Метод опорных векторов (Support Vector Machine)

и его применения в вычислительной биологии

Пятницкий Михаил

Институт Биомедицинской Химии РАМН

Calvin, I'm still confused about **cats** and **dogs**!

OK, then I will explain it once more ...





Задача: научиться распознавать объекты.

Machine Learning, Pattern Recognition

Обучение с учителем

$X = \{x_i, y_i\}_{i=1}^{n}$ обучающее множество

$x_i \in \Re^g$ вектор измерений (признаки объекта)

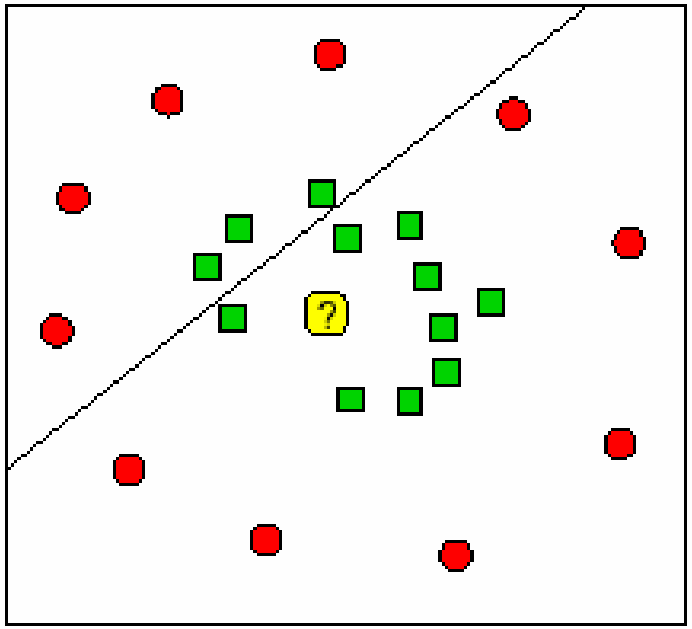
$y_i \in \{+1, -1\}$ метки (класс каждого объекта)

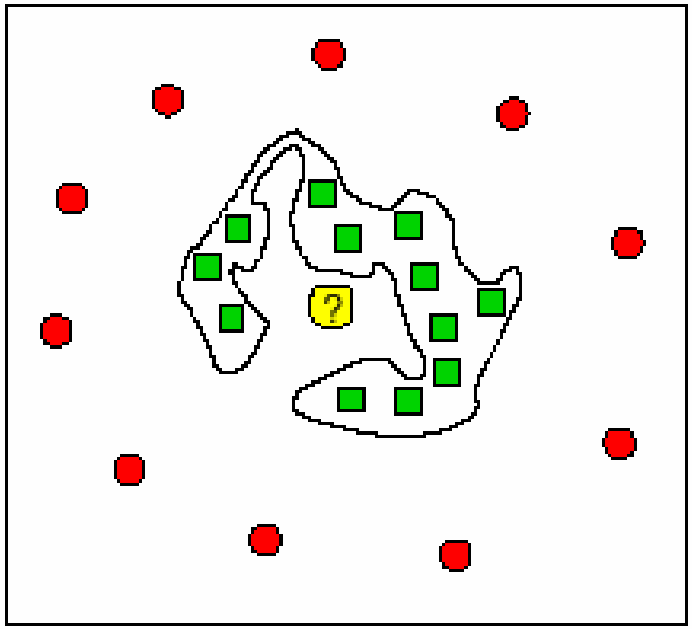
Задача: построить решающее правило, которое хорошо описывает данные, т.е. найти отображение

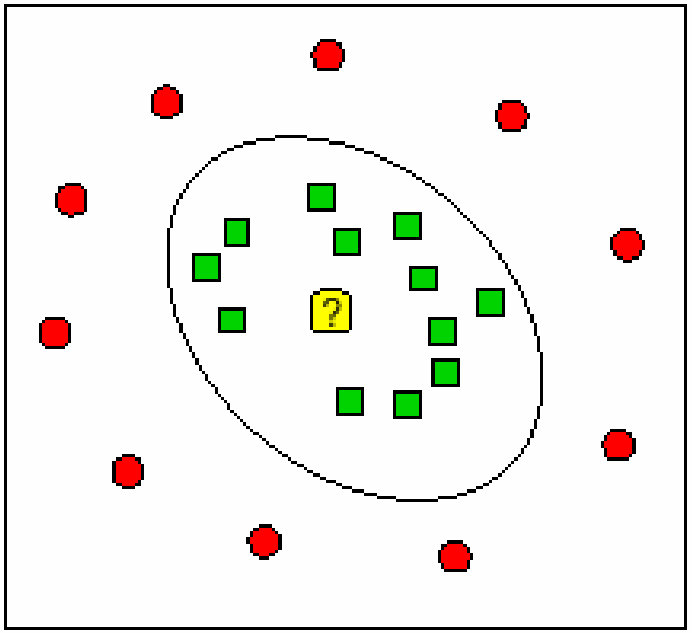
$$f_X : \Re^g \mapsto \{+1, -1\}$$

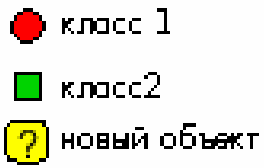
$$\text{Решение} = f_X(\textit{новый объект})$$

Проблема: обобщение









Support Vector Machines

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.

