核方法

刘新旺 教授

https://xinwangliu.github.io/

国防科技大学 计算机学院

计算科学系人工智能与大数据教研室

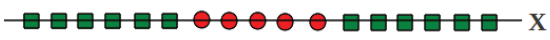
2020 年 10 月 27 日

Content

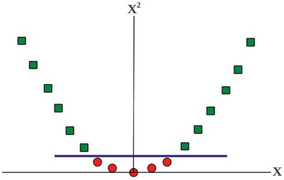
(线性)支持向量机优缺点

1. 优点：算法简单、直观、可理解性强

1. 优点：算法简单、直观、可理解性强
2. 缺点：即便引入松弛变量(即允许违反间距，margin violation)，也没有希望找到一个任何好的线性分类器。
   样例：

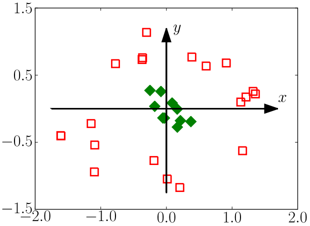


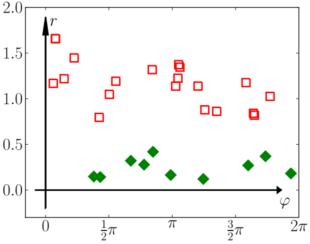
(c) 原始样本

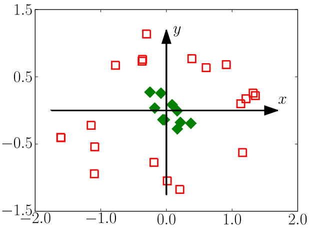


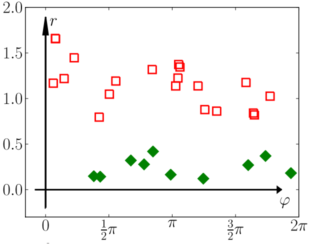
(d) 变换后的样本

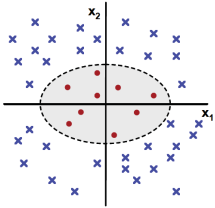
更多样例

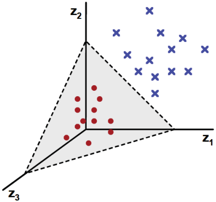


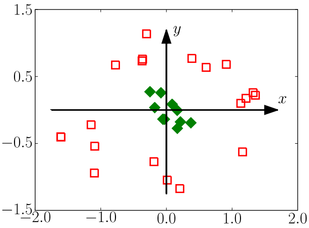


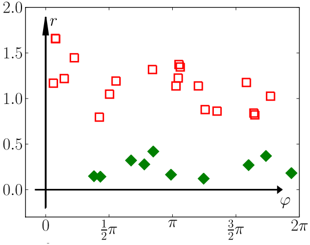

1.5
0.5
−0.5
−1.5
−2.0
−1.0
0.0
1.0
2.0
y
x

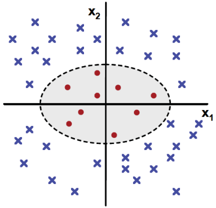

2.0
1.5
1.0
0.5
0.0
r
φ
0
$\frac{1}{2}\pi$
π
$\frac{3}{2}\pi$
2π

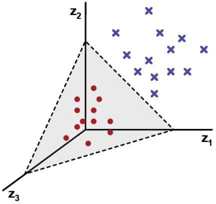

x_2
x_1


z_2
z_1
z_3









在原始空间里，找不到任何超平面将两类样本分开。然而，在某个变换后的空间里，样本变得线性可分。

由现象到通用规律

- 线性模型简单、直观、可理解性强，容易处理
- 表达能力有限

广义的线性分类器

将数据从原始空间变换到某个（高维）空间，然后在变换后的空间使用线性算法。

- 线性模型简单、直观、可理解性强，容易处理
- 表达能力有限

广义的线性分类器

将数据从原始空间变换到某个（高维）空间，然后在变换后的空间使用线性算法。

这就是**核算法**的基本思想。

Content

特征映射

$\psi(\cdot): \mathbf{x} \in \mathcal{X} \mapsto \mathcal{F}$，即把数据由原始空间映射到(高维)空间中的函数，被称为特征映射(feature mapping)。

