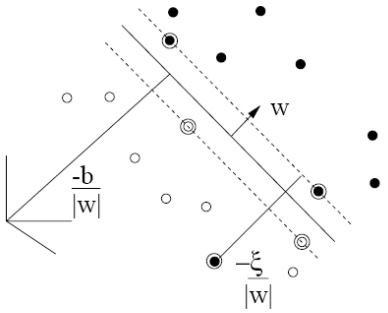
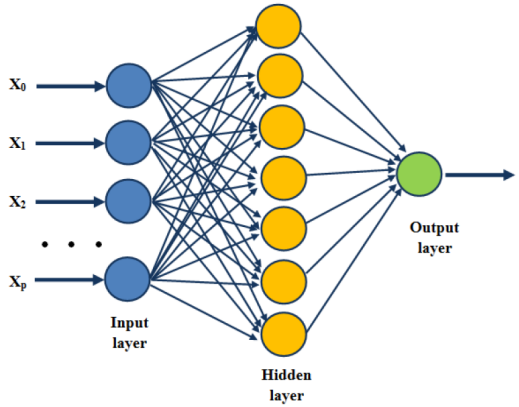
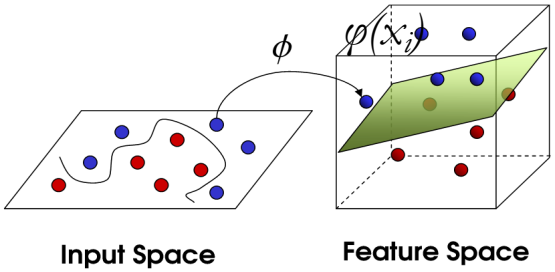
# Введение в химическую информатику

Лекции 11-12




$x_i$



# ХИМИЧЕСКАЯ ИНФОРМАТИКА: РАЗРАБОТКА МОДЕЛЕЙ СТРУКТУРА-СВОЙСТВО

**QSAR Modeling: Where Have You Been? Where Are You Going To?**
Artem Cherkasov, Eugene N. Muratov, Denis Fourches, Alexandre Varnek, Igor I. Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C. Martin, Roberto Todeschini, Viviana Consonni, Victor E. Kuz'min, Richard Cramer, Romualdo Benigni, Chihae Yang, James Rathman, Lothar Terfloth, Johann Gasteiger, Ann Richard, and Alexander Tropsha
*Journal of Medicinal Chemistry* **2014** *57* (12), 4977-5010
DOI: 10.1021/jm4004285

---

**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

---

**Trust, But Verify: On the Importance of Chemical Structure Curation in Cheminformatics and QSAR Modeling Research**
Denis Fourches, Eugene Muratov, and Alexander Tropsha
*Journal of Chemical Information and Modeling* **2010** *50* (7), 1189-1204
DOI: 10.1021/ci100176x

---

# ВОЗМОЖНЫЕ СЛОЖНОСТИ ПРИ МОДЕЛИРОВАНИИ

**Возможная разнородность данных**

**Выбор корректного дескрипторного описания данных (коллинеарность дескрипторов, ошибки в значениях, некорректный выбор дескрипторов)**

**Воспроизводимость моделей**

**Определение области применимости моделей**

**Возможные ошибки (необоснованный отброс данных, наличие дубликатов соединений, переобучение, отсутствие интерпретации моделей,…)**

**Учет диапазона изменения значения свойства**

**Корректная процедура валидации моделей**

**…**

Cherkasov et al **QSAR Modeling: Where Have You Been? Where Are You Going To?**
*Journal of Medicinal Chemistry* **2014** *57* (12), 4977-5010

# ЭТАПЫ РАЗРАБОТКИ МОДЕЛЕЙ

## Подготовка данных

- Сбор  и проверка экспериментальных данных (одинаковые экспериментальные условия или введение дополнительных параметров, позволяющих нивелировать различия, удаление нежелательных соединений, нормализация специфических хемотипов, таутомеров)
- Выбор дескрипторов, получение исходного набора дескрипторов  и их нормализация (при необходимости)

## Разработка моделей

- Выбор метода машинного обучения/отбор переменных
- Проверка прогнозирующей способности моделей (разделение исходного набора данных на обучающую и тестовую выборки или кросс-валидация)

## Применение моделей

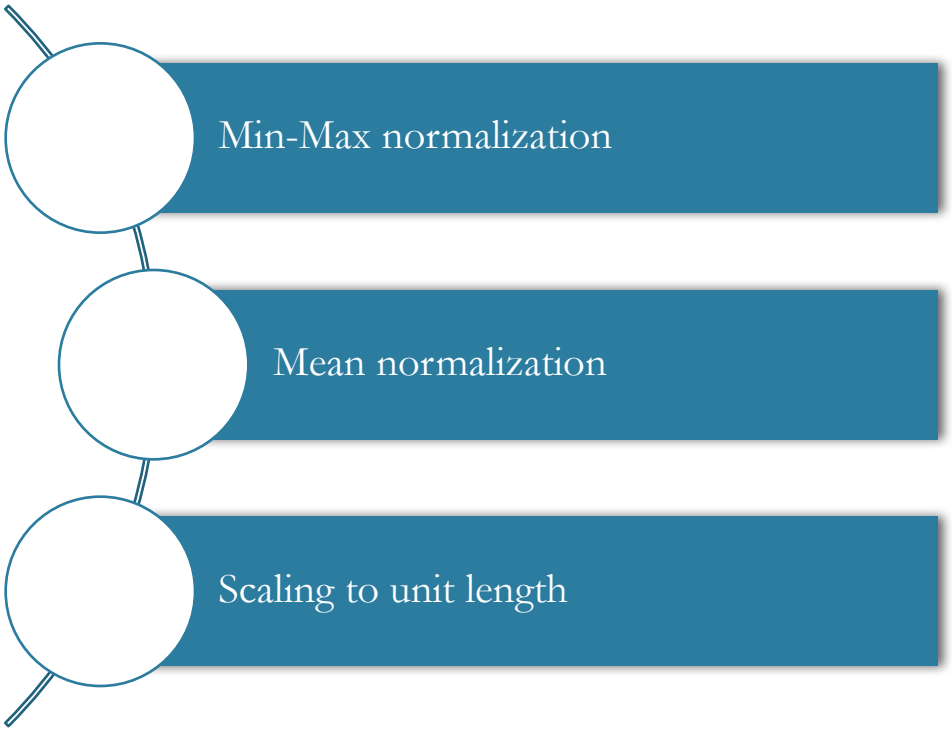
- Определение области применимости моделей
- Использование методов Data Domain Adaptation

# ПОДГОТОВКА ДАННЫХ: ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

| procedures | software |
|---|---|
| inorganics removal | ChemAxon/Standardizer<br>OpenEye/Filter |
| structure normalization (fragment removal, structural curation, salt neutralization) | ChemAxon/Standardizer<br>OpenBabel<br>Molecular Networks/CHECK,TAUTOMER |
| duplicate removal | ISIDA/Duplicates<br>HiT QSAR<br>CCG/MOE |
| SDF management/viewer file format converter | ISIDA/EdiSDF<br>Hyleos/ChemFileBrowser<br>OpenBabel<br>ChemAxon/MarwinView<br>CambridgeSoft/ChemOffice<br>Schrödinger/Canvas<br>ACD/ChemFolder<br>Symyx/Cheminformatics<br>CCG/MOE<br>Accelrys/Accord<br>Tripos/Benchware Pantheon |

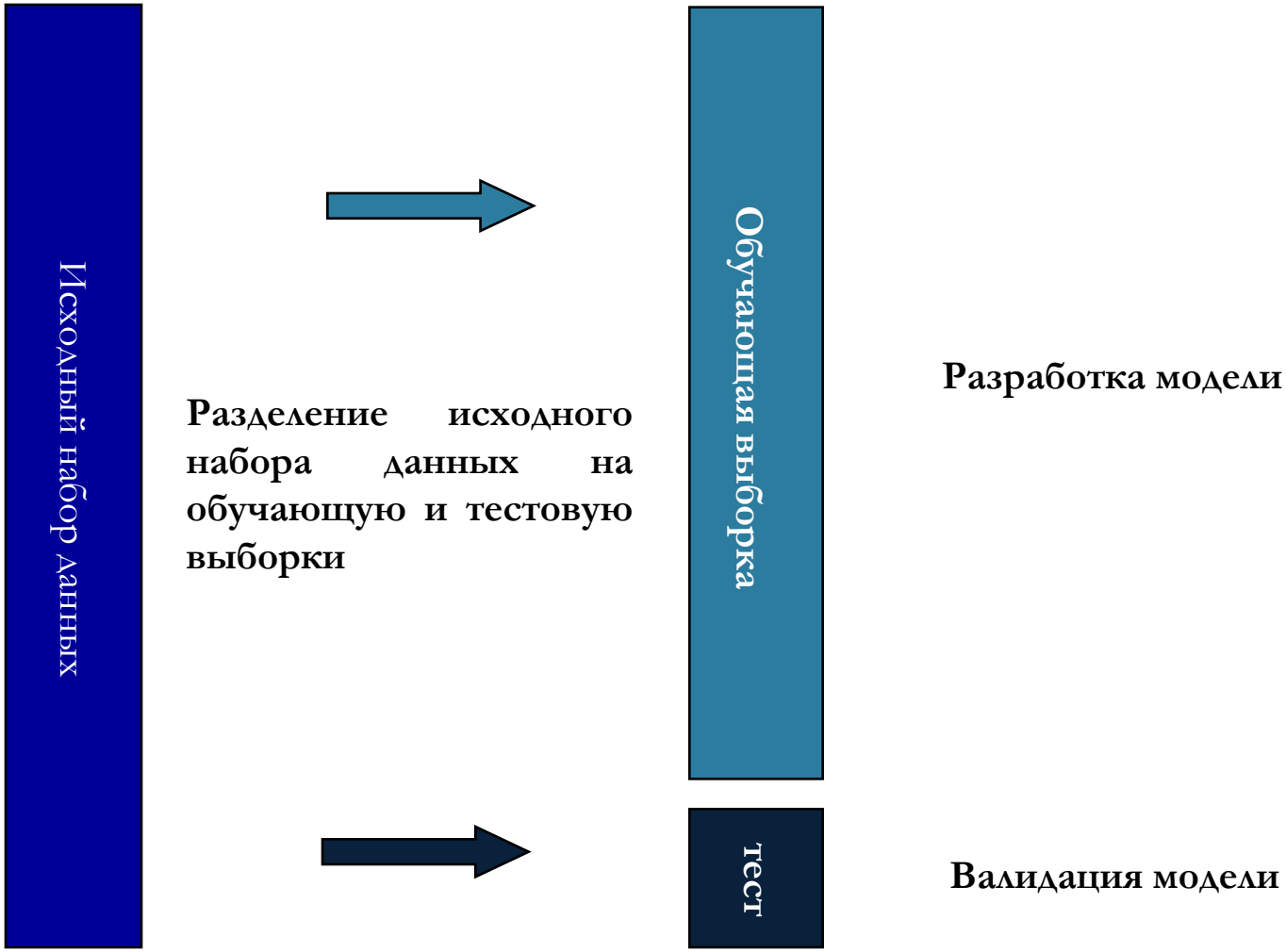
Fourches et al Trust, But Verify: On the Importance of Chemical Structure Curation in Cheminformatics and QSAR Modeling Research *JCIM* 2010 *50* (7), 1189-1204

