn representation

ML4Physics @IPAM 2011: Part I

Klaus-Robert Müller, Matthias Rupp

Anatole von Lilienfeld and Alexandre Tkachenko



Machine Learning for chemical compound space

Ansatz:

$$\{Z_I, \mathbf{R}_I\} \xmapsto{\mathrm{ML}} E$$

instead of

$$\hat{H}(\{Z_I, \mathbf{R}_I\}) \xmapsto{\Psi} E$$

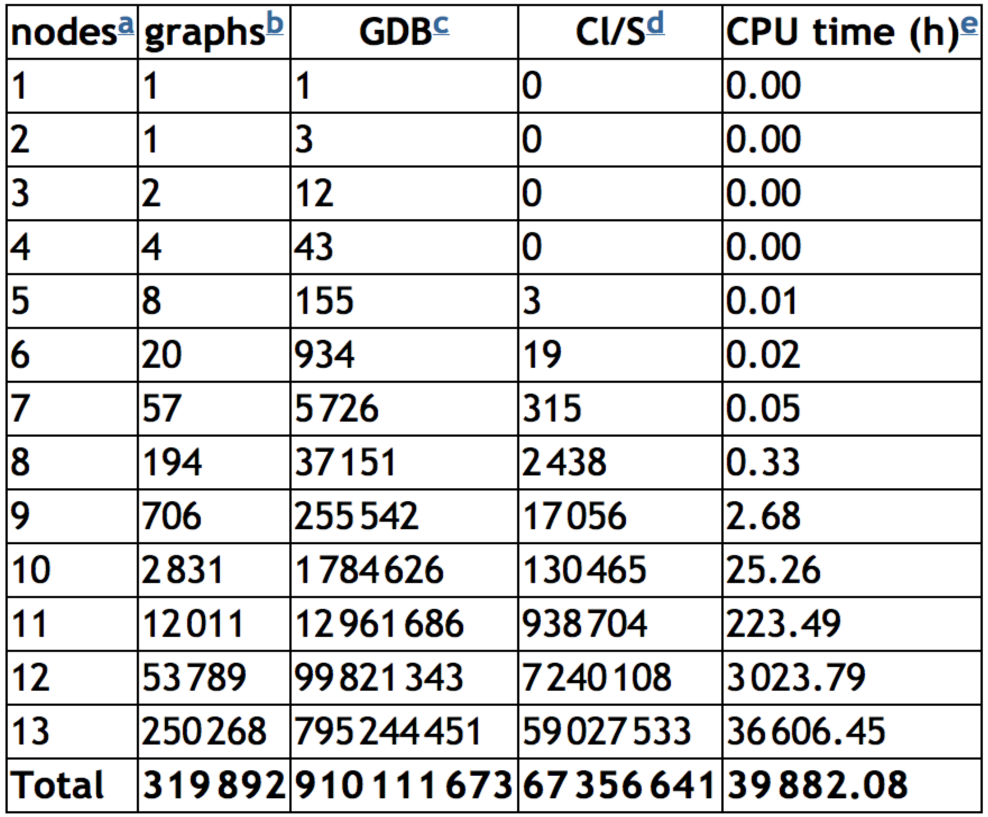
$$\hat{H}\Psi = E\Psi$$

[from von Lilienfeld]

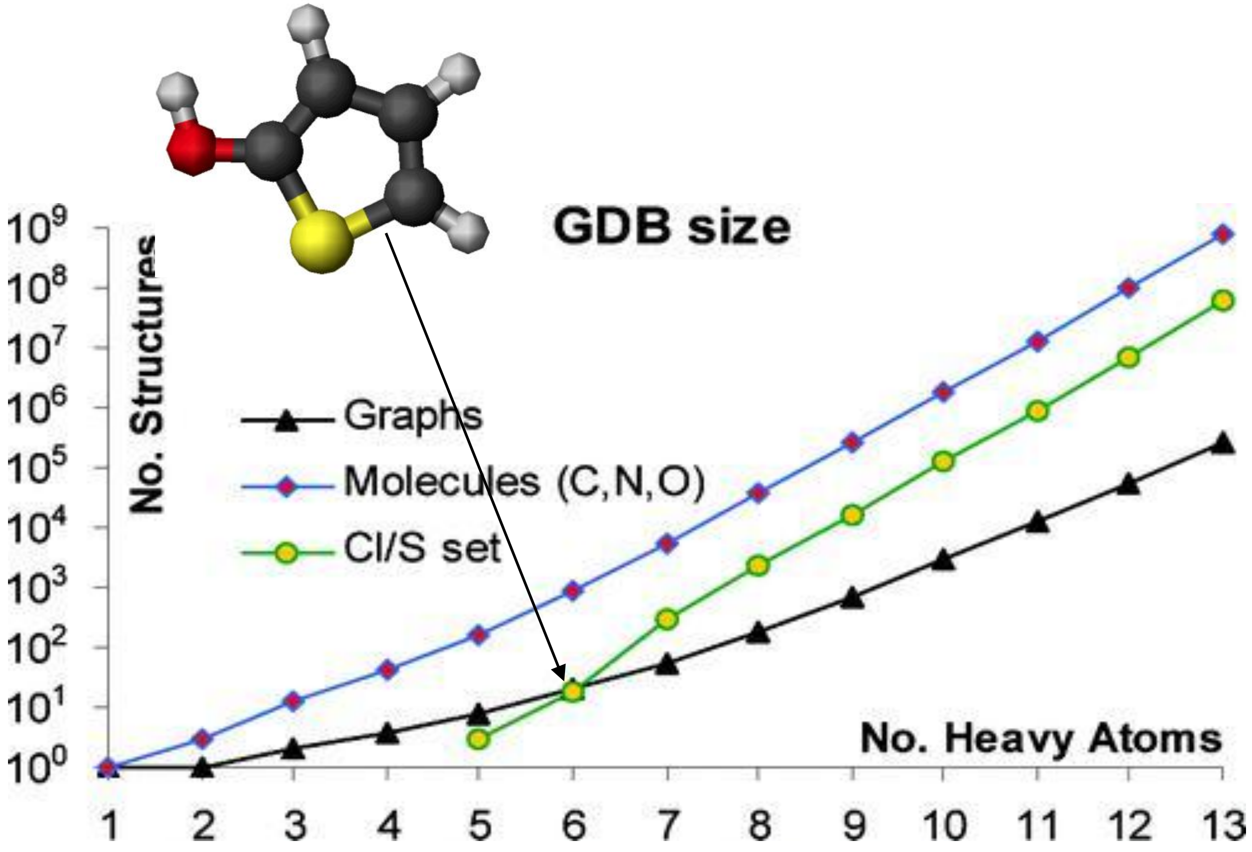
The data

GDB-13 database of all organic molecules (within stability & synthetic constraints) of 13 heavy atoms or less: 0.9B compounds

Table 1. Structure Generation Statistics for GDB-13



nodes[a]	graphs[b]	GDB[c]	Cl/S[d]	CPU time (h)[e]
1	1	1	0	0.00
2	1	3	0	0.00
3	2	12	0	0.00
4	4	43	0	0.00
5	8	155	3	0.01
6	20	934	19	0.02
7	57	5726	315	0.05
8	194	37151	2438	0.33
9	706	255542	17056	2.68
10	2831	1784626	130465	25.26
11	12011	12961686	938704	223.49
12	53789	99821343	7240108	3023.79
13	250268	795244451	59027533	36606.45
Total	**319892**	**910111673**	**67356641**	**39882.08**



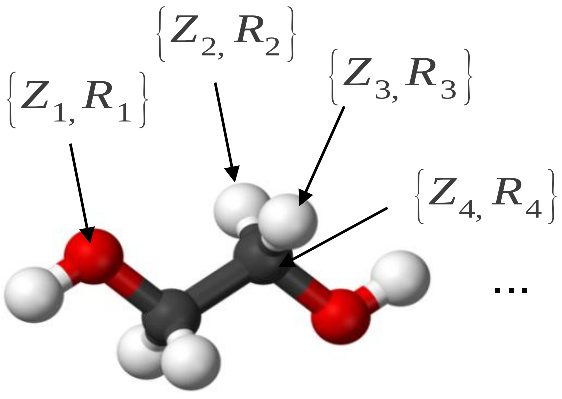
Blum & Reymond, *JACS* (2009)

[from von Lilienfeld]

