$\|A - X^*\|_F = \min_{X \in M} \|A - X\|_F$

$A = U\Sigma V^T = (U_1, U_2)\begin{bmatrix}\Sigma_1 & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^T \\ V_2^T\end{bmatrix}$, $\Sigma_1 = diag(\sigma_1, \sigma_2, \cdots, \sigma_r)$

gdiplus

BitmapData

DrawBitmap

ProjectPoint

DrawTrackingValues

表示自动垃圾回收

PixelFormat32bppRgb

32位,Rgb

8位留用

1024

1101

40075

mon8341

Pt

$$\text{rotmat} = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & 1-2(x^2+z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & 1-2(x^2+y^2) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

CODE OF

$e_n = (y - NNet(x_n))^2 = (y - s_1^{(L)})^2 = \left(y_n - \sum_{i=0}^{d^{(L-1)}} w_{i1}^{(L)} x_i^{(L-1)}\right)^2$

Last Layer $\dfrac{\partial e_n}{\partial w_{i1}^{(L)}} = 2(y - s_1^{(L)}) \cdot (-x_i^{(L-1)})$ $\delta_1^{(L)} = -2(y_n - s_1^{(L)})$

$$\frac{\partial e_n}{\partial w_{ij}^{(l)}} = \frac{\partial e_n}{\partial s_j^{(l)}} \cdot \frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}} = \boxed{\delta_j^{(l)}} \cdot (x_i^{(l-1)})$$

$$\delta_j^{(l)} = \frac{\partial e_n}{\partial s_j^{(l)}} = \sum_{k=1}^{d^{(l+1)}} \frac{\partial e_n}{\partial s_k^{(l+1)}} \cdot \frac{\partial s_k^{(l+1)}}{\partial x_j^{(l)}} \cdot \frac{\partial x_j^{(l)}}{\partial s_j^{(l)}}$$

$$= \sum_k (\delta_k^{(l+1)})(w_{jk}^{(l+1)})(\tanh'(s_j^{(l)}))$$

$$\delta^{(1)} \cdots\cdots \delta^{(l)} \Longleftarrow \delta^{(l+1)}$$

Back propagation Algroithm (BP Algroithm)

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta x_i^{(l-1)} \delta_j^{(l)}$$

regulation

$$\sum \frac{(w_{ij}^{(l)})^2}{1 + (w_{ij}^{(l)})^2}$$

初始小权重，随机小权重.

Soft-Margin　C越大越复杂，容易overfit

primal: $\min_{b,w,\varepsilon} \frac{1}{2}w^Tw + C\sum_{n=1}^{N}\varepsilon_n$, st $y_n(w^Tz_n+b) \geq 1-\varepsilon_n$, $\varepsilon_n \geq 0$

$$L(b,w,\varepsilon,\alpha,\beta) = \frac{1}{2}w^Tw + C\sum_{n=1}^{N}\varepsilon_n + \sum_{n=1}^{N}\alpha_n(1-\varepsilon_n-y_n(w^Tz_n+b)) + \sum_{n=1}^{N}\beta(-\varepsilon_n)$$

dual: $\max_{\alpha_n\geq 0,\beta_n\geq 0}\left(\min_{b,w,\varepsilon}\left(L(b,w,\varepsilon,\alpha,\beta)\right)\right)$

$$\frac{\partial L}{\partial \varepsilon_n} = C-\alpha_n-\beta_n = 0 \Rightarrow \beta_n = C-\alpha_n ,\quad 0\leq\alpha_n\leq C$$

$$\Rightarrow \max_{0\leq\alpha_n\leq C,\ \beta_n=C-\alpha_n}\left(\min_{b,w}\left(\frac{1}{2}w^Tw+\sum_{n=1}^{N}\alpha_n(1-y_n(w^Tz_n+b))\right)\right)$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{n=1}^{N}\alpha_n y_n z_n$$

α_n有上限C

$$\left.\begin{array}{l}\alpha_n(1-\varepsilon_n-y_n(w^Tz_n+b))=0 \\ (C-\alpha_n)\underline{\underline{\varepsilon_n}}=0\end{array}\right\} \Rightarrow SV(\alpha_s>0)$$