# НОРМАЛИЗАЦИЯ ДЕСКРИПТОРОВ



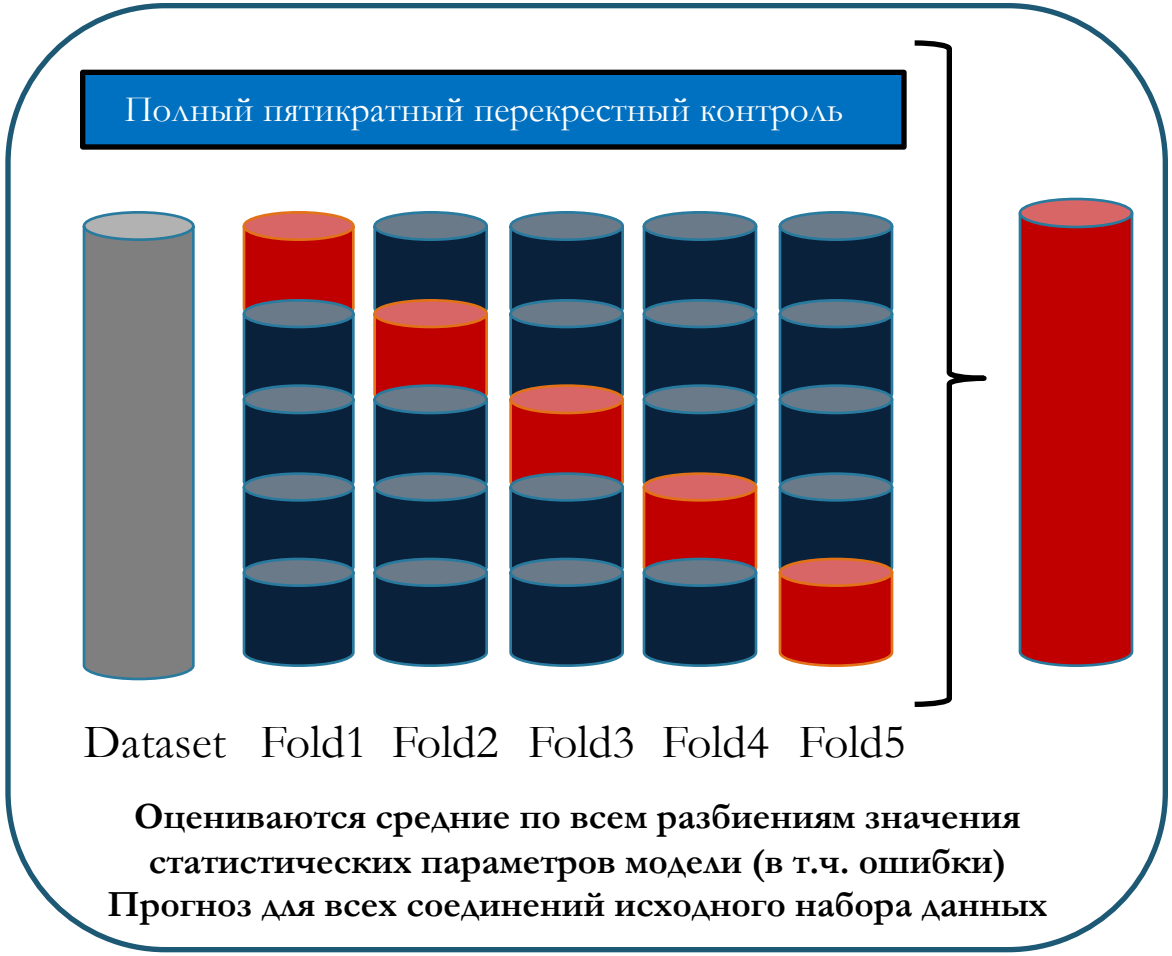
# МЕТОДЫ ВАЛИДАЦИИ МОДЕЛЕЙ

# ПОДГОТОВКА ОБУЧАЮЩЕЙ И ТЕСТОВОЙ ВЫБОРОК



# ПРОЦЕДУРА ПЕРЕКРЕСТНОГО КОНТРОЛЯ

**Процедура оценки обобщающей способности алгоритмов (прогностической способности модели)**

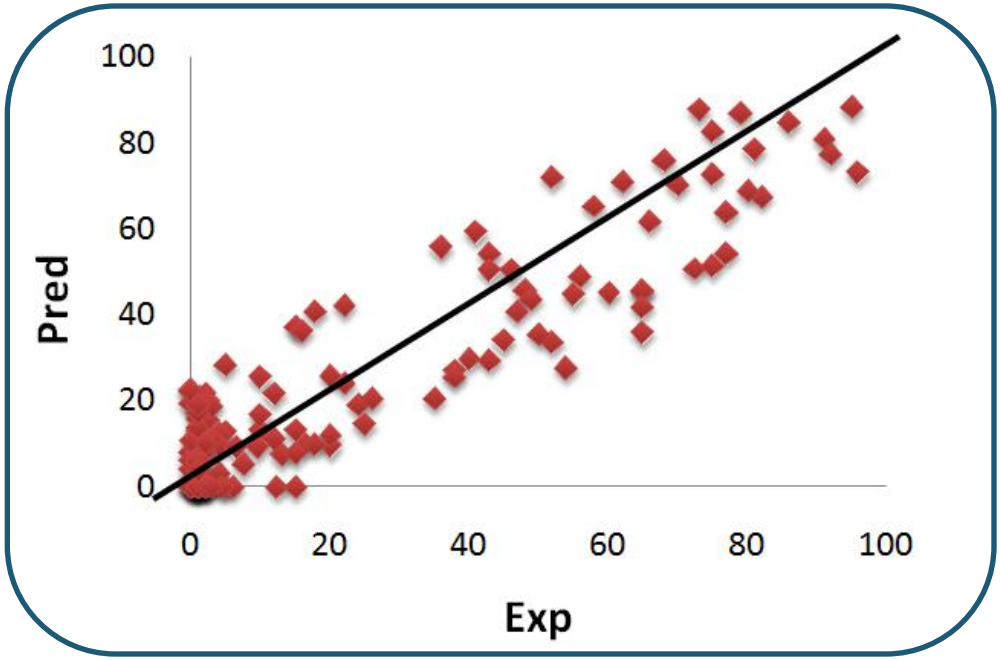




**Оцениваются средние по всем разбиениям значения статистических параметров модели (в т.ч. ошибки)**
**Прогноз для всех соединений исходного набора данных**

**Виды перекрестного контроля:**

- **Полный скользящий контроль**
- **Случайные разбиения**
- **Контроль по отдельным объектам**
- **Контроль по блокам**

# СТАТИСТИЧЕСКИЕ ПАРАМЕТРЫ ОЦЕНКИ ПРОГНОСТИЧЕСКОЙ СПОСОБНОСТИ РЕГРЕССИОННЫХ МОДЕЛЕЙ

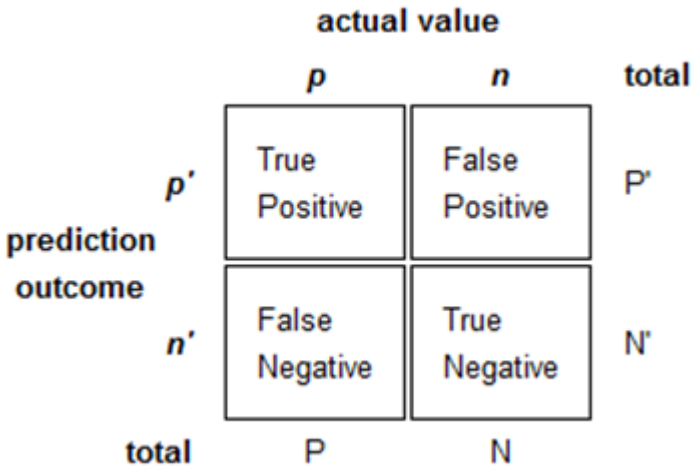



$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_{pred,i} - y_{exp,i}\right)^2}{\sum_{i=1}^{n}\left(y_{exp,i} - \overline{y}_{exp,i}\right)^2}$$

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

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

# СТАТИСТИЧЕСКИЕ ПАРАМЕТРЫ КЛАССИФИКАЦИИ: ТАБЛИЦА СОПРЯЖЕННОСТИ (CONFUSION MATRIX)




- TP (True Positives) – верно классифицированные положительные примеры (так называемые истинно положительные случаи);
- TN (True Negatives) – верно классифицированные отрицательные примеры (истинно отрицательные случаи);
- FN (False Negatives) – положительные примеры, классифицированные как отрицательные (ошибка I рода) - «ложный пропуск»
- FP (False Positives) – отрицательные примеры, классифицированные как положительные (ошибка II рода) – «ложное обнаружение»

***Чувствительность*** (Sensitivity) = true positive rate (TPR) = hit rate = ***recall***

$TPR = TP / P = TP / (TP + FN)$

false positive rate (FPR)

$FPR = FP / N = FP / (FP + TN)$

***Специфичность*** (Specificity) = True Negative Rate

$SPC = TN / N = TN / (FP + TN) = 1 - FPR$

positive predictive value (PPV) = ***precision***

$PPV = TP / (TP + FP)$

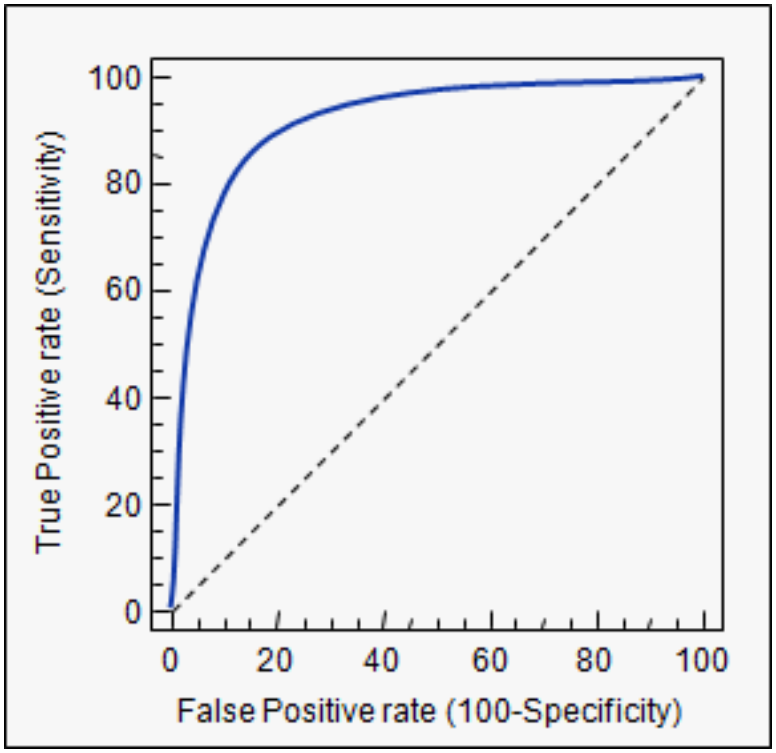
***Точность*** (Accuracy)

$ACC = (TP + TN) / (P + N)$

***Сбалансированная точность*** (Balanced Accuracy)

$BA = (\textbf{sensitivity} + \textbf{specificity}) / 2 = (TP / (TP + FN) + TN / (FP + TN)) / 2$

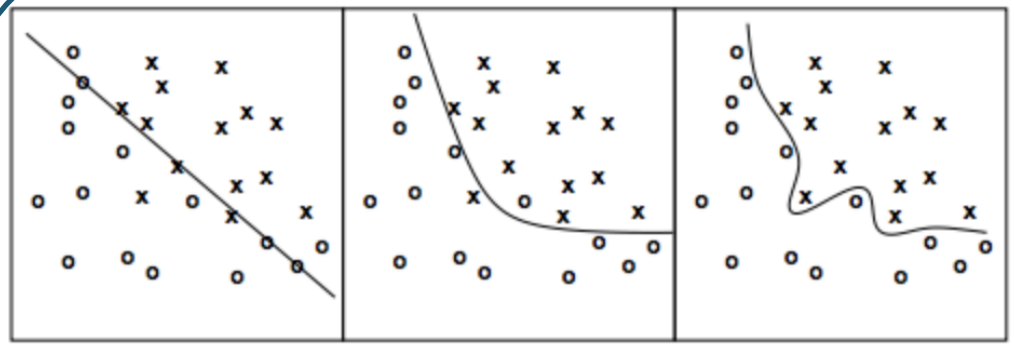
# ROC-кривая (Receiver Operator Characteristic)

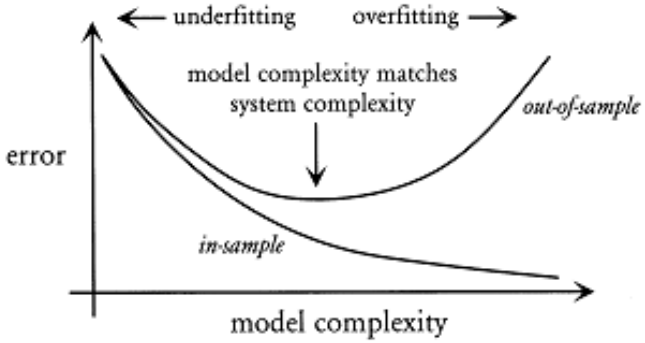



| Sensitivity | False Positive |
|---|---|
| 0.0 | 0.0 |
| s1 | fp1 |
| s2 | fp2 |
| … | … |
| 1.0 | 1.0 |

ROC-кривая отражает зависимость количества верно классифицированных положительных примеров от количества неверно классифицированных отрицательных примеров.

