Tangent Cones and Normal Cones in RCDD

Charles J. Geyer

February 27, 2008

1 Introduction

The *tangent cone* of a convex set C at a point $x \in C$ is given by

$$T_C(x) = \operatorname{cl}\{\, s(y - x) : y \in C \text{ and } s \geq 0 \,\}$$

(Rockafellar and Wets, 2004, Theorem 6.9). Note that the closure is necessary (consider a closed ball). It is not necessary to consider arbitrary sequences; as always in convex analysis it is enough to consider line segments. The *normal cone* of a convex set C at a point $x \in C$ is given by

$$\begin{aligned} N_C(x) &= T_C(x)^* \\ &= \{\, w : \langle w, y - x \rangle \leq 0,\ y \in C \,\} \end{aligned}$$

(Rockafellar and Wets, 2004, Theorem 6.9).

Specializing these to the polyhedral convex set

$$C = \{\, x : \langle a_i, x \rangle \leq b_i,\ i \in I, \text{ and } \langle a_i, x \rangle = b_i,\ i \in E \,\}$$

we have

$$T_C(x) = \{\, x : \langle a_i, x \rangle \leq 0,\ i \in A, \text{ and } \langle a_i, x \rangle = 0,\ i \in E \,\}$$

where

$$A = \{\, i \in I : \langle a_i, x \rangle = b_i \,\}$$

is the active set of inequality constraints (in essence, we drop the inactive constraints and make x the new origin). We also have

$$N_C(x) = \operatorname{pos}\bigl(\{\, a_i : i \in A \cup E \,\} \cup \{\, -a_i : i \in E \,\}\bigr)$$

In the language of the `rcdd` package, we have an H-representation for the tangent cone and a V-representation of the normal cone.

2 Example: Cube

Let us consider a very simple example, the unit cube in three dimensions. Let us start with a V-representation for it.

```
> v <- rbind(c(0, 0, 0), c(0, 0, 1), c(0, 1, 0), c(0,
+     1, 1), c(1, 0, 0), c(1, 0, 1), c(1, 1, 0), c(1,
+     1, 1))
> m <- cbind(0, 1, v)
> m
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    1    0    0    0
[2,]    0    1    0    0    1
[3,]    0    1    0    1    0
[4,]    0    1    0    1    1
[5,]    0    1    1    0    0
[6,]    0    1    1    0    1
[7,]    0    1    1    1    0
[8,]    0    1    1    1    1
```

Then we convert to the H-representation.

```
> library(rcdd)
> out <- scdd(m, representation = "V")
> out$output
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0    1
[2,]    0    0    1    0    0
[3,]    0    0    0    1    0
[4,]    0    1    0    0   -1
[5,]    0    1    0   -1    0
[6,]    0    1   -1    0    0
attr(,"representation")
[1] "H"
```

A cube has six facets, and we have one constraint for each.

2.1 At a Vertex

Now suppose we want to calculate the tangent cone at $(1, 1, 1)$. We need to calculate the active set.

```
> l <- out$output[, 1]
> b <- out$output[, 2]
> v <- out$output[, -c(1, 2)]
> active <- v %*% c(1, 1, 1) + b == 0
> active
```

```
      [,1]
[1,] FALSE
[2,] FALSE
[3,] FALSE
[4,]  TRUE
[5,]  TRUE
[6,]  TRUE
```

Hence the tangent cone at $(1, 1, 1)$ has the following H-representation.

```
> tan111 <- out$output[active, ]
> tan111[, 2] <- 0
> tan111
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0   -1
[2,]    0    0    0   -1    0
[3,]    0    0   -1    0    0
```

The cone represented is the negative orthant.

To turn this into the V-representation of the normal cone, we must obtain the vectors a_i, which are the rows of A where the H-representation has the form $\begin{pmatrix} l & b & -A \end{pmatrix}$. Since we also want $(0, 0)$ for the first two columns of the V-representation (to represent rays) we can just negate the H-representation

```
> nor111 <- -tan111
> nor111 <- redundant(nor111, representation = "V")
> nor111
```

```
$output
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0    1
[2,]    0    0    0    1    0
[3,]    0    0    1    0    0
attr(,"representation")
```

```
[1] "V"

$new.position
[1] 1 2 3
```

(We ran through the `redundant` function just in case.)