$b = y_s - y_s\boxed{\varepsilon_s} - w^Tz_s$

在边界上的点

如何计算ε_s

SV中$\alpha_s>0$，找到 free SV 即 $\alpha_s<C$

则此时$\varepsilon_s=0$，则 $b=y_s-w^Tz_s$

$$b = y_s - \sum_{SV\ indices\ n}\alpha_n y_n K(x_n,x_s)$$

non SV: $\alpha_n=0$, $\xi_n=0$ 在边界外

free SV: $0<\alpha_n<C$, $\xi_n=0$ 在边界上, 用以确定 b

bounded SV: $\alpha_n=C$, $\xi_n=1-y_n(w^Tz_n+b)$, violation amount 违反

即 在边界内的(容忍点)

Tip:

SV 数目少一点比较好. $E_{loocv} \le \frac{|SV|}{N}$

$\begin{cases} \text{lib linear} \\ \text{libsvm} \end{cases}$

autoencode

denoising autoencode
artificial noise/hint as regularization.

linear autoencode

exclude x_0

constrain $w_{ij}^{(1)} = w_{ji}^{(2)} = w_{ij}$: regularization $W=[w_{ij}]$ of size $d\times\tilde{d}$

assume $\tilde{d}<d$: ensure non-trivial solution.

$$h_k(x) = \sum_{j=0}^{\tilde{d}} w_{kj}\left(\sum_{i=1}^{d} w_{ij} x_i\right) = WW^T x$$

$\mathrm{rank}(WW^T) \le \tilde{d}$
$\Downarrow$
$\mathrm{rank}(\Gamma) \le \tilde{d}$

$$E_{in}(h) = E_{in}(W) = \frac{1}{N}\sum_{n=1}^{N} \|x_n - WW^T x_n\|^2$$

eigen-decompose $WW^T = V\Gamma V^T$, V is orthogonal $\Rightarrow VV^T = V^T V = I$

Γ is diagonal $\le \tilde{d}$ non-zero

$\therefore WW^T x_n = V\Gamma V^T x_n$ 将x_n中$d-\tilde{d}$维去除(为0)

V^T: rotate or reflect

Γ: set $d-\tilde{d}$ components to 0

V: back-rotate

$$\min_{V}\min_{\Gamma} \frac{1}{N}\sum_{n=1}^{N} \|VIV^T x_n - V\Gamma V^T x_n\|^2$$

want $I-\Gamma$ many 0

Γ max

$$\Rightarrow \min_{V}\min_{\Gamma} \sum_{n=1}^{N} \|(I-\Gamma)V^T x_n\|^2$$

$\Gamma = \begin{bmatrix} I_{\tilde{d}} & 0 \\ 0 & 0 \end{bmatrix}$

$$\Rightarrow \min_{V} \sum_{n=1}^{N} \left\| \begin{bmatrix} 0 & 0 \\ 0 & I_{d-\tilde{d}} \end{bmatrix} V^T x_n \right\|^2$$

$$\Rightarrow \max_{V} \sum_{n=1}^{N} \left\| \begin{bmatrix} I_{\tilde{d}} & 0 \\ 0 & 0 \end{bmatrix} V^T x_n \right\|^2 \Rightarrow \max_{V} \sum_{n=1}^{N} V^T x_n x_n^T V$$

subject to $VV^T = 1$

autoencode

denoising autoencode
artificial noise/hint as regularization.

linear autoencode

exclude x_0

constrain $w_{ij}^{(1)} = w_{ji}^{(2)} = w_{ij}$: regularization $W = [w_{ij}]$ of size $d \times \tilde{d}$

assume $\tilde{d} < d$: ensure non-trivial solution.

$$h_k(x) = \sum_{j=0}^{\tilde{d}} w_{kj} \left(\sum_{i=1}^{d} w_{ij} x_i \right) = WW^T x$$

