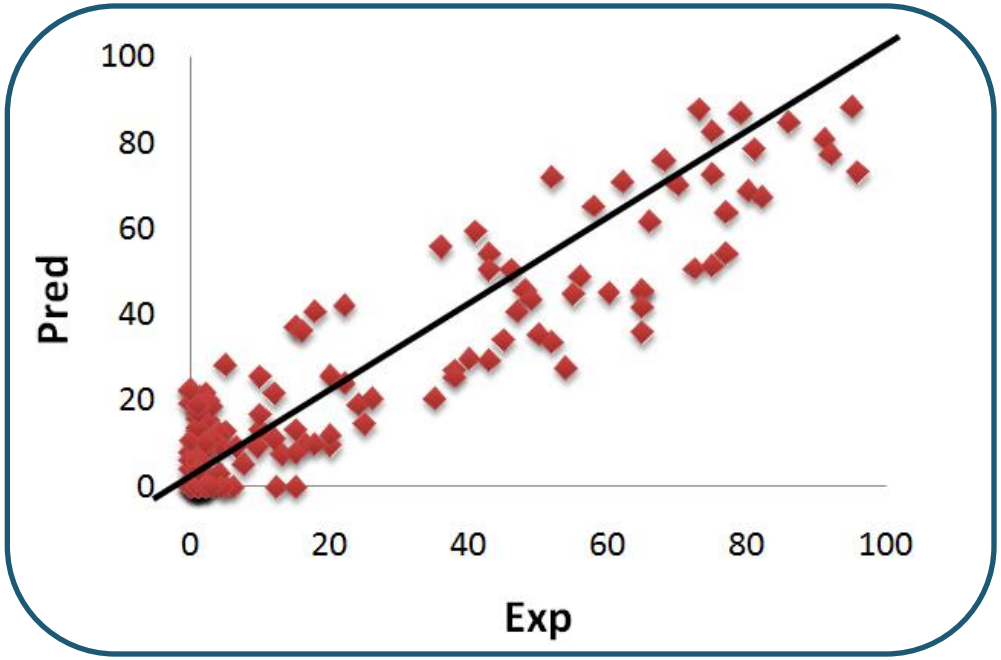
# РЕГРЕССИОННЫЕ МЕТОДЫ

# СТАТИСТИЧЕСКИЕ ПАРАМЕТРЫ ОЦЕНКИ ПРОГНОСТИЧЕСКОЙ СПОСОБНОСТИ РЕГРЕССИОННЫХ МОДЕЛЕЙ

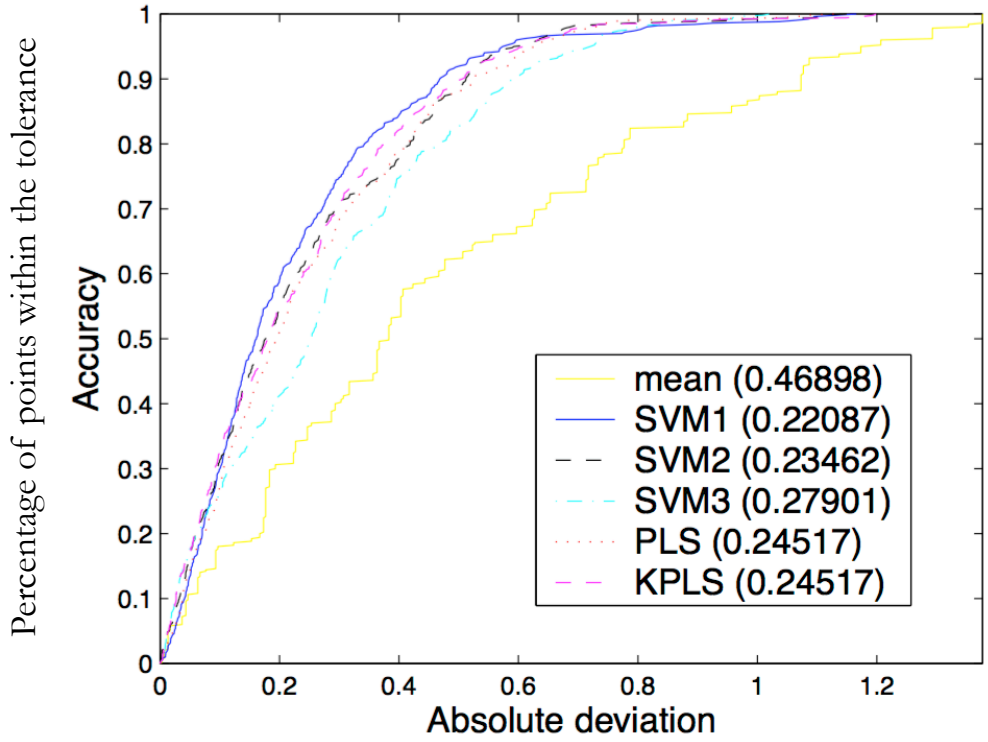



$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_{pred,i} - y_{exp,i}\right)^2}{\sum_{i=1}^{n}\left(y_{exp,i} - \overline{y}_{exp,i}\right)^2}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_{pred,i} - y_{exp,i}\right)^2}{n}}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|\left(y_{pred,i} - y_{exp,i}\right)\right|}{n}$$

# СТАТИСТИЧЕСКИЕ ПАРАМЕТРЫ ОЦЕНКИ ПРОГНОСТИЧЕСКОЙ СПОСОБНОСТИ РЕГРЕССИОННЫХ МОДЕЛЕЙ: REGRESSION ERROR CHARACTERISTIC (REC) CURVE




$$x^m = (x_i, y_i)_{i=1}^m \qquad x_i \in \mathbb{R}^n \qquad y \in \mathbb{R}$$

$$acc(\epsilon) := \frac{|\{(x,y): loss(f(x_i), y_i) \leq \epsilon, i = 1, \ldots, m\}|}{m}$$

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

# МНОЖЕСТВЕННАЯ ЛИНЕЙНАЯ РЕГРЕССИЯ

$$f(x,\alpha) = \sum_{j=1}^{n} \alpha_j f_j(x) \qquad \alpha \in R^N$$

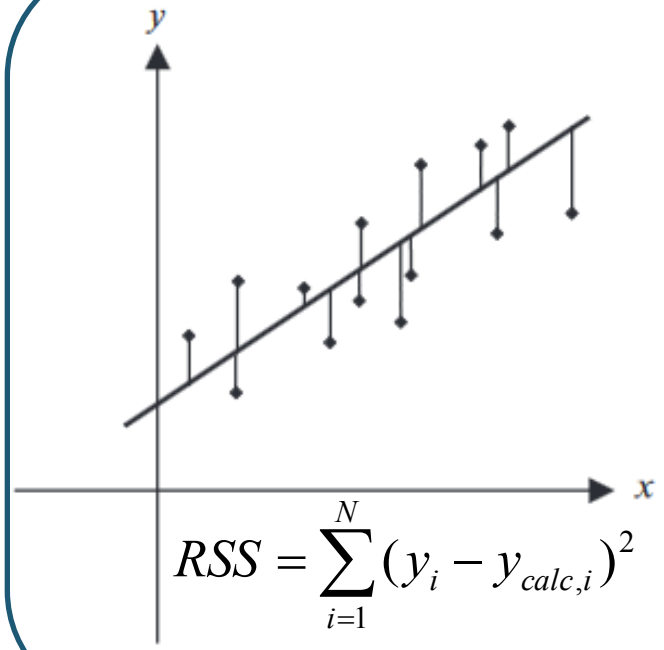
**Минимизируется функционал квадрата ошибки**

$$Q(\alpha, x^l) = \sum_{i=1}^{l} (f(x_i,\alpha) - y_i)^2 = \|F\alpha - y\|^2 \longrightarrow min$$

$$\frac{\partial Q}{\partial \alpha}(\alpha) = 2F^T(F\alpha - y) = 0$$

$$F^T F\alpha = F^T y$$

$$\alpha = (F^T F)^{-1} F^T y$$




**Сингулярное разложение**

$$F = VDU^T$$

$$V^T V = I$$

$$U^T U = I$$

$$D = diag(\sqrt{\lambda_i}, \ldots, \sqrt{\lambda_n}) \qquad (\lambda_j \geq 0 \text{ собственные значения матриц } FF^T \text{ и } F^T F)$$

$$\alpha^* = UD^{-1}V^T y = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j (v_j^T y)$$

# ГРЕБНЕВАЯ РЕГРЕССИЯ

**Множественная линейная регрессия**

$$\alpha^{*} = UD^{-1}V^{T}y = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j \left(v_j^{T} y\right)$$

**Введение «гребня» (штрафа за увеличение нормы весов)**

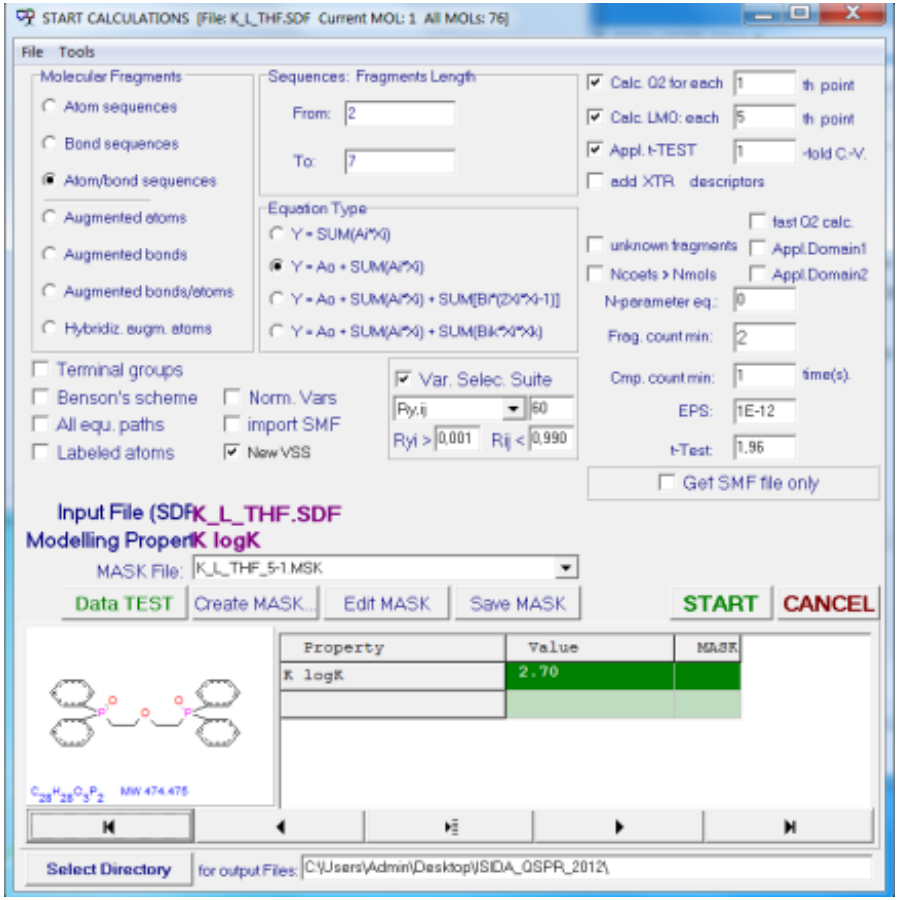
$$\alpha = (F^{T}F + \tau I)^{-1}F^{T}y$$

$$\alpha^{*} = U(D^{2}+\tau I)^{-1}DV^{T}y = \sum_{j=1}^{n} \frac{\sqrt{\lambda_j}}{\lambda_i + \tau} u_j \left(v_j^{T} y\right)$$

