二 次 规 划 算 法

在人脸表情动画的研究中，大部分工作都是通过采集每一时刻的面部运动数据，并求出该数据在表情基中的线性组合。而这个计算问题是一个典型的二次规划问题，如下面的式子所示。

对于人脸动画计算权重的问题则为：



s.t．

其中为需要我们求解的凸组合系数。

上式转化为一个二次规划问题如下：

，



由于是输入数据，为一个常数，对整个式子求最小值不构成影响。去掉常数项之后的式子为：



s.t．

与二次规划问题具有相同的形式。

通过上述问题求出的结果（即每个表情基对应的权重）作用与各个表情基上就能实现逼真的表情动画了。而求解二次规划的方法有很多，下面重点介绍有效集方法的理论并进行相应的代码实现。

一般的二次规划的形式如下：



s.t． 



其中为阶对称矩阵。

记下面给出上述二次规划的一个最优性充要条件。

定理1：是二次规划问题的局部极小点当且仅当

1. 存在，使得



（2）记



对任意的

容易发现，公式（1）是凸二次规划的充分必要条件是是半正定的。因此当为半正定的时候，定理1的第二部分也自然成立。因此凸优化问题的局部极小点也是全局极小点。

定理2：是凸二次规划问题的全局极小点的充要条件是满足条件，即存在，使得



下面要用有效集方法来解二次规划，首先需要确定处的有效集，并证明在有效集中，原始问题与如下的等式约束二次规划问题的解是一致的，具体证明参考文献（倪勤，最优化方法与程序设计。北京：科学出版社 2009.）

定理3,设是一般凸二次规划问题的全局极小点，并且在的有效集为则也是下列等式约束凸二次规划：



s.t．

的全局极小点。

从定理3可以看出，有效集方法的最大难点是事先不知道有效集，因此，只有想办法构造一个集合序列去逼近它，即，从开始点出发，计算有效集，解对应的等式约束子问题，重复这一过程，直到，以获得原问题的最优解。

**有效集的算法步骤：**

1. 选取初值，给定初始可行点
2. 解子问题，确定相应的有效集，求解子问题。



s.t．

可以得到极小点和拉格朗日乘子向量。若转步骤3；否则转步骤2。（等式约束的二次规划的求解在后面介绍...）

1. 检验终止准则。计算拉格朗日乘子



其中







令 

若则是全局极小点，停止计算；否则，若则令，转步骤1。

1. 确定步长，令

其中 

令

1. 若则令否则，若则令：其中满足
2. 令，转步骤1。

下面是matlab实现的有效集解法：

function [x,lamk,exitflag,output]=qpact(H,c,Ae,be,Ai,bi,x0)

% 功能: 用有效集方法解一般约束二次规划问题:

% min f(x)=0.5\*x'\*H\*x+c'\*x,

% s.t. a'\_i\*x-b\_i=0,(i=1,...,l),

% a'\_i\*x-b\_i>=0,(i=l+1,...,m)

%输入: x0是初始点, H, c分别是目标函数二次型矩阵和向量；

% Ae=(a\_1,...,a\_l)', be=(b\_1,...,b\_l)';

% Ai=(a\_{l+1},...,a\_m), bi=(b\_{l+1},...,b\_m)'.

%输出: x是最优解， lambda是对应的乘子向量；output是结构

% 变量, 输出极小值f(x), 迭代次数k等信息, exitflag是算法终止类型

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 初始化

%epsilon 是一个很小的数，用于辅助不等式判断，

%如在进行 >= 的判断时，往往是 x>=b+epsilon，err也是一个类似的量

epsilon=1.0e-9; err=1.0e-6;

k=0; x=x0; %k为迭代次数，x0为初始点

n=length(x); kmax=1.0e3; %kmax最大迭代次数

ne=length(be); ni=length(bi); lamk=zeros(ne+ni,1);

index=ones(ni,1);

for (i=1:ni)

if(Ai(i,:)\*x>bi(i)+epsilon), index(i)=0; end

end

%算法主程序

while (k<=kmax)

%求解子问题

Aee=[ ];

if(ne>0), Aee=Ae; end

for(j=1:ni) %不等式约束的个数

if(index(j)>0), Aee=[Aee; Ai(j,:)]; end %将不等式约束和等式约束的稀疏矩阵合并

end

% min f(x)=0.5\*x'\*H\*x+c'\*x,

% s.t. a'\_i\*x-b\_i=0,(i=1,...,l，l+1,...,m),

% 将不等式约束改为等式约束求解子问题。

gk=H\*x+c;

[m1,n1] = size(Aee);

[dk,lamk]=qsubp(H,gk,Aee,zeros(m1,1)); %计算出极小点dk和拉格朗日乘子向量lamk

if(norm(dk)<=err) %dk为0的时候转入第二步

y=0.0;

if(length(lamk)>ne)

[y,jk]=min(lamk(ne+1:length(lamk))); %确定有效集中lamda的最小元素。

end

if(y>=0)

exitflag=0; %如果每个lamda都大于零，这dk为全局极小点，

else %否则减去lamda对应的有效集元素，形成新的有效集

exitflag=1;

for(i=1:ni)

if(index(i) & (ne+sum(index(1:i)))==jk) %如果lamda对应的有效集位置为jk，且索引为1，则将索引置0

index(i)=0; break; %确保在之后的计算中，不在计算当前不等式约束。

end

end

end

k=k+1;

else %如果dk不等于0，转入第三步

exitflag=1; %确定步长alpha

%求步长

alpha=1.0; tm=1.0;

for(i=1:ni)

if((index(i)==0)&(Ai(i,:)\*dk<0))

tm1=(bi(i)-Ai(i,:)\*x)/(Ai(i,:)\*dk); %alpha的计算见第三步的具体公式

if(tm1<tm)

tm=tm1; ti=i;

end

end

end

alpha=min(alpha,tm); %选取最小的alpha，一般为边界点对应的alpha

x = x+alpha\*dk; %确定新的x位置。

%修正有效集

if(tm<1), index(ti)=1; end

end

if(exitflag==0), break; end

k=k+1;

end

output.fval=0.5\*x'\*H\*x+c'\*x;

output.iter=k;

%%%%%%%% 求解子问题 %%%%%%%%%%%%%%% 求法见附A

function [x,lambda]=qsubp(H,c,Ae,be) %求解子问题即，求解一个等式约束的二次规划，

ginvH=pinv(H);

[m,n]=size(Ae);

if (m>0)

rb = Ae\*ginvH\*c + be;

lambda = pinv(Ae\*ginvH\*Ae')\*rb;

x = ginvH\*(Ae'\*lambda-c);

else

x = -ginvH\*c;

lambda = zeros(m,1);

end

附A

等式约束的二次规划求解



s.t．

使用拉格朗日乘子法解这个问题，可以写成如下形式：



令： 

得到方程组：





写成矩阵的形式如下：



记 

由恒等式关系



可得到

 

 

得出









即：求出相应的拉格朗日乘子和未知数。