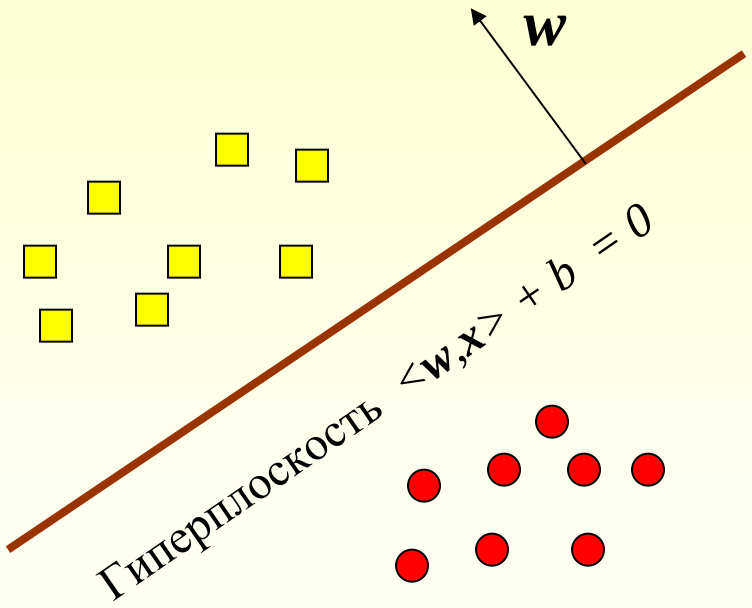
  Statistical Learning Theory, Wiley, NY, 1998

Линейно разделимый случай

Самый простой классификатор - гиперплоскость

$$S = \left\{ \mathbf{x} \mid \left\langle \mathbf{w}, \mathbf{x} \right\rangle + b = 0 \right\}$$

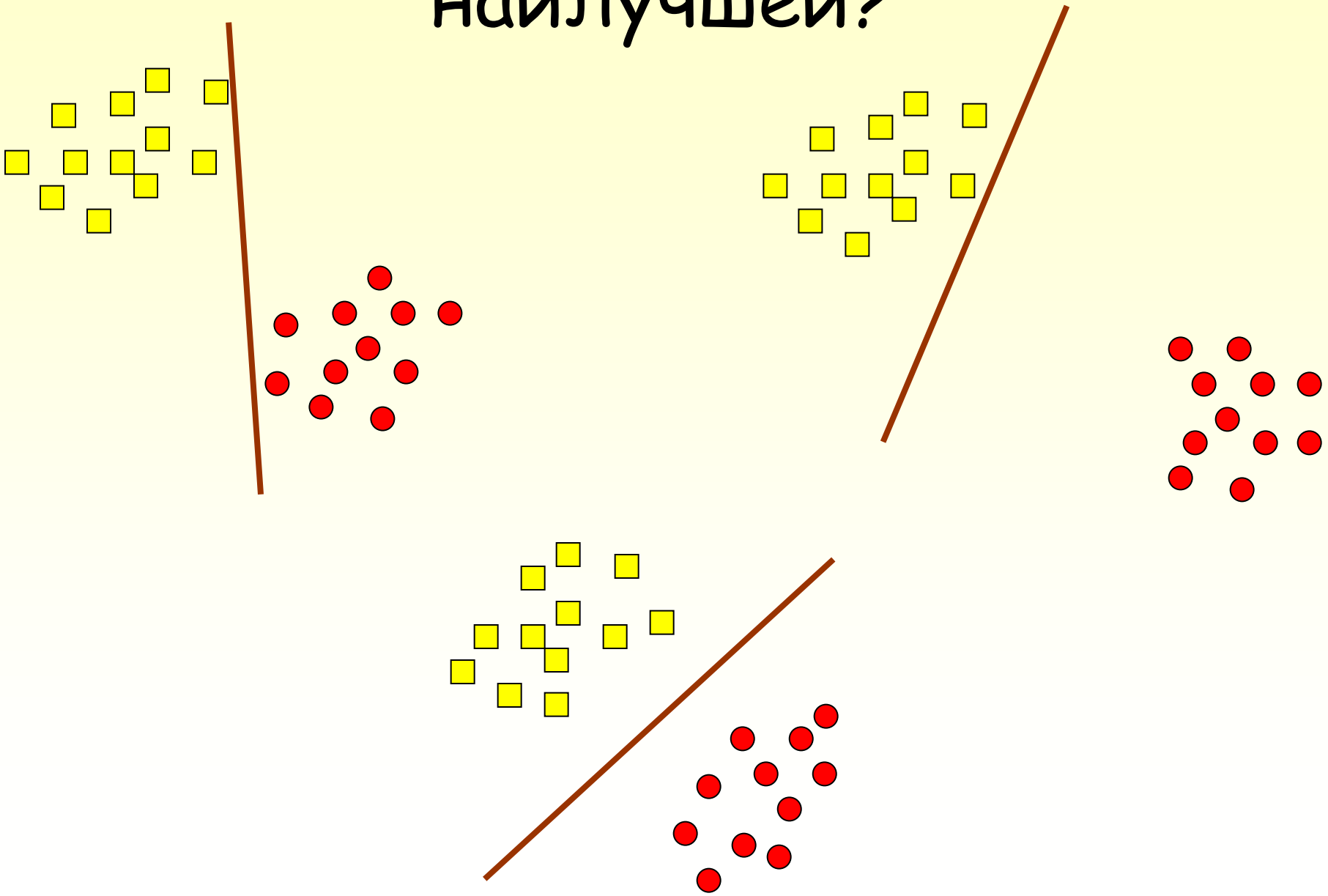
Выбрать **w** и **b** исходя из информации содержащейся в обучающем множестве