$\tau -$ неотрицательный параметр регуляризации

**Величина τ может быть подобрана при помощи процедуры перекрестного контроля**

# Множественная линейная регрессия: ISIDA QSPR




http://infochim.u-strasbg.fr/recherche/Download/Download.php

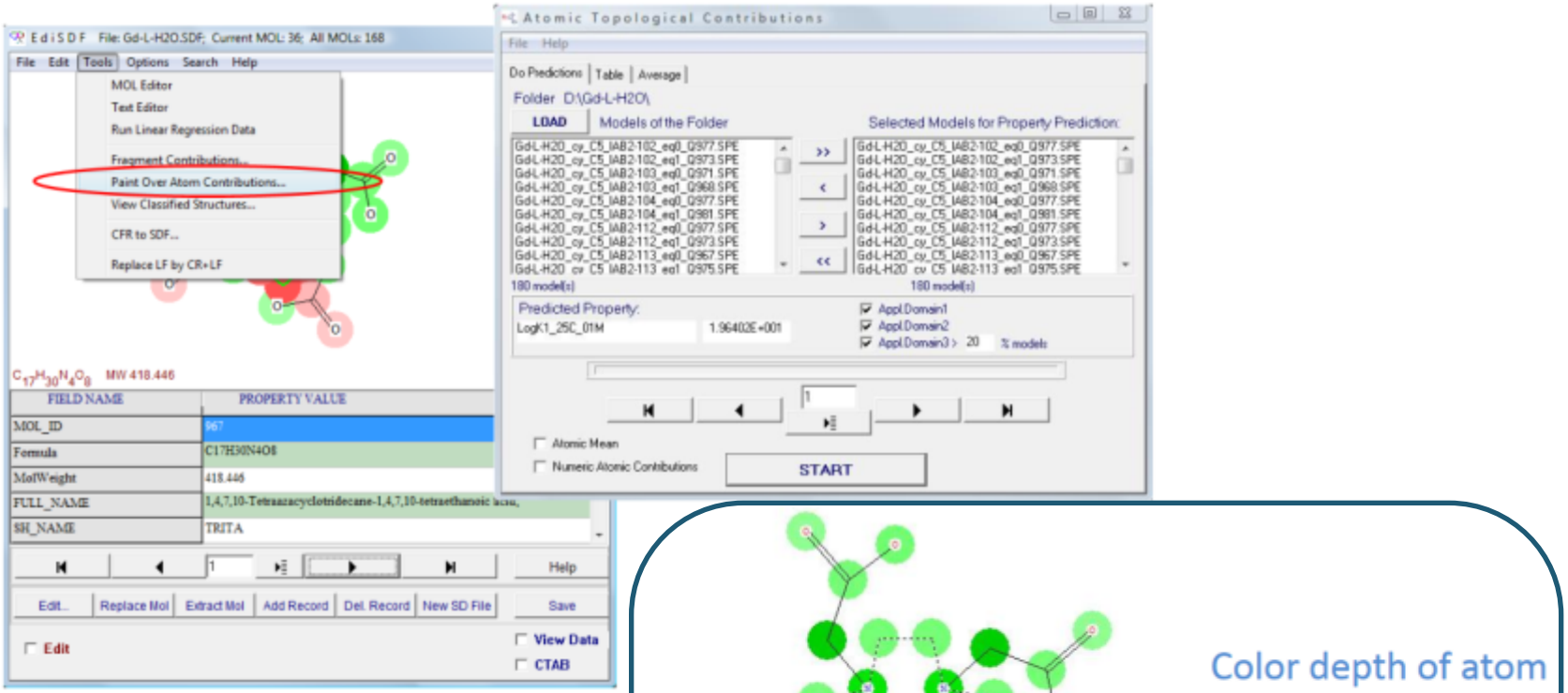
http://vpsolovev.ru/programs/

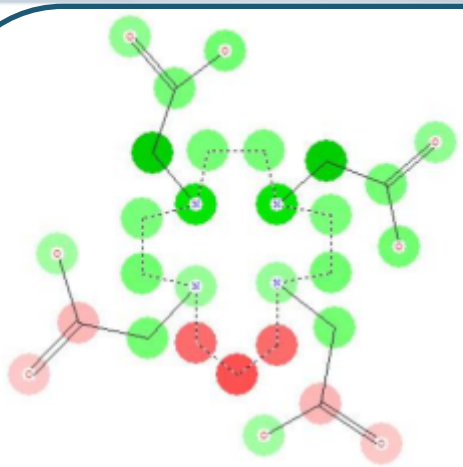

Varnek A., Solov'ev V. Rev. in Book: *Ion Exchange and Solvent Extraction, A Series of Advances*, **2009,** *19*, pp. 319-358

Katritzky A.,Dobchev D.,Fara D.,Hur E.,Tamm K.,Kurunczi L.,Karelson M.,Varnek A.,Solov'ev V. *J.Med. Chem.*, **2006,** *49*, p. 3305

De Luca A., Horvath D., Marcou G., Solov'ev V., Varnek A. *J. Chem. Inf. Model.*, **2012**, *52*, pp. 2325–2338

# Множественная линейная регрессия: ISIDA QSPR



Color depth of atom

$$c_A = \sum_{j=1}^{M} \sum_{A \in F_j} a_{ij}$$

equilibrium

$Gd^{3+} + L = (Gd^{3+})L$

stability constant

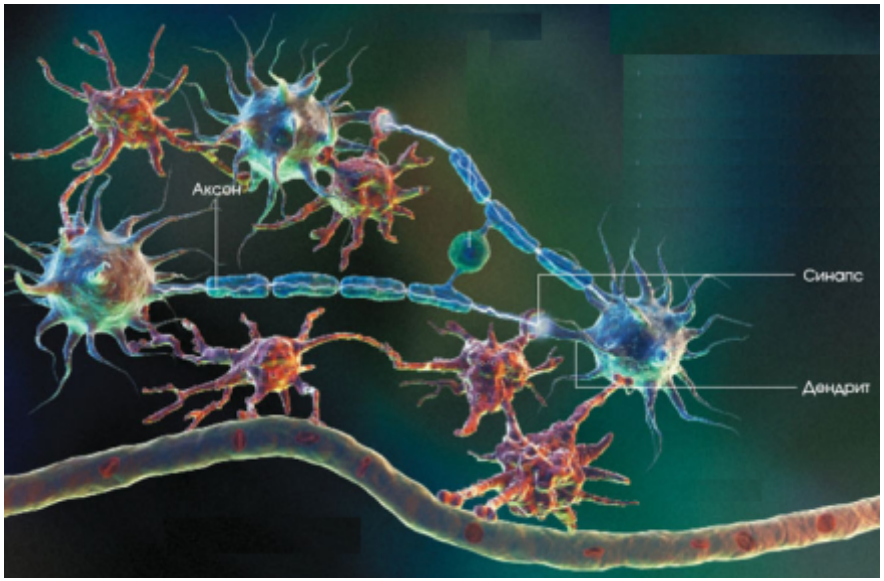
$\log K_{exp} = 19.2$

topological contributions of ligand atoms

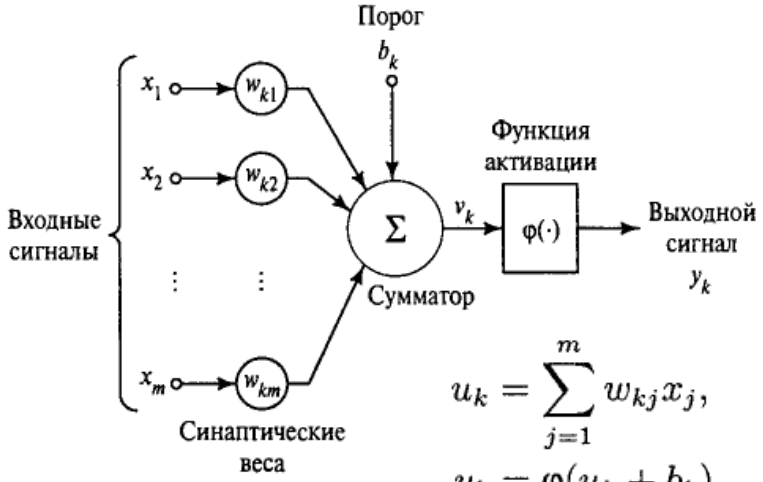
Слайд из доклада В.П. Соловьева на летней школе-конференции по хемоинформатике в Казани (2013)

# Искусственные нейронные сети: биологический и искусственный нейрон

Искусственные нейронные сети (ИНС) — математические модели, построенные по принципу организации и функционирования биологических нейронных сетей, моделирующие способ обработки мозгом конкретной задачи.




