

基于支持向量机的机器学习策略

机器学习策略简介

什么是机器学习策略？

从广义上来说,机器学习是一种能够赋予机器学习的能力以此让它完成直接编程无法完成的功能的方法.但从实践的意义来说,机器学习是一种通过利用数据,训练出模型,然后使用模型预测的一种方法.而机器学习策略,则是通过机器学习方法,通过输入训练数据与模型来获得对未来数据的价格或者涨跌的预期判断,进而进行投资的量化策略.

支持向量机简介

什么是支持向量机？

支持向量机(Support Vector Machines, SVM) 被Vapnik与他的合作者提出于1995年,基础为统计学习理论和结构风险最小化原则.支持向量机具有完备的理论基础和出色的学习能力,是借助于最优化方法解决有限样本机器学习问题的数据挖掘出色方法之一.

支持向量机的原理是？

假设给定一个特征空间上的训练集:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)\}; x_i \in X = \mathbb{R}^n, y_i \in Y = \{+1, -1\}; i = 1, 2, \dots, N$$

其中 x_i 为第 i 个特征向量, y_i 为 x_i 的类标记,当 $y_i = +1$ 时, x_i 称为正例,当 $y_i = -1$ 时, x_i 称为负例.

线性可分SVM

假设训练数据集线性 T 可分,通过间隔最大化或等价求解相应凸二次规划问题得到分离超平面 $w^* \cdot x + b^* = 0$ 和相应的分类决策函数 $f(x) = \text{sign}(w^* \cdot x + b^*)$ 为线性可分SVM.

为了量化分类的正确性和确信度,引入函数间隔的概念.

对于给定的训练数据集 T 和超平面 (w, b) ,定义超平面关于样本点 (x_i, y_i) 的函数间隔为:

$$\hat{\gamma}_i = y_i(w \cdot x_i + b)$$

定义超平面 (w, b) 关于训练数据集 T 的函数间隔为关于 T 中所有样本点 (x_i, y_i) 的函数间隔的最小值:

$$\hat{\gamma} = \min_{i=1, \dots, N} \hat{\gamma}_i$$

为了取消成比例改变 w, b 导致函数间隔变化但超平面不变的问题,引入规范化 $\|w\|$,此时函数间隔变为几何间隔.

对于给定的训练数据集 T 和超平面 (w, b) ,定义超平面关于样本点 (x_i, y_i) 的几何间隔为:

$$\tilde{\gamma}_i = y_i \left(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right)$$

定义超平面 (w, b) 关于训练数据集 T 的函数间隔为关于 T 中所有样本点 (x_i, y_i) 的几何间隔的最小值:

$$\tilde{\gamma} = \min_{i=1, \dots, N} \tilde{\gamma}_i$$

可知函数间隔和几何间隔的关系为:

$$\gamma_i = \frac{\hat{\gamma}_i}{\|w\|}, \gamma = \frac{\hat{\gamma}}{\|w\|} \quad (1)$$

硬间隔最大化

通过寻找最大化几何间隔的分离超平面可以以充分大的确信度对训练数据进行分类,最大化几何间隔又称为硬间隔最大化,此时的约束最优化问题为:

$$\begin{aligned} & \max_{w,b} \gamma \\ & s.t. y_i \left(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right) \geq \gamma, i = 1, 2, \dots, N \end{aligned}$$

即最大化超平面 (w, b) 关于训练数据集 T 的几何间隔 γ

考虑(1),问题可改写为:

$$\begin{aligned} & \max_{w,b} \frac{\hat{\gamma}}{\|w\|} \\ & s.t. y_i (w \cdot x_i + b) \geq \hat{\gamma}, i = 1, 2, \dots, N \end{aligned}$$