常见的特征映射(假设$\mathbf{x} = [x_1, x_2]^\top$)包括：

- 极坐标

$$\psi(\cdot): \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} \sqrt{x_1^2 + x_2^2} \\ \arctan \frac{x_1}{x_2} \end{pmatrix} \tag{1}$$

- 二次多项式

$$\psi(\cdot): \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \left(1, x_1, x_2, x_1 x_2, x_1^2, x_2^2\right)^\top \tag{2}$$

使用特征映射后，对应的决策函数变为：

$$\hat{h}'(\mathbf{x}) = \hat{\boldsymbol{\omega}}^\top \psi(\mathbf{x}) + \hat{b} \tag{3}$$

对应的优化问题变为：

$$\begin{aligned} \min_{\hat{\boldsymbol{\omega}}, \hat{b}, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\hat{\boldsymbol{\omega}}\|^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_i\left(\hat{\boldsymbol{\omega}}^\top \psi(\mathbf{x}_i) + \hat{b}\right) \geq 1 - \xi_i,\ \xi_i \geq 0,\ \forall i \end{aligned} \tag{4}$$

- $\psi(\cdot)\ \mathbf{x} \mapsto \mathcal{F}$为特征映射
- 允许训练过程中有误差
- C为正则化参数—调节模型复杂度与训练误差

使用特征映射面临的两大难题

- 如何计算特征映射
  - 计算效率。
- 如何选择合适的特征映射
  - 特征映射可视为原始数据的一种表示，它的选择极大地影响所学习到的分类器的性能。

使用特征映射面临的两大难题

- 如何计算特征映射
  - 计算效率。
- 如何选择合适的特征映射
  - 特征映射可视为原始数据的一种表示，它的选择极大地影响所学习到的分类器的性能。

$\Longrightarrow$核函数的出现很好地解决了上述难题。

核函数$\kappa(\cdot,\cdot): \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$被定义为样本在特征空间$\mathcal{F}$(由特征映射$\psi(\cdot)$ $\mathbf{x} \mapsto \mathcal{F}$确定)中的内积，即

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle = \psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j) \tag{5}$$

$$
\begin{aligned}
\mathcal{L}(\hat{\boldsymbol{\omega}}, \hat{b}, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2}\|\hat{\boldsymbol{\omega}}\|^2 + C\sum_{i=1}^{n}\xi_i \\
&\quad - \sum_{i=1}^{n}\alpha_i\left(y_i\left(\hat{\boldsymbol{\omega}}^\top\psi(\mathbf{x}_i)+\hat{b}\right)-1+\xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i
\end{aligned} \tag{6}
$$

其中$\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^\top$, $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_n]^\top$ 为Lagrange乘子。

KKT条件

$$
\begin{cases}
\nabla_{\hat{\boldsymbol{\omega}}}\mathcal{L} = \hat{\boldsymbol{\omega}} - \sum_{i=1}^{n}\alpha_i y_i\psi(\mathbf{x}_i) = 0 & \Longrightarrow \hat{\boldsymbol{\omega}} = \sum_{i=1}^{n}\alpha_i y_i\psi(\mathbf{x}_i) \\
\nabla_{\hat{b}}\mathcal{L} = -\sum_{i=1}^{n}\alpha_i y_i = 0 & \Longrightarrow \sum_{i=1}^{n}\alpha_i y_i = 0 \\
\nabla_{\xi_i}\mathcal{L} = C - \alpha_i - \beta_i & \Longrightarrow \alpha_i + \beta_i = C \\
\forall i \;\; \alpha_i\left(y_i\left(\hat{\boldsymbol{\omega}}^\top\mathbf{x}_i+\hat{b}\right)-1+\xi_i\right) = 0 & \Longrightarrow \alpha_i \vee y_i\left(\hat{\boldsymbol{\omega}}^\top\mathbf{x}_i+\hat{b}\right) = 1-\xi_i \\
\forall i \;\; \beta_i\xi_i = 0 & \Longrightarrow \beta_i = 0 \vee \xi_i = 0
\end{cases} \tag{7}
$$