**Нелинейная модель нейрона**




$$u_k = \sum_{j=1}^{m} w_{kj} x_j,$$

$$y_k = \varphi(u_k + b_k)$$

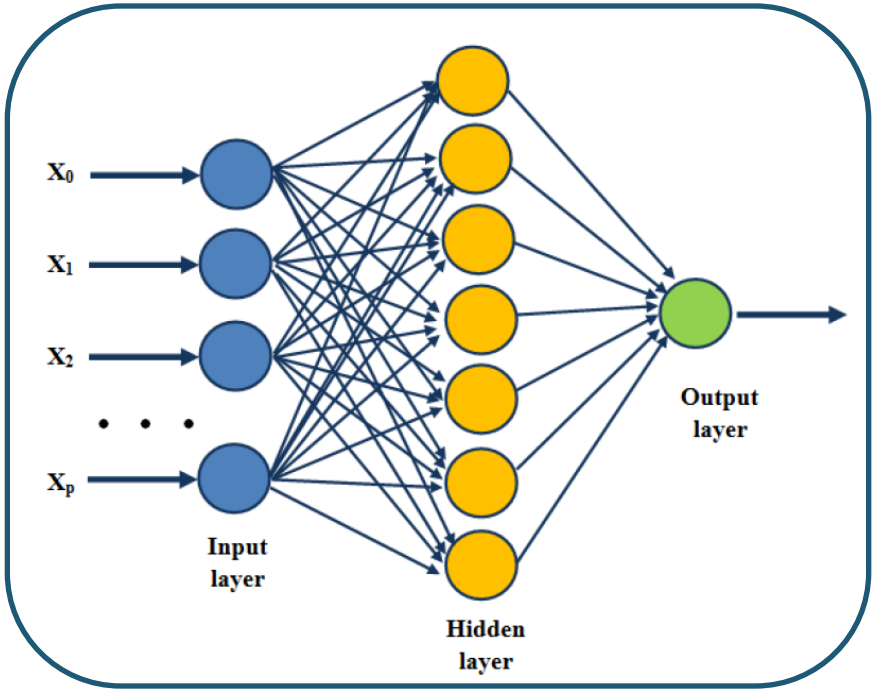
$$v_k = u_k + b_k.$$

**Три основных элемента:**

- Набор синапсов или связей, каждый из которых характеризуется своим весом
- Сумматор складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона (линейная комбинация)
- Функция активации ограничивает амплитуду входного сигнала (также называется функцией сжатия)

# Нейронные сети: многослойный персептрон

**Многослойный персептрон – многослойная сеть прямого распространения, состоящая из множества входных узлов, формирующих входной слой, одного или нескольких скрытых слоев вычислительных нейронов и одного выходного слоя**

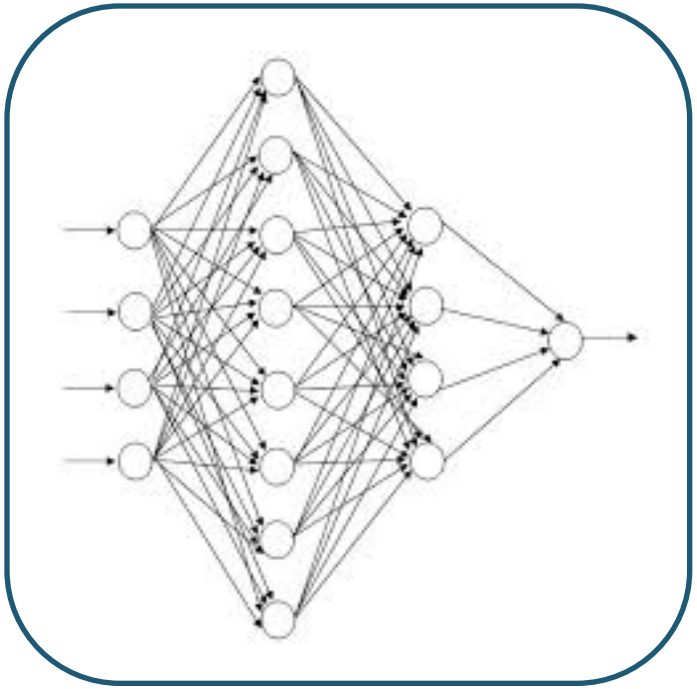


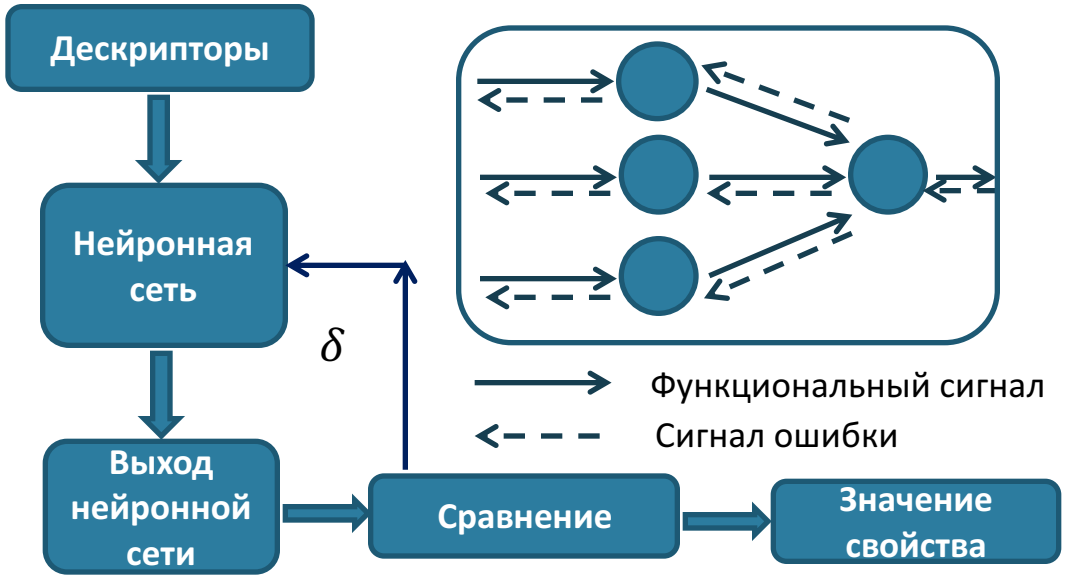

- Нейроны входного слоя соответствуют дескрипторам, нейроны выходного – прогнозируемым свойствам, нейроны скрытого слоя – нелинейным латентным переменным
- Обучение выполняется при помощи алгоритма обратного распространения ошибки

Обучение предполагает два прохода по всем слоям сети (прямого и обратного):

- При прямом проходе входной вектор подается на входной слой и сигнал распространяется от слоя к слою, что генерирует выходной сигнал (при прямом проходе все синаптические веса фиксированы).
- Во время обратного прохода осуществляется настройка синаптических весов (сравнение прогнозируемого и целевого значений свойства для оценки значения ошибки, их разность (сигнал ошибки) распространяется в обратном направлении)

# Многослойный персептрон: алгоритм обратного распространения






Два типа вычислений нейронов

**Вычисление функционального сигнала на выходе нейрона**

- Реализуемое в виде непрерывной нелинейной функции от входного сигнала и синаптических весов, связанных с данным нейроном

**Вычисление оценки вектора градиента**

- Вычисление градиента поверхности ошибки по синаптическим весам, связанным со входом данного нейрона

# Многослойный персептрон: алгоритм обратного распространения

**Общая энергия ошибки сети:**

$$\mathbf{E}(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n)$$

$$e_j(n) = d_j(n) - y_j(n)$$

сигнал ошибки выходного нейрона *j* на итерации *n* (соответствующей *n*-му примеру обучения):

*N* – общее число образов

**Текущая сумма квадратов ошибок или энергия среднеквадратичной ошибки (функция стоимости) по всей выборке:**

$$\mathbf{E}_{\text{av}}(n) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}(n)$$

**Индуцированное локальное поле (взвешенная сумма всех синаптических входов плюс порог):**

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n)$$

m – общее число входов

**Функциональный сигнал $y_j(n)$ на выходе нейрона *j* на итерации *n*:** $y_j(n) = \varphi_j(v_j(n))$

**Применение к синаптическому весу коррекции, пропорциональной частной производной $\partial \mathbf{E}(n)/\partial w_{ji}(n)$ согласно дельта-правилу:**

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathbf{E}(n)}{\partial w_{ji}(n)}$$

где $\eta$ – параметр скорости обучения

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

# Многослойный персептрон: алгоритм обратного распространения

## Инициализация

Генерация синаптических весов и пороговых значений

## Предъявление примеров обучения

Сети передаются входные вектора, для каждого из которых последовательно выполняется прямой и обратный проходы (последовательно вычисляются индуцированные локальные поля нейронов)

## Прямой проход

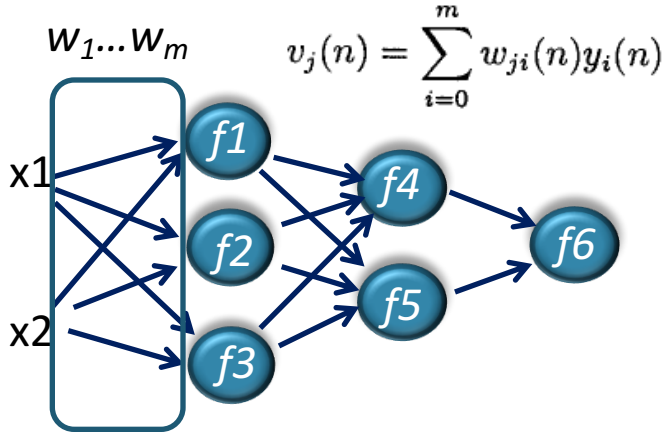
Вычислить выход сети, вычислить сигнал ошибки

## Обратный проход

Корректировка весов для минимизации ошибки

## Итерации

Последовательно выполнить прямой и обратный проходы до достижения критерия останова




Выходной сигнал нейрона *j* слоя *l*:

$$y_j^{(l)}(n) = \varphi_j(v_j(n))$$

Если нейрон находится в первом скрытом слое:

$$y_j^{(0)}(n) = x_j(n)$$

Если нейрон находится в выходном слое:

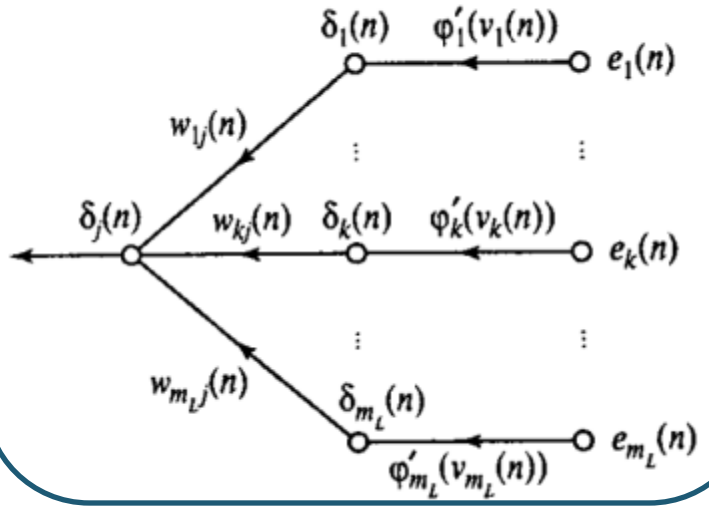
$$y_j^{(L)}(n) = o_j(n)$$

Сигнал ошибки:

$$e_j(n) = d_j(n) - o_j(n)$$

# Алгоритм обратного распространения: локальный градиент

**Граф передачи сигнала для процесса обратного распространения ошибки**




$$\Delta w_{ji}(n) =$$

**Параметр скорости обучения** $\eta$ • **Локальный градиент** $\delta$ • **Входной сигнал нейрона** $y_i(n)$

Нейрон ***j*** – выходной (известен желательный отклик):

$$\delta_j(n) = -\frac{\partial \mathbf{E}(n)}{\partial v_j(n)} = -\frac{\partial \mathbf{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n)\varphi_j'(v_j(n))$$

Нейрон ***j*** – скрытый (необходимость идентифицировать вклад отдельных скрытых нейронов в величину общей ошибки):

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

$\varphi_j'(v_j(n))$ - производная функции активации

$\sum_k \delta_k(n) w_{kj}(n)$ - взвешенная сумма градиентов, вычисленных для нейронов следующего слоя

