Support vector machines

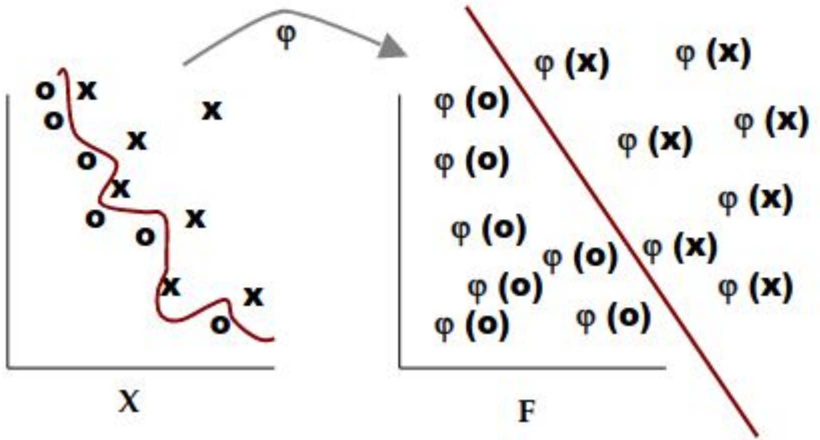
Андрей Рудометов, 23221

Основные идеи

- “kernel functions” - основанное на скалярном произведении обобщение “схожести” для эффективного разделения в нелинейных областях.
- Задача квадратичного программирования для избежания локальных минимумов

Основные идеи

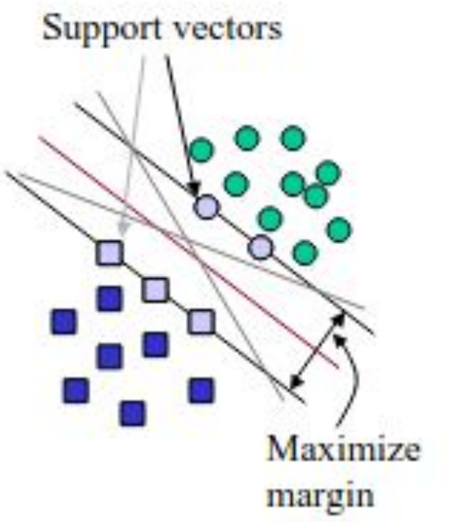
- Поиск гиперплоскости, если линейно разделимо
- Преобразование в другое пространство и поиск там, если нет ("kernel functions")



Опорные векторы

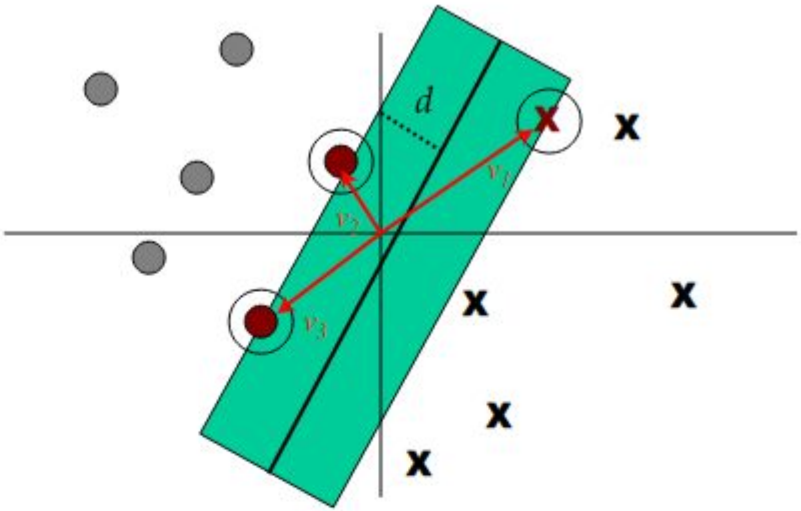
- для разделения достаточно лишь крайних точек, а не всего класса
- хотим максимизировать расстояние от гиперплоскости до ближайшей точки