# ОБОБЩАЮЩАЯ СПОСОБНОСТЬ МОДЕЛЕЙ






**Переобученность (потеря обобщающей способности модели):**
Вероятность ошибки на обучающей выборке существенно ниже, чем на тестовой вследствие использования сложной модели, с подстройкой на конкретные примеры
**Модель хорошо воспроизводит объекты обучающего набора данных, но плохо работает на новых примерах**

**Недообучение:**
Недостаточно сложные для описания данных модели

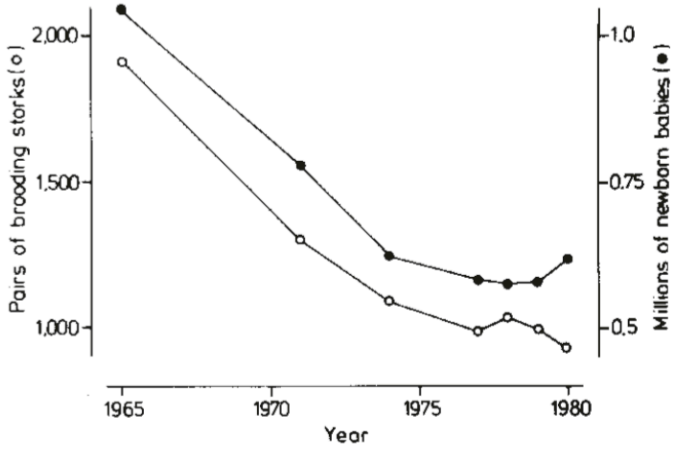
**Методы предотвращения:**

- Использование процедуры перекрестной валидации (Использование тестового сета для контроля качества модели (перекрестный контроль) и финальной оценки (внешний тестовый набор данных)
- Регуляризация (штраф за сложность модели)
- Ранняя остановка обучения
- Вербализация нейронных сетей (прореживание нейронных сетей)
- …

# Проблема “случайной корреляции”: Y-scrambling

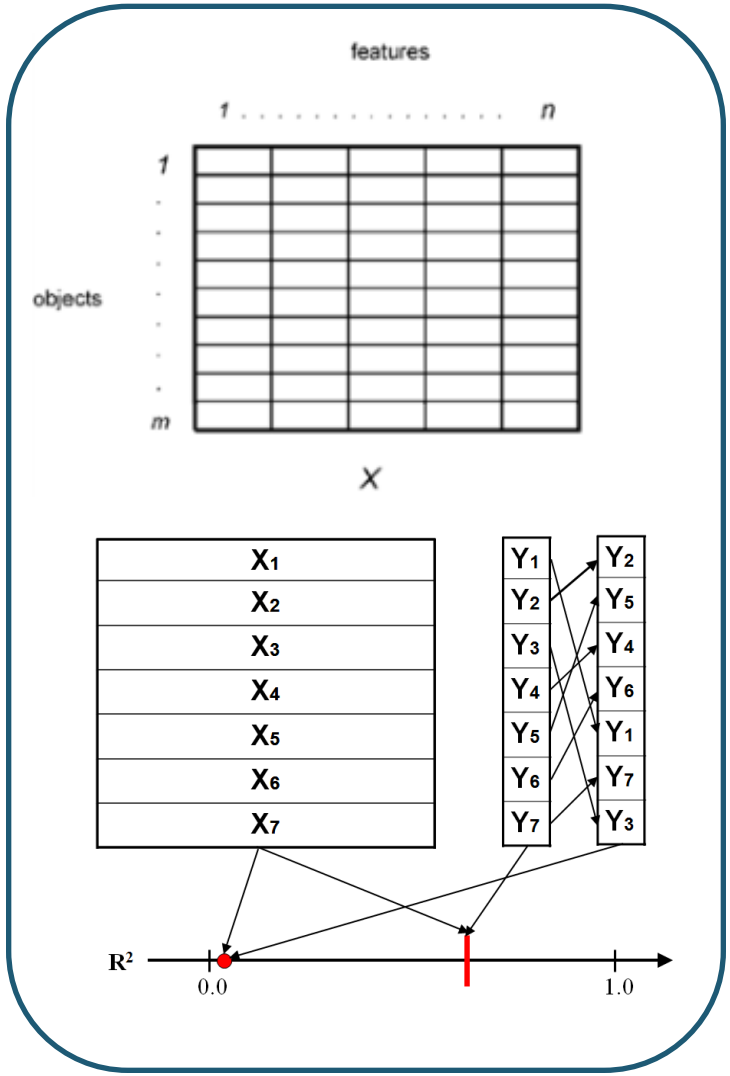
## A new parameter for sex education

SIR—There is concern in West Germany over the falling birth rate. The accompanying graph[1,2] might suggest a solution that every child knows makes sense.




HELMUT SIES
*Institut für Physiologische Chemie 1, Universität Düsseldorf, Moorenstrasse 5, D-4000 Düsseldorf, FRG*

1. *Fachserie Gebiet und Bevölkerung* (Statistisches Bundesamt, Kohlhammer, Stuttgart, 1984).
2. Bauer, S. & Thielcke, G. *Die Vogelwarte* **31**, 183–191 (1982).




Nature 332 7 (1988)

# Некоторые типы машинного обучения

**Обучение с учителем (Supervised Learning):**

В процессе обучения для примеров обучающей выборки известно значение свойства (принадлежность к классу).

**Обучение без учителя (Unsupervised learning):**

Информация о значении свойства (принадлежности к классу) не участвует в процессе разработки моделей.

**Обучение с частичным привлечением учителя (Semi-supervised learning):**

Обучение с учителем + дополнительные данные без информации о значении свойства

**Многозадачное обучение (Multi-Task Learning)**

одновременное обучение нескольким взаимосвязанным задачам

# КЛАССИФИКАЦИОННЫЕ МЕТОДЫ

Разработка моделей, которые на основе заданного векторного описания данных $x$ выдают прогноз в виде принадлежности к классу $\hat{t}$ или в виде вектора вероятностных оценок принадлежности к каждому из классов

# ЛИНЕЙНЫЕ КЛАССИФИКАТОРЫ: ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

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

**Линейная модель классификации:**

$$a(x, w) = sign\langle x, w\rangle$$

**Функция потерь:**

$$Q(w) = \sum_{i=1}^{l} [a(x_i, w)y_i < 0] \leq \sum_{i=1}^{l} \mathcal{L}(\langle x_i, w\rangle y_i)$$

**Логарифмическая функция потерь:**

$$\mathcal{L}(M) = log(1 + e^{-M})$$

**Отступ объекта:**

$$M_i(w) = \langle x_i, w\rangle y_i$$

Эквивалентность функции потерь принципу максимума правдоподобия вероятностной модели позволяет определить апостериорную принадлежность к классу каждого объекта

# ЛИНЕЙНЫЕ КЛАССИФИКАТОРЫ: ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

**Оптимизация параметров логистической регрессии:**

**Метод стохастического градиента**

$w^{(t+1)} := w^{(t)} + \eta_t y_i x_i (1 - \sigma_i)$

$\eta_t$ – градиентный шаг

$\sigma_i = \sigma\big((x_i, w) y_i\big) = P(y_i | x_i)$ - вероятность правильной классификации

**Метод Ньютона-Рафсона**

$w^{(t+1)} := w^{(t)} + \eta_t (F^T \Lambda F)^{-1} F^t \tilde{y}$

$F$- матрица объекты-признаки

$\tilde{y} = \big(y_i(1 - \sigma_i)\big)$

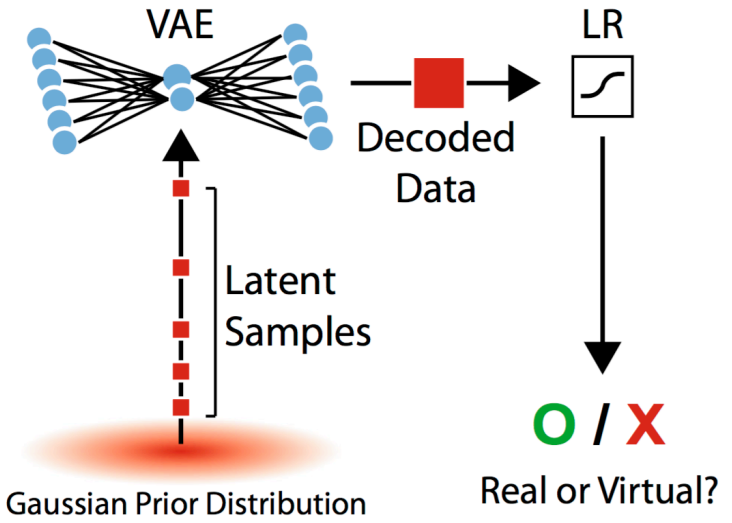
$\Lambda = diag\big((1 - \sigma)/\sigma_i\big)$

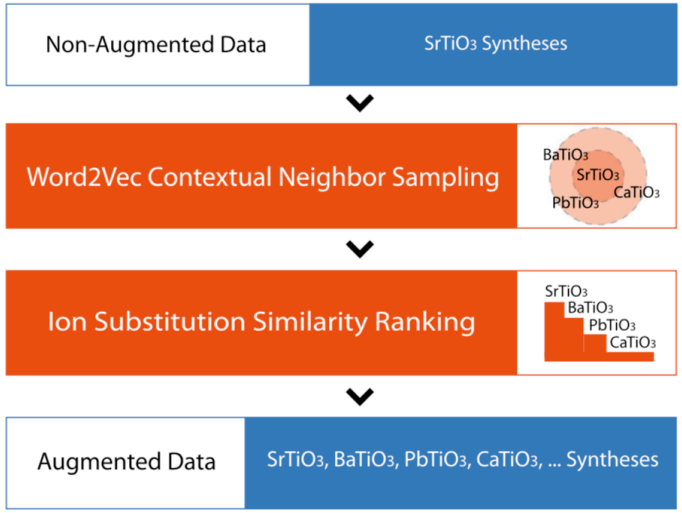
**Метод Левенберга-Маквардта**

# VIRTUAL SCREENING OF INORGANIC MATERIALS SYNTHESIS PARAMETERS WITH DEEP LEARNING

Descriptors:
processing (synthesis) information – sintering and calcination temperature and time, method of synthesis, solvent

Tasks:
- SrTiO3/BaTiO3 synthesis details discriminating
- MnO2 polymorph elucidation




