Script for Ndaona demo

Dinoj Surendran
Department of Computer Science
University of Chicago
dinoj@cs.uchicago.edu

This demo requires Matlab (at least 6.5) running Windows[1], then you can download Ndaona from

http://people.cs.uchicago.edu/ dinoj/ndaona/

That page also has several 3d models made by Ndaona, documentation, and a couple of movies made using the 3d models. Each demo consists of some data in a Matlab .mat file, commands to call Ndaona from Matlab, and a zip file with the files created by Ndaona for Partiview and the Partiview binaries.

To run an unzipped demo, the user just needs to click on a .bat file on Windows[2]. This brings up a 3d model that the user can spin, etc. The key is moving the mouse with the left or right mouse button pressed — the Ndaona page has more details.

The demo proposed for HLT-NAACL 2006 just uses a couple of the available demos, both showing visualizations of classification results.

1 Demo 1 : Dialog Act Classification

Installation consists of just adding the location of ndaona.m to your Matlab path.

Download the file http://people.cs.uchicago.edu/∼dinoj/ndaona/downloads/damaptask_mat.zip . This unzips to a file `damaptask.mat`

```
>> load damaptask
>> whos
  Name               Size                    Bytes  Class

  L              14810x1                   118480  double array
  ProbEst        14810x13                 1540240  double array
  RawText        14810x1                  1964842  cell array
  daclassnames       1x13                     940  cell array
  SomePros       14810x7                   829360  double array
  SomeProsNames      1x7                      514  cell array
```

This data has 14810 (test) examples of dialog acts (DAs) from the MapTask corpus. Each DA is tagged with one of 13 tags, and was classified by a support vector machine (SVM) that produced probability outputs[3].

[1] It often runs on Unix-based systems too, more on this later.

[2] Linux users need to make the *.csh file executable before running it (there have been problems reported with the Linux binary, however, in which case the user will have to compile Partiview using the source code available on the Partiview website. OS X users can try clicking on the corresponding *.command file, but this does not always work.

[3] Specifically, it was a combination of six binary support vector machine with a linear kernel. Each SVM output was Platt-

The names of the j-th class, $j = 1, \ldots, 13$, is `daclassnamesj` while the i-th DA has manual transcription `RawTexti` e.g.

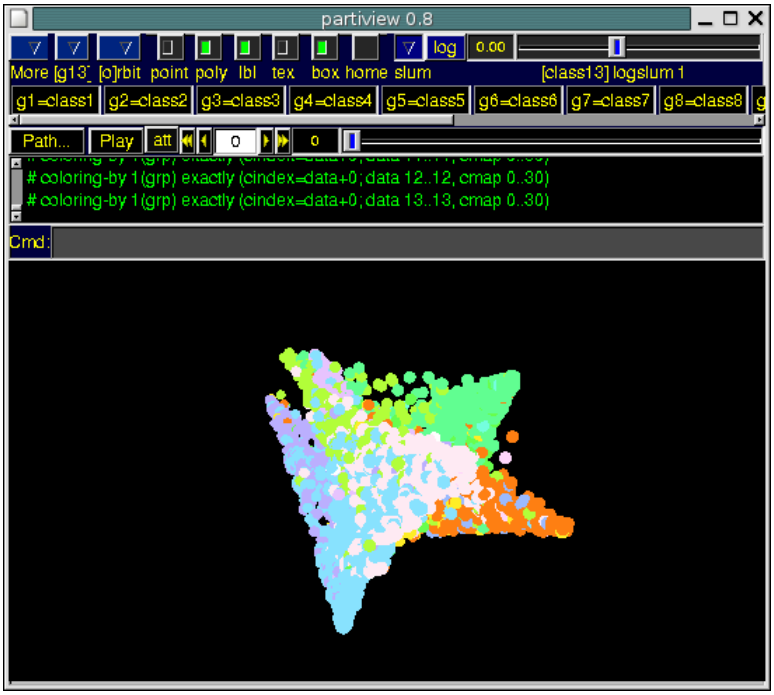
```
>> RawText{1031}

ans =

uh-huh NOISE(NONVOCAL) right
```

The i-th DA has true label `L(i)` and has been predicted to have label j with probability `ProbEst(i,j)` by the classification algorithm. Each DA also has some associated prosodic features with the k-th such feature `SomeProsNamesk` for the i-th DA in `SomePros(i,k)`. For the record, `SomeProsNames` is {`'meanf0'`, `'stdvf0'`, `'maxf0'`, `'minf0'`, `'rangef0'`, `'meanenergy'`, `'duration'`}.