# Многослойный персептрон: алгоритм обратного распространения

## Инициализация

Генерация синаптических весов и пороговых значений

## Предъявление примеров обучения

Сети передаются входные вектора, для каждого из которых последовательно выполняется прямой и обратный проходы (последовательно вычисляются индуцированные локальные поля нейронов)

## Прямой проход

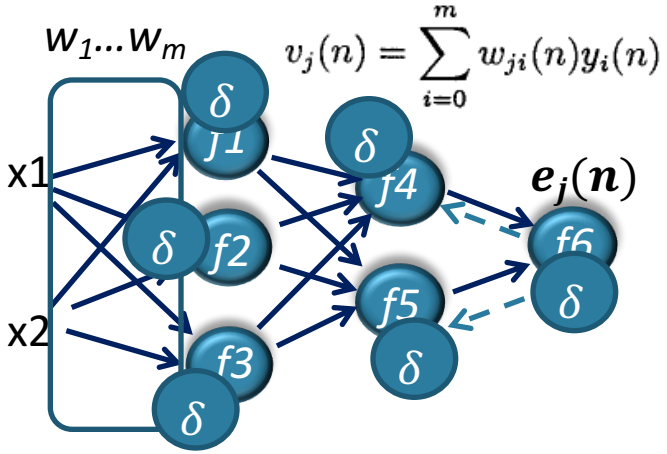
Вычислить выход сети, вычислить сигнал ошибки

## Обратный проход

Корректировка весов для минимизации ошибки

## Итерации

Последовательно выполнить прямой и обратный проходы до достижения критерия останова




Выходной сигнал нейрона *j* слоя *l*:

$$y_j^{(l)}(n) = \varphi_j(v_j(n))$$

Если нейрон находится в первом скрытом слое:

$$y_j^{(0)}(n) = x_j(n)$$

Если нейрон находится в выходном слое:

$$y_j^{(L)}(n) = o_j(n)$$

Сигнал ошибки:

$$e_j(n) = d_j(n) - o_j(n)$$

# Регрессия метода опорных векторов

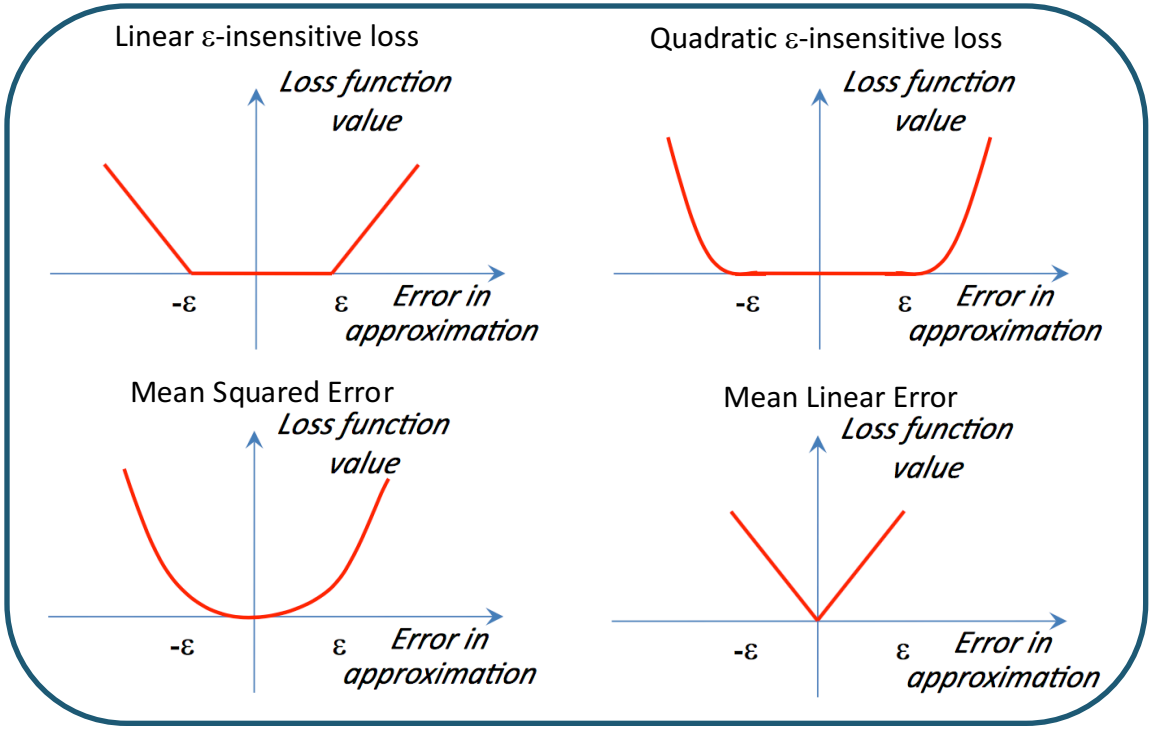
$$d = f(x) + \vartheta \qquad \{(x_i, d_i)\}_{i=1}^{N}$$

$$y = \sum_{j=0}^{m_1} w_j \varphi_j(x) = w^T \varphi(x)$$

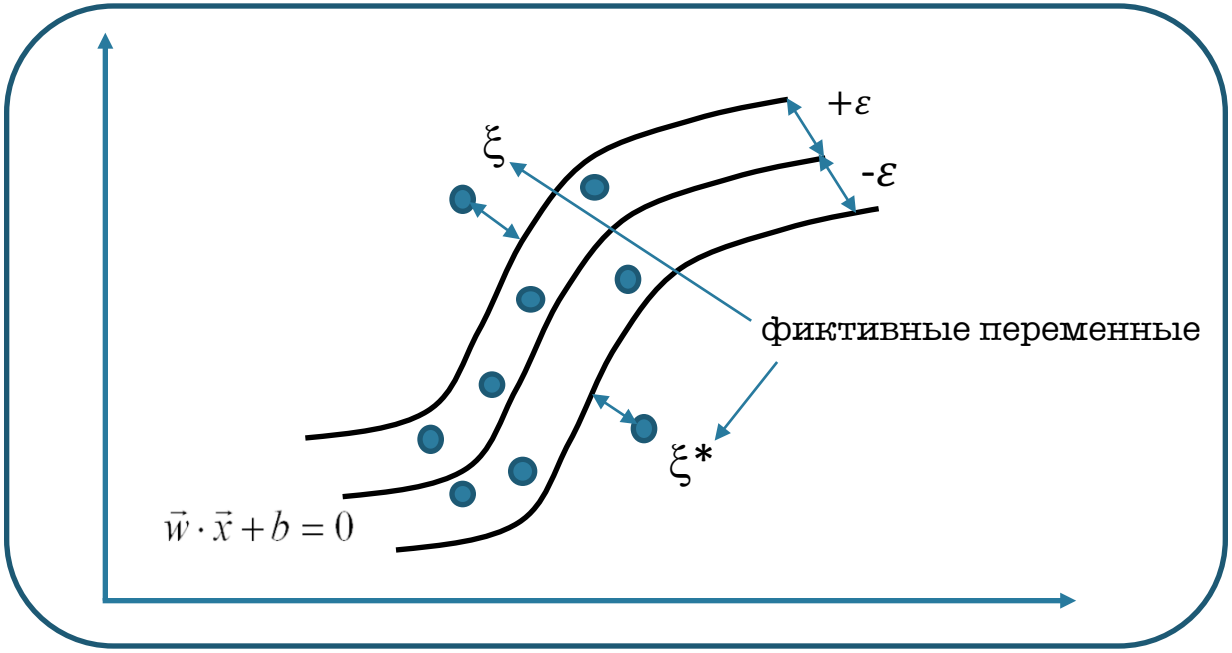
$$R_{emp} = \frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(d_i, y_i) \qquad \text{при условии } w^2 < c_0$$

Введение ε-нечувствительной функции потерь: $L(d, y) = |d - y|$

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon \text{ для } |d - y| \geq \varepsilon \\ 0 \text{ в остальных случаях} \end{cases}$$



# Регрессия метода опорных векторов




$$\min\frac{1}{2}w^T w + C\left(\sum_{i=1}^{N}(\xi_i + \xi_i^*)\right)$$

$$y_i - (\vec{w}\cdot\vec{x}+b) \le \varepsilon + \xi_i$$

$$y_i - (\vec{w}\cdot\vec{x}+b) \ge -\varepsilon - \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$

# Случайные процессы

Случайный процесс - индексированное множество случайных величин

$$\xi(\omega) = \{\xi_t(\omega) | t \in T\}$$

Если $T \subset \mathbb{R}$, а переменная t ассоциировалась со временем, то случайный процесс удобно представлять как случайную величину, являющуюся функцией от времени

Если $T \in \mathbb{R}^d$, случайный процесс может быть представлен как случайная величина, меняющаяся в пространстве и обычно называется случайным полем

Случайный процесс называется стационарным, если все его вероятностные характеристики не зависят от времени (стационарный в узком смысле),

$$p_{t_1,\ldots,t_n}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = p_{t_1+\tau,\ldots,t_n+\tau}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$$

и называется стационарным в широком смысле, если математическое ожидание и дисперсия имеют постоянные значения

# Гауссовские процессы (Gaussian Processes) в задачах регрессии

Случайный процесс, все конечномерные распределения которого нормальные, вероятностная модель которого, моделирует распределение функций: распределение в каждой точке, распределение всей функции

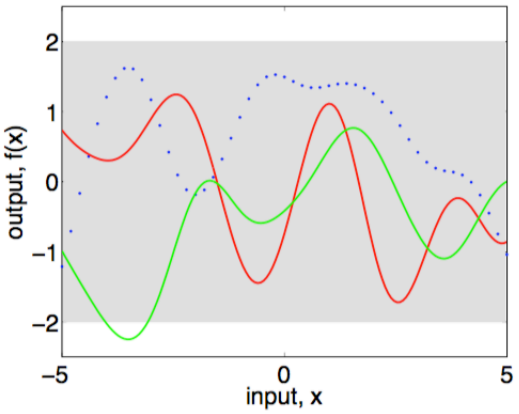
Задачи:

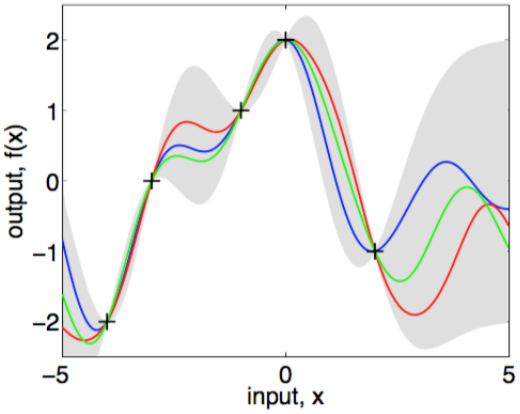
- ❖ Аппроксимация функции
- ❖ Определение доверительного интервала (степень точности предсказания значения функции в конкретной точке *x*)