Kim et al npj Computational Materials (2017) 53

# Метод опорных векторов

Владимир Вапник,
AT&T Research Laboratories

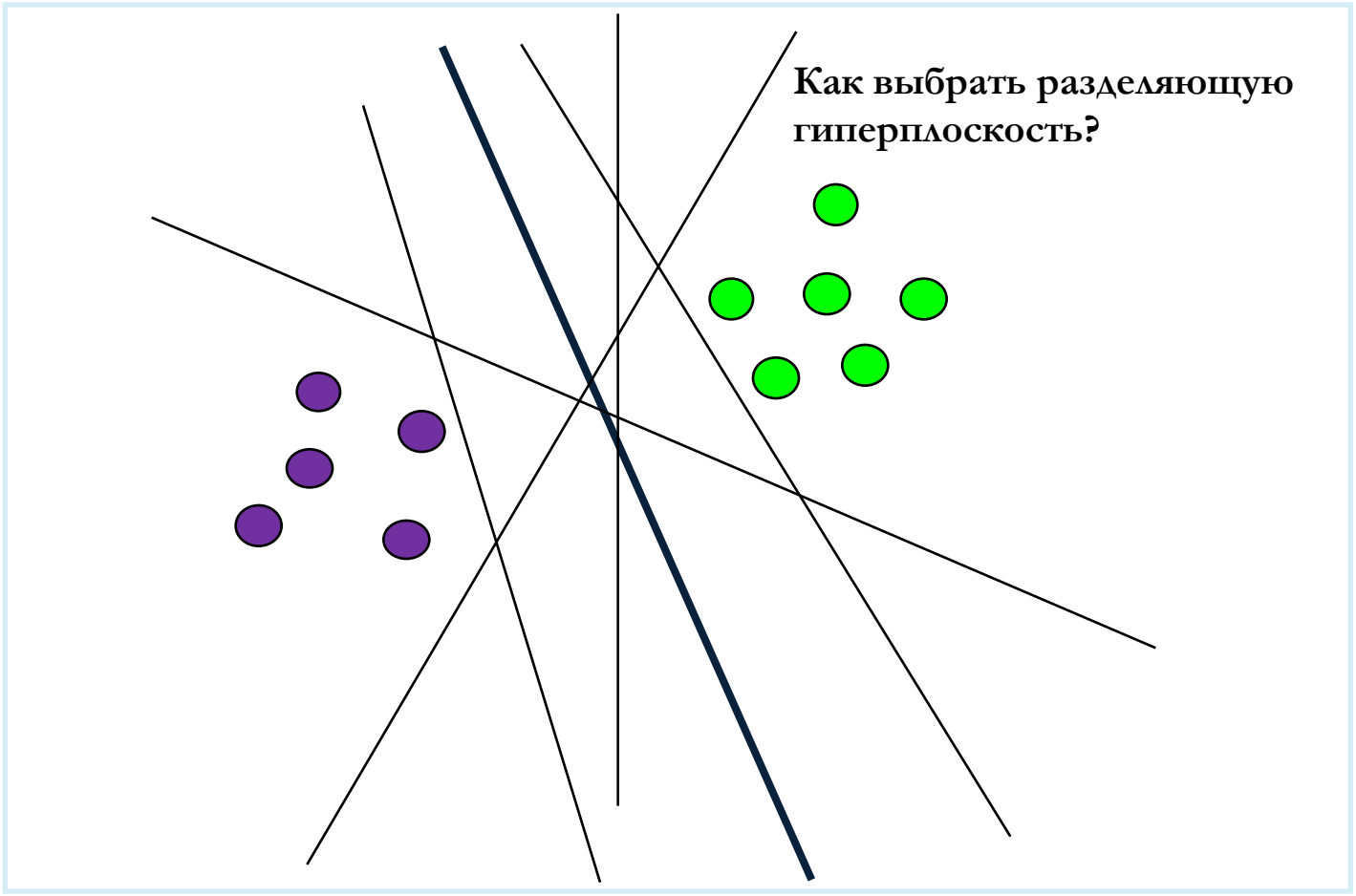
Первое упоминание об SVM в 1992 году - Vapnik et al
- Общая формулировка - 1995
- http://www.kernel-machines.org
- Vladimir Vapnik **Statistical Learning Theory, Wiley, NY, 1998**

N.Chen et al **Support Vector Machines in Chemistry, World Scientific, 2004**

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: КЛАССИФИКАЦИЯ

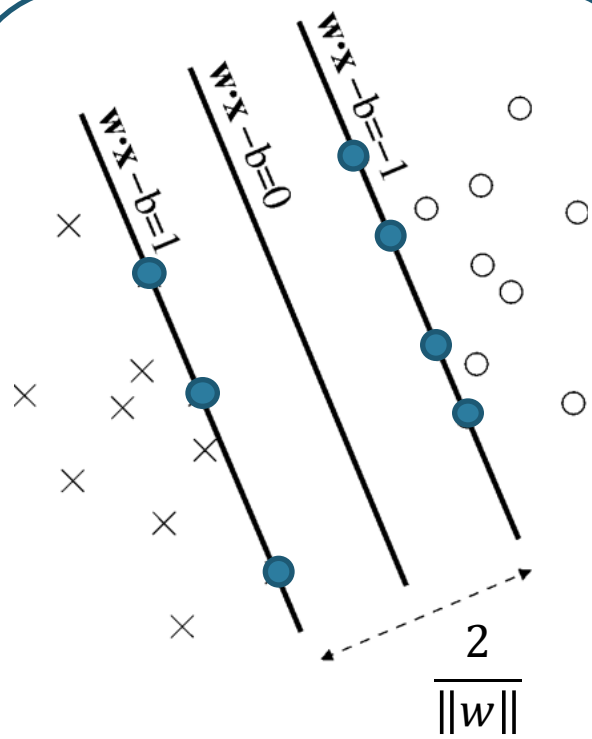
**Оптимальная гиперплоскость для линейно-разделимых объектов**

**Даны два класса. Каждый объект классификации является вектором (точкой) в n-мерном пространстве. Проведем линию, разделяющую эти два класса. Все новые точки автоматически классифицируются в соответствии с расположением относительно этой прямой**

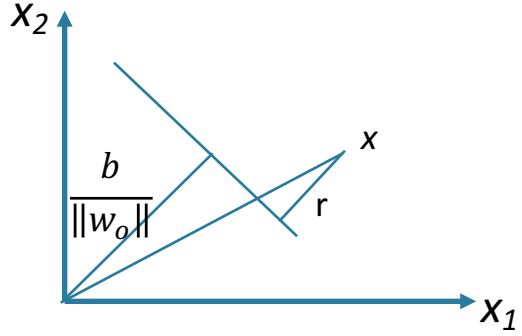



**Оптимальная гиперплоскость - расстояние от которой до каждого класса максимально**

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: КЛАССИФИКАЦИЯ (СЛУЧАЙ ЛИНЕЙНОЙ РАЗДЕЛИМОСТИ КЛАССОВ)




Зазор или граница разделения – минимальное расстояние до разделяющей гиперплоскости




$x^l = (x_i, y_i)_{i=1}^l \quad x_i \in \mathbb{R}^n \quad d_i \in \{-1, +1\}$

**Уравнение поверхности решений в виде гиперплоскости:**

$$\mathbf{w}^T\mathbf{x} + b = 0$$

$$\mathbf{w}^T\mathbf{x}_i + b \geq 0 \text{ для } d_i = +1$$

$$\mathbf{w}^T\mathbf{x}_i + b < 0 \text{ для } d_i = -1$$

**Точки, для которых первое и второе ограничения выполняются со знаком равенства – опорные вектора**

**Дискриминантная функция** $g(\mathbf{x}) = \mathbf{w}_o^T\mathbf{x} + b_o$

**Определяет меру расстояния от точки x до оптимальной гиперплоскости**

$$g(\mathbf{x}_p) = 0$$

$$g(\mathbf{x}) = \mathbf{w}_o^T\mathbf{x} + b_o = r\,\|\mathbf{w}_0\| \qquad r = \frac{g(\mathbf{x})}{\|\mathbf{w}_o\|}$$

**Осуществляется поиск $w_0$ и $b_0$, максимизирующих расстояние до каждого класса (зазор)**

$$r = \frac{g(\mathbf{x}^{(s)})}{\|\mathbf{w}_o\|} = \begin{cases} \frac{1}{\|\mathbf{w}_o\|}, \text{ если } d^{(s)} = +1, \\ -\frac{1}{\|\mathbf{w}_o\|}, \text{ если } d^{(s)} = -1, \end{cases} \qquad \rho = 2r = \frac{2}{\|\mathbf{w}_o\|}$$

**Максимизация границы разделения между классами (поиск оптимальной гиперплоскости) эквивалентна минимизации Евклидовой нормы вектора весов w**

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: КЛАССИФИКАЦИЯ (СЛУЧАЙ ЛИНЕЙНОЙ РАЗДЕЛИМОСТИ КЛАССОВ)

**Поиск для данного обучающего множества $\boldsymbol{T = \{(x_i, d_i)\}_{i=1}^{N}}$ оптимального значения вектора весовых коэффициентов $w$ и порога $b$, удовлетворяющих условию:**

$$d_i(w^T x_i + b) \geq 1 \text{ для } i = 1, 2, \dots, N$$

**и минимизирующих функцию стоимости:**

$$\Phi(x) = \frac{1}{2} w^T w$$

**Функция Лагранжа, которую необходимо минимизировать по w и b одновременно максимизируя по $\boldsymbol{\alpha}$:**

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i [d_i(w^T x_i + b) - 1] \qquad \text{где } \alpha_i - \text{множители Лагранжа}$$

$$\frac{\partial J(w, b, \alpha)}{\partial w} = 0 \qquad w = \sum_{i=1}^{N} \alpha_i d_i x_i$$

$$\frac{\partial J(w, b, \alpha)}{\partial b} = 0 \qquad \sum_{i=1}^{N} \alpha_i d_i = 0$$

$$\alpha_i [d_i(w^T x_i + b) - 1] = 0 \text{ для } i = 1, 2, .., N$$

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: КЛАССИФИКАЦИЯ (СЛУЧАЙ ЛИНЕЙНОЙ РАЗДЕЛИМОСТИ КЛАССОВ)

$$J(w,b,\alpha)=\frac{1}{2}w^Tw-\sum_{i=1}^{N}\alpha_i[d_i(w^Tx_i+b)-1]$$ где $\alpha_i$ – множители Лагранжа

$$J(w,b,\alpha)=\frac{1}{2}w^Tw-\sum_{i=1}^{N}\alpha_i d_i w^Tx_i-b\sum_{i=1}^{N}\alpha_i d_i+\sum_{i=1}^{N}\alpha_i$$

$$\sum_{i=1}^{N}\alpha_i d_i=0$$

$$w^Tw=\sum_{i=1}^{N}\alpha_i d_i w^Tx_i=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j x_i^Tx_j$$ где $\alpha_i$ неотрицательны

$$\max Q(A)=\sum_{i=1}^{N}\alpha_i\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j x_i^Tx_j$$

$$\sum_{i=1}^{N}\alpha_i d_i=0$$

$\alpha_i\geq 0$ для i =1, 2, ..N

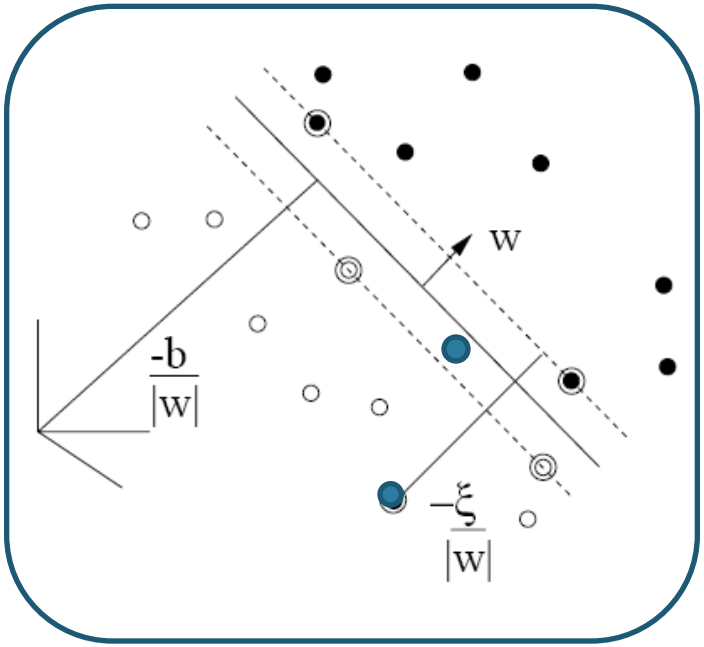
$$w_o=\sum_{i=1}^{N}\alpha_{0,i}d_i x_i \qquad b_0=1-w_0^Tx^{(s)} \text{ для } d^{(s)}=1$$