The simplest command to start with is

```
ndaona('CLASSPROBS',ProbEst,'CLASSES',L);
```

Now Ndaona finds an embedding (using the Parametric Embedding algorithm (Iwata et al., 2004)) so that points with similar class probabilities are closer together, estimates graphical parameters required by Partiview, creates a temporary directory in the current directory, writes/copies about twenty files into there, and then starts up the model viewer so one see this:



This can be rotated, etc. The points seem larger than required, so we press ESCAPE to exit and return to Matlab decide to reduce them five times in size by saying

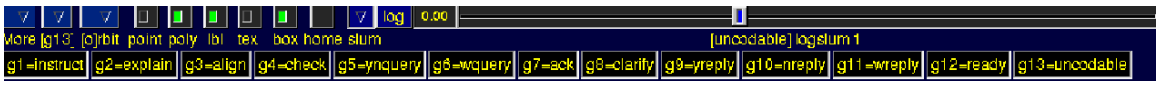
```
ndaona('CLASSPROBS', ProbEst,'CLASSES',L, 'GLYPHSIZE',0.2);
```

scaled (Platt, 2000) and combined with the procedure in (Wu et al., 2004) to produce a 13-class probability estimate $P(tone = i|syllable), i = 1, \ldots, 13$.

This produces a nicer scatterplot, so we can notice some more things. Each point is colored according to its class. There are also buttons at the top, with labels 'g1=class1', 'g2=class2', ... , 'g13=class13'. But we know the names of each class, so we can press ESC again and type in Matlab:

```
ndaona('CLASSPROBS', ProbEst,'CLASSES',L, 'GLYPHSIZE',0.2,
 'CLASSNAMES',daclassnames);
```

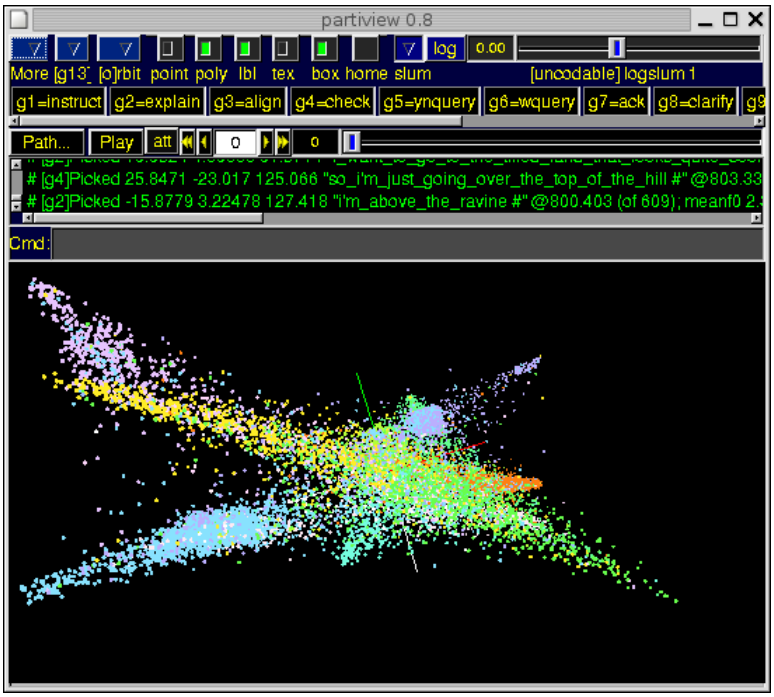
Now the menu looks like this. Pressing, say, the `g4=align` button toggles the visibility of all points corresponding to alignment dialog acts.



The problem so far is that we do not know what each point is. Thus we link each point with its raw text, and with some prosodic features/attributes associated with it. Pressing ESC again and typing in Matlab:

```
ndaona('CLASSPROBS', ProbEst,'CLASSES',L, 'GLYPHSIZE',0.2,
 'CLASSNAMES',daclassnames, 'INFO',RawText, 'ATTRIBUTES',SomePros,
 'ATTRIBUTENAMES', SomeProsNames);
```

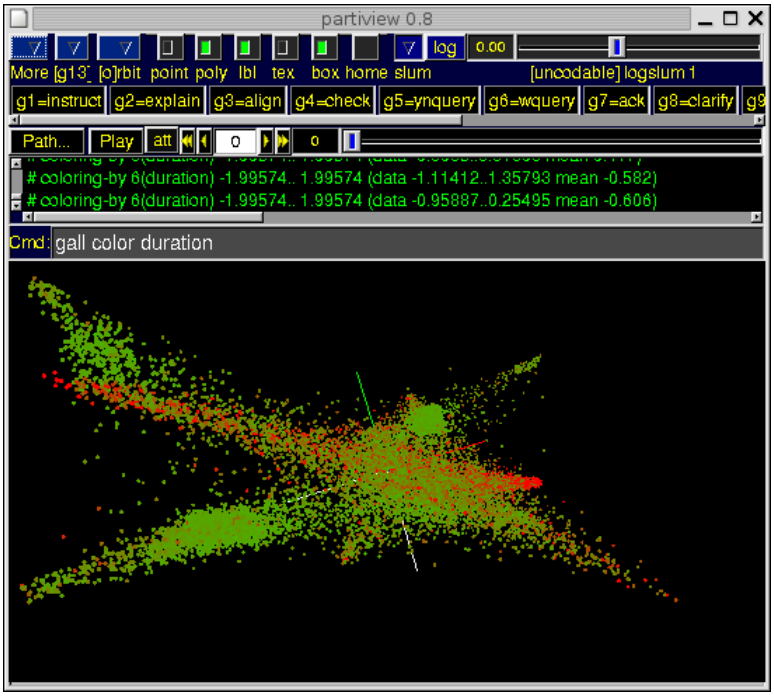
Now if you click on any point and press 'P', information will come up in the command window about its associated manual transcription.