$\text{rank}(WW^T) \le \tilde{d}$
$\Downarrow$
$\text{rank}(\Gamma) \le \tilde{d}$

$$E_{in}(h) = E_{in}(W) = \frac{1}{N} \sum_{n=1}^{N} \| x_n - WW^T x_n \|^2$$

eigen-decompose $WW^T = V \Gamma V^T$, V is orthogonal $\Rightarrow VV^T = V^T V = I$

$\therefore WW^T x_n = V \Gamma V^T x_n$ $\Rightarrow$ 将x_n中$d-\tilde{d}$维去除(为0) Γ is diagonal $\le \tilde{d}$ non-zero

V^T : rotate or reflect

Γ : set $d-\tilde{d}$ components to 0

V : back-rotate

$$\min_V \min_\Gamma \frac{1}{N} \sum_{n=1}^{N} \| V I V^T x_n - V \Gamma V^T x_n \|^2$$

$\Rightarrow$ want $I - \Gamma$ many 0

$$\Rightarrow \min_V \left[\min_\Gamma \sum_{n=1}^{N} \| (I - \Gamma) V^T x_n \|^2 \right]$$

Γ max $\tilde{d}$ 1

$\Gamma = \begin{bmatrix} I_{\tilde{d}} & 0 \\ 0 & 0 \end{bmatrix}$

$$\Rightarrow \min_V \sum_{n=1}^{N} \left\| \begin{bmatrix} 0 & 0 \\ 0 & I_{d-\tilde{d}} \end{bmatrix} V^T x_n \right\|^2$$

$$\Rightarrow \max_V \sum_{n=1}^{N} \left\| \begin{bmatrix} I_{\tilde{d}} & 0 \\ 0 & 0 \end{bmatrix} V^T x_n \right\|^2 \Rightarrow \max_V \sum_{n=1}^{N} V^T x_n x_n^T V$$

subject to $VV^T = 1$

$$L(v, x, \lambda) = \sum_{n=1}^{N} v^T x_n x_n^T v - \lambda (v^T v - 1)$$

$$\frac{\partial L}{\partial v} = \sum_{n=1}^{N} v^T x_n x_n^T - \lambda v^T = 0 \Rightarrow \sum_{n=1}^{N} x_n x_n^T v = \lambda v$$

选出 topmost ~~的~~ eigenvectors, at most $\tilde{d}$

Linear Autoencoder or PCA

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n, \quad x_n = x_n - \bar{x}$$

calculate $\tilde{d}$ top eigenvectors $w_1, w_2, \cdots w_{\tilde{d}}$ of $X^T X$

return feature transform $\Phi(x) = W(x - \bar{x})$

生成线性子空间 spanned subspace

$W = \mathrm{span}\{x_1, x_2, \cdots, x_n\} = \{k_1x_1 + k_2x_2 + \cdots + k_nx_n \mid k_i \in F\}$

为 $x_1, x_2, \cdots, x_n$ 向量组的线性包

定理：$B \in R_r^{m\times r}$，$C \in R_r^{r\times n}$；B^TB 与 CC^T 都非奇异；即 B^TB, CC^T 可逆

以B为例：$(B^TB)^{-1}B^TB = I$，则 $B_L^{-1} = (B^TB)^{-1}B^T$

$A \in R^{m\times n}$，$G \in R^{n\times m}$

满足 $AGA = A$，$GAG = G$，$(AG)^T = AG$，$(GA)^T = GA$ 时，G为A的 Moore-Penrose 广义逆 A^+. $\boxed{\text{存在且唯一}}$

$\mathrm{rank}(A^+) = \mathrm{rank}(AA^+) = \mathrm{rank}(A^+A) = \mathrm{rank}(A)$

$x_1, x_2, \cdots x_n$ 正交 $\|x_1 + x_2 + \cdots x_n\|^2 = \|x_1\|^2 + \|x_2\|^2 + \cdots + \|x_n\|^2$

酉空间为复线性空间

Schmidt 正交化

$q_1 = \frac{q_1'}{r_{11}}$，$r_{11} = \|q_1'\|$，其中 $q_1' = x_1$　　复习 PPT 2.4 28~30

