Sum-Product Networks (SPN)
User Guide

Hoifung Poon
hoifung.poon@gmail.com

September 24, 2011

1 Introduction

In the following sections, we will provide step-by-step instructions for using the SPN system in image completion, as described in [2]. Throughout this document, we assume that the default directory is where you unpack `spn-release.tar.gz`. We will use the experiments in [2] as a running example.

2 Files

`poon11.pdf`: our UAI paper on sum-product networks and the SPN system.

`code/src/`: source code for SPN.

`code/bin/`: executable bytecode for SPN.

`data/`: default data directory with the input data files for replicating the SPN experiments in [2]. It contains the image files from Caltech-101 [1] and Olivetti [3].

`results/`: default directory for the output of SPN.

3 Input and Output

In the current experiments, we assume the input are pixelwise gray-scale intensities, stored under `data/`.[1] Each file simply lists intensities as space-delimited integers. The class `spn.Dataset` contains methods that input the images and

[1] Additional channels like color and depth can be incorporated in a similar way.

divide them into training and test sets. We used the last third of each category (up to 50 images) for test, and the rest for training. Because the images in Caltech-101 and Olivetti are ordered by objects, this ensures that the objects in the test images have minimal overlap with the training ones.

The Olivetti images are 64×64 in size, and we used them as is. The Caltech-101 images vary widely in size. We wanted the input to be fixed-size square images. So we rescaled the original images so that the longer side is of length N, and then used the inner $M \times M$ portions for the experiments. We set $N = 100, M = 64$ in our experiments.

After learning, the learned model is in `results/<DOMAIN>/models`. In addition, the left or bottom completions for test images are output to `results/<DOMAIN>/completions`. The completions contain both the original and completed images, separated by 10-pixel gaps with zero intensity.

4 MPI

The SPN code uses the message-passing interface (MPI) to parallelize computation. In particular, it uses MPJ Express, an open-source API for using MPI in Java. To install MPJ Express, go to `mpj-express.org` and follow the instructions.

5 Running SPN

The experiments in [2] were conducted in a cluster of machines with 8 cores (Intel Xeon 2.3GHz) and 16 GB of memory. To run SPN, first make sure that the MPI environment is properly set and the ring is alive, then go to `code/bin`, and run:

```
   mpjrun.sh -np [NUM_PROCESSOR] -dev niodev -mx8000m eval.Run
[SPN_OPTIONS] > [LOG_FILE]
```

The number of processes depends on the number of slave processes for each image category, and the number of slave groups (which controls the number of image categories that are concurrently processed). Specially,

$$numProc = (numSlavePerCat + 1) \times numSlaveGroup$$

For example, to replicate the Caltech experiments (with one group and fifty slave processes), run:

```
   mpjrun.sh -np 51 -dev niodev -mx8000m eval.Run -d C >
../../results/runcal.log
```

To replicate the Olivetti experiments, run:

```
   mpjrun.sh -np 51 -dev niodev -mx8000m eval.Run -d O >
../../results/runolive.log
```

In our cluster, the Olivetti experiment, including both learning and completion, finished within a few minutes. The Caltech one finished in about five hours for all 101 categories.

6 Parameters

SPN accepts the following parameters:

`Convergence threshold`: EM terminates if the change in log likelihood is beneath this value. Default 0.1; set by `-ct`.

`Sparse prior`: exponential penalty on each non-zero parameter. Default 1; set by `-sp`.

`Number of Gaussians for each pixel variable`: Default 4; set by `-ncv`.

`Number of sum nodes assigned to each region`: Default 20; set by `-nsr`.

`Mini-batch size`: Used in online EM learning. Default 50; set by `-bs`.

`Number of slave processes per image category`: Used in concurrently processing examples in mini-batches in EM learning. Should not be higher than mini-batch size. Default 50; set by `-ns`

`Number of slave groups`: Used in concurrently processing multiple image categories in Caltech experiments. Default 1; set by `-nsg`.

7 Compute Mean-Square Errors

Once the SPN experiments finish, the mean-square errors of completed pixels can be computed by running:

```
java spn.Eval
```

This assumes that the resulting completions are in the default directories for Caltech and Olivetti.

References

[1] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples. In *Proc. CVPR Wkshp. on Generative Model-Based Vision*, 2004.

[2] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty Seventh Conference on Uncertainty in Artificial Intelligence*, 2011.

[3] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proc. 2nd IEEE Wkshp. on Applications of Computer Vision*, 1994.