基于与引入规范化 $\|w\|$ 同样的原因, $\hat{\gamma}$ 的取值对结果没有影响,故取 $\hat{\gamma} = 1$,考虑

$$\max_{w,b} \frac{1}{\|w\|} \Leftrightarrow \min_{w,b} \frac{1}{2} \|w\|^2$$

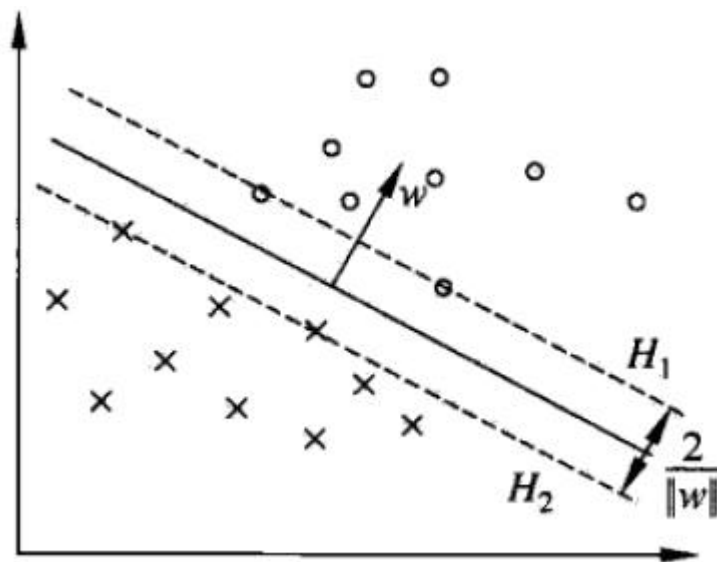
有以下线性可分SVM学习的最优化问题:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (2)$$

$$s.t. y_i (w \cdot x_i + b) - 1 \geq 0, i = 1, 2, \dots, N \quad (2)$$

该问题是一个凸二次规划问题.可以证明,若训练数据集 T 线性可分,那么可将训练数据集中的样本点完全正确分开的最大间隔超平面存在且唯一.

在线性可分的情况下,训练数据集的样本点与分离超平面距离最近的样本点的示例称为支持向量,如下图的 H_1 和 H_2 上的点:



H_1 和 H_2 之间的距离称为间隔,为 $\frac{2}{\|w\|}$, H_1 和 H_2 称为间隔边界.

线性可分SVM对偶学习算法

通过运用拉格朗日对偶性,可以得到原始问题的对偶问题.对偶问题往往更容易求解,也便于引入核函数推广到非线性分类问题.拉格朗日对偶性在此不赘述,但有个重要定理需要特别指出:对于原始问题和对偶问题,在满足特定条件时,则 x^* . α^* . β^* 分别为原始问题和对偶问题的解的充要条件为 x^* . α^* . β^* 满足KKT条件.KKT条件表述如下:

$$\nabla_x L(x^*, \alpha^*, \beta^*) = 0$$

$$\alpha_i^* c_i(x^*) = 0, i = 1, 2, \dots, k \text{ (KKT的对偶互补条件)}$$

$$c_i(x^*) \leq 0, i = 1, 2, \dots, k$$

$$\alpha_i^* \geq 0, i = 1, 2, \dots, k$$

$$h_j(x^*) = 0, j = 1, 2, \dots, l$$

向(2)引入拉格朗日乘子 $\alpha_i \geq 0, i = 1, 2, \dots, N$ 定义拉格朗日函数

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^N \alpha_i \quad (3)$$

其中, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ 为拉格朗日乘子向量.

根据拉格朗日对偶性可以得到原始问题的对偶问题即极大极小问题:

$$\max_{\alpha} \min_{w, b} L(w, b, \alpha)$$

先求 $\min_{w, b} L(w, b, \alpha)$,分别让(3)对 w . b 的偏导等于0,有:

$$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^N \alpha_i x_i y_i = 0$$

$$\nabla_b L(w, b, \alpha) = - \sum_{i=1}^N \alpha_i y_i = 0$$

$$w = \sum_{i=1}^N \alpha_i x_i y_i \quad (4)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (4)$$

把(4)代入(3)整理可得:

$$\min_{w, b} L(w, b, \alpha) = -\frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^N \alpha_i$$

接下来再求 $\max_{\alpha} \min_{w, b} L(w, b, \alpha)$,其对偶问题为:

$$\max_{\alpha} -\frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^N \alpha_i \quad (5)$$

$$s. t. \sum_{i=1}^N \alpha_i y_i = 0 \quad (5)$$

$$\alpha_i \geq 0, i = 1, 2, \dots, N \quad (5)$$

(5)等价于下面的最优化问题:

$$\min_{\alpha} \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i \quad (6)$$

$$s. t. \sum_{i=1}^N \alpha_i y_i = 0 \quad (6)$$

$$\alpha_i \geq 0, i = 1, 2, \dots, N \quad (6)$$

可以证明以下定理:设 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_i^*)^T$ 为对偶优化问题(6)的解,则 $\exists j, \alpha_j^* > 0$ 且原始最优化问题可按式求解 w^*, b^* :