- в двумерных примерах гиперплоскость - линия, но вообще говоря нет



Опорные векторы

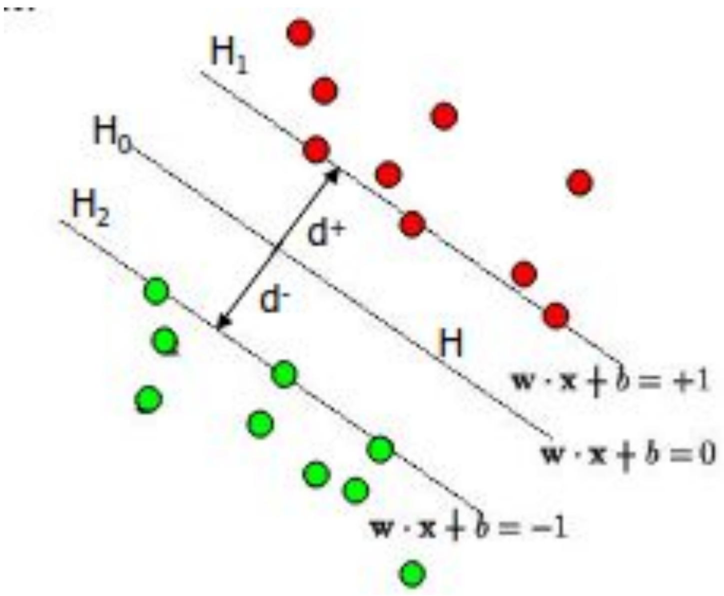
- Почему векторы, а не точки?



Как делить?

- w - нормаль к плоскости
- несложно посчитать, что расстояние H_1 - H_2 это 2/ ||w||
- Значит, надо минимизировать ||w||
- При условии, что между плоскостями нет точек:

$$\left.\begin{array}{l} \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \text{ when } y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \text{ when } y_i = -1 \end{array}\right\}$$



Что решать?

- x - вектор, c - класс (например, -1 или 1)

$$\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_n, c_n)\}$$

- задача с ограничением

$$\begin{cases} \|\mathbf{w}\|^2 \to \min \\ c_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1, \quad 1 \leq i \leq n. \end{cases}$$

- двойственная задача поиска седловой точки функции Лагранжа

$$\begin{cases} \mathbf{L}(\mathbf{w}, \mathbf{b}; \lambda) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \lambda_{\mathrm{i}}(c_i((\mathbf{w} \cdot \mathbf{x_i}) - b) - 1) \to \min_{w,b} \max_{\lambda} \\ \lambda_{\mathrm{i}} \geq 0, \quad 1 \leq i \leq n \end{cases} \quad (2)$$

Как использовать ответ?

- с только двойственными переменными

$$\begin{cases} -\mathbf{L}(\lambda) = -\sum_{i=1}^{n} \lambda_\mathbf{i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_\mathbf{i}\lambda_\mathbf{j} c_i c_j (\mathbf{x_i} \cdot \mathbf{x_j}) \to \min_\lambda \\ \lambda_\mathbf{i} \geq 0, \quad 1 \leq i \leq n \\ \sum_{i=1}^{n} \lambda_\mathbf{i} c_i = 0 \end{cases} \qquad (3)$$

- тогда найдем вектора плоскости как

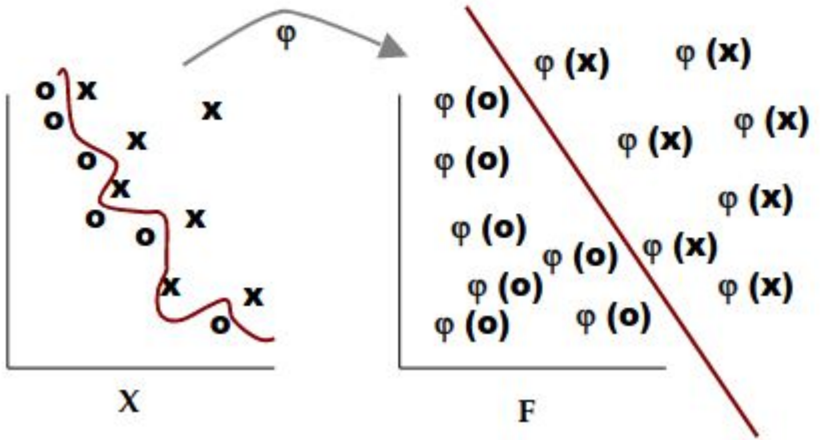
$$\mathbf{w} = \sum_{i=1}^{n} \lambda_\mathbf{i} c_i \mathbf{x_i} \qquad \mathbf{b} = \mathbf{w} \cdot \mathbf{x_i} - c_i, \quad \lambda_i > 0$$