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big)$$

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})],$$ - Математическое ожидание функции при заданных значениях в точках обучающей выборки

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$ - ковариационная функция




| covariance function | expression |
|---|---|
| constant | $\sigma_0^2$ |
| linear | $\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$ |
| polynomial | $(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$ |
| squared exponential | $\exp(-\frac{r^2}{2\ell^2})$ |
| Matérn | $\frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{\ell} r\right)$ |
| exponential | $\exp(-\frac{r}{\ell})$ |
| $\gamma$-exponential | $\exp\left(-(\frac{r}{\ell})^\gamma\right)$ |
| rational quadratic | $(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$ |
| neural network | $\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$ |

C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006

# Гауссовские процессы (Gaussian Processes) в задачах регрессии

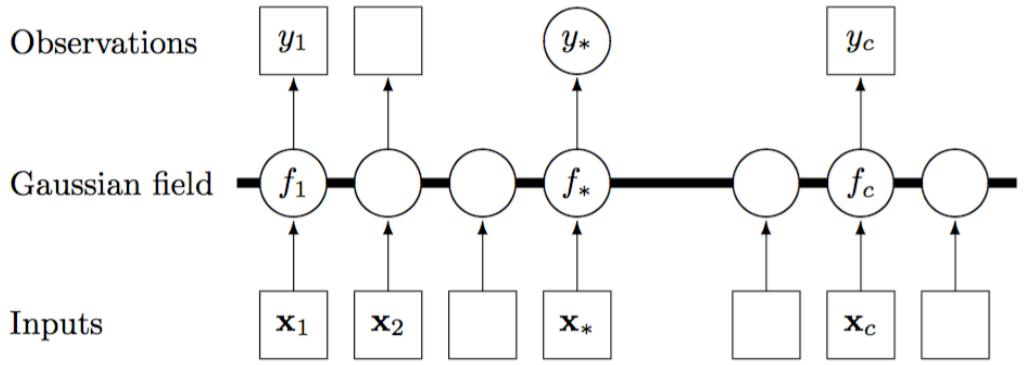
Итоговая аппроксимация – взвешенная сумма от ковариационных функций, где

$$\mu = \boldsymbol{k}^T C^{-1} \boldsymbol{t}$$

$$\sigma^2 = C(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^T C^{-1} \boldsymbol{k} = s^2 - \boldsymbol{k}^T C^{-1} \boldsymbol{k},$$

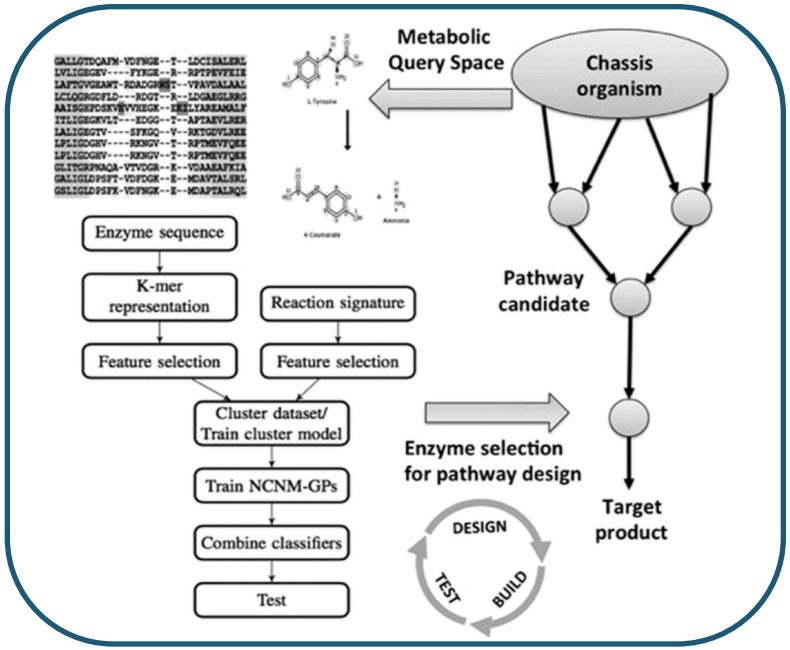
$S^2$- дисперсия случайного поля

Графическая вероятностная модель (Graphical Model) гауссовских процессов

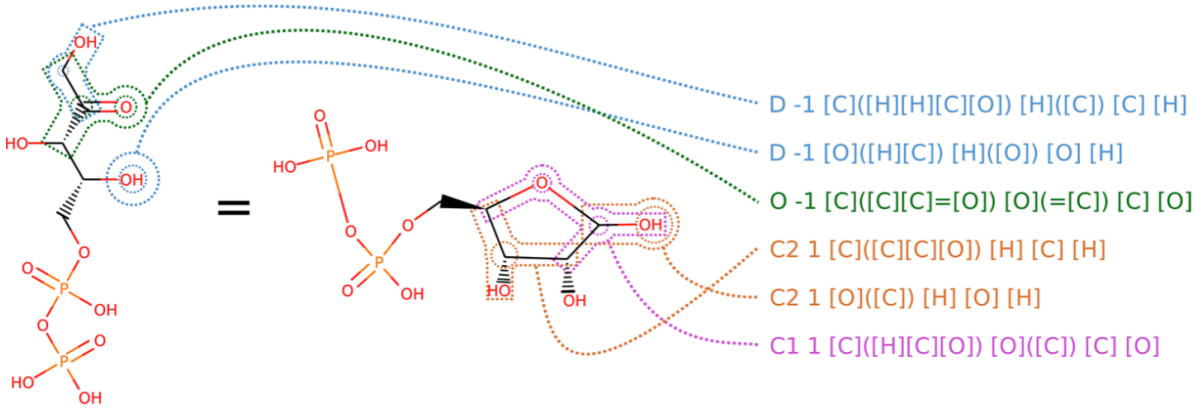



C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006

# Semi-supervised Gaussian Process for Automated Enzyme Search




Отпечатки, характеризующие реакционные центры




Mellor et al Semi-supervised Gaussian Process for Automated Enzyme Search ACS Synth. Biol. 2016, 5, 518–528

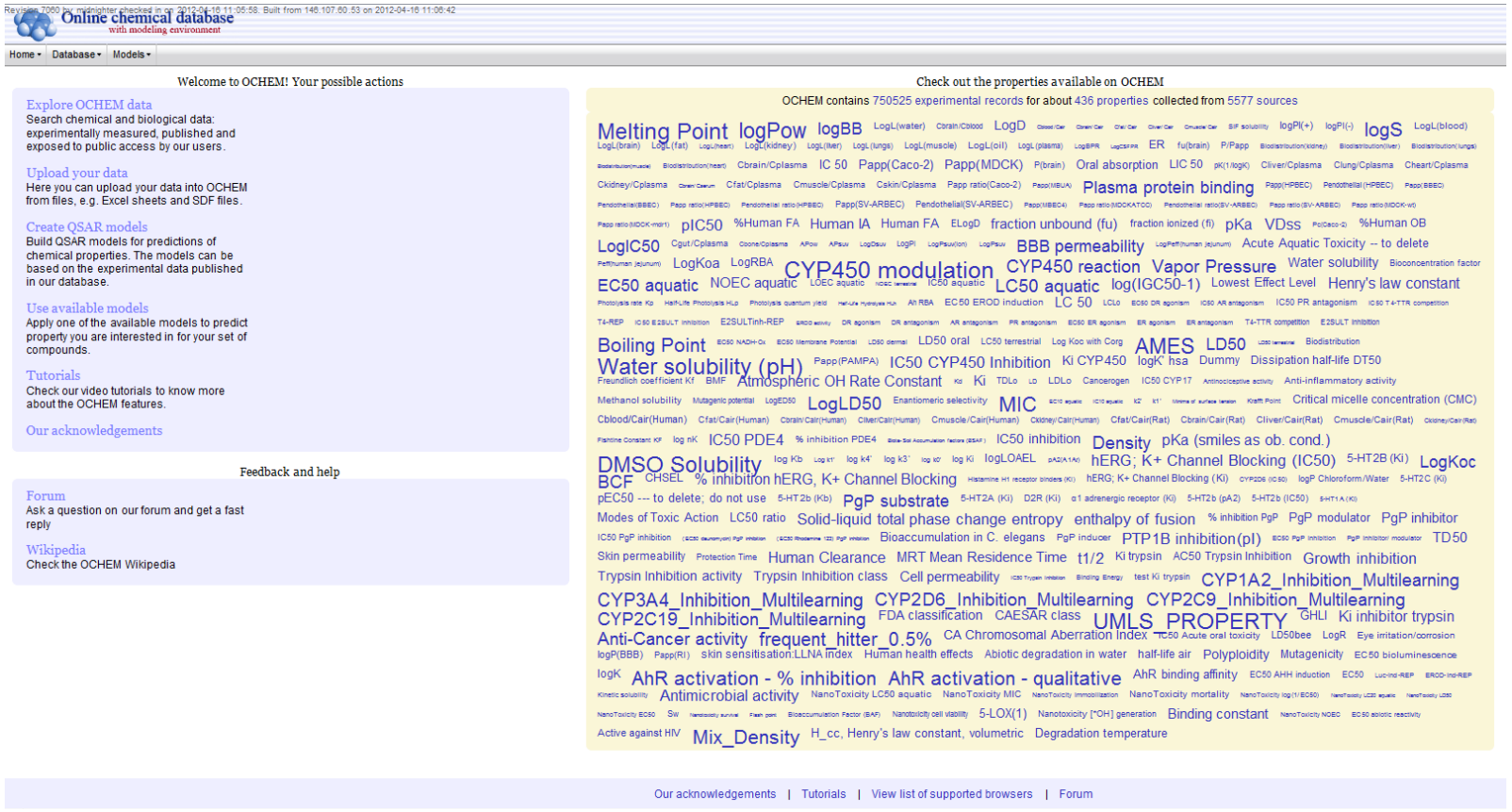
# Дополнительная рекомендуемая литература

**Quantitative Correlation of Physical and Chemical Properties with Chemical Structure: Utility for Prediction**
Alan R. Katritzky, Minati Kuanar, Svetoslav Slavov, C. Dennis Hall, Mati Karelson, Iiris Kahn, and Dimitar A. Dobchev
*Chemical Reviews* **2010** *110* (10), 5714-5789
DOI: 10.1021/cr900238d

**Machine Learning Methods for Property Prediction in Chemoinformatics: Quo Vadis?**
Alexandre Varnek and Igor Baskin
*Journal of Chemical Information and Modeling* **2012** *52* (6), 1413-1437
DOI: 10.1021/ci200409x

# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

# Онлайн-ресурсы




Online CHemical Modeling (OCHEM) - информационный и вычислительный ресурс, позволяющий работать через Web-интерфейс с базой данных по органическим соединениям и их свойствам, пополнять ее, осуществлять в ней поиск и формировать выборки, рассчитывать широкий набор молекулярных дескрипторов, строить количественные модели структура-свойство и применять их для прогнозирования свойств новых соединений.

http://ochem.eu/home/show.do

# Онлайн-ресурсы

CHEM BENCH

HOME | MY BENCH | DATASET | MODELING | PREDICTION | CECCR BASE

## ACCELERATING CHEMICAL GENOMICS RESEARCH BY CHEMINFORMATICS

Chembench is a free portal that enables researchers to mine available chemical and biological data. Chembench can help researchers rationally design or select new compounds or compound libraries with significantly enhanced hit rates in screening experiments.