Coulomb representation of molecules

$$M_{ii} = Z_i^{2.4}$$

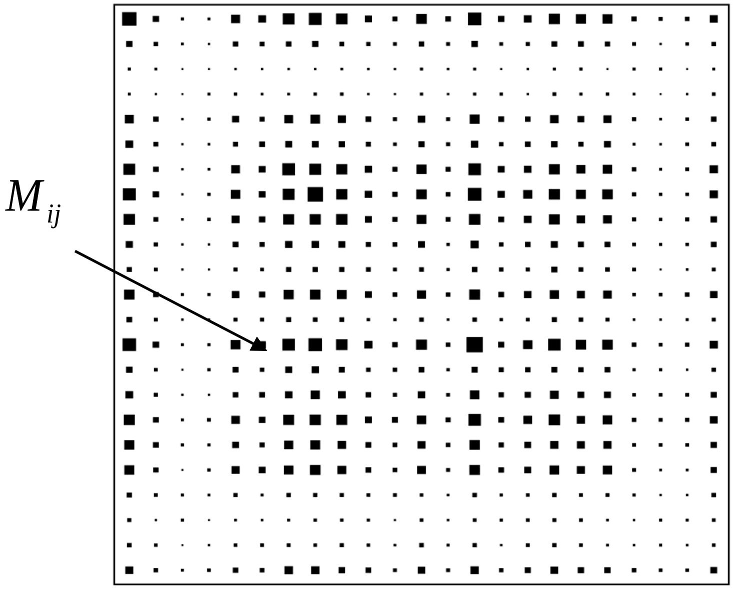
$$M_{ij} = \frac{Z_i Z_j}{\left\| R_i - R_j \right\|}$$



+ phantom atoms

$\{0, R_{21}\}$ $\{0, R_{22}\}$ $\{0, R_{23}\}$

$M \in {}^{23\times 23}$



Coulomb Matrix (Rupp, Müller et al 2012, PRL)

$$d(\mathbf{M}, \mathbf{M}') = \sqrt{\sum_{IJ} |M_{IJ} - M'_{IJ}|^2}$$

Kernel ridge regression

Distances between **M** define Gaussian kernel matrix **K**

$$k(\mathbf{M},\mathbf{M}') \;=\; \exp\left(-\frac{d(\mathbf{M},\mathbf{M}')^2}{2\sigma^2}\right)$$

Predict energy as sum over weighted Gaussians

$$\boxed{E^{est}(\mathbf{M}) \;=\; \sum_i \alpha_i k(\mathbf{M},\mathbf{M}_i) + b}$$

using weights that minimize error in training set

$$\min_{\alpha} \qquad \sum_i \left(E^{est}(\mathbf{M}_i) - E_i^{ref}\right)^2 + \lambda \sum_i \alpha_i^2$$

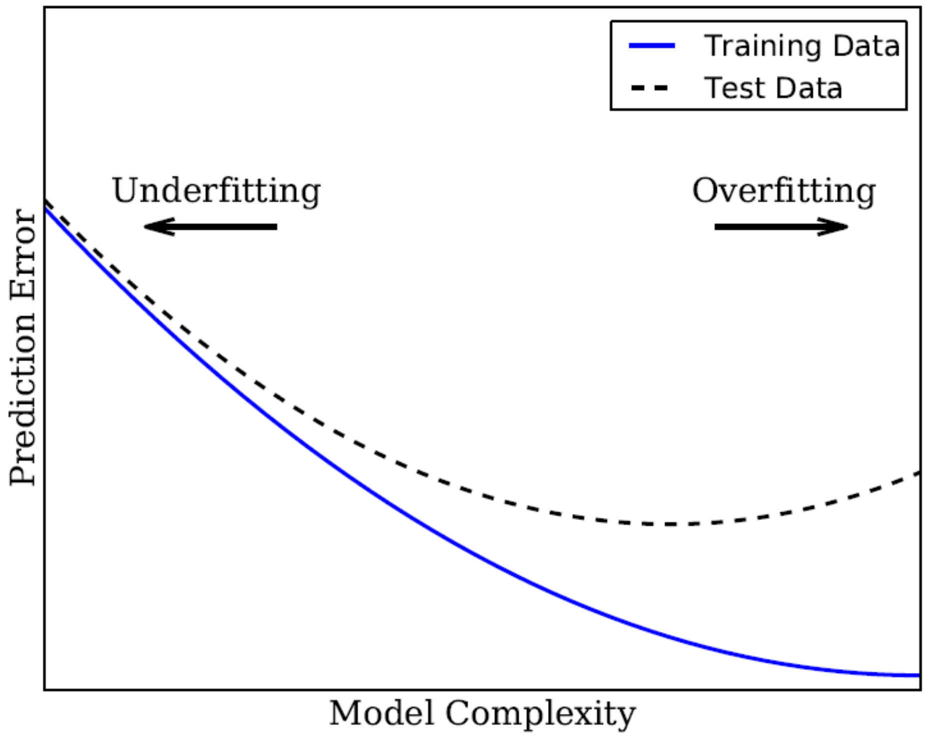
$$\alpha \;=\; (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{E}^{ref}$$

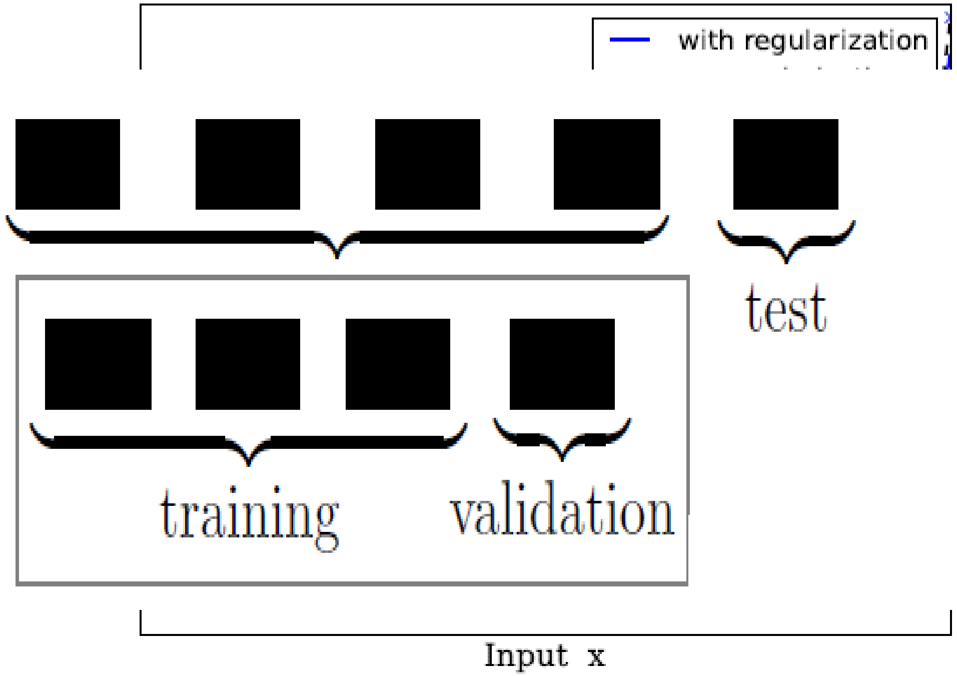
Exact solution

As many parameters as molecules + 2 global parameters, characteristic length-scale or kT of system (σ), and noise-level (λ)

[from von Lilienfeld]

Remarks on Generalization and Model Selection in ML





Kernel Ridge Regression Model

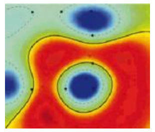
$$E^{est}(\mathbf{M}) = \sum_i \alpha_i k(\mathbf{M}, \mathbf{M}_i) + b$$

$$\min_{\alpha} \quad \sum_i \left(E^{est}(\mathbf{M}_i) - E_i^{ref}\right)^2 + \lambda \sum_i \alpha_i^2$$