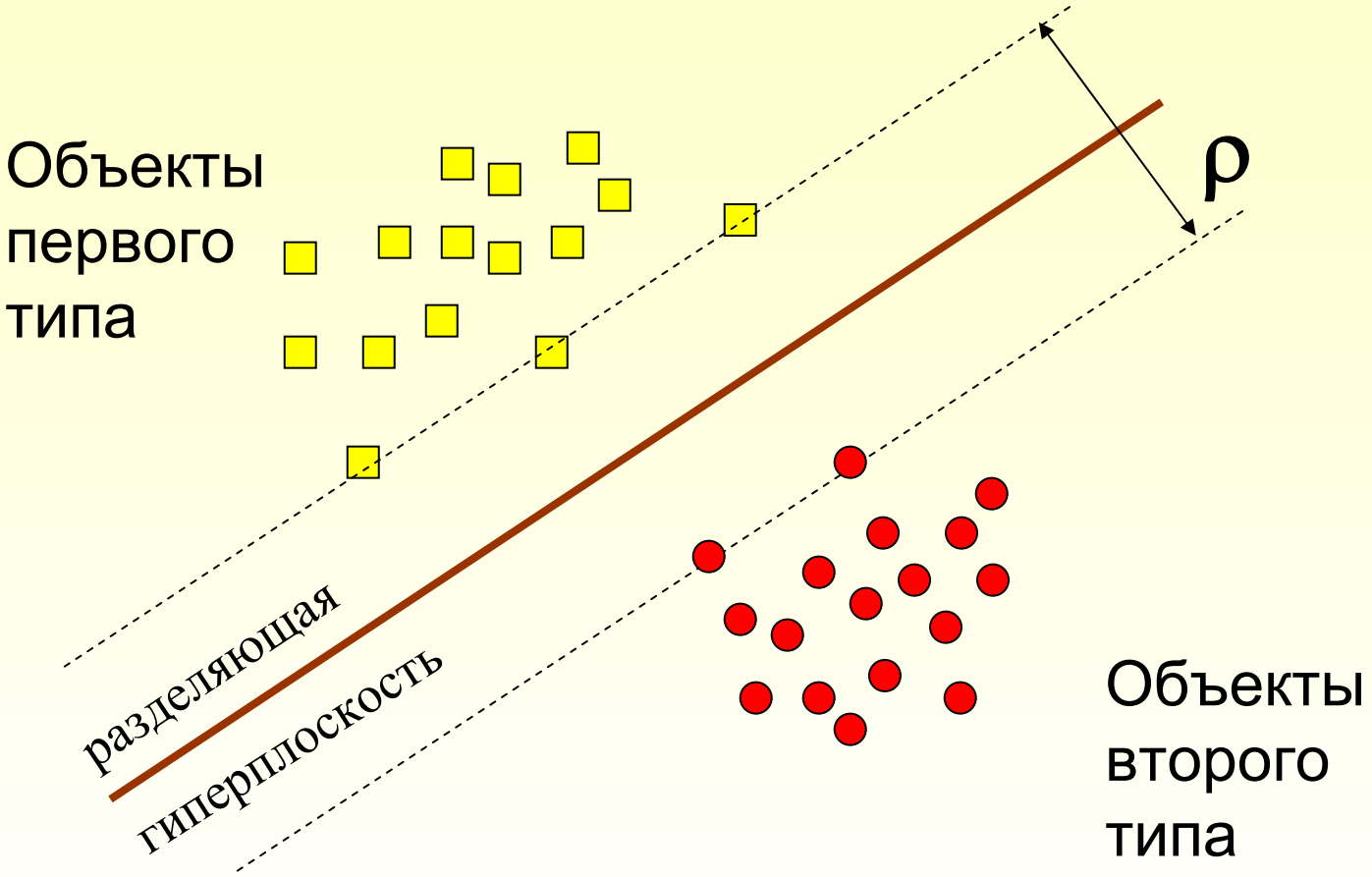
Предсказание: с какой стороны гиперплоскости будет лежать новая точка?

$$f_X\left(x_{новый}\right) = sign\left(\left\langle w, x_{новый} \right\rangle + b\right)$$

Какая гиперплоскость является наилучшей?



SVM: идея №1



SVM строит гиперплоскость с максимальной шириной разделяющей полосы

Постановка задачи

$$\left.\begin{array}{l}\langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq 1 \ \ для \ \ y_i = +1 \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leq -1 \ \ для \ \ y_i = -1\end{array}\right\} \ y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1$$

определение линейно-разделимых множеств

$$\rho(\mathbf{w}, b) = \min_{\{\mathbf{x}: y=1\}} \frac{\langle \mathbf{x}, \mathbf{w} \rangle}{\|\mathbf{w}\|} - \max_{\{\mathbf{x}: y=-1\}} \frac{\langle \mathbf{x}, \mathbf{w} \rangle}{\|\mathbf{w}\|}$$

$$\rho(\mathbf{w}_o, b_o) = \frac{2}{\|\mathbf{w}_o\|}$$

Найти w_o *и* b_o *удовлетворяющие* $y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1$ *для* $i = 1, 2, ..., N$

и минимизирующие функцию стоимости $\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$