It provides cheminformatics research support to molecular modelers, medicinal chemists and quantitative biologists by integrating robust model builders, property and activity predictors, virtual libraries of available chemicals with predicted biological and drug-like properties, and special tools for chemical library design. Chembench was initially developed to support researchers in the **Molecular Libraries Probe Production Centers Network (MLPCN)** and the Chemical Synthesis Centers.

Please cite this website using the following URL: **http://chembench.mml.unc.edu**

The Carolina Cheminformatics Workbench (Chembench) is developed by the Carolina Exploratory Center for Cheminformatics Research (CECCR) with the support of the **National Institutes of Health** (grants **P20HG003898** and **R01GM066940**) and the Environmental Protection Agency (RD83382501 and RD832720). This website has been developed using grants from the EPA and NIH. Therefore Chembench adheres to their required terms of use.

### Please login

Username:

Password: login

Or, **login as a guest**

Forget your password? **click here**

### New Users

Please **register here**

### Help & Links

**Chembench Overview**
**Chembench Workflows & Methodology**
**Links to More Cheminformatics Tools**

### Statistics

Visitors: 344967
Users: 567
Jobs completed: 21188
Compute time used: 23.799 years
Current Users: 6
Running Jobs: 99

http://chembench.mml.unc.edu/

# Программное обеспечение, основанное на принципе поточной обработки данных

- KNIME

https://www.knime.org/

- Orange

http://orange.biolab.si/

- Tanagra

http://eric.univ-lyon2.fr/~ricco/tanagra/

- Taverna

http://www.taverna.org.uk/

- ELKI

http://elki.dbs.ifi.lmu.de/

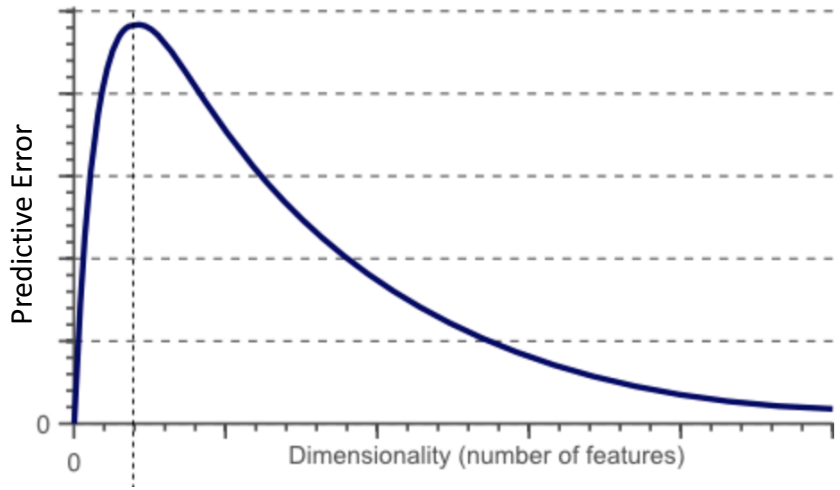
# МЕТОДЫ ОТБОРА ДЕСКРИПТОРОВ (ПЕРЕМЕННЫХ)

# Проклятие размерности (Curse of dimensionality)

Необходимое число примеров (для достижения той же точности) растет экспоненциально с числом переменных

На практике: число обучающих примеров фиксировано

Возможные проблемы:
Трудоемкость вычисление
Увеличение количества шумов
Точность метода может уменьшаться для большого количества дескрипторов

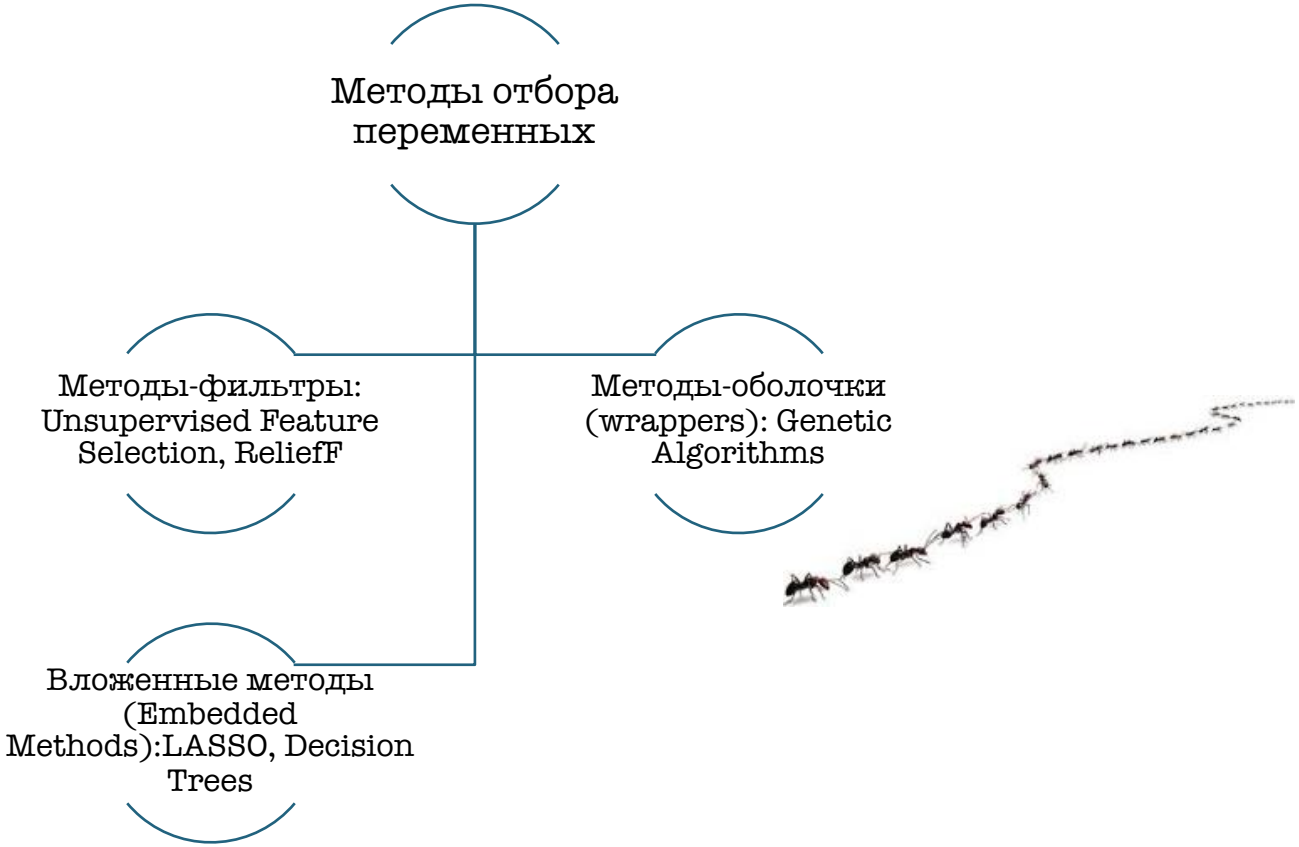



Возможное решение: отбор переменных

# Отбор переменных (Feature Subset Selection)

Необходимо определить:

- Критерий оценки качества набора дескрипторов (scoring function)
- Стратегия поиска поднабора дескрипторов



# Отбор переменных (Feature Subset Selection): Фильтры

Характерные особенности:

- Обычно используются в качестве шага предварительной обработки
- Отличаются высоким быстродействием
- Пытаются *a-priori* выявить дескрипторы, содержащие полезную информацию

Наиболее распространенные типы:

- Фильтры, основанные на корреляции
- Фильтры, основанные на теории информации (рассчитывают вариативность молекулярных дескрипторов). Представители: Shannon Entropy Filter

# Отбор переменных (Фильтры): независимый прямой отбор (Unsupervised Forward Selection)

1. Выбрать две переменные с наименьшим коэффициентом корреляции между ними
2. Отбросить переменные, для которых коэффициент корреляции с отобранными превышает заданное пороговое значение *rsqmax*
3. Выбрать следующую переменную с наименьшим коэффициентом корреляции с уже отобранными
4. Повторить шаг 2
5. Повторять шаги 3 - 4 пока все переменные не будут выбраны или отброшены

D. C. Whitley, M.G. Ford and D. J. Livingstone Unsupervised forward selection: a method for eliminating redundant variables. J. Chem. Inf. Comp. Sci., 2000, **40**, 1160-1168

# Отбор переменных (Feature Subset Selection): методы-оболочки

Характерные особенности:

- Прогностическая способность оценивается на одиночной тестовой выборке или процедурой перекрестного контроля
- Методы-оболочки универсальны и просты
- Недостаток: времязатратность

Представители:

- Процедуры прямого и обратного отбора переменных
- Генетические алгоритмы
- Метод муравьиных колоний
- ...

# Генетический алгоритм

***Генетический алгоритм (John Holland 1975*** "репродуктивный план Холланда"***)***
алгоритм оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов естественного отбора, напоминающих биологическую эволюцию:

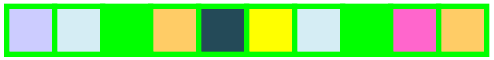
- Наследования
- Мутации
- Отбор
- Перекрёст

***Основан на 2 принципах:***

- “Выживает наиболее приспособленный”
- “Генетическая разнородность”

# Генетический алгоритм

- Генетический алгоритм стартует со случайного набора решений (переменные, характеризующие решение, представлены в виде генов в хромосоме, хромосомы формируют популяцию). Для хромосомы могут использоваться любые обозначения (числа, символы), но на практике чаще используются бинарные