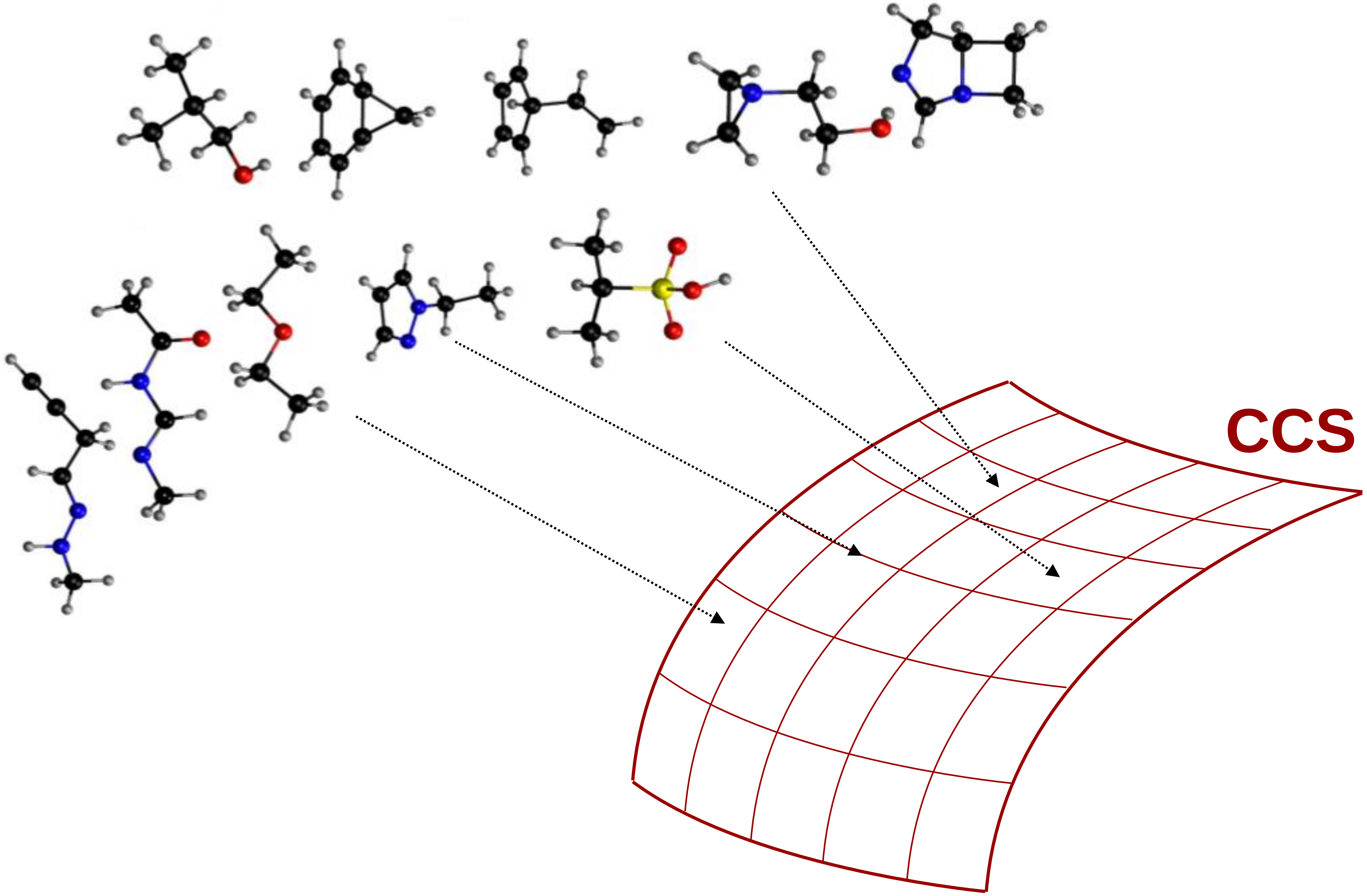
$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{E}^{ref}$$

ML4Physics: Part II Representations

Gregoire Montavon, Klaus-Robert Müller, Katja Hansen, Siamac Fazli, Franziska Biegler, Andreas Ziehe, Matthias Rupp, Anatole von Lilienfeld and Alexandre Tkachenko



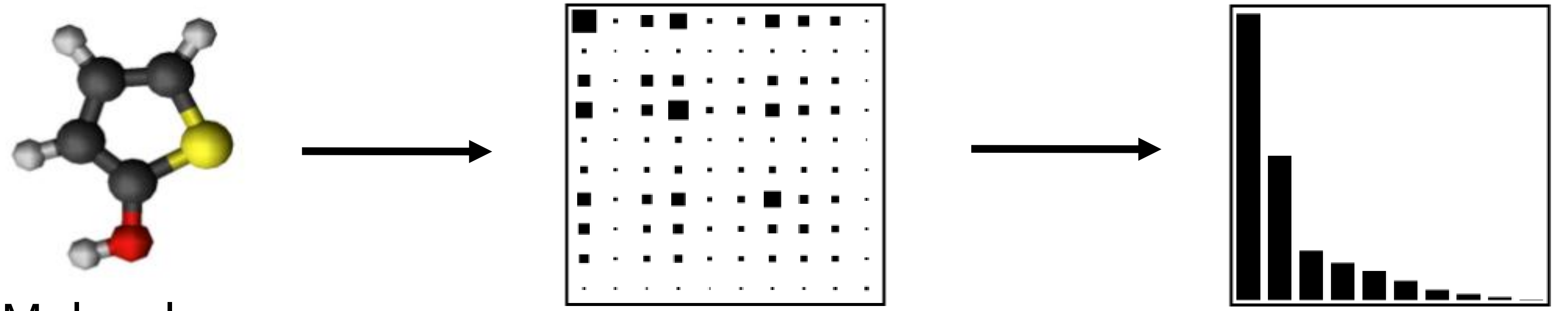
The chemical compound space (CCS)



Coulomb Eigenspectrum (Rupp et al. 12)

- For each Coulomb matrix C, compute its eigenspectrum λ, i.e. solutions to the eigenvalue problem:

$$Cx=\lambda x \quad \text{where} \quad \lambda_i \geq \lambda_{i+1}$$



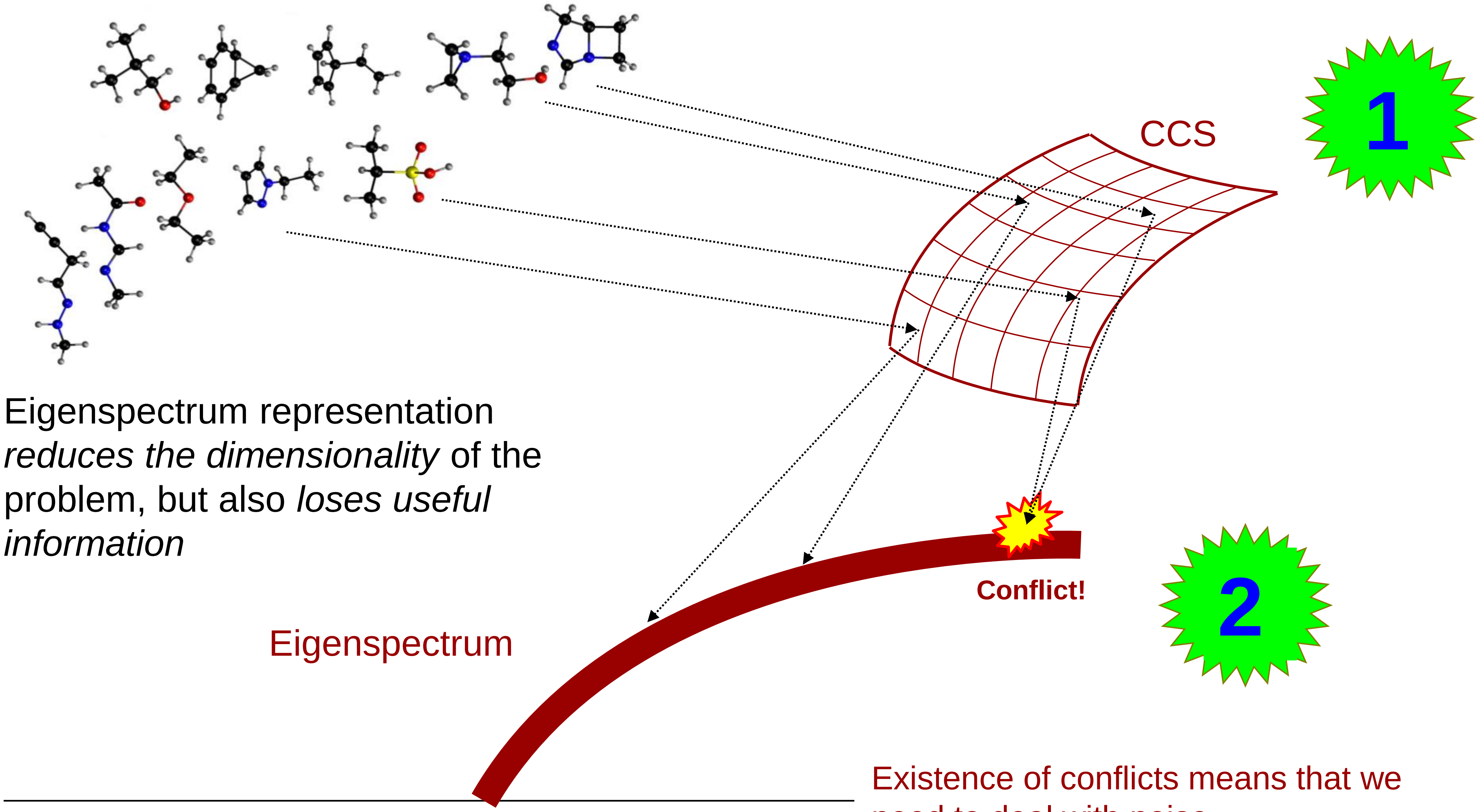
Molecule

Coulomb matrix

Eigenspectrum

- The eigenspectrum λ has only the square root of the number of dimensions of C.
- The eigenspectrum is invariant to permutation of atoms indices.

Coulomb Eigenspectrum

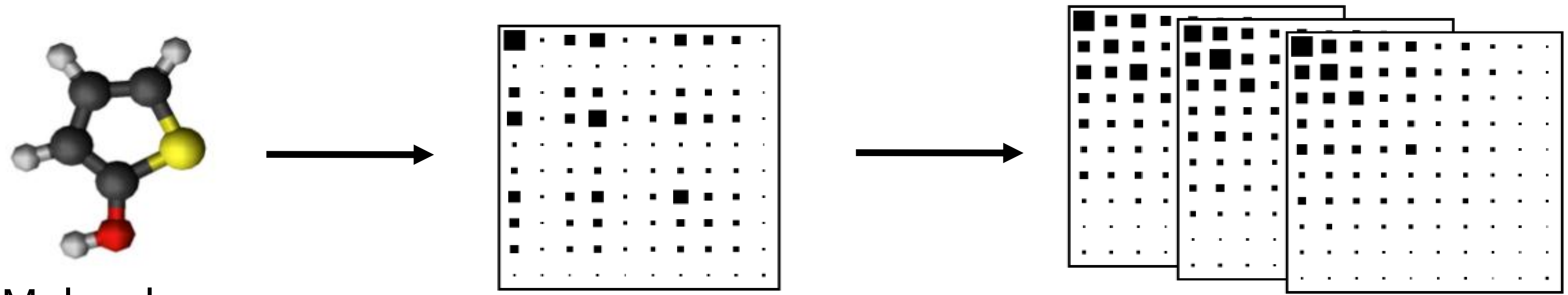


Eigenspectrum representation *reduces the dimensionality* of the problem, but also *loses useful information*

Existence of conflicts means that we need to deal with noise
→ impossible to learn in deep.

Coulomb sets (Montavon et al. 12)

- For each molecule, we collect a set of valid Coulomb matrices:

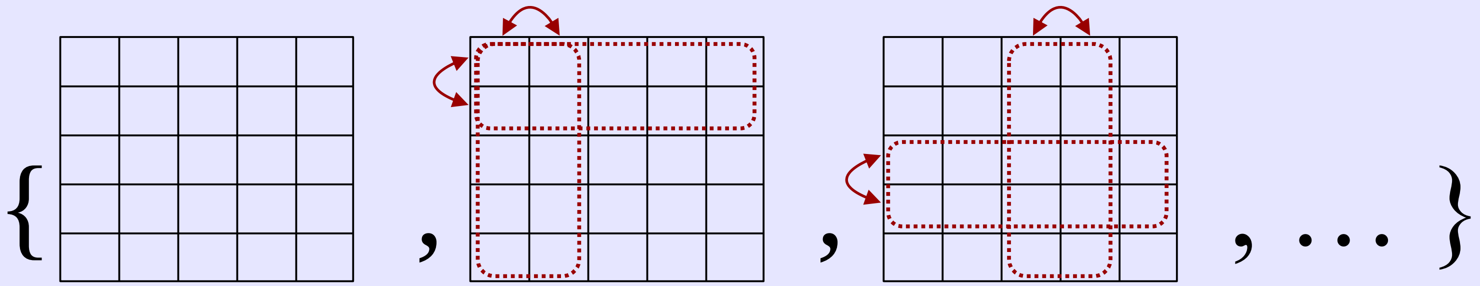


Molecule

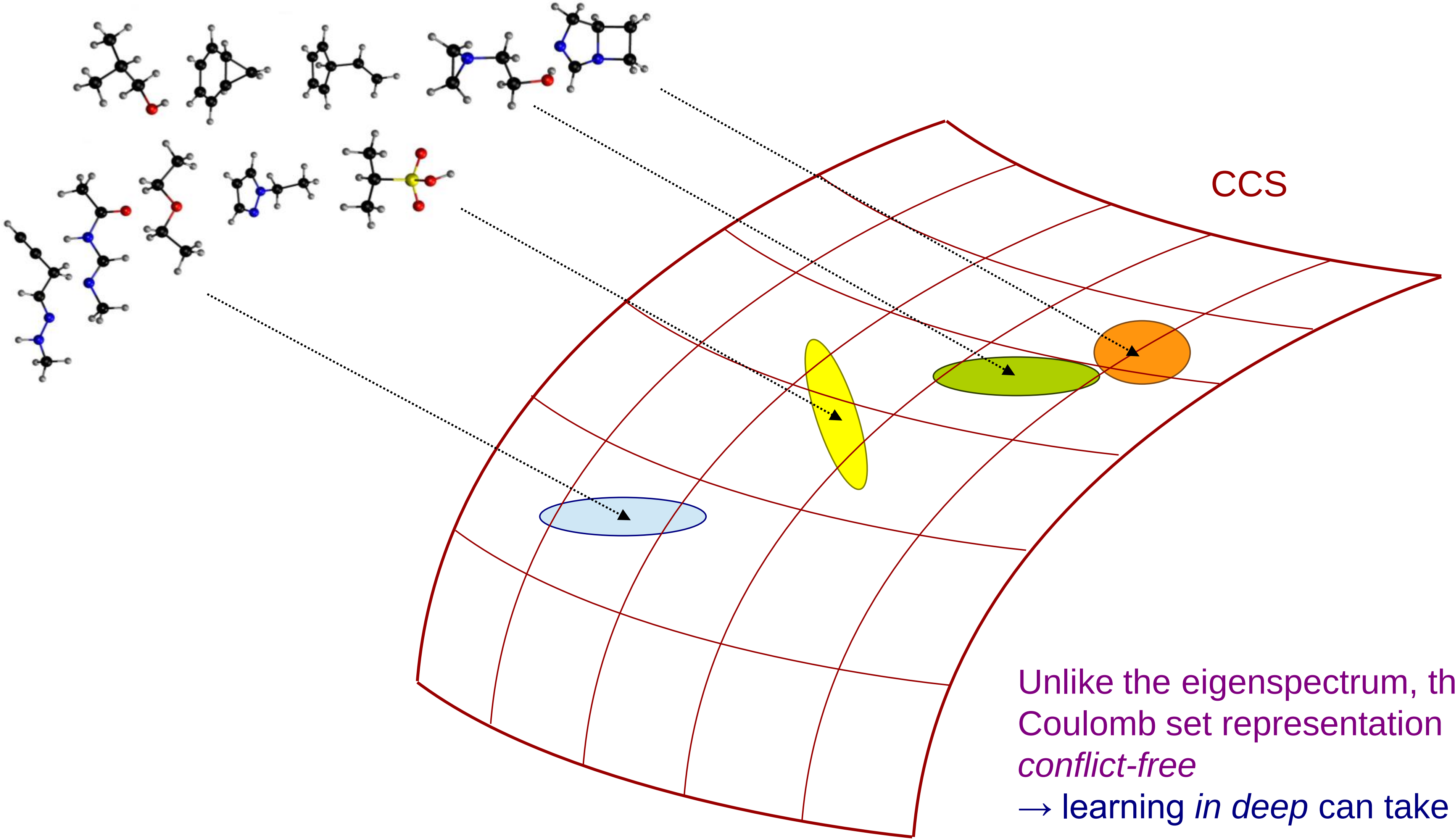
Coulomb matrix

Coulomb set

Coulomb set =



Coulomb sets

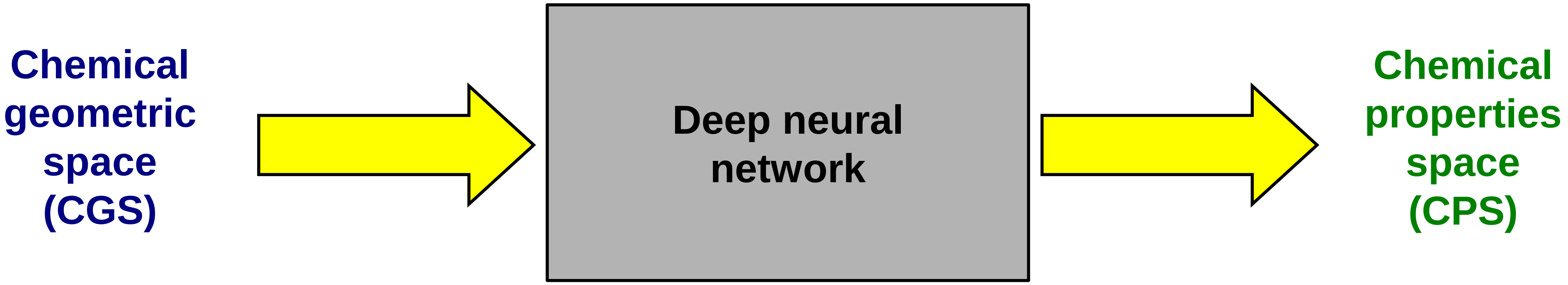


Unlike the eigenspectrum, the Coulomb set representation is *conflict-free*

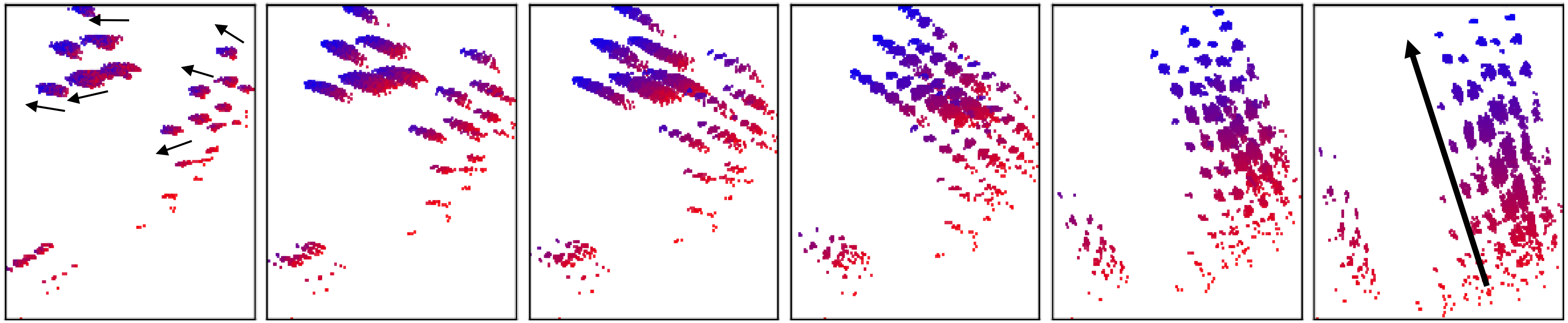
→ learning *in deep* can take place.

Deep neural networks

- Sequence of slight transformation of the representation implemented by artificial neurons.
- Each layer of the deep neural network encodes a slight **deformation** of the chemical compound space.
- Multiple layers progressively transform the representation from the input (molecular geometries) to the output (molecular properties).


Chemical geometric space (CGS)
Deep neural network
Chemical properties space (CPS)

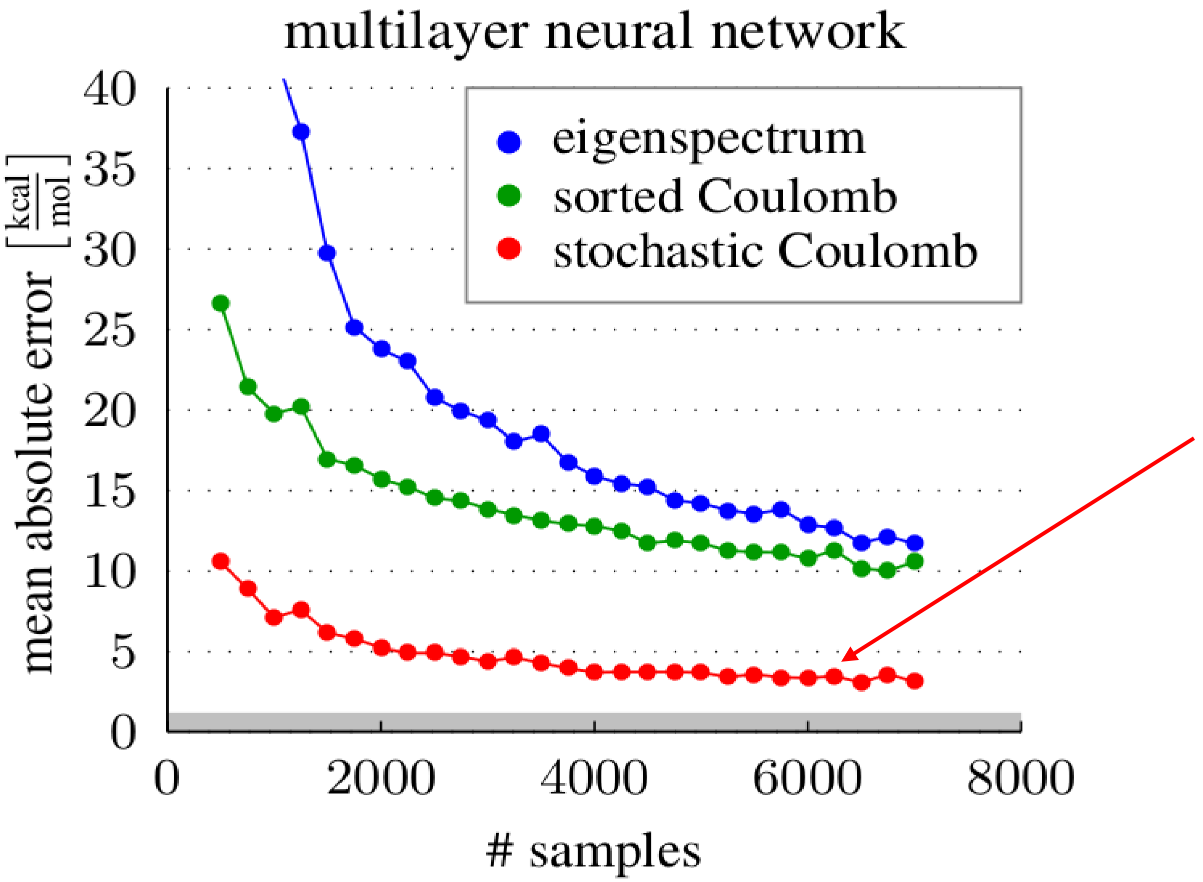
From geometries to energies



Input:
molecular geometries

Output:
molecular energies

Results



March 2012
Rupp et al., PRL
9.99 kcal/mol
(kernels + eigenspectrum)

December 2012
Montavon et al., NIPS
3.51 kcal/mol
(Neural nets + Coulomb sets)

Alex T. will show 1kcal/mol result

Prediction considered chemically accurate when MAE is below **1 kcal/mol**

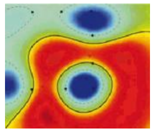
Dataset available at http://quantum-machine.org

ML4Physics @IPAM 2011 : Part III – Particles in a box

Demonstrate for a *very* simple system, we can ‘learn’ the exact kinetic energy functional