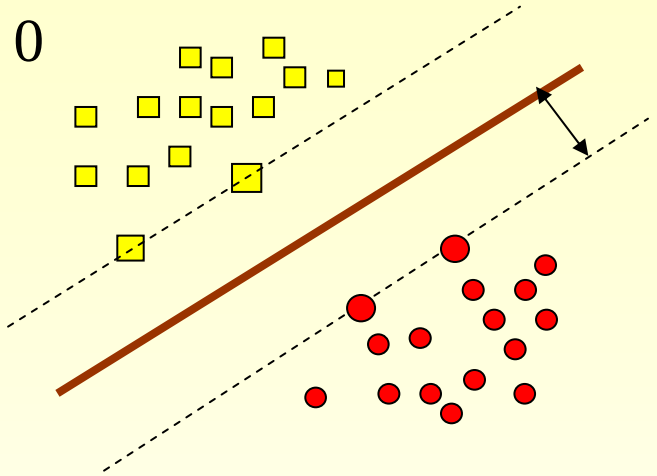
Подобная задача квадратичной оптимизации с линейными ограничениями может быть решена с использованием метода множителей Лагранжа

Лагранжиан

$$L(\mathbf{w},b,\alpha)=\frac{1}{2}\mathbf{w}^T\mathbf{w}-\sum_{i=1}^{N}\alpha_i\left[y_i\left(\langle\mathbf{w},\mathbf{x}\rangle+b\right)-1\right],\quad \alpha_i\geq 0$$

$$\frac{\partial L(\mathbf{w},b,\alpha)}{\partial \mathbf{w}}=0 \qquad \mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i$$

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



В седловой точке для каждого α_i произведение этого множителя на соответствующее ограничение сходится к нулю, т.е.

$$\alpha_i\left[y_i\left(\langle\mathbf{w},\mathbf{x}\rangle+b\right)-1\right]=0 \quad для\ i=1,\ 2,\ ...,\ N$$

Следовательно, ненулевые α будут иметь только те вектора обучающего набора, которые точно соответствуют условию. $y_i\left(\langle\mathbf{w},\mathbf{x}_i\rangle+b\right)=1$

Это *опорные векторы* (support vectors). Только они влияют на положение гиперплоскости

Решение оптимизационной задачи

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \qquad b = y_k - \mathbf{w}^T \mathbf{x}_k \quad для \quad \forall \mathbf{x}_k : \alpha_k \neq 0$$

- Каждый ненулевой α_i соответствует опорному вектору $\mathbf{x_i}$.
- Решающая функция $f(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$
- Зависит только от скалярного произведения между новым вектором **x** и опорными векторами $\mathbf{x_i}$
- Решение задачи оптимизации также требует только вычисления скалярных произведений $\mathbf{x_i^T x_j}$ между всеми парами обучающих векторов.
- Опорные вектора являются критическими элементами обучающего множества, все остальные вектора могут быть убраны без изменения решения

Линейно неразделимый случай

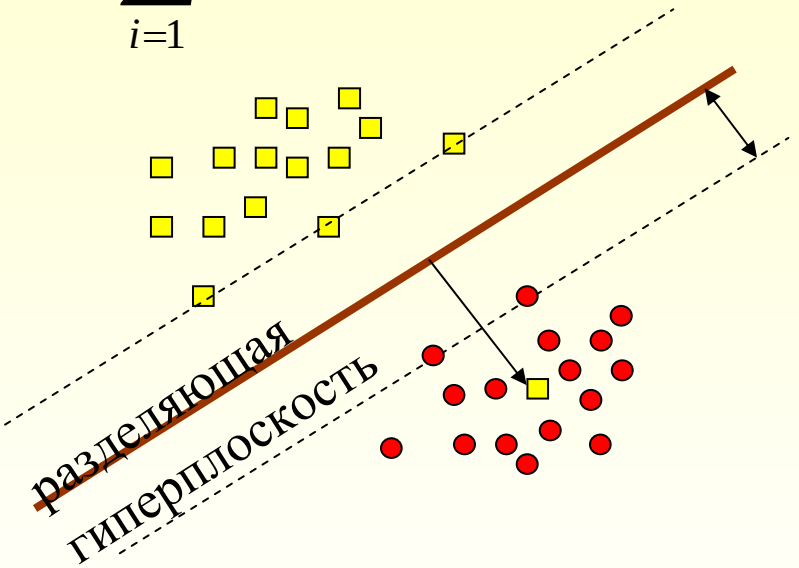
Использовать линейный классификатор, но учитывать ошибки обучения.

$$\textit{минимизировать } \Phi(\mathbf{w},\xi)=\frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{N}\xi_i$$