$q_k = \frac{q_k'}{r_{kk}}$，$r_{kk} = \|q_k'\|$，$r_{ik} = (x_k, q_i)$，$q_k' = x_k - r_{1k}q_1 - r_{2k}q_2 - \cdots - r_{k-1,k}q_{k-1}$

正交变换

$(T(x), T(y)) = (x, y)$，T是V上正交变换

$\forall x \in V, (T(x), T(x)) = (x, x)$

标准正交基 $\xrightarrow{T}$ 标准正交基

$A^TA = AA^T = I$，$A^T = A^{-1}$，$A = [e_1, e_2, \cdots e_n]$

生成线性子空间 spanned subspace

$W = \mathrm{span}\{x_1, x_2, \cdots, x_n\} = \{k_1x_1 + k_2x_2 + \cdots + k_nx_n \mid k_i \in F\}$

为 $x_1, x_2, \cdots, x_n$ 向量组的线性包

定理: $B \in R_r^{m\times r}$, $C \in R_r^{r\times n}$; B^TB与CC^T都非奇异；即B^TB, CC^T可逆

以B为例 $(B^TB)^{-1}B^TB = I$, 则 $B_L^{-1} = (B^TB)^{-1}B^T$

$A \in R^{m\times n}, G \in R^{n\times m}$

满足 $AGA = A$, $GAG = G$, $(AG)^T = AG$, $(GA)^T = GA$ 时, G为A的

Moore-Penrose广义逆 A^+. $\boxed{\text{存在且唯一}}$

$\mathrm{rank}(A^+) = \mathrm{rank}(AA^+) = \mathrm{rank}(A^+A) = \mathrm{rank}(A)$

$x_1, x_2, \cdots x_n$ 正交 $\|x_1 + x_2 + \cdots x_n\|^2 = \|x_1\|^2 + \|x_2\|^2 + \cdots + \|x_n\|^2$

酉空间为复线性空间

Schmidt正交化

$q_1 = \frac{q_1'}{r_{11}}$, $r_{11} = \|q_1'\|$, 其中 $q_1' = x_1$ 课件PPT 2(4) 28~30

$q_k = \frac{q_k'}{r_{kk}}$, $r_{kk} = \|q_k'\|$, $r_{ik} = (x_k, q_i)$, $q_k' = x_k - r_{1k}q_1 - r_{2k}q_2 \cdots - r_{k-1,k}q_{k-1}$

正交变换

$(T(x), T(y)) = (x, y)$, T是V上正交变换

$\forall x \in V, (T(x), T(x)) = (x, x)$

标准正交基 $\xrightarrow{T}$ 标准正交基

$A^TA = AA^T = I$, $A^T = A^{-1}$, $A = \begin{bmatrix} (\varepsilon_1,\varepsilon_1) & \cdots \\ \vdots & (\varepsilon_n,\varepsilon_n) \end{bmatrix}$

EnableStream (TYPE, H, W, fps)

PXCSenseManager senseMgr = session -> CreateSenseManger();

PXCSession session

PXCSenseManager

senseMgr

PXCCaptureManager

Init()

EnableTracker()

QueryTracker()

captureMgr

FilterByDeviceInfo(device, 0, 0).

QueryDevice()

PXCTracker::

PTracker

PXCTracker

PTracker

SetCameraParameters(g_calibrationFile)

createProject

配置senseMgr

captureMgr

PTracker

Init() 初始化

PXCProject

PTracker { Set2DTrackFromFile

Set3DTrack

Set3DInstantTrack (   )

Set2DTrackFromFile(model_filename, &cosID) 引用

\- private

\# protected

\+ public

Intel China Research Center Ltd.
8F, Raycom Infotech Park A
No.2 Kexueyuan South Road,
Zhong Guan Cun, Haidian District,
Beijing China100190

英特尔（中国）研究中心有限公司
北京市海淀区中关村科学院南路2号
融科资讯中心A座8层
电话/Tel：(86-10) 8261-1515
传真/Fax：(86-10) 8286-1400