Klaus-Robert Müller, Matthias Rupp, Katja Hansen

Kieron Burke, John Snyder



ML4Physics @IPAM 2011 : Part IV

Zach Pouzon, Katja Hansen, Dan Sheppard,

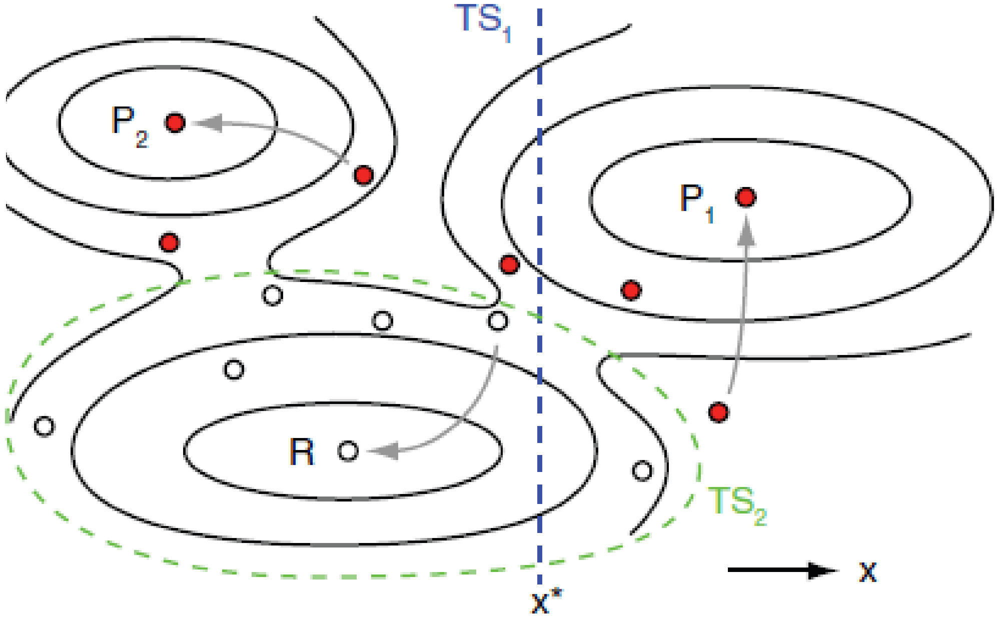
Matthias Rupp, Klaus-Robert Müller, Graeme Henkelman

Optimizing Transition State Theory with ML

• Within transition state theory the description of rare events is transformed from a problem of kinetics to one of equilibrium statistical mechanics by constructing a hypersurface that separates a reactant state from product states.

• Rate of reaction can be approximated by equilibrium flux out of this hypersurface

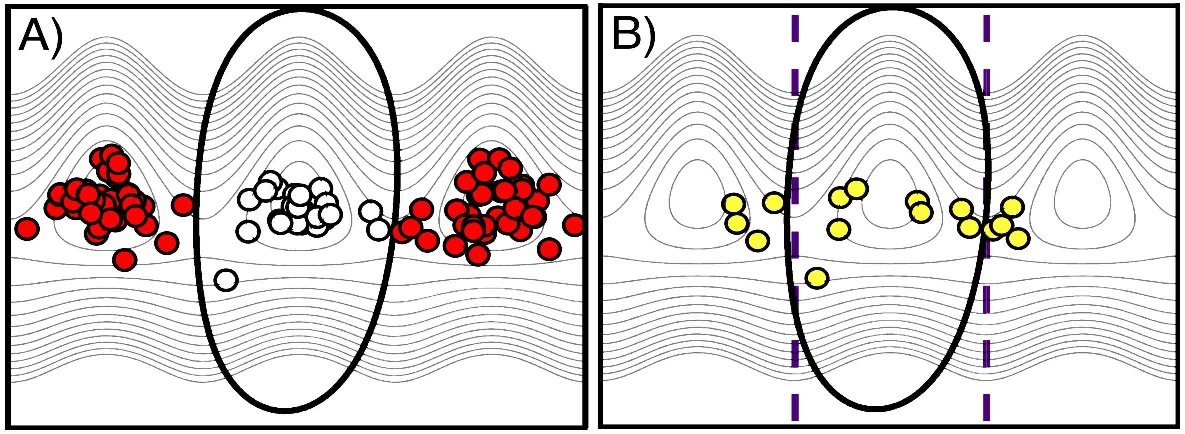
$$k_{\mathrm{TST}} = \frac{1}{2}\langle \delta(x - x^*)|\bar{v}|\rangle_{\mathrm{R}},$$



[Pozun et al 2012]

Our Approach

1. Run some high-temperature MD and generate an initial surface

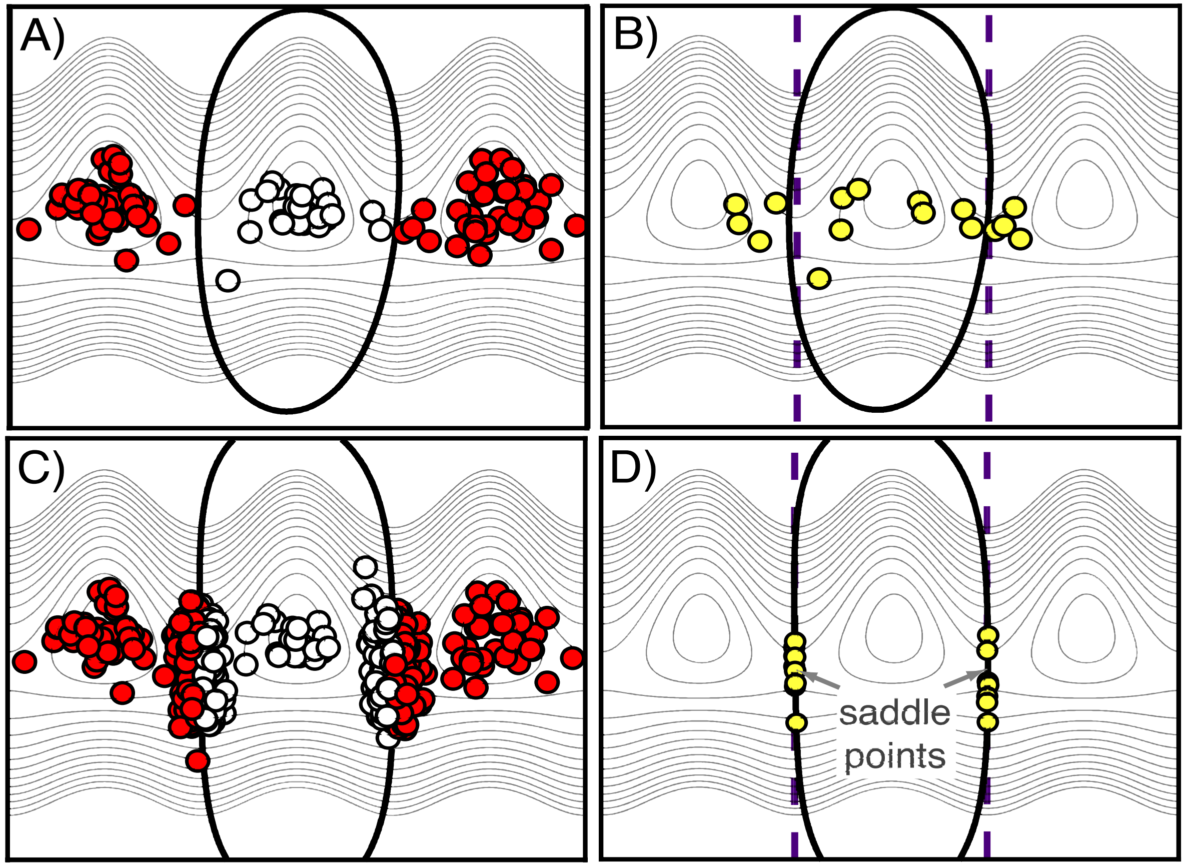


Potential from: A. F. Voter, J. Chem. Phys. **106**, 4665 (1997).

Our Approach

1. Run some high-temperature MD and generate an initial surface

2. Evaluate the gradients and attach a spring to the surface and continually sample and re-learn



Two parameters: C and γ

Potential from: A. F. Voter, J. Chem. Phys. **106**, 4665 (1997).