$$w^* = \sum_{i=1}^N \alpha_i^* x_i y_i$$

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j)$$

此时有分离超平面:

$$\sum_{i=1}^N \alpha_i^* y_i (x \cdot x_i) + b^* = 0$$

分类决策函数:

$$f(x) = \text{sign}[\sum_{i=1}^N \alpha_i^* y_i (x \cdot x_i) + b^*]$$

考虑原始最优化问题(2)和对偶最优化问题(6),将数据集中对应 $\alpha_i^* > 0$ 的样本点 (x_i, y_i) 的实例 $x_i \in \mathbb{R}^n$ 称为支持向量.可证明支持向量 x_i 一定在间隔边界上.

线性可分SVM学习算法

构造求解约束最优化问题

$$\min_{\alpha} \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i$$

$$s. t. \sum_{i=1}^N \alpha_i y_i = 0$$

$$\alpha_i \geq 0, i = 1, 2, \dots, N$$

求得最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_i^*)^T$

计算 $w^* = \sum_{i=1}^N \alpha_i^* x_i y_i$ 并选取 $\alpha_j^* > 0$ 计算 $b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j)$.

求得分离超平面: $w^* \cdot x + b^* = 0$

分类决策函数: $f(x) = \text{sign}(w^* \cdot x + b^*)$

软间隔最大化线性SVM

假设训练数据集 T 线性不可分且存在特异点,除特异点以外的数据线性可分.

为了使得线性不可分而不满足(2)中约束条件的训练数据集 T 可被训练,对每个样本点 (x_i, y_i) 引入松弛变量 $\xi \geq 0$,并支付相应的距离代价 ξ_i ,使得(2)变为:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$

$$s. t. y_i (w \cdot x_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N$$

其中 $C > 0$,称为惩罚系数,取值由实际问题决定.这个思路称为软间隔最大化.

此时线性不可分SVM的学习问题变为如下的凸二次规划原始问题:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (7)$$

$$s. t. y_i(w \cdot x_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N \quad (7)$$

$$\xi_i > 0, i = 1, 2, \dots, N \quad (7)$$

可以证明 $\exists(w, b, \xi)$ 为(7)的解,且 w 唯一, b 不唯一且存在于一个区间中.

(7)的对偶问题是:

$$\min_{\alpha} \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i \quad (8)$$

$$s. t. \sum_{i=1}^N \alpha_i y_i = 0 \quad (8)$$

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, N \quad (8)$$

线性SVM学习算法

选择惩罚参数 $C > 0$,构造并求解凸二次规划问题

$$\min_{\alpha} \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i$$

$$s. t. \sum_{i=1}^N \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0, i = 1, 2, \dots, N$$

求得最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_i^*)^T$

计算 $w^* = \sum_{i=1}^N \alpha_i^* x_i y_i$ 并选取 $C > \alpha_j^* > 0$ 计算 $b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j)$.

求得分离超平面: $w^* \cdot x + b^* = 0$

分类决策函数: $f(x) = \text{sign}[\sum_{i=1}^N \alpha_i^* y_i (x \cdot x_i) + b^*]$

非线性SVM

假设训练数据集 T 能用 \mathbb{R}^n 中的一个超曲面将正负例分开,这种分类问题被称为非线性可分问题.

对于这种问题,需要应用SVM的核技巧如下:

通过非线性变换将输入空间(欧式空间 \mathbb{R}^n 或离散集合)对应于一个特征空间(希尔伯特空间 \mathbb{H})

输入空间 \mathbb{R}^n 中的超曲面模型对应于特征空间 \mathbb{H} 中的超平面模型(SVM)

通过求解特征空间 \mathbb{H} 中的线性SVM完成非线性可分分类问题的学习任务.

设 \mathbf{X} 为输入空间(欧式空间 \mathbb{R}^n 的子集或离散集合),又称 \mathbf{X} 为特征空间(希尔伯特空间),若 $\exists \phi(\mathbf{x}) : \mathbf{X} \rightarrow \mathbb{H}$ 使得 $\forall \mathbf{x}, \mathbf{z} \in \mathbf{X}, K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$,则称 $K(\mathbf{x}, \mathbf{z})$ 为核函数, $\phi(\mathbf{x})$ 称为映射函数. $\phi(\mathbf{x}) \cdot \phi(\mathbf{z})$ 称为 $\phi(\mathbf{x})$ 和 $\phi(\mathbf{z})$ 的内积.