PXCTracker::

TrackingValues trackData.

pTracker→QueryTrackingValues(iter→cosID, trackData)

查询追踪值

SenseMgr→QueryTrackerSample()

PXCCapture::Sample

PXCImage

color
depth
ir
left
right
reserved [STREAM_limit]

cosIDs 3D 有多个点

vector<Model>

将文件名作为目标压入 g_targets

object_tracker.cpp

object_tracker.h

PXCMSenseManager AcquireFrame { true / false aligned or not

EnableStream (TYPE, H, W, fps)

define some Handler function

PXCMSenseManage. Handler handler = new ——

PXCMSenseManage sm ;

sm. init (handler);

Intel China Research Center Ltd.
8F, Raycom Infotech Park A
No.2 Kexueyuan South Road,
Zhong Guan Cun, Haidian District,
Beijing China100190

英特尔（中国）研究中心有限公司
北京市海淀区中关村科学院南路2号
融科资讯中心A座8层
电话/Tel：(86-10) 8261-1515
传真/Fax：(86-10) 8286-1400

算图像P的高斯差

尺度空间函数

PXCMSession

ImplDesc 描述信息.

ImageInfo

PXCMImage

CreateImage(PXCMImage.ImageInfo info).

CreateImage(ImageInfo info, ImageData data).

ImageData.

PixelForment. 枚举类型.

Int32[] pitches

IntPtr[] planes

Intel China Research Center Ltd.
8F, Raycom Infotech Park A
No.2 Kexueyuan South Road,
Zhong Guan Cun, Haidian District,
Beijing China100190

英特尔（中国）研究中心有限公司
北京市海淀区中关村科学院南路2号
融科资讯中心A座8层
电话/Tel：(86-10) 8261-1515
传真/Fax：(86-10) 8286-1400

计算图像P的高斯差

$$D(x,y,\sigma) = \frac{(g(x,y,k\sigma) - g(x,y,\sigma)) * P}{k\sigma - \sigma} = \underbrace{L(x,y,k\sigma)}_{\text{尺度空间函数}} - L(x,y,k)$$

$$M_{SIFT}(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta_{SIFT}(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right)$$

Intel China Research Center Ltd.
8F, Raycom Infotech Park A
No.2 Kexueyuan South Road,
Zhong Guan Cun, Haidian District,
Beijing China100190

英特尔（中国）研究中心有限公司
北京市海淀区中关村科学院南路2号
融科资讯中心A座8层
电话/Tel：(86-10) 8261-1515
传真/Fax：(86-10) 8286-1400

A为$m\times n$实矩阵, $\mathrm{rank}(A)=r$, A^TA与AA^T有相同非零特征值 $\sigma_1^2, \sigma_2^2, \cdots, \sigma_r^2$

$$V^TA^TAV=\begin{bmatrix}\Sigma_r^2 & 0\\ 0 & 0\end{bmatrix},\quad 其中\ \Sigma_r=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_r)$$

$$\Downarrow$$

$$\begin{bmatrix}V_1^T\\ V_2^T\end{bmatrix}\qquad V_1^TA^TAV_1=\Sigma_r^2 \Rightarrow \Sigma_r^{-1}V_1^TA^TAV_1\Sigma_r^{-1}=I_r$$

$$U_1=AV_1\Sigma_r^{-1},\ \Rightarrow U_1^TU_1=I_r \qquad U正交.$$

$$U^TAV=\begin{bmatrix}\Sigma_r & 0\\ 0 & 0\end{bmatrix}$$

动态链接

Linux: .so　Win: .dll

gcc -shared xx.c -o libxx.o

dlopen(　) 动态加载

dlsym(　)

dlclose(　)

库名 xxx，文件 libxxx.so

可重定位：.o

可执行：可运行

共享文件：static / dynamic

readelf -h xxx

获取 ELF 头部

gcc -L 指定头文件路径

-l 库文件

(默认找 LD_LIBRARY_PATH)