To see the variation in some prosodic feature, say `duration` (as named in `SomeProsNames`) across all points, we need to type in the Partiview command window

```
gall color duration
```

The `gall` means 'apply the following command to all groups'. This changes the colors of groups as shown below; all points are given a color on the red-green continuum with red colored points having longer duration. One can also say `gall color meanf0` to color points by their mean pitch value, etc. One can still determine the true label of a point using the group buttons or clicking on a point and pressing 'P'.



2 Demo 2 : Mandarin Tone Recognition

Download and unzip the file http://people.cs.uchicago.edu/∼dinoj/ndaona/downloads/xutf_libsvm_linkernel_mat.zip to give the file xutf_libsvm_linkernel.mat before typing

```
>>load xutf_libsvm_linkernel.mat
>> whos
  Name         Size                    Bytes  Class

  CPtest    2880x4                     92160  double array
  Ltest        1x2880                  23040  double array
  Xtest     2880x20                   460800  double array
```

This dataset has the results of a 4-class Mandarin tone classification experiment from (Surendran et al., 2005) on 2880 test examples. Each example represents a syllable of elicited Mandarin speech by a native speaker, and is represented by a 20-dimensional vector and has a label between 1 and 4 inclusive[4]

The classification algorithm was a SVM as before. The vector for the i-th test example is in `Xtest(i,:)` and its true label is `Ltest(i)` $\in \{1, \ldots, 4\}$, while the SVM predicted that it had class $j \in \{1, \ldots, 4\}$ with probability `CPtest(i,j)`.

This example shows one thing that the previous example does not, which is the usefulness of glyphs. Here each point is associated with a dense low-dimensional vector with a plot of the vector providing useful information to the domain expert. Thus we can use this Matlab command to create 2880 small pictures of each vector. (This takes a while, and will be done before the public demo.)

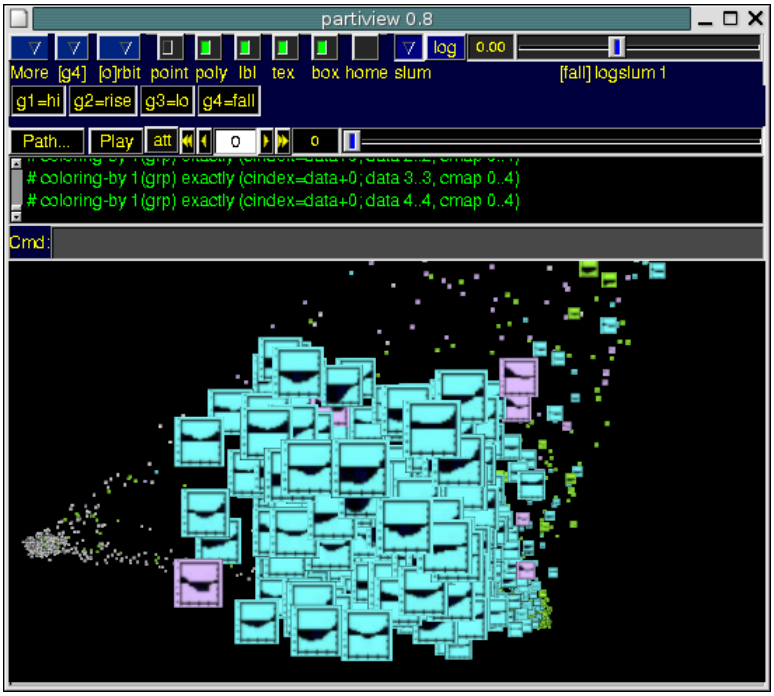
```
opts.filename = 'xutfsyll';
pix = makebarglyphs(Xtest,opts);
```

[4]The vector has the pitch contour of the syllable, normalized by time (all syllables are resampled to have the same duration) and speaker (the speaker's mean pitch has been subtracted and the result divided by the standard deviation of the speaker's pitch).

This creates 2880 SGI images in directory 'xutfsyll' and returns their names in the variable `pix`. Now the command

```
ndaona('CLASSPROBS',CPtest,'CLASSES',Ltest,'PIC',pix,
  'GLYPHSIZE',0.2,'PICDIR', 'images_xutfsyll');
```

produces a 3d model where each syllable is represented by a picture of its vector.



References

T Iwata, K Saito, N Ueda, S Stromsten, T L Griffiths, and Joshua B Tenenbaum. 2004. Parametric embedding for class visualization. In *Advances in Neural Information Processing Systems 17*.

John Platt. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A.J. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74.

Dinoj Surendran, Gina-Anne Levow, and Yi Xu. 2005. Tone recognition in mandarin using focus. In *Proceedings of the 9th European Conference of Speech Communication and Technology*.

Ting-Fan Wu, Chih-Jin Lin, and Ruby C. Weng. 2004. Probability estimates for multi-class classification for pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005.