用核函数 $K(\mathbf{x}, \mathbf{z})$ 代替对偶问题(8)中的目标函数和分类决策函数的内积,有新的目标函数:

$$W(\alpha) = \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i$$

和新的分类决策函数:

$$f(\mathbf{x}) = \text{sign}[\sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*]$$

若使用多项式核函数 $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z} + 1)^p$,对应的SVM为 p 次多项式分类器,分类决策函数成为:

$$f(\mathbf{x}) = \text{sign}[\sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x} + 1)^p + b^*]$$

若使用高斯核函数 $K(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2})$,对应的SVM为高斯径向基函数(RBF)分类器,分类决策函数成为:

$$f(\mathbf{x}) = \text{sign}[\sum_{i=1}^N \alpha_i^* y_i \exp(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}) + b^*]$$

非线性SVM学习算法

选取适当核函数 $K(\mathbf{x}, \mathbf{z})$ 和适当的惩罚参数 $C > 0$,构造并求解最优化问题

$$\min_{\alpha} \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i \quad (9)$$

$$s.t. \sum_{i=1}^N \alpha_i y_i = 0 \quad (9)$$

$$C \geq \alpha_i \geq 0, i = 1, 2, \dots, N \quad (9)$$

求得最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_i^*)^T$

并选取 $C > \alpha_j^* > 0$ 计算 $b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}_j)$.

构造分类决策函数: $f(\mathbf{x}) = \text{sign}[\sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}, \mathbf{x}_i) + b^*]$

可以证明,当 $K(\mathbf{x}, \mathbf{z})$ 为正定核函数时,(9)为凸二次规划问题并有解.

附录

B. Gram矩阵

定义: n 维欧式空间任意 $k(k \leq n)$ 个向量 $\alpha_1, \alpha_2, \dots, \alpha_k$ 的内积组成的矩阵

$$\Delta(\alpha_1, \alpha_2, \dots, \alpha_k) = \begin{pmatrix} (\alpha_1, \alpha_1) & (\alpha_1, \alpha_2) & \dots & (\alpha_1, \alpha_k) \\ (\alpha_2, \alpha_1) & (\alpha_2, \alpha_2) & \dots & (\alpha_2, \alpha_k) \\ \dots & \dots & \dots & \dots \\ (\alpha_k, \alpha_1) & (\alpha_k, \alpha_2) & \dots & (\alpha_k, \alpha_k) \end{pmatrix}$$

称为 k 个向量 $\alpha_1, \alpha_2, \dots, \alpha_k$ 的Gram矩阵,行列式 $G(\alpha_1, \alpha_2, \dots, \alpha_k) = \Delta(\alpha_1, \alpha_2, \dots, \alpha_k)$ 称为Gram行列式.

定理B-1:欧式空间中向量 $\alpha_1, \alpha_2, \dots, \alpha_k$ 的Gram矩阵 $\Delta(\alpha_1, \alpha_2, \dots, \alpha_k)$ 必为半正定矩阵, $\Delta(\alpha_1, \alpha_2, \dots, \alpha_k)$ 为正定矩阵 $\Leftrightarrow \alpha_1, \alpha_2, \dots, \alpha_k$ 线性无关.

证明:对向量组 $\alpha_1, \alpha_2, \dots, \alpha_k$ 应用Schmidt正交化过程,得到一组正交向量组 $\beta_1, \beta_2, \dots, \beta_k$,把 $\alpha_1, \alpha_2, \dots, \alpha_k$ 用 $\beta_1, \beta_2, \dots, \beta_k$ 表示如下:

$$\alpha_1 = \beta_1 (B-1)$$

$$\alpha_2 = b_{12}\beta_1 + \beta_2 (B-1)$$

$$\dots (B-1)$$

$$\alpha_k = b_{1k}\beta_1 + b_{2k}\beta_2 + \dots + b_{k-1k}\beta_{k-1} + \beta_k (B-1)$$

$$(\alpha_1, \alpha_2, \dots, \alpha_k) = (\beta_1, \beta_2, \dots, \beta_k) \begin{vmatrix} 1 & b_{12} & \dots & b_{1k} \\ 0 & 1 & \dots & b_{2k} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}$$