# ПОИСК ОПТИМАЛЬНОЙ ГИПЕРПЛОСКОСТИ ДЛЯ НЕРАЗДЕЛИМЫХ ОБРАЗОВ

**Для неразделимых образов невозможно построить разделяющую гиперплоскость полностью исключающую ошибки классификации**

**Граница разделения классов считается «мягкой», если некоторая точка нарушает следующее условие:**

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, i = 1, 2, \ldots, N.$$




**Переформулируем оптимизационную задачу, допустив ошибку, но штрафуя за неё:**

$$y_i\left(w \cdot x_i - b\right) \geq 1 - \xi_i$$

$\xi$ **— фиктивная переменная (slack variable), определяющая отклонение точек от состояния линейной разделимости**

**Задача сводится к поиску разделяющей гиперплоскости, минимизирующей суммарную ошибку классификации:**

$$\Phi(\xi) = \sum_{i=1}^{N} I(\xi_i - 1) \qquad I(\xi) = \begin{cases} 0, & \xi \leq 0, \\ 1, & \xi > 0. \end{cases}$$

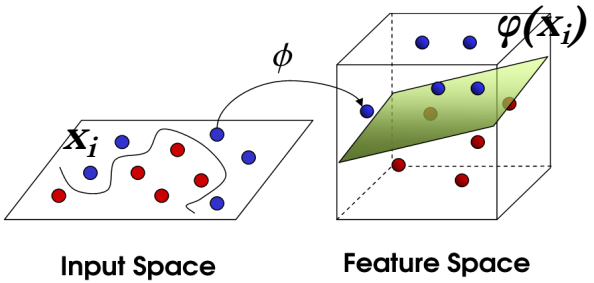
**функция индикатора**

$$\Phi(\xi) = \sum_{i=1}^{N} \xi_i \qquad w = \arg\min\left\{ \tfrac{1}{2}\|w\|^2 + C\sum_i \xi_i \right\}$$

**Коэффициент C — оптимизируемый параметр метода, который позволяет регулировать отношение между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки**

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕЛИНЕЙНОЕ ПРЕОБРАЗОВАНИЕ (KERNEL TRICK)

**Исходное пространство может быть отображено в пространство более высокой размерности, где множество станет линейно-разделимым (теорема Ковера (Cover's theorem)).**




**Условия:**

- **Преобразование должно быть нелинейным**
- **Размерность пространства должна быть достаточно большой**

**Для множества нелинейных преобразований можно определить гиперплоскость:**

$$\sum_{j=1}^{m_1} w_j \varphi_j(\mathbf{x}) + b = 0 \qquad \boldsymbol{\varphi}(\mathbf{x}) = [\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \ldots, \varphi_{m1}(\mathbf{x})]^T$$

***m* – размерность вектора из исходного пространства**

**Поверхность решений в общем виде и адаптируя к задаче линейного разделения векторов в пространстве признаков**

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) = 0 \qquad \mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \varphi(\mathbf{x}_i)$$

$$K(\mathbf{x}, \mathbf{x}_i) = \varphi^T(\mathbf{x})\varphi(\mathbf{x}_i) = \sum_{j=0}^{m_1} \varphi_j(\mathbf{x})\varphi_j(\mathbf{x}_i)$$

**Ядро скалярного произведения характеризует расстояние между двумя векторами и может использоваться для построения оптимальной гиперплоскости в пространстве признаков, не представляя его в явном виде**

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕЛИНЕЙНОЕ ПРЕОБРАЗОВАНИЕ (KERNEL TRICK)

**Наиболее часто используемые ядра классификатора**

Gaussian RBF $$K(x,x') = \exp(\frac{-\|x-x'\|^2}{2\sigma^2})$$

Polynomial $$K(x,x') = (< x,x' > +\theta)^d$$

Sigmoidal $$K(x,x') = \tanh(\kappa < x,x' > +\theta)$$

Inverse multi-quadratic $$K(x,x') = \frac{1}{\sqrt{(x-x')^2 + c^2}}$$

# СВОЙСТВА SVM

- **Возможность выбора различных функций близости (ядер)**
- **Разреженность решения при работе с большими объемами обучающих данных**

  **- только опорные вектора используются при построении разделяющей гиперплоскости**

  **- возможность работы с данными больших размерностей**
- **Переобучение может контролироваться использованием штрафа**
- **Математически удобно: оптимизационная задача гарантировано сходится к одному глобальному минимуму**
- **Возможен отбор значащих для распознавания переменных**
- **Геометрически наглядная интерпретация (в отличие от ANN)**

# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

| | |
|---|---|
| **Kernel Machines** | http://www.kernel-machines.org/software |
| **LIBSVM -- A Library for Support Vector Machines** | http://www.csie.ntu.edu.tw/~cjlin/libsvm/ |
| **Shogun - A Large Scale Machine Learning Toolbox** | http://www.shogun-toolbox.org/ |
| **Shark Machine Learning Library** | http://shark-project.sourceforge.net/ |
| **SVM light** | http://svmlight.joachims.org/ |

# МЕТРИЧЕСКИЕ КЛАССИФИКАТОРЫ: ВЫБОР МЕТРИКИ

**Метрика должна удовлетворять следующим условиям:**

1. $\mathrm{d}(\mathrm{x},\mathrm{y}) = 0 \Leftrightarrow \mathrm{x} = \mathrm{y}$ – аксиома тождества
2. $\mathrm{d}(\mathrm{x},\mathrm{y}) > 0$ – неотрицательность
3. $\mathrm{d}(\mathrm{x},\mathrm{y}) = \mathrm{d}(\mathrm{y},\mathrm{x})$ – аксиома симметрии
4. $\mathrm{d}(\mathrm{x},\mathrm{y}) \leq \mathrm{d}(\mathrm{x},\mathrm{z}) + \mathrm{d}(\mathrm{z},\mathrm{y})$ – неравенство треугольника

**Евклидово расстояние:**

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

**Расстояние Минковского:**

$$d(x,y) = \left(\sum_i |x_i - y_i|^r\right)^{\frac{1}{r}}$$

**Расстояние Хэмминга:**

$$d(x,y) = \sum_i |x_i - y_i|$$

- Влияние отдельных выбросов уменьшается

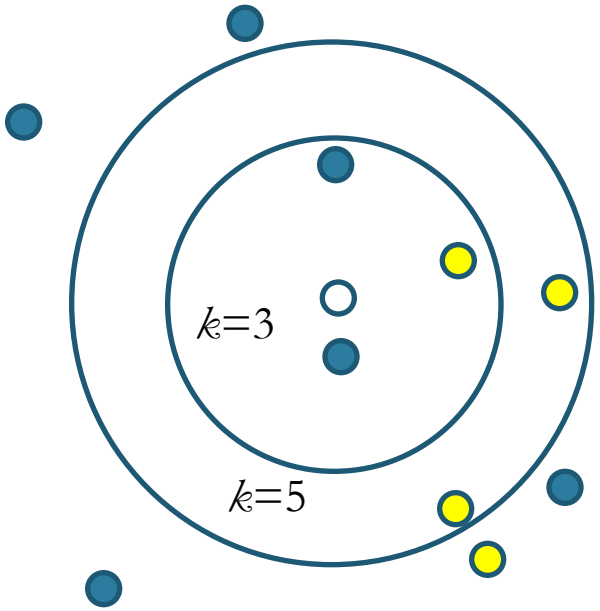
**Расстояние Махаланобиса:**

$$d(x,y) = \sqrt{(x - y)^{\mathrm{T}}\mathbf{M}(x - y)}$$

- Учитывает корреляции между переменными
- Инвариантно к масштабу

# Метод *k* ближайших соседей

**Объект относится к тому классу, к которому принадлежит большинство из его соседей — ближайших к нему объектов обучающей выборки. В задачах с двумя классами число соседей берут нечётным, чтобы не возникало ситуаций неоднозначности, когда одинаковое число соседей принадлежат разным классам.**




- **Вычислить расстояние до каждого из объектов обучающей выборки**
- **Отобрать k объектов обучающей выборки, расстояние до которых минимально**
- **Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей**
- **В случае использования метода для регрессии, объекту присваивается среднее значение по ближайшим**

**Взвешенный способ**

$$Q_j = \sum_{i=1}^{n} \frac{1}{d(x, a_i)^2}$$

***d(x, a)*** **— дистанция от нового значения *x* до объекта *a***

# Метод $k$ ближайших соседей: ограничения

- **Выбор числа соседей $k$**

Оптимальное значение параметра определяют кросс-валидацией моделей.

- **Проблема выбора метрики**

В практических задачах классификации Евклидово расстояние не всегда является наилучшей функцией расстояния.

- **Работа с большими наборами данных**

Метод ближайших соседей основан на явном хранении всех обучающих объектов:
Необходимо работать с большим объёмом данных
Быстро осуществлять поиск ближайших соседей произвольного объекта.

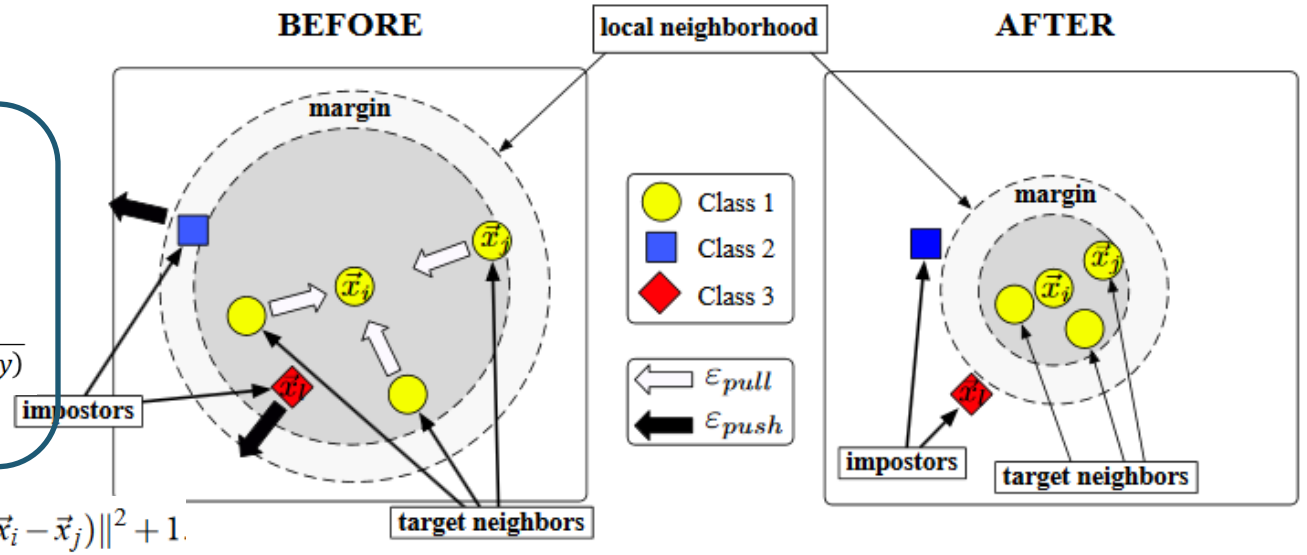
Проблема решается применением специальных индексов и эффективных структур данных для быстрого поиска ближайших соседей .