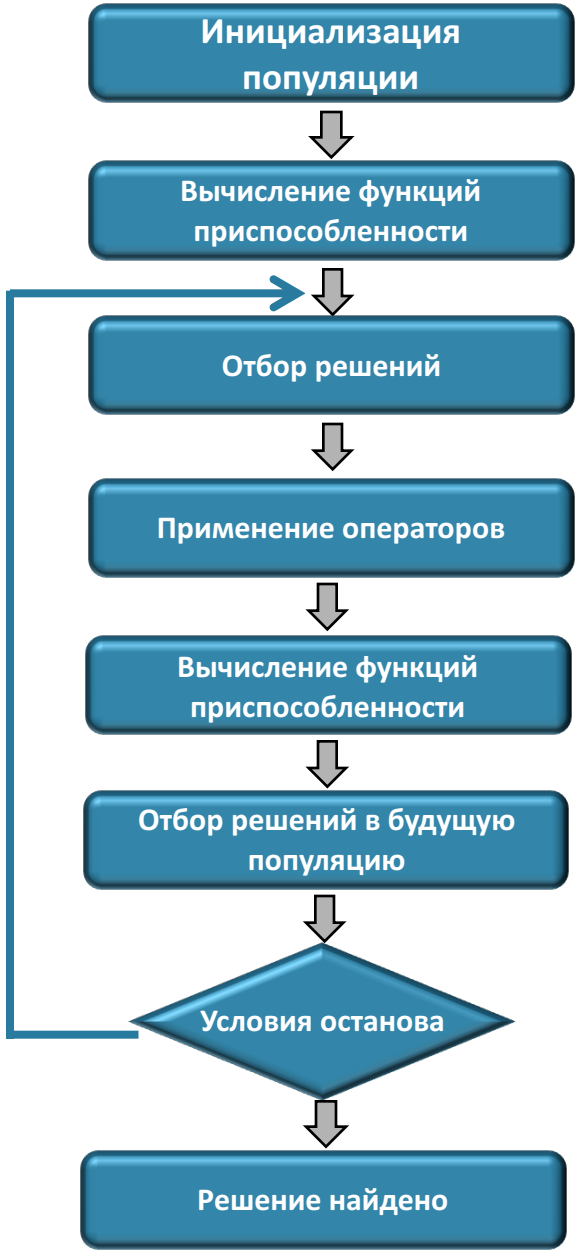


- Каждое решение характеризуется функцией приспособленности (fitness function): максимальное значение функции соответствует лучшему решению

- На основе значения этой функции, отбираются решения-«родители» для генерации следующего поколения, являющегося комбинацией двух «родительских» решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение.

- Критерием останова алгоритма могут быть:

  - нахождение глобального, либо локального решения;
  - исчерпание числа поколений, отпущенных на эволюцию;
  - исчерпание времени, отпущенного на эволюцию.

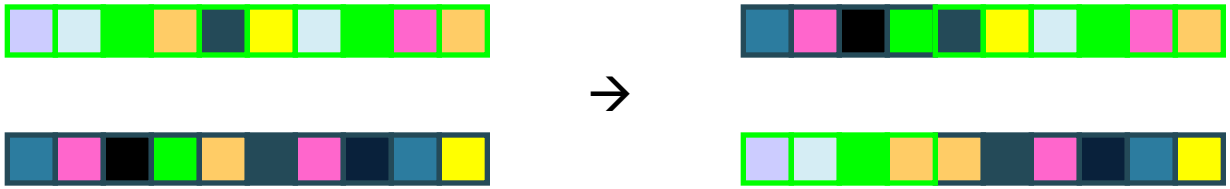


# Генетический алгоритм: операторы

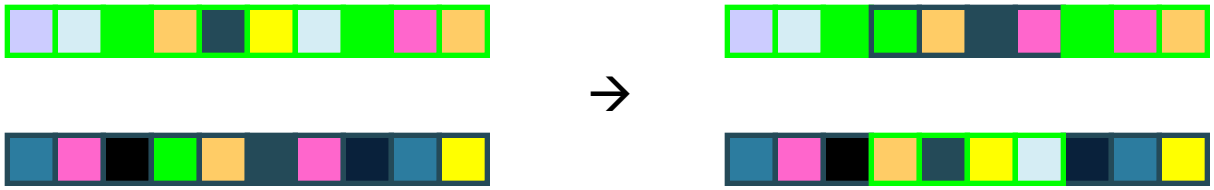
## Перекрёст

Два родителя формируют два новых решения

***Single point crossover***

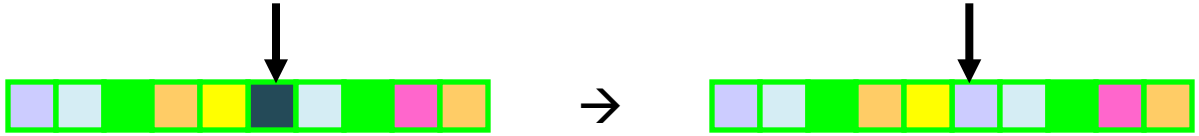


***Two points crossover***



## Мутация

Случайное изменение гена в хромосоме



# Генетический алгоритм: отбор решений

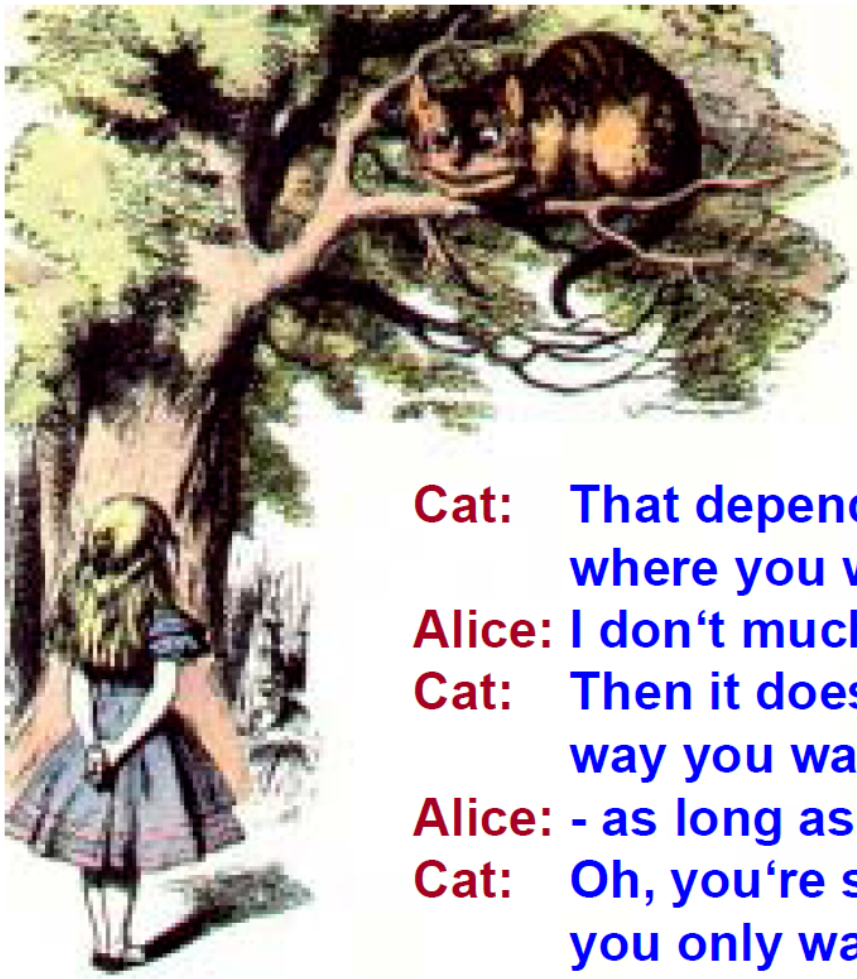
## Принцип рулетки:

площадь сегмента колеса рулетки, сопоставленного конкретной хромосоме, пропорциональна величине её относительной функции приспособленности. В данной стратегии сначала определяются приспособленности особей в популяции. Затем создается некое подобие круговой диаграммы, сектора которой раздаются особям, причем, чем больше приспособленность у особи, тем больше у нее сектор.

## Принцип турнира:

(tournament selection) реализует n турниров, чтобы выбрать n особей. Каждый турнир построен на выборке k элементов из популяции, и выбора лучшей особи среди них. Наиболее распространен турнирный отбор с k=2.

# Генетический алгоритм



Lewis Carroll

**Alice in Wonderland**

**Alice:** Would you tell me, please, which way I ought to walk from here?

**Cat:** That depends a good deal on where you want to get to.

**Alice:** I don‘t much care where - .

**Cat:** Then it doesn‘t matter which way you walk.

**Alice:** - as long as I get *somewhere.*

**Cat:** Oh, you‘re sure to do that, if you only walk long enough.

Слайд из лекций Хьюго Кубиньи

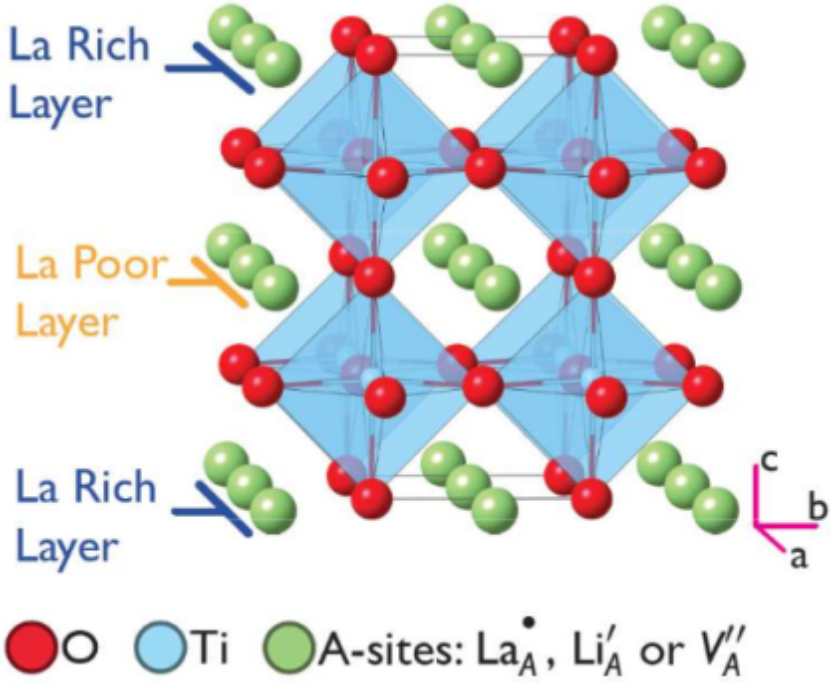
# Genetics of superionic conductivity in lithium lanthanum titanates

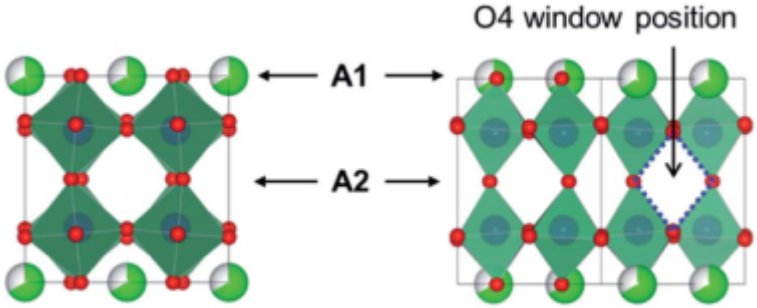
## Perovskites: $ABO_3$

A site ions alkaline-earth or rare-earth elements (12-fold coordinated)

B site cations (6-fold coordinated)