$$G = (\alpha_1, \alpha_2, \dots, \alpha_k) = \begin{vmatrix} (\alpha_1, \alpha_1) & (\alpha_1, \alpha_2) & \dots & (\alpha_1, \alpha_k) \\ (\alpha_2, \alpha_1) & (\alpha_2, \alpha_2) & \dots & (\alpha_2, \alpha_k) \\ \dots & \dots & \dots & \dots \\ (\alpha_k, \alpha_1) & (\alpha_k, \alpha_2) & \dots & (\alpha_k, \alpha_k) \end{vmatrix} = \begin{vmatrix} \alpha_1^T \\ \alpha_2^T \\ \dots \\ \alpha_k^T \end{vmatrix} (\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_k)$$

$$= \begin{vmatrix} (\beta_1, \beta_2, \dots, \beta_k) \end{vmatrix} \begin{vmatrix} 1 & b_{12} & \dots & b_{1k} \\ 0 & 1 & \dots & b_{2k} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}^T * (\beta_1, \beta_2, \dots, \beta_k) \begin{vmatrix} 1 & b_{12} & \dots & b_{1k} \\ 0 & 1 & \dots & b_{2k} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}$$

$$= \begin{vmatrix} 1 & b_{12} & \dots & b_{1k} \\ 0 & 1 & \dots & b_{2k} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}^T * \begin{vmatrix} \beta_1^T \\ \beta_2^T \\ \dots \\ \beta_k^T \end{vmatrix} (\beta_1 \quad \beta_2 \quad \dots \quad \beta_k) * \begin{vmatrix} 1 & b_{12} & \dots & b_{1k} \\ 0 & 1 & \dots & b_{2k} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}$$

$$= \begin{vmatrix} \begin{pmatrix} \beta_1^T \\ \beta_2^T \\ \dots \\ \beta_k^T \end{pmatrix} (\beta_1 \quad \beta_2 \quad \dots \quad \beta_k) \end{vmatrix} = G(\beta_1, \beta_2, \dots, \beta_k)$$

由 $\beta_1, \beta_2, \dots, \beta_k$ 的正交性可得 $G(\alpha_1, \alpha_2, \dots, \alpha_k) = G$

$$(\beta_1, \beta_2, \dots, \beta_k) = \begin{vmatrix} (\beta_1, \beta_1) & & & \\ & (\beta_2, \beta_2) & & \\ & & \dots & \\ & & & (\beta_k, \beta_k) \end{vmatrix} = \prod_{i=1}^k |\beta_i|^2 \geq 0$$

因为 $\Delta(\alpha_1, \alpha_2, \dots, \alpha_k)$ 的 k 阶主子阵也是Gram矩阵,所以行列式的值大于等于零,因此 Δ 也是半正定矩阵.

$\forall k, G(\alpha_1, \alpha_2, \dots, \alpha_k) \geq 0$,对于线性相关向量组 $\alpha_1, \alpha_2, \dots, \alpha_k$,必有一向量是其余向量的线性组合,反之亦真.设 α_k 是 $\alpha_1, \alpha_2, \dots, \alpha_{k-1}$ 的线性组合,有:

$$\alpha_k = \sum_{i=1}^{k-1} \alpha_i \alpha_j$$

把 $G(\alpha_1, \alpha_2, \dots, \alpha_k)$ 的第 k 列减去第 i 列的 α_i 倍($i = 1, 2, \dots, k-1$),由内积的线性性质有 $(\alpha_i, 0) = 0 (i = 1, 2, \dots, k)$,故有:

$$G(\alpha_1, \alpha_2, \dots, \alpha_k) = \begin{vmatrix} (\alpha_1, \alpha_1) & (\alpha_1, \alpha_2) & \dots & (\alpha_1, 0) \\ (\alpha_2, \alpha_1) & (\alpha_2, \alpha_2) & \dots & (\alpha_2, 0) \\ \dots & \dots & \dots & \dots \\ (\alpha_k, \alpha_1) & (\alpha_k, \alpha_2) & \dots & (\alpha_k, 0) \end{vmatrix} = 0$$

综上可得, $\Delta(\alpha_1, \alpha_2, \dots, \alpha_k)$ 为正定矩阵 $\Leftrightarrow \alpha_1, \alpha_2, \dots, \alpha_k$ 线性无关.

证毕.