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left(\psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j)\right) - \sum_{i=1}^{n}\alpha_i \\
s.t. \quad & \sum\nolimits_{i=1}^{n} \alpha_i y_i = 0,\ C \geq \alpha_i \geq 0,\ \forall i.
\end{aligned} \tag{8}
$$

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j)\right) - \sum_{i=1}^{n}\alpha_i \qquad (8)$$

$$s.t. \ \sum\nolimits_{i=1}^{n}\alpha_i y_i = 0, \ C \geq \alpha_i \geq 0, \ \forall i.$$

即

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i \qquad (9)$$

$$s.t. \ \sum\nolimits_{i=1}^{n}\alpha_i y_i = 0, \ C \geq \alpha_i \geq 0, \ \forall i.$$

- 核函数$\kappa(\mathbf{x}_i,\mathbf{x}_j) = \psi(\mathbf{x}_i)^\top\psi(\mathbf{x}_j)$
- EQ.(9)中的优化变量$\boldsymbol{\alpha} \in \mathbb{R}^n$只与样本个数相关，与样本维数无关!

- 支持向量(Support Vectors): 对应$\alpha_i > 0\,(\forall i)$的样本
- 怎样计算$\hat{b}$?

- 支持向量(Support Vectors): 对应$\alpha_i > 0\ (\forall i)$的样本
- 怎样计算$\hat{b}$?
  任取$(\alpha_i > 0)$对应的样本，计
  算$\hat{b} = y_i - \sum_{j=1}^{n} \alpha_j y_j \big(\psi(\mathbf{x}_j)^\top \psi(\mathbf{x}_i)\big) = y_i - \sum_{j=1}^{n} \alpha_j y_j \kappa(\mathbf{x}_j, \mathbf{x}_i)$

- 支持向量(Support Vectors): 对应$\alpha_i > 0\,(\forall i)$的样本
- 怎样计算$\hat{b}$?
  任取$(\alpha_i > 0)$对应的样本，计
  算$\hat{b} = y_i - \sum_{j=1}^{n} \alpha_j y_j \left(\psi(\mathbf{x}_j)^\top \psi(\mathbf{x}_i)\right) = y_i - \sum_{j=1}^{n} \alpha_j y_j \kappa(\mathbf{x}_j, \mathbf{x}_i)$
- 决策函数：
  $\hat{h}(\mathbf{x}) = \operatorname{sgn}\left(\hat{\boldsymbol{\omega}}^\top \mathbf{x} + \hat{b}\right) = \operatorname{sgn}\left(\sum_{j=1}^{n} \alpha_j y_j \left(\psi(\mathbf{x}_j)^\top \psi(\mathbf{x})\right) + \hat{b}\right) =$
  $\operatorname{sgn}\left(\sum_{j=1}^{n} \alpha_j y_j \kappa(\mathbf{x}_j, \mathbf{x}) + \hat{b}\right)$

下面哪一个论断是正确的?

给定假说集合$\mathcal{H}$和数据集$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$，学习算法$\mathcal{A}$从假说集$\mathcal{H}$中挑选一个假说$h$以使得$h \approx f$，其中$f$是真实（存在但未知）的假说。下面那个论断是正确的?

1. 线性SVM与核SVM的区别在于学习算法不同
2. 线性SVM与核SVM的区别在于假说集不同
3. 线性SVM与核SVM的区别在于优化方法不同
4. 以上论断都不正确

下面哪一个论断是正确的?

给定假说集合$\mathcal{H}$和数据集$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$， 学习算法$\mathcal{A}$从假说集$\mathcal{H}$中挑选一个假说$h$以使得$h \approx f$，其中$f$是真实（存在但未知）的假说。下面那个论断是正确的?