ML4Physics @ Halle: Materials

Kristof Schütt, Felix Brockherde, Wiktor Pronobis, Klaus-Robert Müller

and Henning Glawe, Antonio Sanna, Hardy Gross

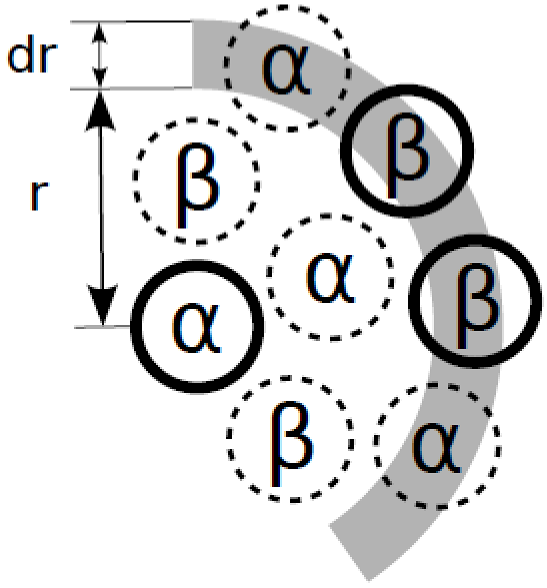
ML on Materials

Data: 5519 Materials with up to 8 atoms per cell, elements from spd

Features

Distribution of pair-wise distances for a pair of elements:

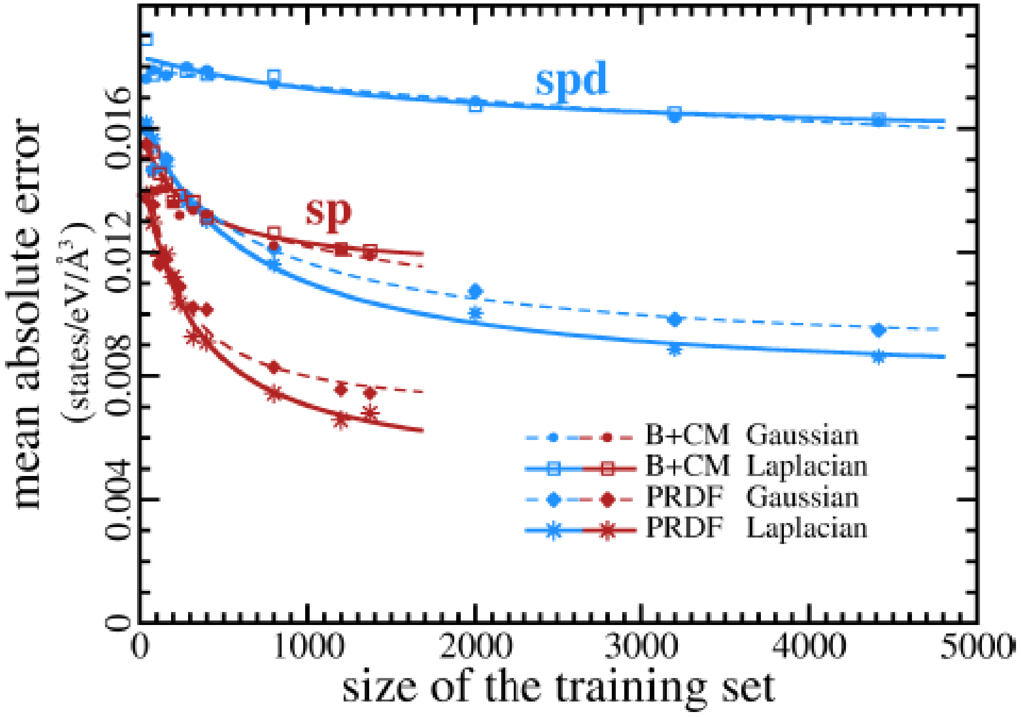
$$g_{\alpha\beta}(r) = \frac{1}{N_\alpha V_r} \sum_{i\in\alpha} \sum_{j\in\beta} \int_{r}^{r+dr} \delta(d_{ij} - s)\,ds$$



[Schütt et al 2012]

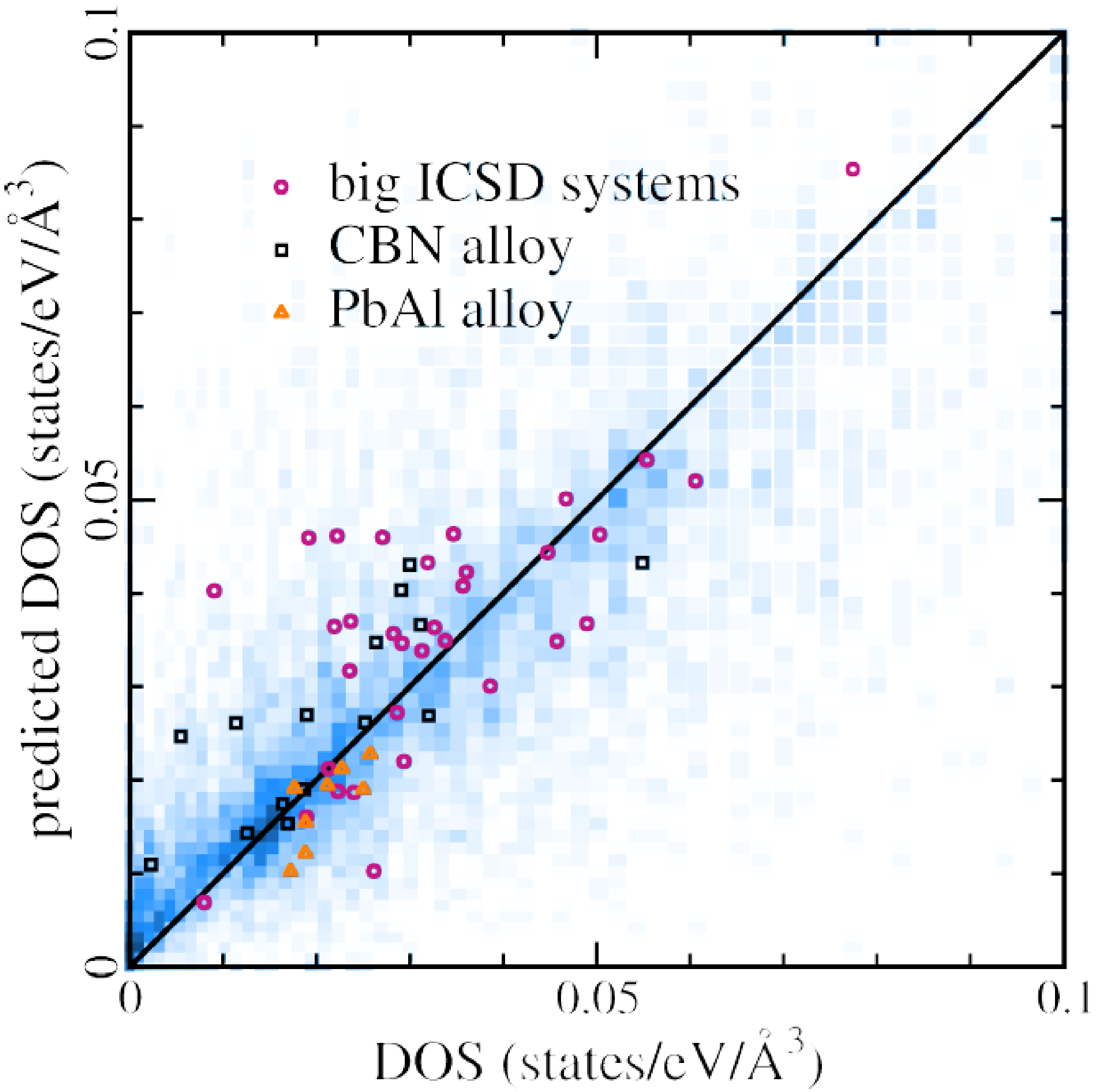
Lerning Curves

- Kernel Ridge Regression
- Gaussian / Laplacian Kernel
- Data set
  - 5519 Materials with up to 8 atoms / cell
  - elements from spd
- DFT-calculations of DOS at E_F



[Schütt et al 2012]

Results superconductors



[Schütt et al 2012]

Representations - remarks

- representations derived/learned by first principles information (unbiased)

    - Coulomb matrix, EVs, permuted coulomb matrix (Rupp et al, Montavon et al, Hansen et al.)
    - Fourier representation (Lilienfeld et al)
    - Bag of bonds (Hansen et al)
    - SOAP (Csanyi et al)
    - Neural Networks (Behler et al, Montavon et al)
    - Partial Radial Distribution functions (Schütt et al)

- representations using derived physical variables – using prior knowledge (biased)

    - feature selection from very large variable set (Ramprasad et al.)
    - feature selection from predefined physical variable set (Scheffler et al.)

Challenge: How to gain better **understanding** from ML representation 4 Physics, see Bag of bonds!