при условиях

$$y_i\left(\langle \mathbf{w},\mathbf{x}_i\rangle+b\right)\geq 1-\xi_i,$$

$$\xi_i \geq 0$$



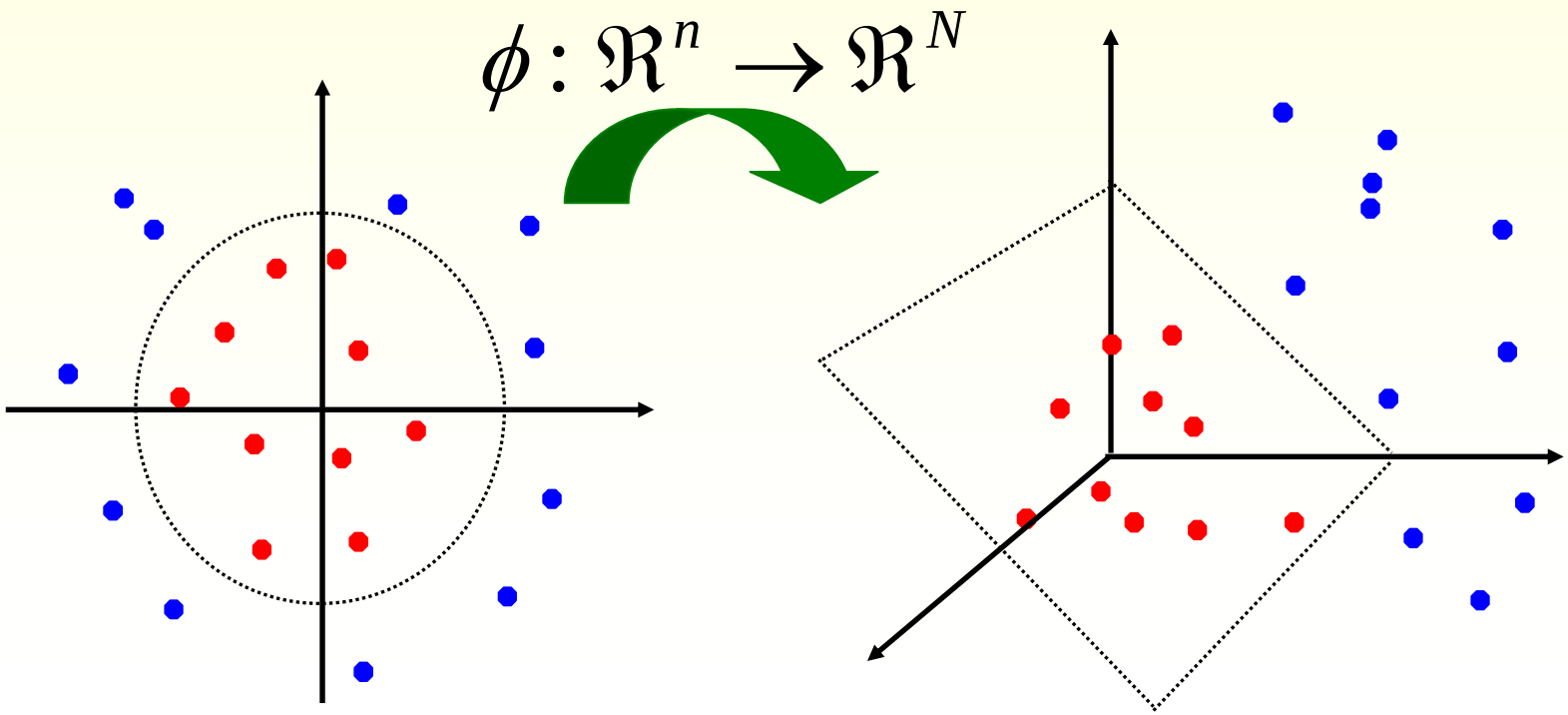
Решение отличается введением более жесткого условия:

$$0 \leq \alpha_i \leq C$$

Штраф за ошибки : расстояние до гиперплоскости умноженное на константу штрафа C

SVM: идея №2

- Исходное пространство может быть отображено в пространство более высокой размерности, где множество станет линейно-разделимым.
- Подтверждение: теорема Ковера (Cover's theorem)
- Ядро: $K(\mathbf{x},\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$ **аналог скалярного произведения, характеризует расстояние между двумя векторами**



Kernel Trick

- Каждое ядро должно быть представимо в виде

$$K(\mathbf{x},\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

- Теорема Мерсера: каждая симметричная, положительно полуопределенная функция является ядром, т.е. матрица К должна быть положительно полуопределенной

$$\mathrm{K} = \begin{array}{|c|c|c|c|c|} \hline K(\mathbf{x}_1,\mathbf{x}_1) & K(\mathbf{x}_1,\mathbf{x}_2) & K(\mathbf{x}_1,\mathbf{x}_3) & \dots & K(\mathbf{x}_1,\mathbf{x}_N) \\ \hline K(\mathbf{x}_2,\mathbf{x}_1) & K(\mathbf{x}_2,\mathbf{x}_2) & K(\mathbf{x}_2,\mathbf{x}_3) & & K(\mathbf{x}_2,\mathbf{x}_N) \\ \hline \dots & \dots & \dots & \dots & \dots \\ \hline K(\mathbf{x}_N,\mathbf{x}_1) & K(\mathbf{x}_N,\mathbf{x}_2) & K(\mathbf{x}_N,\mathbf{x}_3) & \dots & K(\mathbf{x}_N,\mathbf{x}_N) \\ \hline \end{array}$$

Нам не требуется знать как выглядит на самом деле пространство где строится гиперплоскость, нужны только значения ядра как меры близости между двумя векторами

Наиболее часто используемые ядра

- Линейное: $K(\boldsymbol{x}_i,\boldsymbol{x}_j)= \boldsymbol{x}_i^T\boldsymbol{x}_j$ (соответствует классификации в исходном пространстве)

- Полиномиальное степени p:

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)= (1+ \boldsymbol{x}_i^T\boldsymbol{x}_j)^p$$

