Lab 4 - GMMs and HMMs

{bengio,mkeller}@idiap.ch
http://www.idiap.ch/~{bengio,mkeller}

January 12, 2006

GMMs

1. **Let $\theta = \{w_1, \ldots, w_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K\}$ define a gaussian mixture model, where $\forall k$, $\Sigma_k = \sigma \cdot \mathbb{1}$, with σ a fixed parameter. Compare the EM fitting of the parameters to the K-means batch algorithm.**

....

- The gaussian mixture model:

$$p(x|\theta) = \sum_{k=1}^{K} w_k \phi_k(x, \theta) = \sum_{k=1}^{K} w_k \left[\frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\sigma}} \exp\left(-\frac{\|x - \mu_k\|^2}{2\sigma} \right) \right]$$

$$= \left[\frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\sigma}} \right] \sum_{k=1}^{K} w_k \exp\left(-\frac{\|x - \mu_k\|^2}{2\sigma} \right)$$

- **GMM E-step** - $\forall x \in D_{train}$, estimate the posterior probability, sometimes called *responsibility* of each j:

$$\gamma_j(x) = P(j|x, \theta) = \frac{P(j|\theta) \cdot p(x|j, \theta)}{p(x|\theta)} = \frac{w_j \phi_j(x, \theta)}{\sum_{k=1}^{K} w_k \phi_k(x, \theta)}$$

$$= \frac{w_j \exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma} \right)}{\sum_{k=1}^{K} w_k \exp\left(-\frac{\|x - \mu_k\|^2}{2\sigma} \right)}$$

- **K-means 1st step** - "For each prototype μ_k, put in the emptied set S_k the examples of D_{train} that are closer to μ_k than to any other $\mu_{j \neq k}$."

- **GMM M-step** - find parameters θ maximizing the auxiliary function (also called expected complete loglikelihood):

$$\mu_k = \frac{\sum_{i=1}^{n} \gamma_k(x_i) x_i}{\sum_{i=1}^{n} \gamma_k(x_i)}$$

$$w_k = \frac{1}{n} \sum_{i=1}^{n} \gamma_k(x_i)$$

- **K-means 2nd step** - "Re-compute the value of each μ_k as the average of the examples in S_k."

. . . .

HMMs

2. ***Maximum Likelihood & Decoding*** **- Imagine that from the other side of a curtain I tell you that I have 2 biased coins C_1 and C_2, that following a Markov assumption I flip one or the other coin and that I give you the resulting sequence of heads and tails without telling you from which coin each component comes.**

   (a) Design a hidden markov model for this sequence.

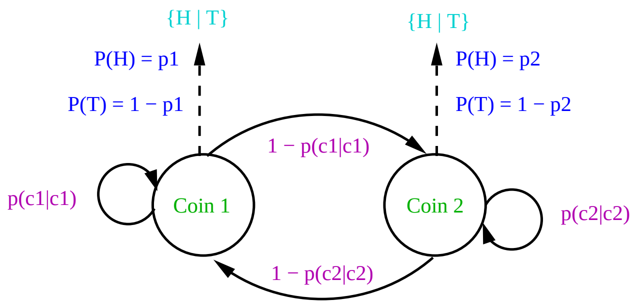
   (b) Using `2coinsgenerator.py` generate a list of sequences coming from a common distribution.

   (c) Using the functions implemented in `hmm.py`, select the parameters which maximize the likelihood of the list of sequences.

   (d) Generate a new sequence with the same distribution, implement the Viterbi algorithm and decode the new sequence. (Hint: Note that the recursive equation of $V(i,t)$ is very similar to the one of $\alpha(i,t)$ and do not forget to keep track of the path).

. . . .

(a)



(b)
```
> data = [gen(100)[0] for i in range(10)]
```

(c)
```
> [logEmission,logTransition,hmm_err] = em_hmm(data,2,30,2)
InitialLogEmission [[-1.77648189,-0.18540528,], [-0.70620657,-0.68025614,]]
InitialLogTransition [[-0.5030033 ,-0.92814015,], [-0.56799312,-0.83623614,]]
iteration 0 hmm_err 79.0057543691
iteration 1 hmm_err 69.1872515038
...
iteration 29 hmm_err 59.7914928982
```

(d) see `hmm_solution.py`.

```
> test = gen(10)
> viterbi(logEmission,logTransition,test[0])
[0, 1, 1, 0, 1, 0, 1, 1, 0, 1]
> test[1]
[1, 1, 0, 1, 0, 1, 1, 0, 1, 0]
```

. . . .

3. ***Classification*** **- Let us be in the same setting as the previous question, but this time I can give you a sequence originating from the flipping of the previous 2 coins or originating from a 3rd coin C_3 with $P(H) = 0.56$. What will you do to decide whether a new sequence comes from the 2 coins process or from C_3?**

. . . .
Use a Bayes Classifier.