1. 线性SVM与核SVM的区别在于学习算法不同
2. 线性SVM与核SVM的区别在于假说集不同
3. 线性SVM与核SVM的区别在于优化方法不同
4. 以上论断都不正确

Reference Answer: ②

$$\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sgn}[\boldsymbol{\omega}^\top \psi(\mathbf{x}) + b] : \boldsymbol{\omega} \in \mathbb{R}^{|\mathcal{F}|}, b \in \mathbb{R}\}.$$

Content

由于$\kappa(\mathbf{x}_j, \mathbf{x}_i) = \psi(\mathbf{x}_j)^\top \psi(\mathbf{x}_i)$，引入核函数$\kappa(\mathbf{x}_j, \mathbf{x}_i)$并不能改进我们学习算法的性能。但是，核化具有计算效率和灵活性等优势：

计算效率

相对于$\psi(\mathbf{x}_j)^\top \psi(\mathbf{x})$与$\kappa(\mathbf{x}_j, \mathbf{x}_i)$，后者的计算开销要小得多。

以$x \in R^1$和二次多项式特征映射为例。对于每个样本，计算$\psi(\cdot) : x \mapsto (1, \sqrt{2}x, x^2) \in \mathbb{R}^3$需要2个乘法操作，对于整个数据集则需要$2n$个乘法。随后，计算内积需要5个操作（3个乘法和2 个加法），对于整个数据集则需要$\frac{5n}{2}(n+1)$。一共需要$\frac{5n}{2}(n+1) + 2n$ 个操作。

$\kappa(\mathbf{x}_j, \mathbf{x}_i) = \langle \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle = \langle (1, \sqrt{2}x_j, x_j^2)^\top, (1, \sqrt{2}x_i, x_i^2)^\top \rangle = 1 + 2x_ix_j + x_i^2x_j^2 = \left(1 + x_i^\top x_j\right)^2$只需要三个操作（2个乘法1个加法），对于整个数据集则需要$\frac{3n}{2}(n+1)$。可节省40%的操作。

灵活性

可以构造一个核函数，该核函数对应于某个特征映射ψ的内积。无需知道ψ的具体形式，也不需要知道怎么去计算它。

灵活性

可以构造一个核函数，该核函数对应于某个特征映射ψ的内积。无需知道ψ的具体形式，也不需要知道怎么去计算它。

对于任意给定的核函数κ，是否存在某个特征映射ψ使得该核函数等于特征映射的内积？

灵活性

可以构造一个核函数，该核函数对应于某个特征映射ψ的内积。无需知道ψ的具体形式，也不需要知道怎么去计算它。

对于任意给定的核函数κ，是否存在某个特征映射ψ使得该核函数等于特征映射的内积？

Theorem 3.1

Mercer's Condition. 假设$\mathcal{X}$是一个非空集合。对任意**正定核函数**$\kappa(\cdot,\cdot):\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$，总存在一个*Hilbert*空间$\mathcal{H}$和一个特征映射$\psi(\cdot):\mathcal{X}\mapsto\mathcal{H}$，使得$\kappa(\mathbf{x}_i,\mathbf{x}_j)=\langle\psi(\mathbf{x}_i),\psi(\mathbf{x}_j)\rangle_{\mathcal{H}}$，其中$\langle\cdot,\cdot\rangle_{\mathcal{H}}$表示$\mathcal{H}$中的内积。

定义

假设$\mathcal{X}$是一个非空集合。函数$\kappa(\cdot,\cdot):\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$如果满足下述条件

1. κ是对称的，即对所有$\mathbf{x},\mathbf{x}'\in\mathcal{X}$，都有$\kappa(\mathbf{x},\mathbf{x}')=\kappa(\mathbf{x}',\mathbf{x})$
2. 对任意有限个样本$\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n\in\mathcal{X}$，**核矩阵**$K_{ij}=\kappa(\mathbf{x}_i,\mathbf{x}_j)$是半正定的，即对任意$\boldsymbol{\theta}\in\mathbb{R}^n$，有$\boldsymbol{\theta}^\top\mathbf{K}\boldsymbol{\theta}\geq 0$。

则被称之为正定核函数。

由定理3.1，我们可以得到如下结论：

1. 可以将任意函数$\kappa(\cdot,\cdot): \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$应用到支持向量机算法中，只要$\kappa$是正定核函数。最终得到一个对应于高维特征空间的广义线性分类器。
2. 特征映射ψ和Hilbert空间$\mathcal{H}$都是隐含地定义的，即我们知道它们的存在，但不能够、也不需要计算它们。Hilbert空间$\mathcal{H}$通常是高维的，甚至是无限维，但这并不影响我们的优化。因为我们只关注$\mathcal{H}$ 中样本的内积，它们是通过核函数$\kappa(\cdot,\cdot)$来实现。
3. 核算法可以应用到任意输入集$\mathcal{X}$，只要我们能定义该输入集上的核函数。