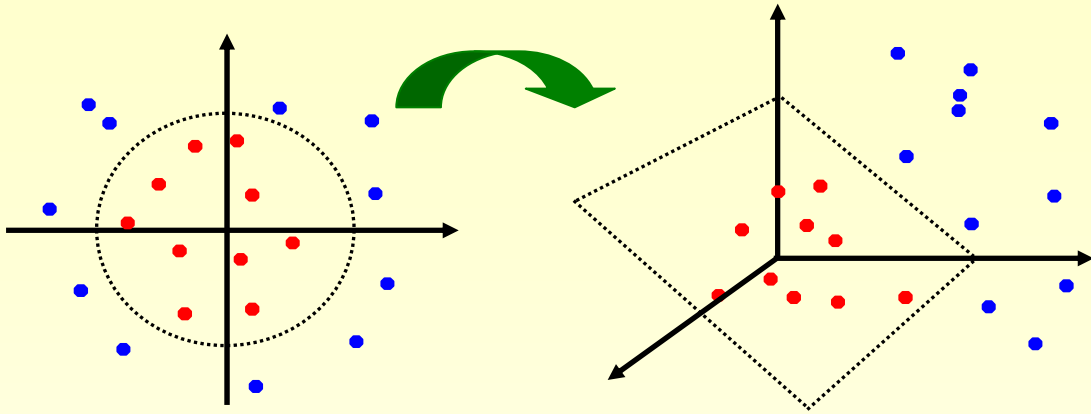
- Гауссово (соответствует RBF-сетям):

$$K(\mathbf{x_i},\mathbf{x_j}) = \exp\left(-\frac{\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{2\sigma^2}\right)$$

- Сигмоидное: $K(\boldsymbol{x}_i,\boldsymbol{x}_j)= \tanh(\beta_0\boldsymbol{x}_i^T\boldsymbol{x}_j + \beta_1)$

Каждое ядро характеризуется параметрами.
Для улучшения качества распознавания эти параметры должны быть оптимальными

Нелинейные SVM



- Решение:

$$f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b$$

- Вычисления в пространствах огромных размерностей становятся возможными благодаря использованию ядер
- Оптимизационный методы для определения α_i остаются неизменными!

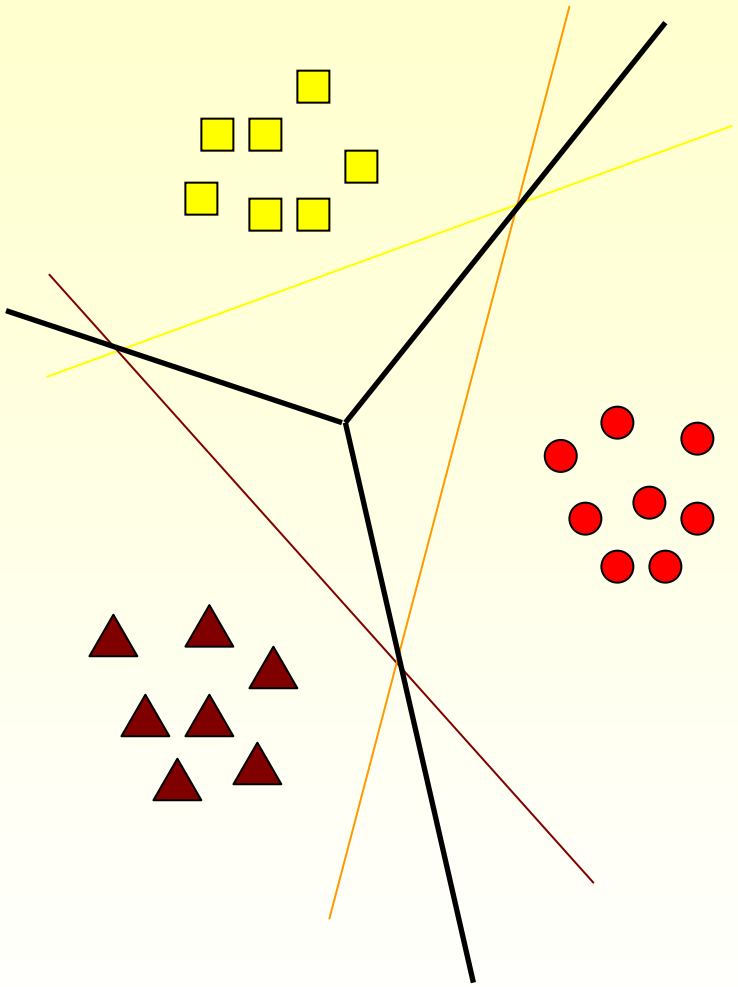
Свойства SVM

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

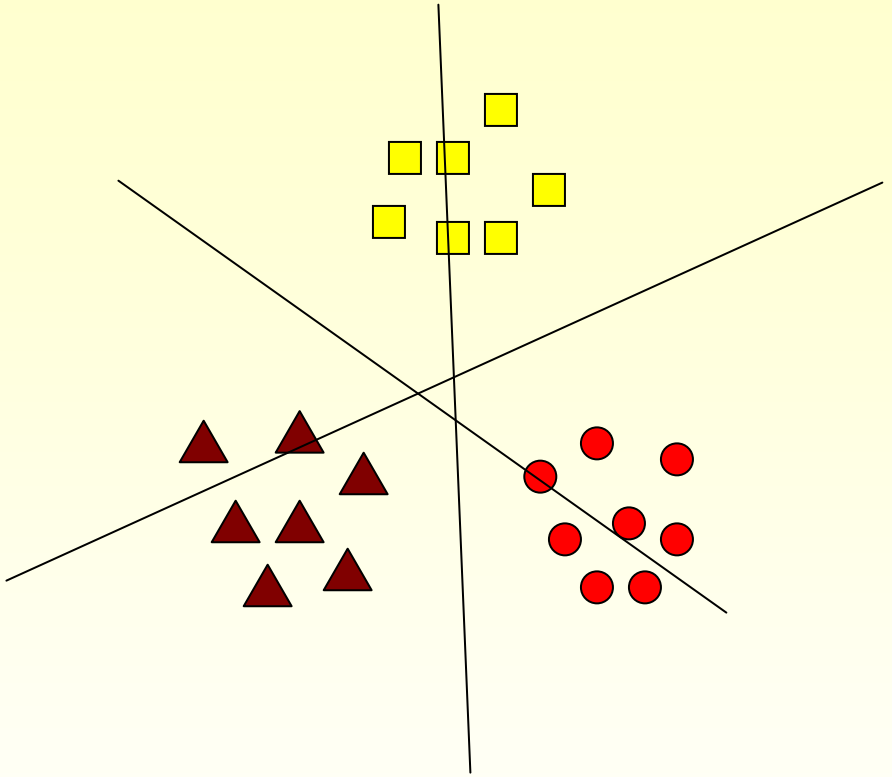
  - только опорные вектора используются при построении разделяющей гиперплоскости