# LARGE MARGIN NEAREST NEIGHBORS

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

$$d(x,y) = \sqrt{(x-y)^{\mathrm{T}}\mathbf{M}(x-y)}$$

$$\boldsymbol{M} = \mathbf{L}^{\mathrm{T}}\mathbf{L}$$




$$\|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2 \leq \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + 1$$

**Идентификация ближайших соседей того же класса для каждого соединения (фиксируется)**

**Через задание числа ближайших соседей определяется радиус окружения, для которого все соединения другого класса будут обозначены как «чужеродные», минимизация числа которых является целью обучения в сочетании с созданием зазора между локальным окружением и соединениями другого класса**

**Линейное преобразование исходного пространства:** $\boldsymbol{x}' = \mathbf{L}\boldsymbol{x}$

**Минимизируем расстояние между целевыми соседями:**

$$\varepsilon_{pull}(\mathbf{L}) = \sum_{ij} \eta_{ij} \left\|\mathbf{L}(x_i - x_j)\right\|^2$$

**Максимизируем расстояния между объектами разных классов:**

$$\varepsilon_{push}(\mathbf{L}) = \sum_{ijl} \eta_{ij}(1 - y_{il}) \left[1 + \left\|\mathbf{L}(x_i - x_j)^2\right\| - \left\|\mathbf{L}(x_i - x_l)^2\right\|\right]_+$$

$$\varepsilon(\mathbf{L}) = (1-\mu)\varepsilon_{pull}(\mathbf{L}) + \mu\varepsilon_{push}(\mathbf{L})$$

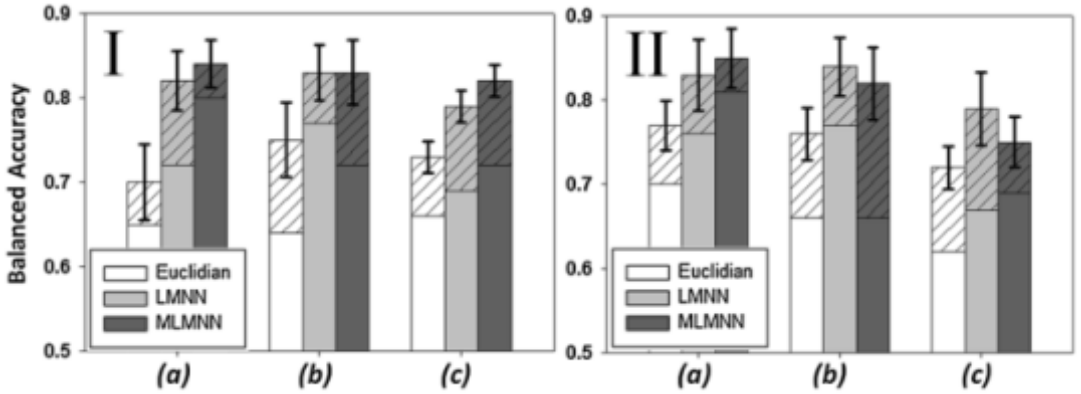
Weinberger et al. (2005) *Advances in neural information processing systems* (pp. 1473-1480)

# LARGE MARGIN NEAREST NEIGHBORS

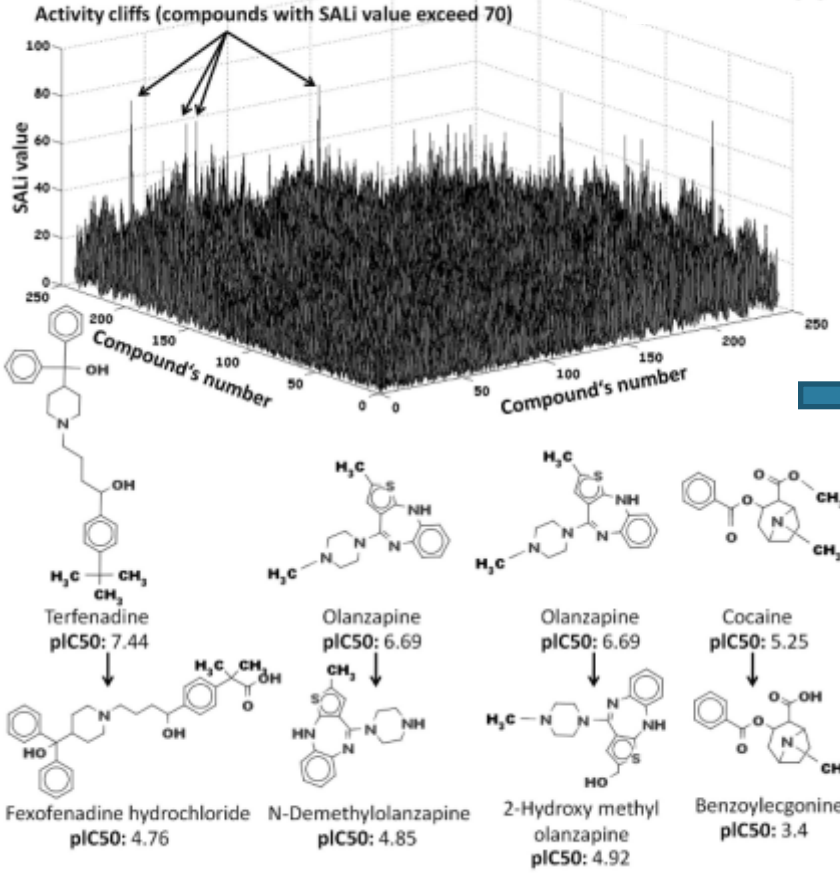
$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

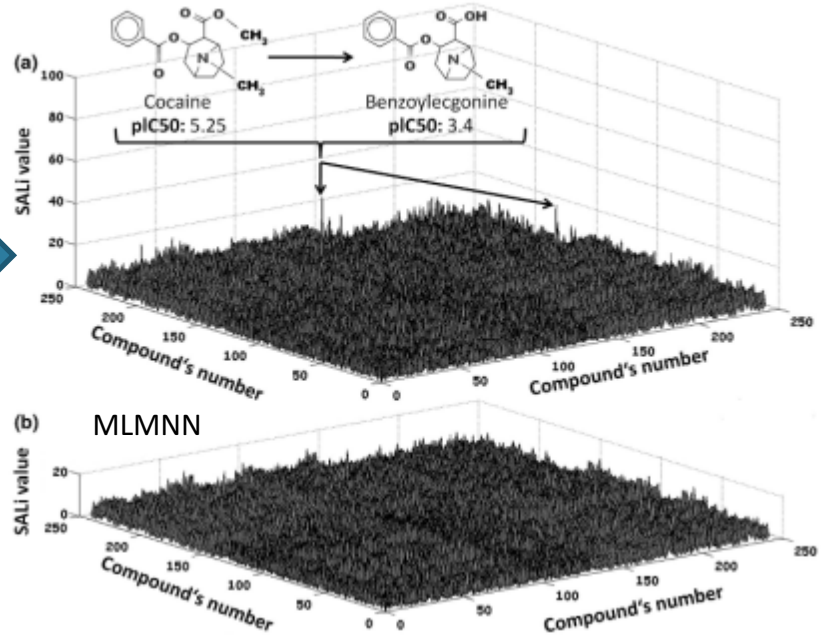
$$d(x,y) = \sqrt{(x - y)^{\mathrm{T}}\mathbf{M}(x - y)}$$

$$\mathrm{SALI}_{i,j} = \frac{|A_i - A_j|}{1 - \mathrm{sim}(i,j)}$$




LMNN




(IC50 - концентрация полумаксимального ингибирования)

# БАЙЕСОВСКАЯ КЛАССИФИКАЦИЯ: НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$ **Теорема Байеса**

$P(c|d)$ - апостериорная вероятность принадлежности данному классу при данном значении признака

$P(c)$ – априорная вероятность данного класса

$P(d|c)$ – правдоподобие (вероятность данного значения признака при данном классе)

$P(d)$ – априорная вероятность данного значения признака

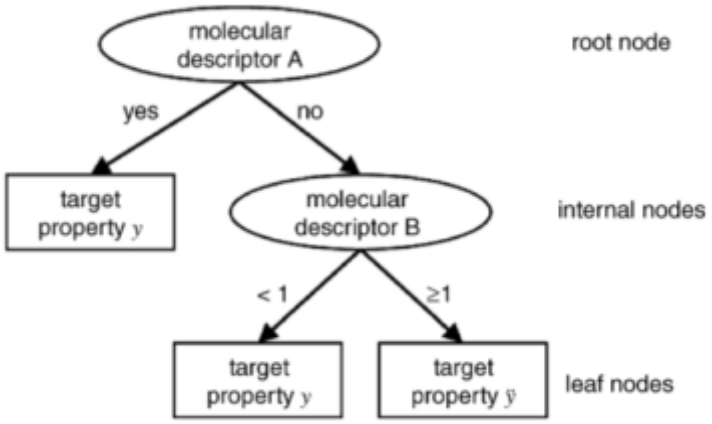
Зная с какой вероятностью причина приводит к некоторому событию можно рассчитать вероятность того, что именно эта причина привела к наблюдаемому событию

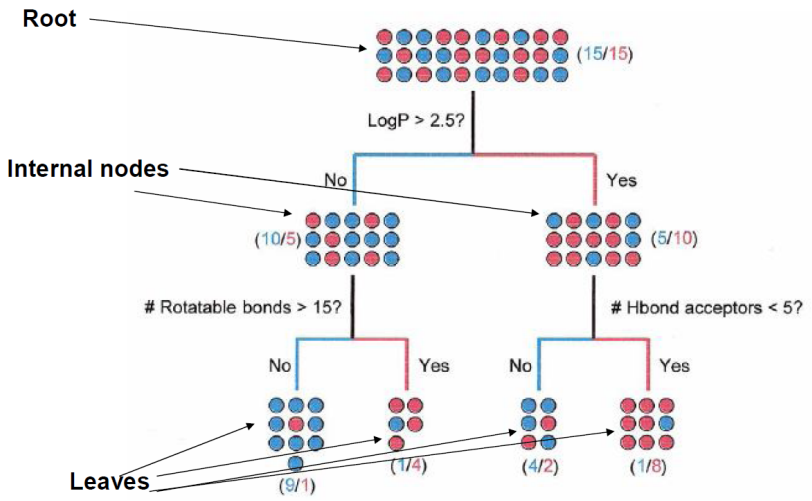
$$c = \max_c P(c|d)$$

# ЛОГИЧЕСКИЕ МЕТОДЫ: ДЕРЕВЬЯ РЕШЕНИЙ

**Для заданного набора данных строится дерево, в узлах которого стоят условия перехода, а в листьях — значения целевой функции**

**Категоризированные данные (например, Druglike/non druglike, активное/неактивное)**
**Построение модели сводится к поиску правила, обеспечивающего статистически наилучшее распределение на классы ( e.g. drug: MW < 500, non drug: MW > 500)**



# Этапы построения деревьев решений

## Выбор критерия, по которому пойдет ветвление

- **Для каждого дескриптора определяется пороговое значение, приводящее к наилучшему делению объектов (соединений) на классы**
- **Дескриптор, обеспечивающий наилучшее разделение выбирается как узел**
- **Рекурсивное применение алгоритма для каждой из ветвей**

## Остановка обучения

- **Использование статистических методов для оценки целесообразности дальнейшего разбиения** (Ограничение глубины: остановка обучения, если разбиение ведет к дереву с глубиной превышающей заданное значение. Разбиение должно быть нетривиальным, т.е. получившиеся в результате узлы должны содержать не менее заданного количества примеров)
- **Пост-процессинг (**построение сначала полного дерева, которое затем уменьшается до оптимального с точки зрения достижения максимальной прогностической способности размера путём объединения некоторых концевых вершин.

# Критерии  оценки качества решений

**Энтропийный индекс неоднородности**
оценка среднего количества информации, необходимого для определения класса соединения.

$$\gamma_e(\widetilde{S}) = -\sum_{i=1}^{L} P_i \ln P_i,$$

$P_i$ - доля объектов  рассматриваемого класса

**Индекс Gini**
Оценка "расстояния" между распределениями классов, на основе идеи уменьшения неопределённости в узле.

$$\gamma_g(\widetilde{S}) = 1 - \sum_{i=1}^{L} P_i^2.$$

# Ансамбли классификаторов

# Ансамбли классификаторов

- **В основе лежит идея обучения набора классификаторов**.
- **Преимущества:** улучшение прогнозирующей способности.
- **Недостаток:** сложность интерпретации итоговой модели.