- итоговый алгоритм классификации:

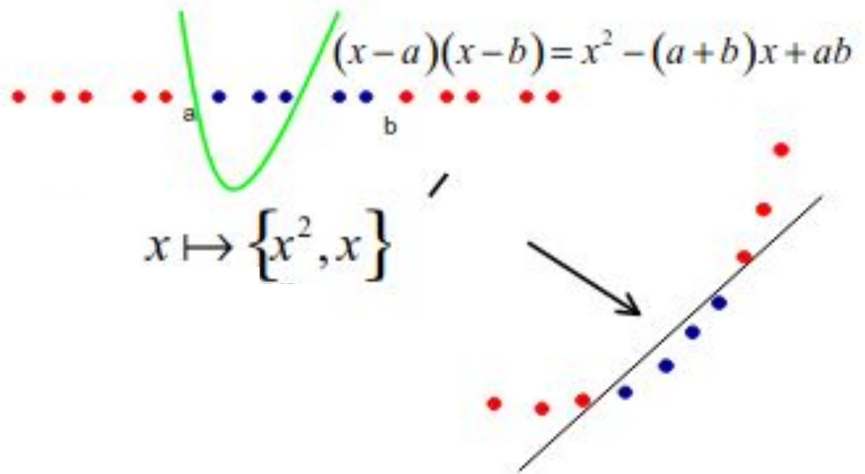
$$a(x) = sign\left(\sum_{i=1}^{n} \lambda_\mathbf{i} c_i \mathbf{x_i} \cdot \mathbf{x} - b\right) \ (4)$$

А что если нелинейно?

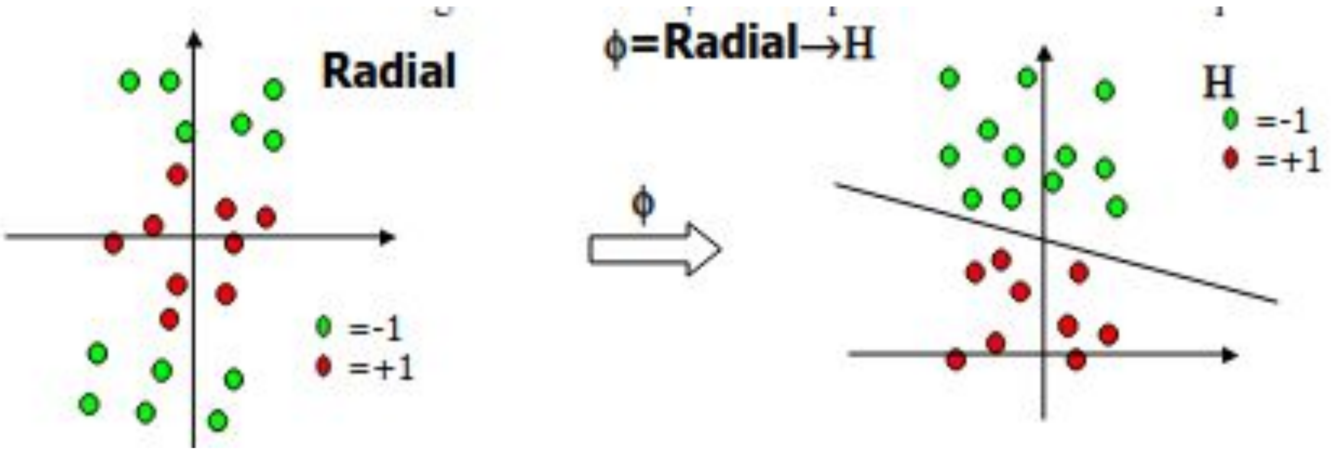
- Поиск гиперплоскости, если линейно разделимо
- Преобразование в другое пространство и поиск там, если нет (“kernel functions”)