2.2 On an Edge

Now let us consider $(0, 0, 1/2)$. We need to calculate the active set.

```
> l <- out$output[, 1]
> b <- out$output[, 2]
> v <- out$output[, -c(1, 2)]
> active <- v %*% c(0, 0, 1/2) + b == 0
> active
```

```
      [,1]
[1,] FALSE
[2,]  TRUE
[3,]  TRUE
[4,] FALSE
[5,] FALSE
[6,] FALSE
```

Hence the tangent cone at $(0, 0, 1/2)$ has the following H-representation.

```
> tan005 <- out$output[active, ]
> tan005[, 2] <- 0
> tan005
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    1    0    0
[2,]    0    0    0    1    0
```

As with the preceding example, the V-representation of the normal cone is just the negative of the H-representation of the tangent cone.

3 Example: Square

In the preceding example, the polyhedron and all of its tangent cones have full dimension. The normal cones may or may not have full dimension (the first example at a vertex does, the second example on an edge doesn't).

Here we consider an example, the unit square in three dimensions, where the polyhedron and its tangent cones do not have full dimension Let us start with a V-representation for it.

```
> v <- rbind(c(0, 0, 0), c(0, 0, 1), c(0, 1, 0), c(0,
+     1, 1))
> m <- cbind(0, 1, v)
> m
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    1    0    0    0
[2,]    0    1    0    0    1
[3,]    0    1    0    1    0
[4,]    0    1    0    1    1
```

Then we convert to the H-representation.

```
> library(rcdd)
> out <- scdd(m, representation = "V")
> out$output
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0    1
[2,]    0    0    0    1    0
[3,]    0    1    0    0   -1
[4,]    0    1    0   -1    0
[5,]    1    0    1    0    0
attr(,"representation")
[1] "H"
```

A cube has six facets, and we have one constraint for each.

3.1 At a Vertex

Now suppose we want to calculate the tangent cone at $(0, 1, 1)$. We need to calculate the active set.

```
> l <- out$output[, 1]
> b <- out$output[, 2]
> v <- out$output[, -c(1, 2)]
> active <- v %*% c(0, 1, 1) + b == 0
> active
```

```
      [,1]
[1,] FALSE
[2,] FALSE
[3,]  TRUE
[4,]  TRUE
[5,]  TRUE
```

Hence the tangent cone at $(0, 1, 1)$ has the following H-representation.

```
> tan011 <- out$output[active, ]
> tan011[, 2] <- 0
> tan011
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0   -1
[2,]    0    0    0   -1    0
[3,]    1    0    1    0    0
```

Note that the code we used is exactly the same as in the other case. Our code is perfectly general (seems to work for all cases). The only difference is that this time our H-representation for the tangent cone has an equality constraint; hence the tangent cone is contained in a hyperplane; hence it is less than full dimension.

To turn this into the V-representation of the normal cone, we must obtain the vectors a_i, which are the rows of A where the H-representation has the form $\begin{pmatrix} l & b & -A \end{pmatrix}$. Since we also want $(0, 0)$ for the first two columns of the V-representation (to represent rays) we can just negate the H-representation

```
> nor011 <- -tan011
> nor011[, 1] <- tan011[, 1]
> nor011 <- redundant(nor011, representation = "V")
> nor011
```

```
$output
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0   -1    0    0
[2,]    0    0    0    0    1
[3,]    0    0    0    1    0
attr(,"representation")
[1] "V"

$new.position
[1] 2 3 1
```

Oops! Our code was not perfectly general here. We should only change the sign of columns 2 on in the H-representation.

The normal cone is full dimensional.

References

Rockafellar, R. T. and Wets, R. J.-B. (2004). *Variational Analysis*, corr. 2nd printing. Berlin: Springer-Verlag.