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

Многоклассовая SVM



Один-против-всех



Один-против-одного

Применения SVM в геномике и протеомике

- ✓ Необходимость работы с многомерными, зашумленными данными
- ✓ Применение ядер: возможность использования любых типов данных (векторы, последовательности, сети, и т.д.)
- ✓ Возможность встраивания априорных биологических знаний (т.к. ядро – показывает меру близости между двумя векторами признаков)
- ✓ Метод SVM хорошо зарекомендовал себя практически в любых задачах классификации и регрессии

Классификация генов и белков

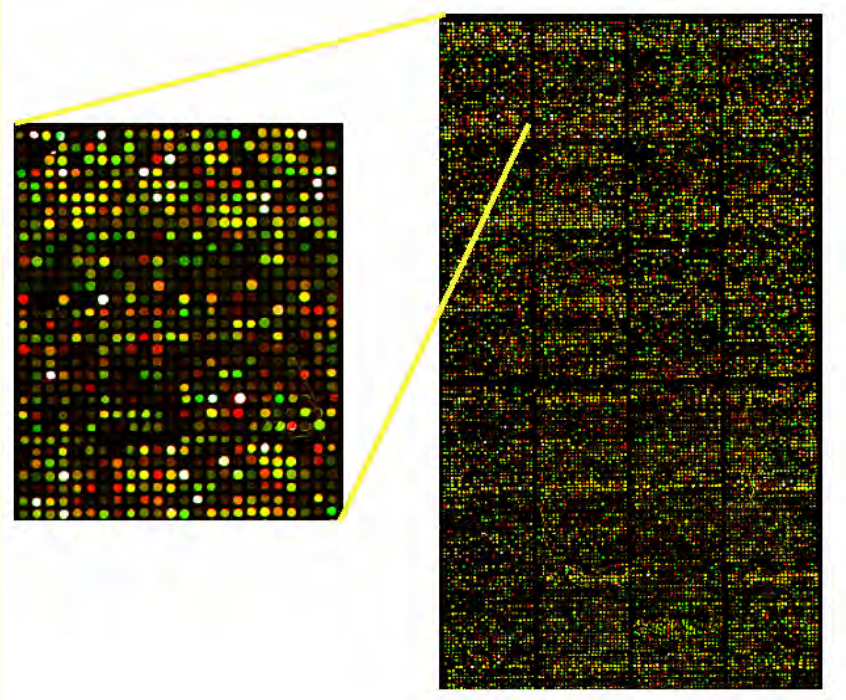
- Классификация промоторов – предсказание функциональной роли белков по промотору (Pavlidis et al, 2001)
- Предсказание функции белка по филогенетическому профилю (Vert, 2002)
- Предсказание внутриклеточной локализации белка (Hua, Sun, 2001)
- Предсказание вторичной структуры белка
- Классификация интрон/экзон (Degroeve et al, 2002)
- Определение места начала трансляции (Zien et al, 2000)
- Предсказание белок-белковых взаимодействий по первичной структуре