0X 8048000 程序加载

0X 8048300 载入地址

int main(int argc, char *argv[], char *envp[])

shell ⇒ execve

ELF: Executable and Linkable Format

可执行与可链接格式

ABI: Application Binary Interface

应用程序二进制接口

动态链接

Linux：.so　Win：.dll

gcc -shared xx.c -o libxx.o

dlopen(　) 动态加载

dlsym(　)

dlclose(　)

库名 xxx，文件 libxxx.so

可重定位：.o

可执行：可运行

共享文件：static / dynamic

---

readelf -h xxx

获取ELF头部

---

gcc -L 指定头文件所在路径

-l 库文件

(默认找 LD_LIBRARY_PATH)

0X 8048000 程序加载的地址

0X 8048300

int main(int argc, char *argv[], char *envp[])

Shell ⟹ execve

int execve(const char *filename …)

ELF：Executable and Linkable Format

可执行与可链接格式

ABI：Application Binary Interface

应用程序二进制接口

动态链接

Linux: .so   Win: .dll

gcc -shared xx.c -o libxx.o

dlopen( ) 动态加载

dlsym( )

dlclose( )

库名 xxx，文件 libxxx.so

可重定位：.o

可执行：可运行.

共享文件：static / dynamic

readelf -h xxx

获取 ELF 头部

gcc -L 指定头文件所在目录

-l 库文件

(默认 LD_LIBRARY_PATH)

0x 8048000 程序加载的地址

0x 8048300

int main(int argc, char *argv[], char *envp[])

shell ⇒ execve

int execve(const char *filename ...)

ELF: Executable and Linkable Format

可执行与可链接格式

ABI: Application Binary Interface

应用程序二进制接口

$L(b,w,\alpha) = \frac{1}{2}w^Tw + \sum_{n=1}^{N} \alpha_n(1-y_n(w^Tz_n+b))$

$$\begin{cases} \frac{\partial L}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n = 0 \Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0 \\ \frac{\partial L}{\partial w_i} = w_i - \sum_{n=1}^{N} \alpha_n y_n z_{n,i} = 0 \Rightarrow w = \sum_{n=1}^{N} \alpha_n y_n z_n \end{cases}$$

$\Rightarrow \max_{\alpha \geq 0} \left(\min_{b,w} L(b,w,\alpha) \right)$

$\Rightarrow \max_{\alpha\geq 0,\ \sum_{n=1}^{N} \alpha_n y_n = 0,\ w = \sum_{n=1}^{N} \alpha_n y_n z_n} \left(-\frac{1}{2}w^2 + \sum_{n=1}^{N} \alpha_n \right)$

$\min_{\alpha} \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m z_n^T z_m - \sum_{n=1}^{N} \alpha_n$

subject to $\sum_{n=1}^{N} y_n \alpha_n = 0$, $\alpha_n \geq 0$

[Standard QP]

QP(Q, p, A, c)

$\min_{\alpha} \frac{1}{2}\alpha^T Q \alpha + p^T \alpha$ subject to $a_i^T \alpha \geq c_i$

$q_{n,m} = y_n y_m z_n^T z_m$

$\begin{cases} a_{\geq} = y,\ a_{\leq} = -y \\ a_n^T = \text{unit direction} \\ c_{\geq} = 0,\ c_{\leq} = 0 \end{cases}$

$p = [-1]_N$

KKT Optimality Conditions

primal feasible: $y_n(w^Tz_n+b) \geq 1$

dual feasible: $\alpha_n \geq 0$

dual-inner optimal: $\begin{cases} \sum y_n \alpha_n = 0 \\ w = \sum \alpha_n y_n z_n \end{cases}$

primal-inner optimal: $\alpha_n(1-y_n(w^Tz_n+b)) = 0$

Karush-Kuhn-Tucker

$\min_{\alpha} \max_{b,w} \geq \max_{b,w} \min_{\alpha}$

dual problem

strong duality "="

$\begin{cases} \text{convex primal} \\ \text{feasible primal (separable)} \\ \text{linear constraints} \end{cases}$

$w = \sum \alpha_n y_n z_n$

$b = y_n - w^T z_n$, when $\alpha_n > 0$

$\alpha_n \Rightarrow z_n$ is on the flat band

SV ($\alpha > 0$)

