Well, there's always RTFM :-)

"Ignoring" values is simple. Simple start your levels at a value greater than the one you want to ignore. Case in point:

n=101 ;number of grid pts

;define the grid
x = findgen(n)/(n-1) & y = findgen(n)/(n-1)
xGrid = x # REPLICATE(1., n)
yGrid = REPLICATE(1., n) # y

z = SQRT(xGrid^2 + yGrid^2) ;make some data

levels = [0., 0.25, 0.5, 0.75, 1.]
;levels = [0.25, 0.5, 0.75, 1.] ;uncomment this out to "ignore" < 0.25
nLevels = N_ELEMENTS(levels)

cgLOADCT, 33, RGB_TABLE=rgb ;get a palette, but don't load the table

colorInd = cgSCALEVECTOR(LINDGEN(nLevels), 0, 255) ;select some colors from the paleete
rgbNew = rgb[colorInd, *] ;get *just* these colors

cgContour, z, x, y, LEVELS=levels, PALETTE=rgbNew, /CELL_FILL ;plot with only the colors you want

tNames = [STRCOMPRESS(levels, /REMOVE), ' '] ;get some ticknames

;plot the colorbar, with your palette
cgCOLORBAR, PALETTE=rgbNew, Divisions = nLevels, /FIT, TICKNAMES=tnames, /DISCRETE

Notice I don't actually load any color tables. There's no need.

Using the palette as I do above allows you to be a little more explicit with the colors you send to (cg)Contour and cgColorBar. I prefer to do these sort of plots in this manner. I also rarely need a continuous color bar - my work is usually much better suited for discrete cb's - so this allows me to ensure the colors match in the contour and cb.