**$La_{2/3-x}Li_{3x}TiO_3$ (LLTO)**



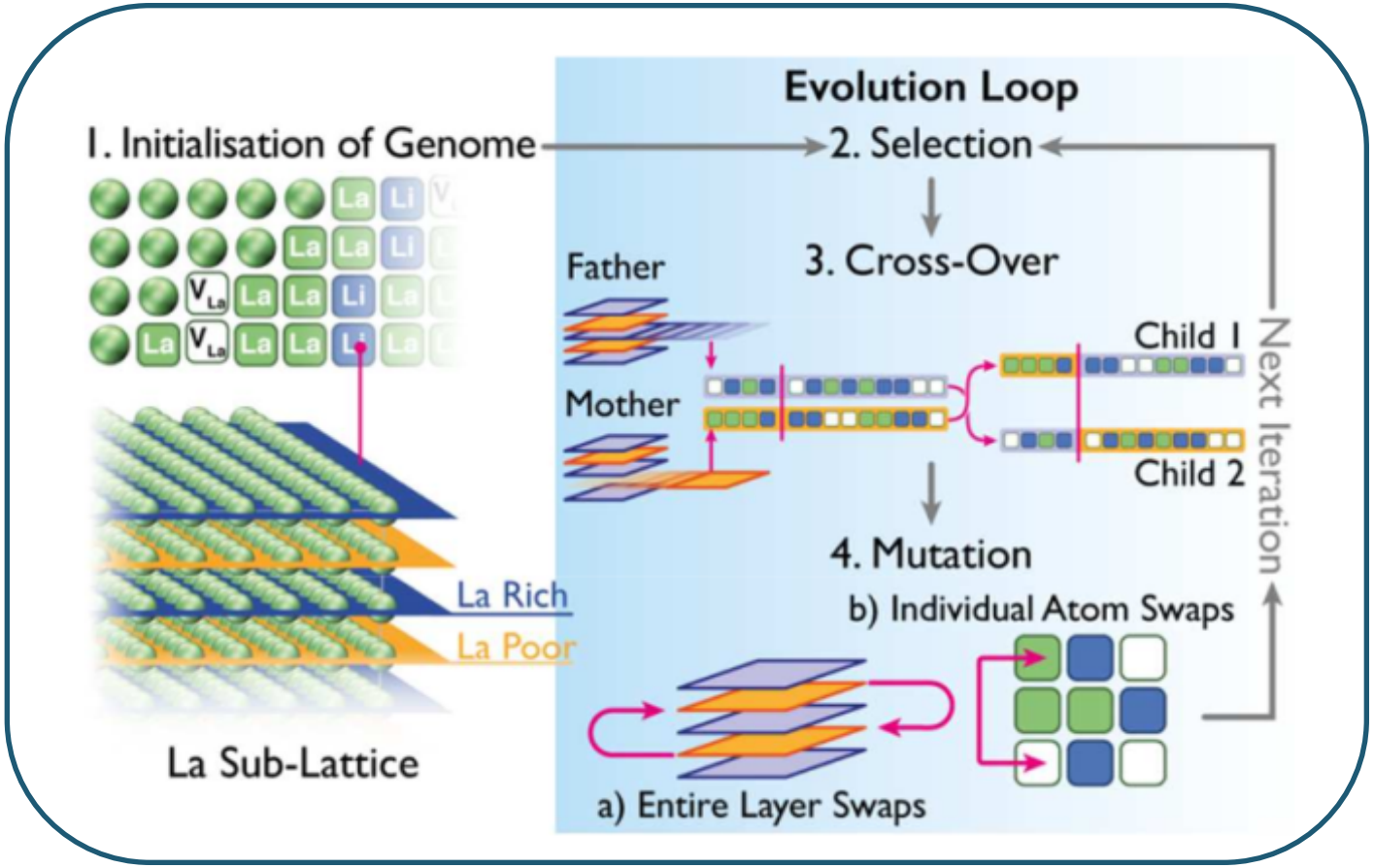




A1 – La-rich layer
A2 – La-poor layer

$$2[V_A''] + [Li_A'] = [La_A^{\bullet}]$$

Jay, E. E., M. J. D. Rushton, et al. (2015). "Genetics of superionic conductivity in lithium lanthanum titanates." Physical Chemistry Chemical Physics **17**(1): 178-183.

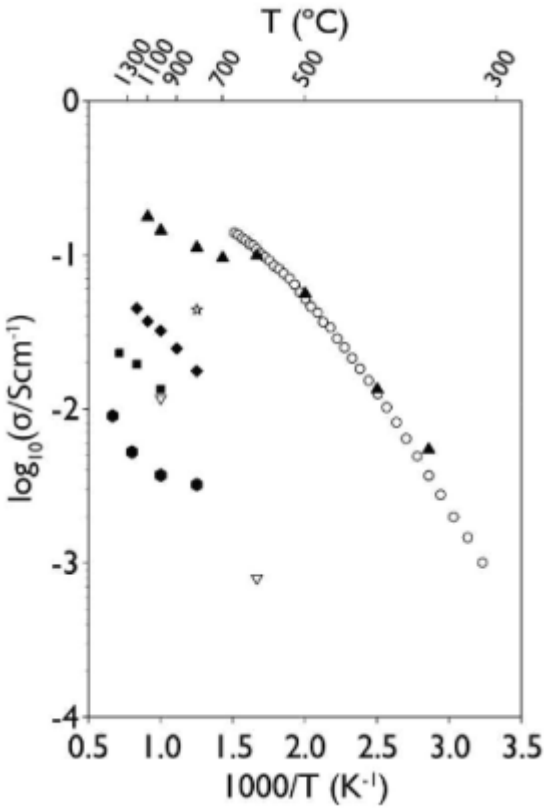
# Genetics of superionic conductivity in lithium lanthanum titanates



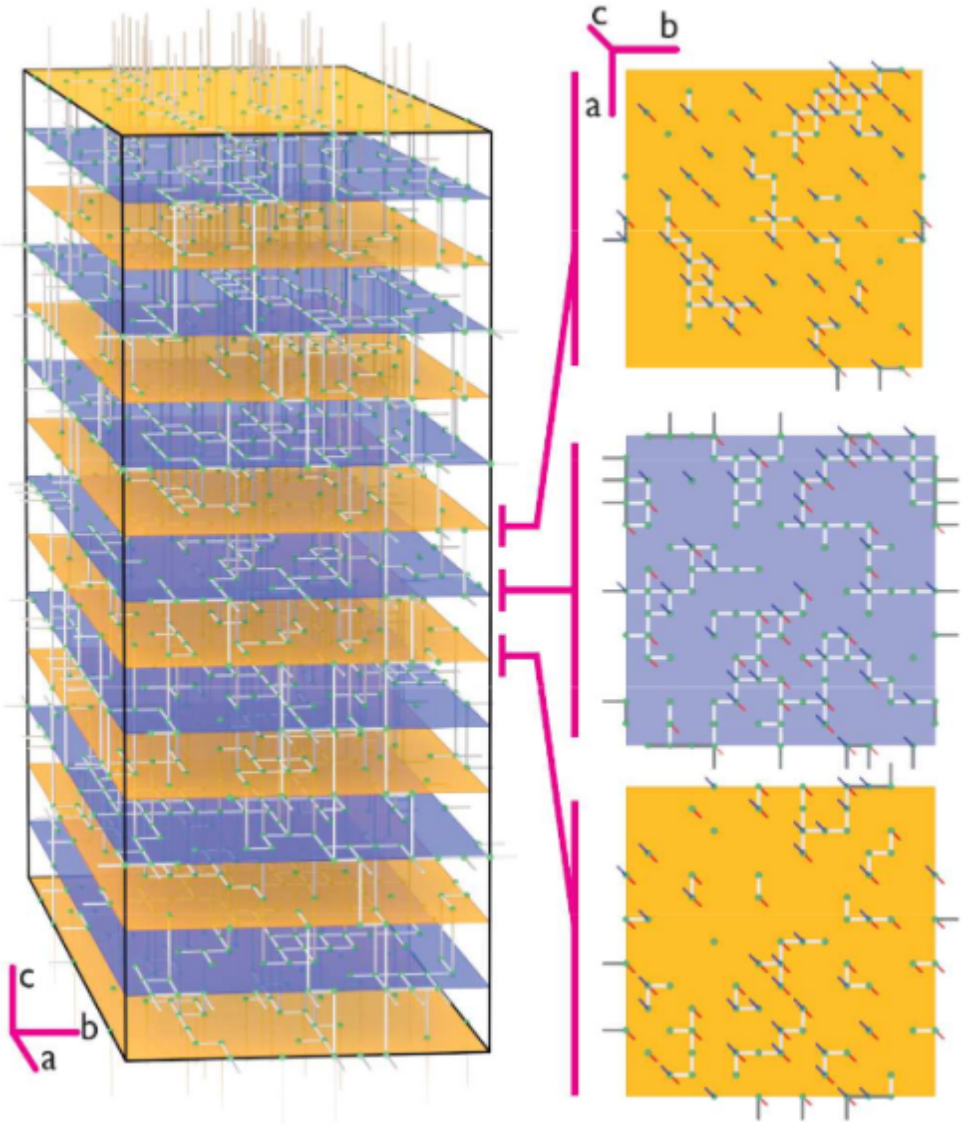

- Подслой лантана: 14 слоёв, 196 сайтов в каждом (образован $La$, $Li_A'$ и $V_A^{''}$ )
- Популяция включает 100 случайным образом сгенерированных конфигураций
- Цель: найти конфигурацию, соответствующую величине проводимости по литию, в качестве критерия для отбора решений используется среднее квадратичное смещение (Mean Squared Displacement)

$$\mathrm{MSD} \equiv \langle (x - x_0)^2 \rangle = \frac{1}{N} \sum_{n=1}^{N} (x_n(t) - x_n(0))^2$$

# Genetics of superionic conductivity in lithium lanthanum titanates




A comparison of Li conductivity produced in this work against other simulated and experimental literature values. The simulation values are for LLTO with $S$=0.2, values are given for the original potential model with random layering (⬢), original potentials with GA optimised structures and random layering (■), original potentials with GA optimised structure and rich-poor layering (◆) and finally GA optimization, rich-poor layering and the potentials derived for this work (▲). Experimental values taken from literature are as follows: Šalkus *et al.*[7] (○), Katsumata *et al.*[32] for $x$ = 0.066 (▽), Hirakuri *et al.*[33] (☆), for $x$ = 0.066.



# Отбор переменных (Feature Subset Selection): вложенные методы

Характерные особенности:

- Совмещены с конкретной обучающей машиной
- Не требуют разделения исходного набора данных на основную (learning set) и вспомогательную (tuning set) выборки
- Отбор переменных осуществляется непосредственно в процессе обучения и не может быть отделен
- Способны получить решение быстрее, чем методы-оболочки за счет отсутствия перебора многочисленных комбинаций параметров

Представители:

- Деревья решений
- Рекурсивное исключение переменных (Recursive Feature Elimination)
- LASSO

# Вопросы?