## Основные представители

- Усреднение по ансамблю
- Бэггинг (Bagging): ресэмплинг (передискретизация) обучающих данных
- Метод усиления (Boosting): изменение «весов» обучающих данных

## Применение ансамблей классификаторов

Использование ансамблей классификаторов (совместное моделирование) может решить следующие проблемы:

- ***Статистическая проблема***

возникает, когда при существовании множества моделей с сопоставимой точностью обучающий алгоритм выбирает лишь одну из них. Существует риск потери точности на внешних данных.

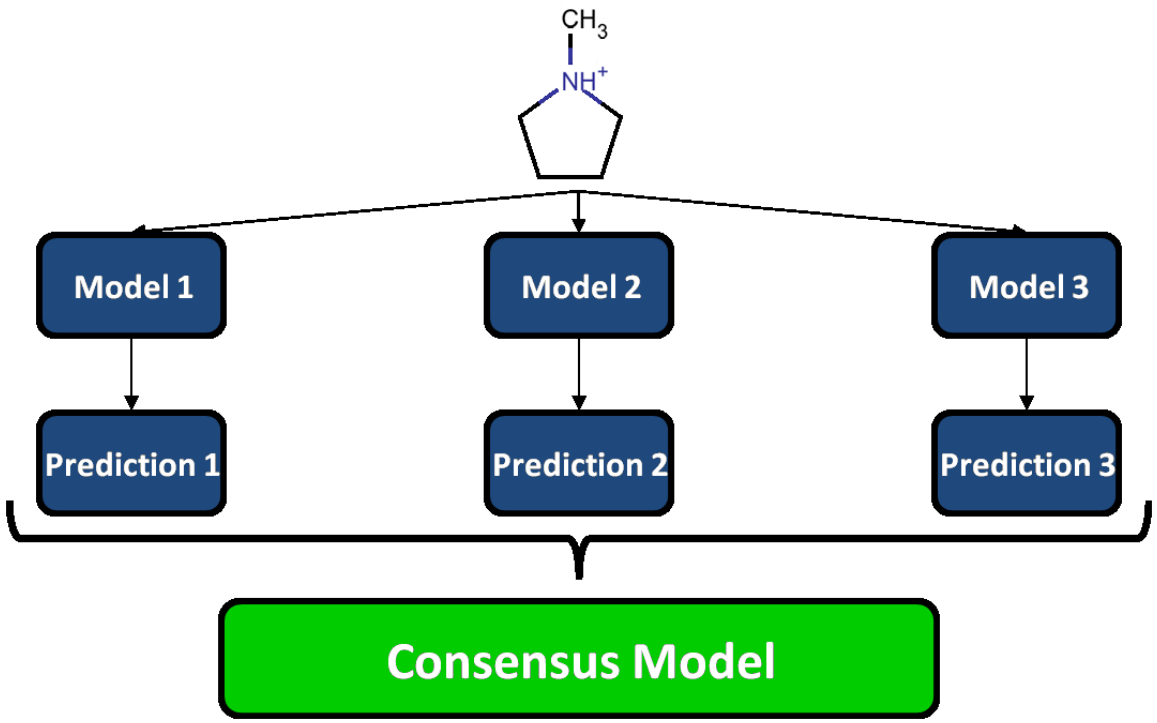
- ***Вычислительная проблема***

возникает, когда обучающий алгоритм не может гарантировать нахождения лучшей модели.

- ***Проблема репрезентативности***

возникает при недостаточном количестве обучающих примеров.

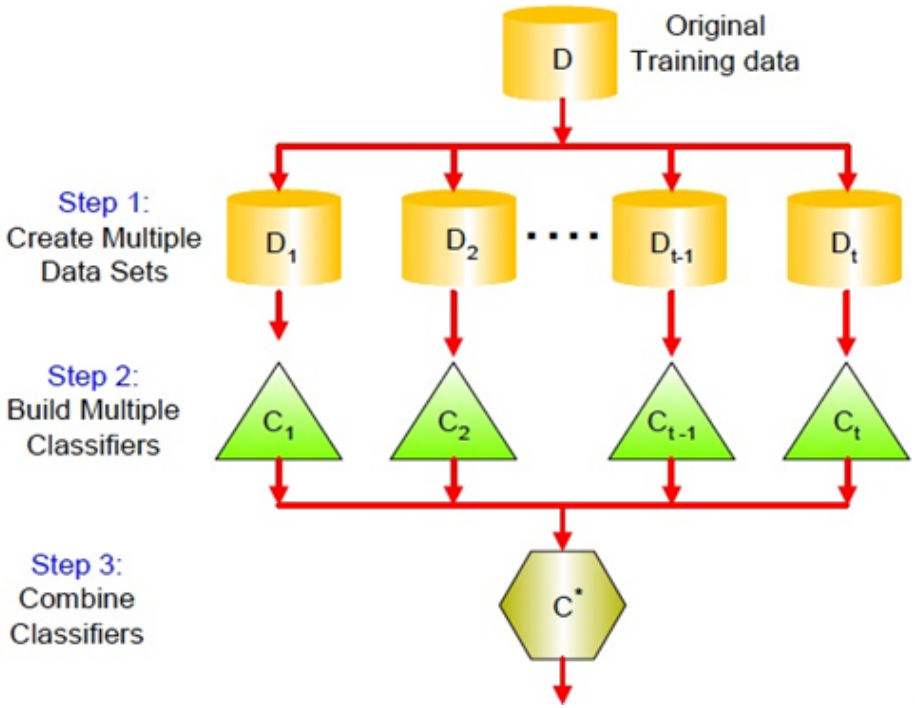
# Ансамбли классификаторов: усреднение по ансамблю



# Бэггинг (Bagging)

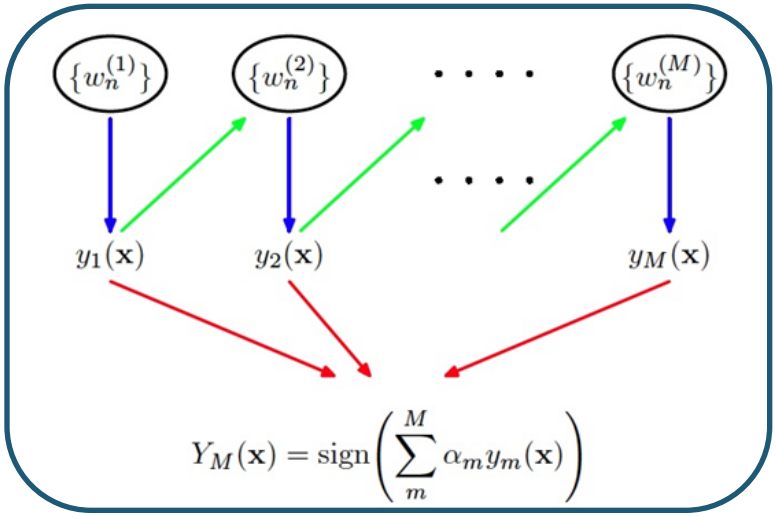
Метод формирования ансамблей классификаторов с использованием случайной выборки с возвратом или бутстрэпа. Название метода произошло от англ. bootstrap aggregating – bagging (Leo Breiman 1994).

- Из исходного набора данных случайным образом отбирается несколько поднаборов, содержащих аналогичное количество соединений: поскольку отбор производится случайно, набор соединений в этих выборках будет различным (некоторые из них могут быть отобраны по несколько раз, а другие – ни разу).
- На основе каждого поднабора строится классификатор и их выходы комбинируются.



# Метод усиления (Boosting)

- **В основе метода усиления лежит идея разработки цепочки (ансамбля) классификаторов, каждый из которых (кроме первого) в обучении использует ошибки предыдущего.**
- **Результат определяется путем простого голосования: пример относится к тому классу, который выдан большинством моделей ансамбля.**




**Способы реализации**

Усиление за счет фильтрации

Отбор примеров обучения различными версиями слабого алгоритма, необходим большой исходный набор данных

Усиление за счет формирования подвыборок

Подвыборки формируются в соответствии с заданным распределением вероятности (AdaBoost)

Усиление путем перевзвешивания

Слабый алгоритм обучения получает «взвешенные» примеры

# Метод усиления (Boosting): усиление за счет фильтрации


Первый классификатор обучается на наборе данных. Обученный первый классификатор используется для фильтрации набора данных. Равновероятны две ситуации:

- набор данных пропускается через классификатор до тех пор, пока не возникнет ошибка. Неверно классифицированный пример добавляется в набор данных для обучения второго классификатора
- набор данных пропускается через классификатор до тех пор, пока пример не будет классифицирован верно (добавляется в набор данных)

Процесс продолжается, пока весь набор не будет отфильтрован.

После обучения второго классификатора следующий набор формируется следующим образом:

- Если прогноз классификаторов совпадает, соединение отклоняется, в противном случае включается в набор данных для обучения третьего классификатора
- После окончания обучения третьего эксперта процесс обучения считается завершенным

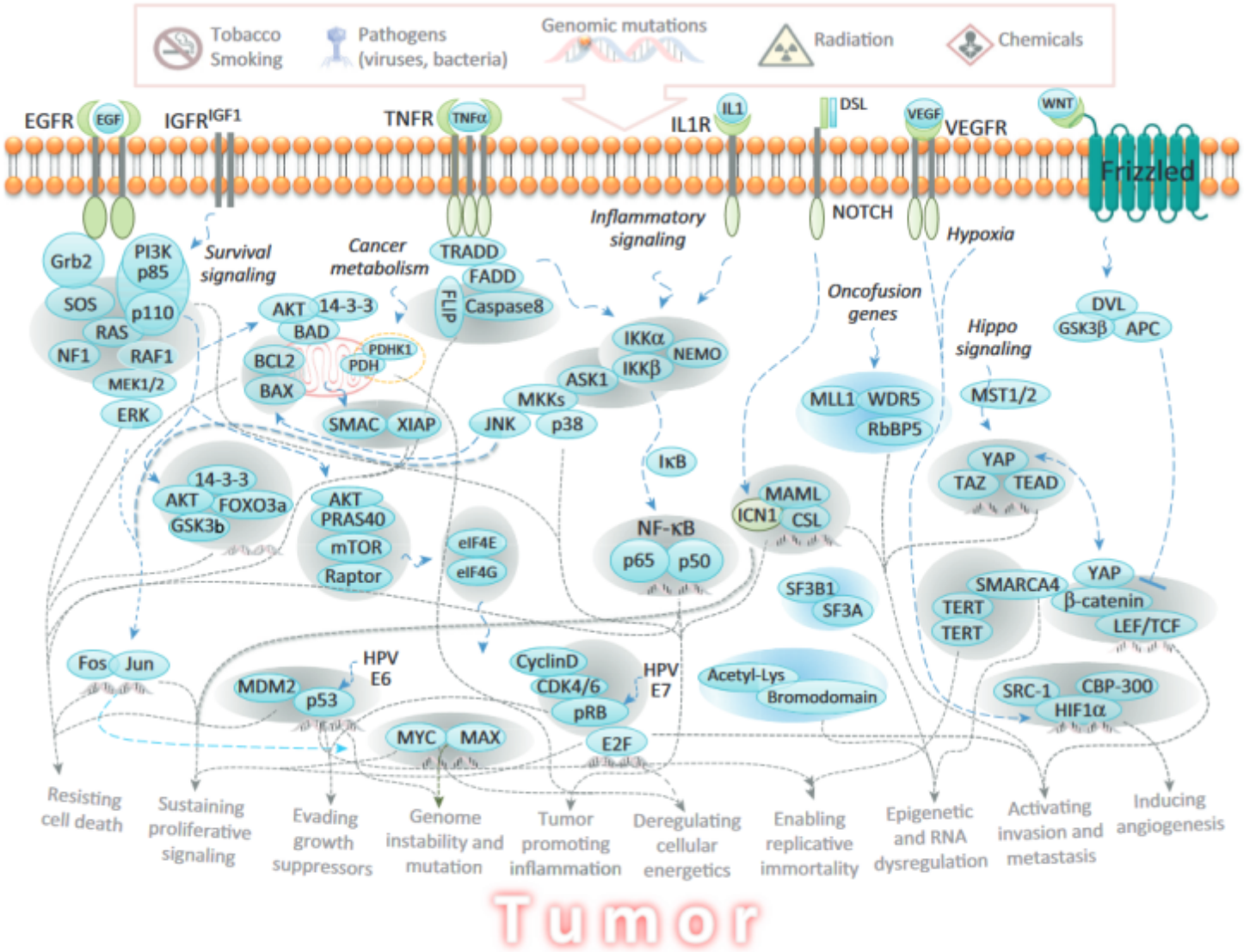
# Случайный лес (Random Forest)

На каждой итерации осуществляется случайная выборка переменных

Для этого нового набора дескрипторов строят дерево принятия решений, при этом производится “bagging”. При этом осуществляется выборка случайным образом двух третей наблюдений для обучения, а оставшаяся треть используется для оценки результата.

