Subject: Re: slow processing of my k-nearest neighour code
Posted by James Kuyper on Tue, 15 Aug 2006 15:14:14 GMT
View Forum Message <> Reply to Message

humphreymurray@gmail.com wrote:
> Wow, that's a great idea to remove most of the code from the loops.
> The only problem is that the code doesn't run on data of any usable
> size. I get a "Unable to allocate memory: to make array" error on this
> line of code:
>
> training_duplicates = REBIN(TRANSPOSE(training_data), $
>   num_attributes, num_training_elements, num_testing_elements)
>
> This code works fine with really small data, but when I'm trying to
> classify a 256x256 pixel image, it is trying to create an array of
> dimensions: [15, 400, 65536]. According to my math, this would be
> about 390 million elements, and assuming that each element takes 1 byte
> of memory, it would use 390mb of ram. My machine at uni only has 512mb
> of ram, so I will try this code at home tonight, where I have 1gb of
> ram.
>
> Would the way to fix this problem be to split the number of training
> pixels up, and process them in small groups? For example, analyse a
> row of pixels at a time?

Yes, one of the key limitations of this technique is that it uses extra memory to achieve faster processing speeds (at least, it should process faster: I'd recommend doing some performance testing with your actual data, to make sure). Breaking up the full test data set into smaller sub-sets is exactly the right way to deal with this problem. I would have mentioned that if I'd realized that you were working with arrays that were big enough for that to be a problem.