$\Rightarrow w = \sum \alpha_n y_n z_n = \sum_{SV} \alpha_n y_n z_n$

$b = y_n - w^T z_n$, any SV (z_n, y_n)

$$L(b, w, \alpha) = \frac{1}{2} w^T w + \sum_{n=1}^{N} \alpha_n (1 - y_n (w^T z_n + b))$$

$$\begin{cases} \frac{\partial L}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n = 0 \Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0 \\ \frac{\partial L}{\partial w_i} = w_i - \sum_{n=1}^{N} \alpha_n y_n z_{n,i} = 0 \Rightarrow w = \sum_{n=1}^{N} \alpha_n y_n z_n \end{cases}$$

$$\Rightarrow \max_{\alpha \geq 0} \left(\min_{b, w} L(b, w, \alpha) \right)$$

$$\Rightarrow \max_{\alpha \geq 0, \sum_{n=1}^{N} \alpha_n y_n = 0, w = \sum_{n=1}^{N} \alpha_n y_n z_n} \left(-\frac{1}{2} w^2 + \sum_{n=1}^{N} \alpha_n \right)$$

$$\min_{\alpha} \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m z_n^T z_m - \sum_{n=1}^{N} \alpha_n$$

subject to $\sum_{n=1}^{N} y_n \alpha_n = 0$, $\alpha_n \geq 0$

[Standard QP]

$QP(Q, p, A, c)$

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha + p^T \alpha$$

subject to $a_i^T \alpha \geq c_i$

$q_{n,m} = y_n y_m z_n^T z_m$

$p = [-1]_N$

$$\begin{cases} a_{\geq} = y, \; a_{\leq} = -y \\ a_n^T = \text{unit direction} \\ c_{\geq} = 0, \; c_{\leq} = 0 \end{cases}$$

KKT Optimality Conditions

primal feasible: $y_n(w^T z_n + b) \geq 1$

dual feasible: $\alpha_n \geq 0$

dual-inner optimal: $\begin{cases} \sum y_n \alpha_n = 0 \\ w = \sum \alpha_n y_n z_n \end{cases}$

primal-inner optimal: $\alpha_n(1 - y_n(w^T z_n + b)) = 0$

Karush-Kuhn-Tucker

$$\min_{\alpha} \max_{b, w} \geq \max_{b, w} \min_{\alpha}$$

dual problem

strong duality "="

$$\begin{cases} \text{convex primal} \\ \text{feasible primal (separable)} \\ \text{linear constraints} \end{cases}$$

$w = \sum \alpha_n y_n z_n$

$b = y_n - w^T z_n$, when $\alpha_n > 0$

$\alpha_n \Rightarrow z_n$ is on the flat boun

SV ($\alpha > 0$)

$\Rightarrow w = \sum \alpha_n y_n z_n = \sum_{SV} \alpha_n y_n z_n$

$b = y_n - w^T z_n$, any SV (z_n, y_n)

SimHash

$d(w_1, w_2, w_3 \cdots\cdots w_n)^{1\times n}$

n维的特征向量

每维随机有一个f维的向量

即 $h(w_i) = (h_1, h_2 \cdots, h_n)$

值为1或-1

表征时可用0,1表示,即0减1

$$H = \begin{bmatrix} h(w_1) \\ h(w_2) \\ \vdots \\ h(w_n) \end{bmatrix}^{n\times f} \qquad d\times H \xRightarrow{\text{二值化}} \text{fingerprint}^{1\times f}$$

$d\times H$ 二值化后得到签名.

fingerprint之间的Hamming距离便可作为相似度.

64位SimHash 在3以内可认为相似度较高.

Stacking

$$f(x) = \sum_{j}^{L} \omega_j \hat{f}_j(x), \quad \omega_j \geq 0, \sum \omega_j = 1$$

$$\min_{\omega} \left[\text{VALIDATION ERROR}\left(\sum_{j}^{L} \omega_j \hat{f}_j \right) \right]$$