Conclusion

- Machine Learning & modern data analysis is of central importance in daily life
- input to ML algorithms can be vectors, matrices, graphs, strings, tensors etc.
- Representation is essential ! Modelselection, Optimization.
- ML 4 XC, ML for reaction transitions, ML for formation energy prediction etc.
- ML challenges from Physics: no noise, high dimensional systems, functionals …
- challenge: learn for Physics from ML representation: towards better understanding

See also: www.quantum-machine.org

Some Publication (see also quantum-machine.org)

Quantum machine

M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld: Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning, Physical Review Letters, 108(5):058301, 2012

G. Montavon, K. Hansen, S. Fazli, M. Rupp, F. Biegler, A. Ziehe, A. Tkatchenko, O. A. von Lilienfeld, K.-R. Müller, Learning Invariant Representations of Molecules for Atomization Energy Prediction, Advances in Neural Information Processing Systems (NIPS), 2012

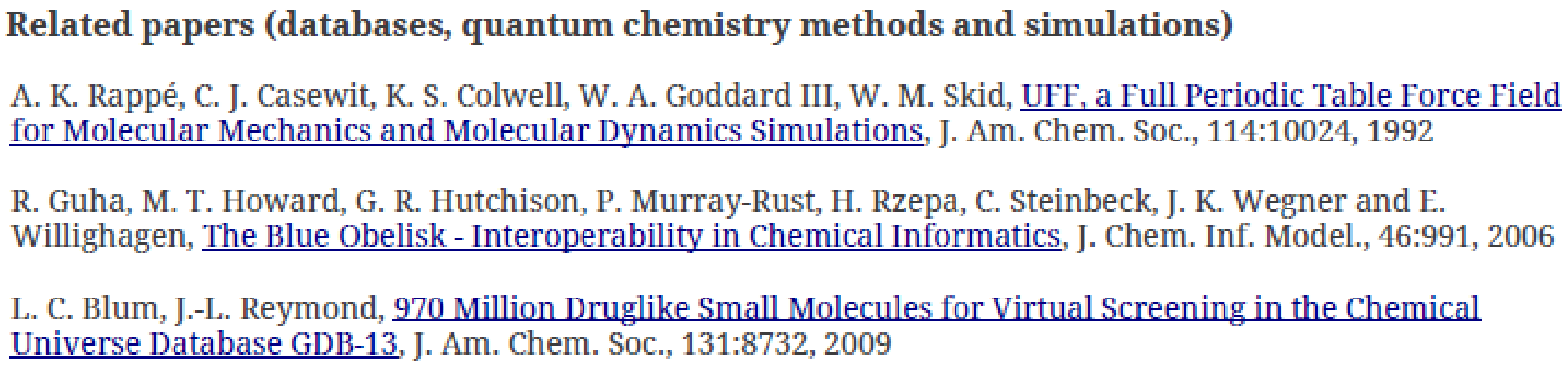
G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, O.A. von Lilienfeld, Machine Learning of Molecular Electronic Properties in Chemical Compound Space, New Journal of Physics, 2013

K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, K.-R. Müller. Assessment and Validation of Machine Learning Methods for Predicting Molecular Energies, J. Chem. Theory Comput., 2013

Snyder, J. C., Rupp, M., Hansen, K., Müller, K. R., & Burke, K. Finding density functionals with machine learning. *Physical review letters*, *108*(25), 253002. 2012.

Pozun, Z. D., Hansen, K., Sheppard, D., Rupp, M., Müller, K. R., & Henkelman, G., Optimizing transition states via kernel-based machine learning. The Journal of chemical physics, 136(17), 174101. 2012 .

K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, How to represent crystal structures for machine learning: Towards fast prediction of electronic properties
Phys. Rev. B 89, 205118 (2014)

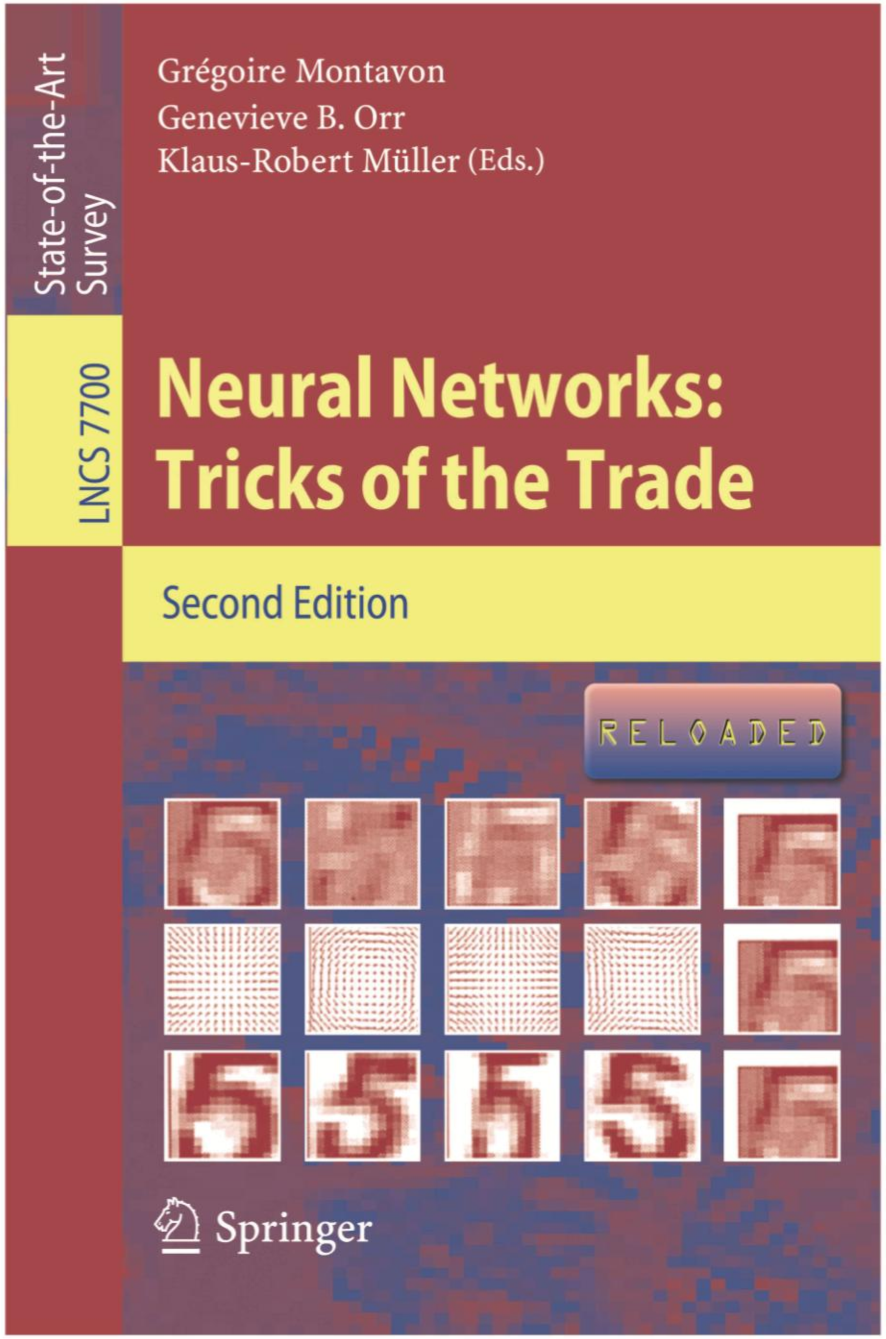


Related papers (databases, quantum chemistry methods and simulations)

A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard III, W. M. Skid, UFF, a Full Periodic Table Force Field for Molecular Mechanics and Molecular Dynamics Simulations, J. Am. Chem. Soc., 114:10024, 1992

R. Guha, M. T. Howard, G. R. Hutchison, P. Murray-Rust, H. Rzepa, C. Steinbeck, J. K. Wegner and E. Willighagen, The Blue Obelisk - Interoperability in Chemical Informatics, J. Chem. Inf. Model., 46:991, 2006

L. C. Blum, J.-L. Reymond, 970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13, J. Am. Chem. Soc., 131:8732, 2009



State-of-the-Art Survey

LNCS 7700

Grégoire Montavon
Genevieve B. Orr
Klaus-Robert Müller (Eds.)

Neural Networks: Tricks of the Trade

Second Edition

Springer