构造核函数

1. 对任意$\psi : \mathcal{X} \mapsto \mathcal{F}$, $\kappa(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$ 是核函数;
2. 如果$d : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$是一个距离函数，即
   - 对所有$\mathbf{x}, \mathbf{x}' \in \mathcal{X}$，都有$d(\mathbf{x}, \mathbf{x}') \geq 0$;
   - 只有当$\mathbf{x} = \mathbf{x}'$时，才有$d(\mathbf{x}, \mathbf{x}') = 0$;
   - 对所有$\mathbf{x}, \mathbf{x}' \in \mathcal{X}$，都有$d(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}', \mathbf{x})$;
   - 对所有$\mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathcal{X}$，$d(\mathbf{x}, \mathbf{x}') \leq d(\mathbf{x}, \mathbf{x}'') + d(\mathbf{x}'', \mathbf{x}')$

   则对任意$\rho > 0$, $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\rho\, d(\mathbf{x}, \mathbf{x}')\right)$ 为核函数。
3. 如果κ是核函数且$\alpha > 0$，则$\kappa + \alpha$和$\alpha\kappa$都是核函数。
4. 如果κ_1和κ_2是核函数，则$\kappa_1 + \kappa_2$和$\kappa_1 \cdot \kappa_2$也是核函数。

常用的核函数包括：

1. 多项式核: $\kappa(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^m$, $m \in \{1, 2, \cdots\}$
2. 高斯核函数: $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$
3. 核函数的线性组合: $\kappa(\mathbf{x}, \mathbf{x}') = \sum_{p=1}^{m} \gamma_p \kappa_p(\mathbf{x}, \mathbf{x}')$，其中$\gamma_p \geq 0$, κ_p $(\forall p)$都是核函数。

二次多项式核函数的支持向量机演示。

Content

在现实应用中，如何使用核函数面临如下困惑

- 选择什么类型的核函数?
- 选定某个类型的核函数后，如何选择“最优”的核函数

在现实应用中，如何使用核函数面临如下困惑

- 选择什么类型的核函数？
- 选定某个类型的核函数后，如何选择“最优”的核函数

通常的解决办法：

1. 选定某个类型的核函数后，通过交叉验证从一组给定的参数集$\{\sigma_1, \sigma_2, \cdots, \sigma_c\}$中选择一个最优的参数$\sigma$
   - $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$
2. 从数据中学习核函数—-多核学习(Multiple Kernel Learning)
   - $\kappa(\mathbf{x}, \mathbf{x}') = \sum_{p=1}^{m} \gamma_p K_p(\mathbf{x}, \mathbf{x}')$，其中$\{\mathbf{K}_p\}_{p=1}^{m}$为预先给定的一组核矩阵，$\gamma \in \Delta = \{\gamma : \sum_{p=1}^{m} \gamma_p = 1, \gamma_p \geq 0\}$

Primal Problem

The primal optimization problem for SVMs[a] is

$$(\textbf{SVMs-P}): \min_{\boldsymbol{\omega},\, b,\, \boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C\sum_{i=1}^{n}\xi_i, \ s.t. \ y_i\left(\boldsymbol{\omega}^\top\phi(\mathbf{x}_i)+b\right) \geq 1-\xi_i, \ \xi_i \geq 0, \tag{10}$$

where $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is a training set, $\boldsymbol{\omega}$, $b, \boldsymbol{\xi}$ denote the normal vector, bias and slack variables, respectively, and $\phi(\cdot): \mathcal{X} \mapsto \mathcal{F}$ is a feature mapping.