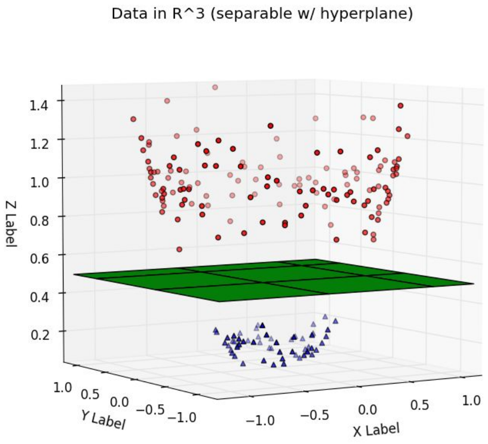
А что если нелинейно?

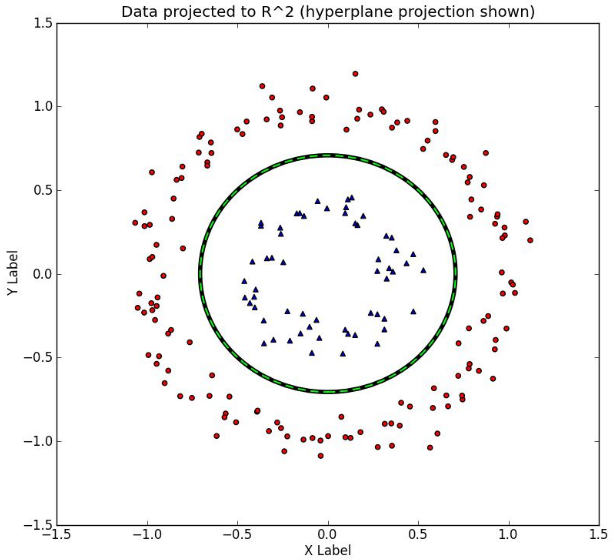


А что если нелинейно?



А что если нелинейно?





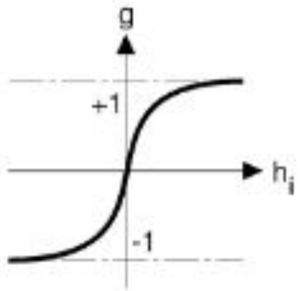
Как еще можно отображать?

- полиномы
- гауссовы функции
- сигмоида

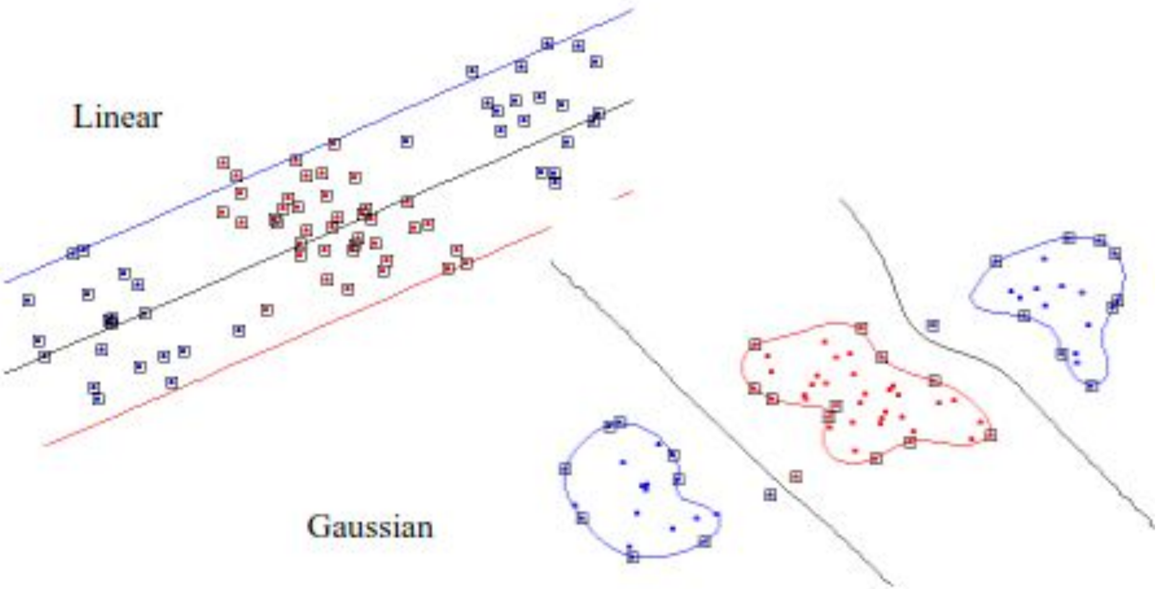
$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^p$$