Результирующая модель будет результатом “голосования” набора полученных при моделировании деревьев.

# IChemPIC: A Random Forest Classifier of Biological and Crystallographic Protein–Protein Interfaces




Ivanov, A. A. et al Targeting Protein-Protein Interactions as an Anticancer Strategy. Trends Pharmacol. Sci. 2013 , 34, 393 – 400.

# IChemPIC: A Random Forest Classifier of Biological and Crystallographic Protein−Protein Interfaces

**Дескрипторное описание белок-белкового взаимодействия:**

- Приблизительное определение интерфейса посредством попарной оценки расстояний между всеми атомами различных цепей, сохраняя только те, для которых не менее 20 значений межатомных расстояний не превышает 5 Å.
- Учет всех межмолекулярных взаимодействий (гидрофобных, ароматических, водородных и ионных связей) между двумя выбранными цепями
- Введение псевдоатома, расположенного посередине расстояния между каждой парой взаимодействующих атомов

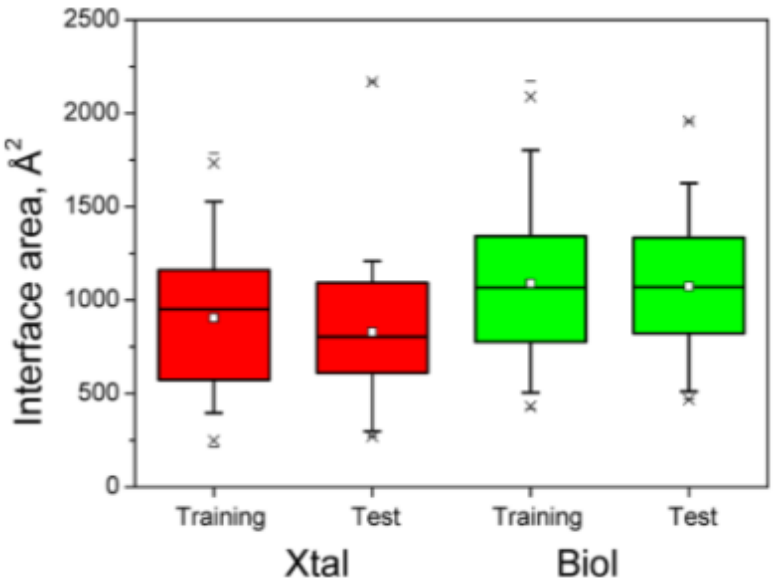
**Итоговый вектор дескрипторов**

- Общее количество взаимодействующих псевдоатомов
- Процент взаимодействий каждого типа (гидрофобные, ароматические, водородные или ионные связи)
- Распределение глубины расположения (buriedness) для каждого типа взаимодействия (Comparison and Druggability Prediction of Protein–Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes J. Desaphy et al *JCIM* 2012 *52* (8), 2287-2299 DOI: 10.1021/ci300184x )

DOI: 10.1021/acs.jcim.5b00190 J. Chem. Inf. Model. 2015, 55, 2005−2014

# IChemPIC: A Random Forest Classifier of Biological and Crystallographic Protein–Protein Interfaces

**Площади интерфейсов**




**Типы взаимодействий**

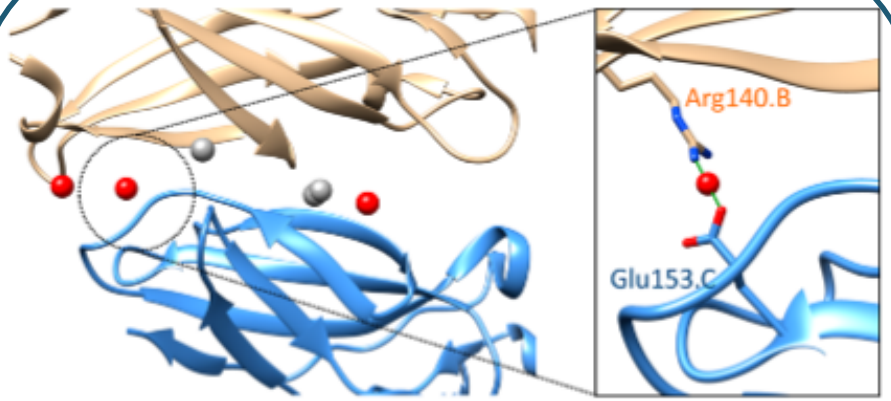



**Table 1. Average Percentage of Interaction Types at Crystallographic and Biological Interfaces**

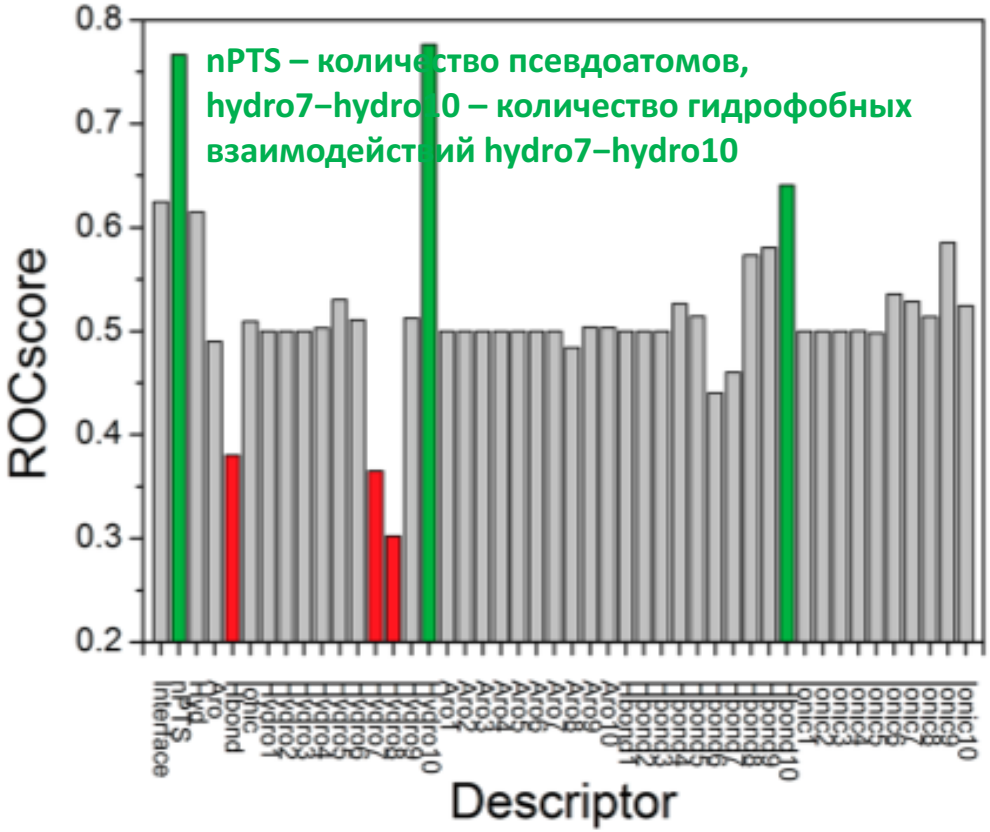
| | protein–protein interface[a] | |
|---|---|---|
| interaction type | crystallographic | biological |
| hydrophobic | 78.06 ± 15.70 | 83.32 ± 9.71 |
| aromatic | 0.24 ± 1.14 | 0.10 ± 0.32 |
| hydrogen bond | 17.97 ± 12.11 | 13.51 ± 7.24 |
| ionic bond | 3.65 ± 5.80 | 3.00 ± 3.87 |

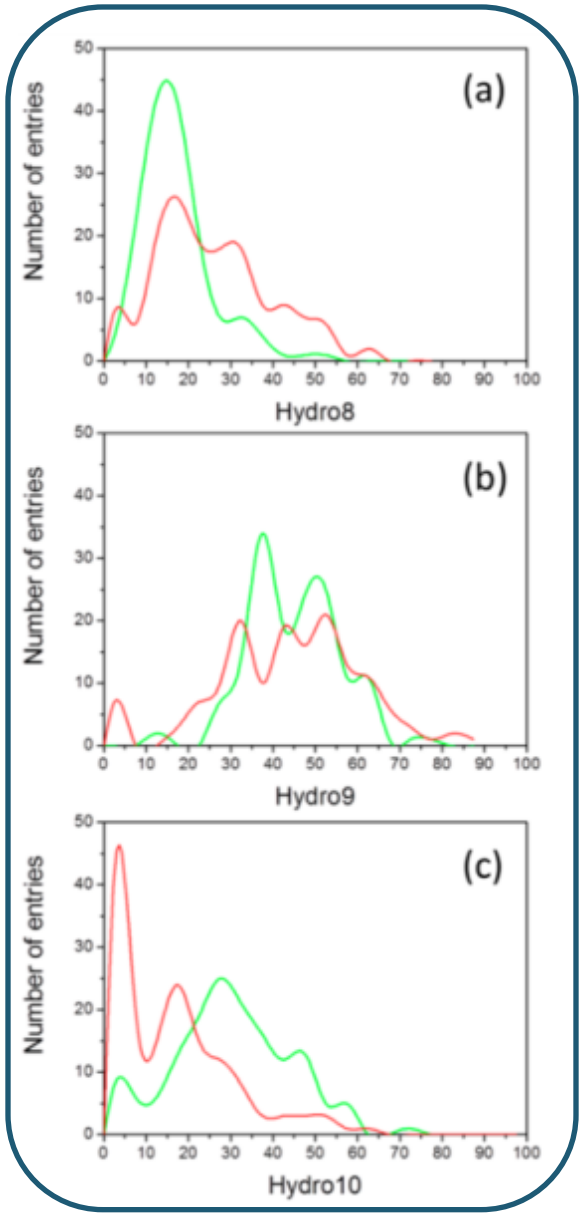
[a]Statistics from 27 186 protein–protein interactions (200 crystallographic and 200 biological interfaces from the FDS set) detected by IChem.[31]

DOI: 10.1021/acs.jcim.5b00190 J. Chem. Inf. Model. 2015, 55, 2005−2014

# IChemPIC: A Random Forest Classifier of Biological and Crystallographic Protein–Protein Interfaces

| parameter | training set ($n$ = 300)[a] | external set ($n$ = 100)[b] |
|---|---|---|
| sensitivity | 0.794 ± 0.017 | 0.728 ± 0.014 |
| precision | 0.759 ± 0.010 | 0.745 ± 0.018 |
| specificity | 0.747 ± 0.014 | 0.750 ± 0.025 |
| accuracy | 0.771 ± 0.009 | 0.739 ± 0.012 |
| F-measure | 0.776 ± 0.009 | 0.736 ± 0.010 |




DOI: 10.1021/acs.jcim.5b00190 J. Chem. Inf. Model. 2015, 55, 2005−2014