[a] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Dual Problem

$$(\textbf{SVMs-D}): \max_{\boldsymbol{\alpha}} \; -\frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})^\top \mathbf{K} (\boldsymbol{\alpha} \odot \mathbf{y}) + \boldsymbol{\alpha}^\top \mathbf{e}, \; s.t. \; \boldsymbol{\alpha}^\top \mathbf{y} = 0, \; \boldsymbol{\alpha} \succeq \mathbf{0}, \quad (11)$$

where $\boldsymbol{\alpha} \in \mathbb{R}_+^n$ is the Lagrange multipliers, $\mathbf{y} = [y_1, y_2, \cdots, y_n]^\top$, $\odot$ denotes componentwise multiplication, $\mathbf{K} \in \mathbb{R}^{n \times n}$ with $K_{ij} = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ and $\mathbf{e} = [1, 1, \cdots, 1]^\top$.

Optimization for MKL

The optimization problem of MKL is formulated as[a],

$$\begin{aligned} (\textbf{MKL-P}):\ \min_{\{\boldsymbol{\omega}_p\}_{p=1}^m,\, b,\, \boldsymbol{\xi}}\ & \frac{1}{2}\left(\sum_{p=1}^m \|\boldsymbol{\omega}_p\|\right)^2 + C\sum_{i=1}^n \xi_i, \\ s.t.\ \ & y_i\left(\sum_{p=1}^m \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i) + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0, \end{aligned} \tag{12}$$

where $\{\phi_p(\cdot)\}_{p=1}^m$ denote m feature mappings corresponding to m pre-specified base kernels $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$.

[a] Gert R. G. Lanckriet et al. "Learning the Kernel Matrix with Semidefinite Programming". In: *JMLR* 5 (2004), 27–72.

$$(\textbf{MKL-D}): \min_{\boldsymbol{\gamma}} \max_{\boldsymbol{\alpha}} \; -\frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})^\top \left(\sum_{p=1}^{m}\gamma_p\mathbf{K}_p\right)(\boldsymbol{\alpha}\odot\mathbf{y}) + \boldsymbol{\alpha}^\top\mathbf{e}$$
$$s.t.\;\; \boldsymbol{\alpha}^\top\mathbf{y}=0,\; \boldsymbol{\alpha}\succeq\mathbf{0},\; \boldsymbol{\gamma}^\top\mathbf{e}=1,\; \boldsymbol{\gamma}\succeq\mathbf{0}. \tag{13}$$

[1] S. Sonnenburg and G. Rätsch and C. Schäfer and B. Schälkopf. "Large scale multiple kernel learning". In: *JMLR* (2006).

[2] Alain Rakotomamonjy et al. "SimpleMKL". In: *JMLR* 9 (2008), 2491–2521.

[3] Zenglin Xu et al. "An Extended Level Method for Efficient Multiple Kernel Learning". In: *NIPS*. 2008, 1825–1832.

[4] Zenglin Xu et al. "Simple and Efficient Multiple Kernel Learning by Group Lasso". In: *ICML*. 2010, 1175–1182.

$$\text{(MKL-D)}: \min_{\boldsymbol{\gamma}} \max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})^\top \left(\sum_{p=1}^{m}\gamma_p\mathbf{K}_p\right)(\boldsymbol{\alpha}\odot\mathbf{y})+\boldsymbol{\alpha}^\top\mathbf{e} \quad (13)$$
$$s.t. \quad \boldsymbol{\alpha}^\top\mathbf{y}=0,\ \boldsymbol{\alpha}\succeq\mathbf{0},\ \boldsymbol{\gamma}^\top\mathbf{e}=1,\ \boldsymbol{\gamma}\succeq\mathbf{0}.$$

1. Semi-infinite Linear Programming[1];
2. Reduced Sub-gradient Descend[2];
3. Extended Level method[3];
4. Closed-form Solution[4];

[1] S. Sonnenburg and G. Rätsch and C. Schäfer and B. Schälkopf. "Large scale multiple kernel learning". In: *JMLR* (2006).

[2] Alain Rakotomamonjy et al. "SimpleMKL". In: *JMLR* 9 (2008), 2491–2521.

[3] Zenglin Xu et al. "An Extended Level Method for Efficient Multiple Kernel Learning". In: *NIPS*. 2008, 1825–1832.

[4] Zenglin Xu et al. "Simple and Efficient Multiple Kernel Learning by Group Lasso". In: *ICML*. 2010, 1175–1182.

总结

- 特征映射
- 核化
- 使用核函数的经验与技巧

Next: Stochastic Gradient Descent