$$K(\mathbf{x},\mathbf{y}) = \exp\left\{-\|\mathbf{x}-\mathbf{y}\|^2 / 2\sigma^2\right\}$$

$$K(\mathbf{x},\mathbf{y}) = \tanh(\kappa\mathbf{x}\cdot\mathbf{y}-\delta)$$

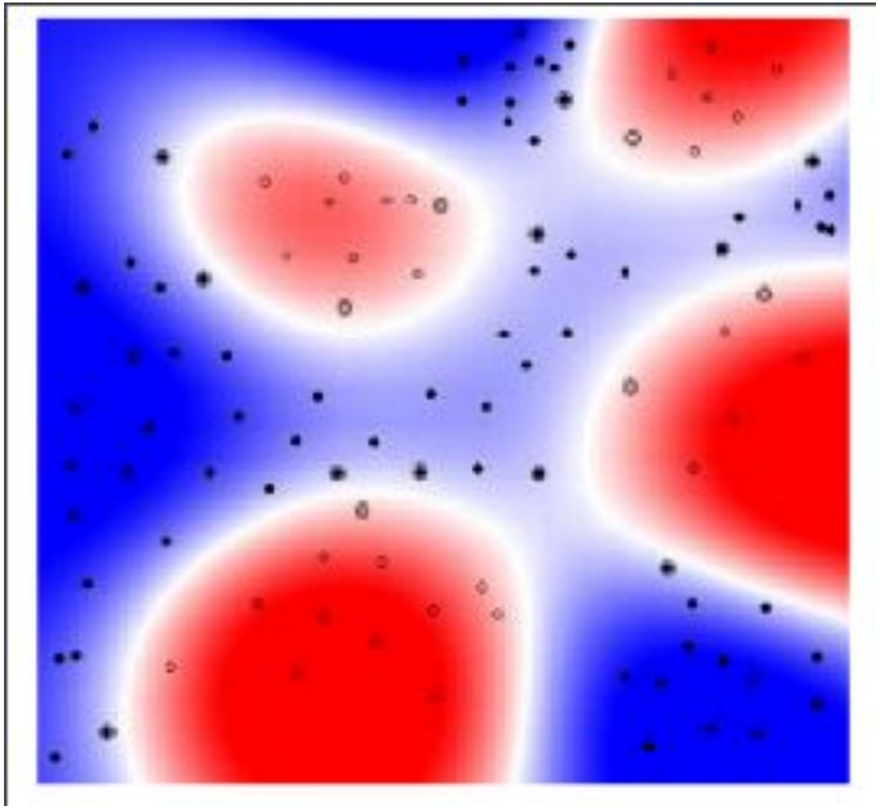


Еще примеры



Еще примеры

- Нелинейная радиально-базисная функция



Преимущества

- при больших размерностях
- R^N > samples
- подмножество samples
- разные определения схожести и комбинации kernel functions для уточнения

Недостатки

- плохо работает при features > samples
- не дает напрямую вероятностей принадлежности классу

Спасибо за внимание