机器学习策略实现（基于掘金量化平台）

策略思想

- 选取7个特征变量:1.收盘价/均值2.现量/均量3.最高价/均价4.最低价/均价5.现量6.区间收益率7.区间标准差
- 用时间窗口为15个交易日的历史数据训练出预测未来5个交易日收益率为上涨或下跌的二分类SVM模型
- 在对行情数据预测为上涨的时候买入,并采取止盈止损操作

策略主要步骤实现

订阅数据

```
subscribe(symbols=context.symbol, frequency='60s')
```

订阅数据需要调用 `subscribe` 函数.

- `symbols` 设置订阅的标的代码.
- `frequency` 设置周期级别,这里设置 `60s` 表示以六十秒为周期.

获取历史数据

```
recent_data = history(symbol=context.symbol, frequency='1d',
                      start_time=start_date, end_time=end_date,
                      fill_missing='Last', df=True)
```

获取历史数据需要调用 `history` 函数.

- `symbols` 需要设置获取数据的标的代码.
- `frequency` 设置周期级别,这里设置 `1d` 表示以一天为周期.
- `start_time` 设置获取数据的开始时间.
- `end_time` 设置获取数据的结束时间.
- `fill_missing` 设置补全数据的方式,这里设置 `Last` 表示用上一个收盘价填充缺失值.
- `df` 设置是否返回DataFrame类型数据,这里设置 `True` 表示返回该类型的数据.

设置SVM的参数并训练

```
context.clf = svm.SVC(C=1.0, kernel='rbf', gamma='auto', shrinking=True, tol=0.001,
                     cache_size=200, verbose=False, max_iter=-1, random_state=None)
context.clf.fit(x_train, y_train)
```

- `C` 设置误差项的惩罚参数.

- `kernel` 设置算法中使用的核函数,这里设置 `rbf` 表示用RBF核函数.
- `gamma` 设置核的函数系数,这里设置 `auto` 表示设置系数为1/特征数.
- `shrinking` 设置是否使用启发式搜索算法,这里设置 `True` 表示为使用.
- `tol` 设置停止训练的精度.
- `cache_size` 设置训练需要的内存.
- `verbose` 设置是否输出详细信息,此处设置 `False` 表示不输出.
- `max_iter` 设置最大的迭代次数,此处设置 `-1` 表示不限制.
- `random_state` 设置随机数种子使得结果可复现,此处设置 `None` 表示随机生成.

获取当前年月日

```
today = bar.bob.strftime('%Y-%m-%d')
```

返回指定日期的上一个交易日

```
last_day = get_previous_trading_date(exchange='SHSE', date=today)
```

- `exchange` 设置交易所,此处设置 `SHSE` 为上交所.
- `date` 设置作为参考的时间点.

获取持有的仓位

```
position = context.account().position(symbol=context.symbol, side=PositionSide_Long)
```

- `symbol` 设置标的.
- `side` 设置仓位方向,此处设置为 `PositionSide_Long` 为多仓.

获取预测的结果

```
prediction = context.clf.predict(features)[0]
```

获取持有的仓位

```
position = context.account().position(symbol=context.symbol, side=PositionSide_Long)
```

- `symbol` 设置标的.
- `side` 设置仓位方向,此处设置为 `PositionSide_Long` 为多仓.

调仓到目标持仓比例

```
order_target_percent(symbol=context.symbol, percent=0.95, order_type=OrderType_Market, position_side=PositionSide_Long)
```

- `symbol` 设置标的.
- `percent` 设置目标比例.
- `order_type` 设置订单类型,此处设置为 `OrderType_Market` 为市价单.
- `side` 设置仓位方向,此处设置为 `PositionSide_Long` 为多仓.

平当前所有可平持仓

```
order_close_all()
```

策略回测分析

回测报告



分析

我们选取了SHESE.60009上海机场2016-04-01到2017-07-30的数据作为训练集,2017年8月1日至2017年9月5日作为回测周期,应用上述的SVM机器学习模型作为买卖的基础.可以看出：

- **胜率**（具有盈利的平仓次数与总平仓次数之比）达到了100%,也即策略在回测期间的每次开仓都是盈利的.
- **卡玛比率**（年化收益率与历史最大回撤之比）是使用**最大回撤率**来衡量风险.采用**最大回撤率**来衡量风险,关注的是**最极端的情况**.卡玛比率越高表示策略承受每单位最大损失获得的报酬越高.在这里卡玛比率达到了19.57.
- **夏普比率**（年化收益率减无风险收益率的差与收益波动率之比）也即每承受一单位的总风险,会产生3.27单位的超额回报.
- **策略收益曲线**总体曲线向上,体现了机器学习在量化投资中的巨大潜力.