Пример: ядра для выявления отдаленных белков-гомологов

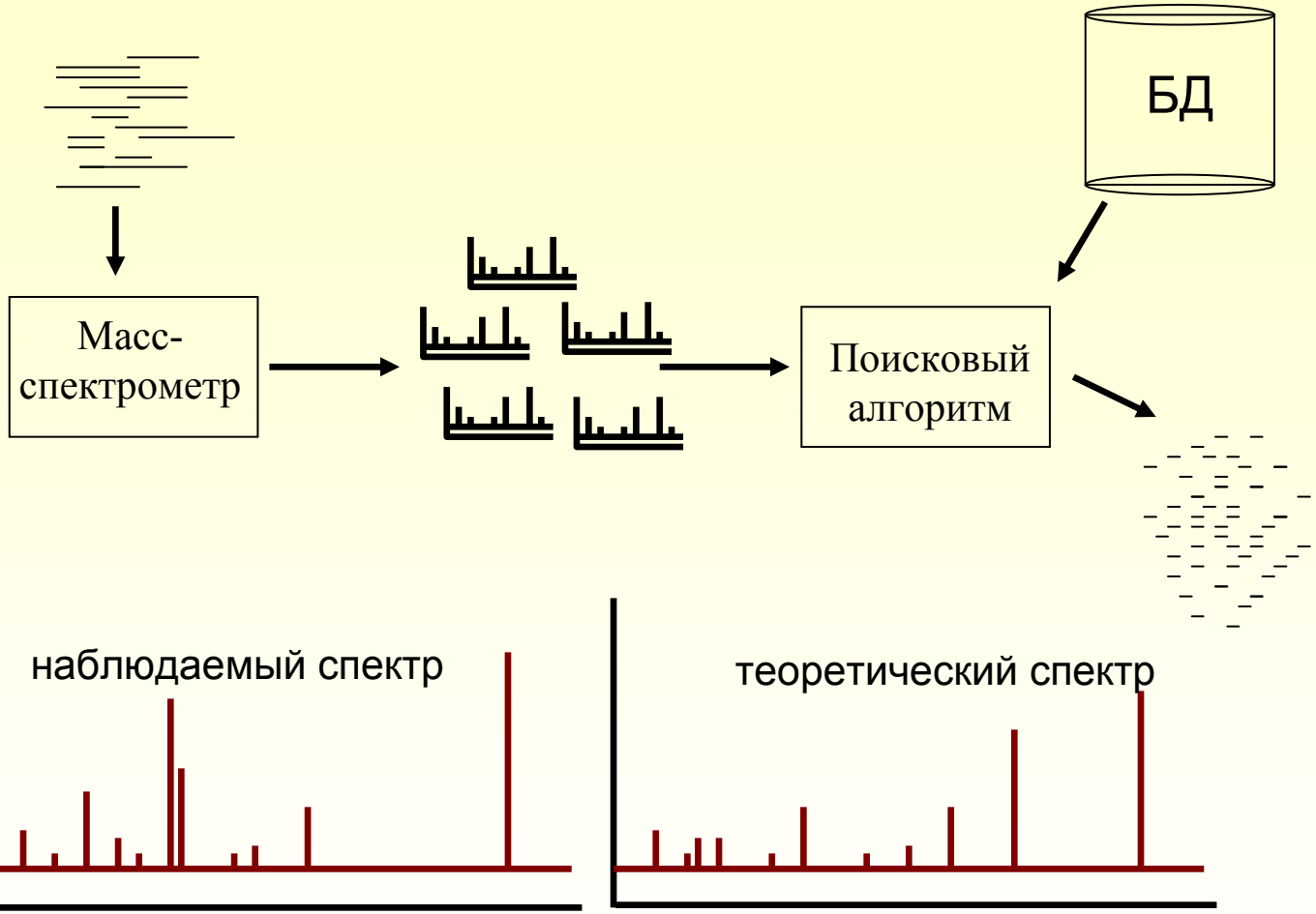
- Fisher kernel – совместное применение SVM и HMM (Jaakkola et al, 1999)
- Composition kernel – каждый белок характеризуется частотой встречаемости в нем аминокислот, гидрофобностью и т.д. (Ding, Dubchak, 2001)
- Motif kernel (Ben-hur, Brutlag, 2003)
- Pairwise composition kernel – использование BLAST, Smith-Watrman (Liao, Noble, 2003).

SVM и микрочипы (microarray)

- Число признаков много больше числа объектов
- SVM позволяет избежать переобучения при таком большом числе признаков
- Отбор важных для классификации генов
- Относительно устойчив к шуму
- Низкий процент ошибок



SVM и масс-спектрометрия



- Классификация результата работы Sequest: false positives/true positives (Anderson et al, 2001)

Объединение разнородных данных (data fusion)

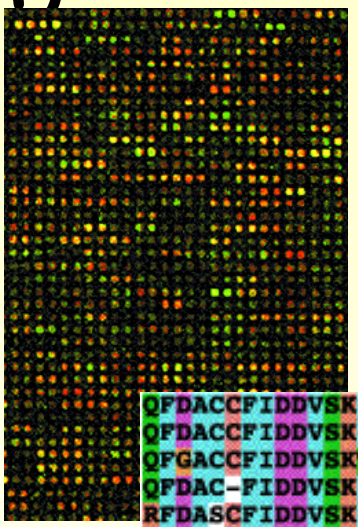
SVM позволяет легко использовать данные самого разнообразного характера (матрицы, последовательности, графы).

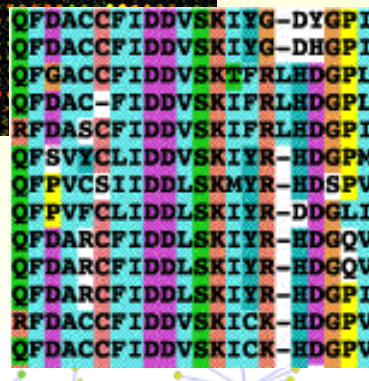
Объединение векторов различных данных:

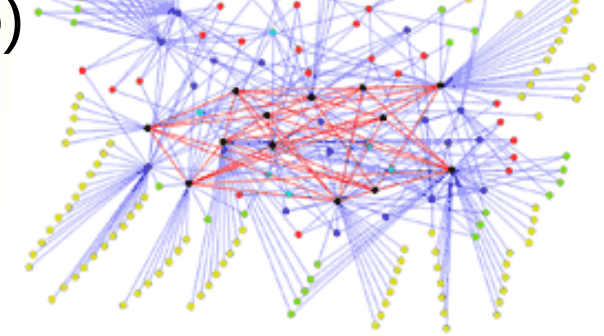
- раннее (2 вектора конкатенируются)
- промежуточное (2 ядра вычисляются отдельно, суммируются, на этом результате обучается SVM)
- позднее (2 SVM обучаются отдельно)

Данные биочипов + филогенетические профили

Данные биочипов + метаболические сети







Еще SVM!

- ✓ Метод выбора информативных признаков, учитывающий их взаимодействие – Recursive Feature Elimination
- ✓ Support Vector Regression – решение задачи регрессии
- ✓ One-class SVM – выявление выбросов в данных
- ✓ Другие методы с применением ядер: kernel Principal Component Analysis (kernel PCA), etc

. . .

Спасибо за внимание!