CS 249 Project: Sampling Image Epitomes

Eagle Jones

June 10, 2003

1 Project Summary

I have implemented a project which generates "epitome" images based on a training set of images. The epitome is generated to match the high-order statistics of a given input image or set of images. The idea is to have the epitome "look like" the training image, so that it appears to come from the same space of images as the training set. Figure 1 shows an example. This project is based on the work of Frey and Jojic [1], who present the epitome as a model for the high-order statistics of natural images. Given an image, or a set of images, they generate a smaller image, using the Expectation Maximization (EM) algorithm [2], such that the probability of a patch drawn from the training (input) images occuring somewhere in the epitome is maximized.

My formulation is somewhat different than what they present. I use a Gibbs sampler to draw images from the space of possible epitomes. The general algorithm for the Gibbs sampler is as follows:

1. Initialize the image. (Randomly, or according to the statistics of the training set.)
2. Select a pixel to update. (Ordered or randomly; typically random selection means faster convergence.)
3. For each possible value of the pixel, calculate a score according to the properties of your model.
4. Update the pixel randomly, according to the distribution defined by the scores from step 3. (The value with the highest score has the highest probability of being selected.) The randomness of this step is controlled by a temperature factor which decreases over time.
5. Continue at step 2 until the image has converged.

The specifics of the scoring process are somewhat complicated. Given a patch from an image of a given size ($N \times N$), that patch is a point in N^2-dimensional space. If we consider the patches in the training set and all patches from the epitome as samples from two distributions, then our goal is to bring the distribution of patches in the epitome "close" to the distribution of patches in the training set. We define closeness

Figure 1: Original Earth image and its epitome.

according to the Kullback-Leibler divergence:

$$D_{KL}(p_1(x), p_2(x)) = \int p_1(x) \ln \frac{p_1(x)}{p_2(x)} dx \tag{1}$$

Of course, we don't have the proper distributions, so we estimate them by centering a gaussian of fixed variance at each patch in N^2-space. However, calculating the KL divergence over the entirety of the two distributions is prohibitively expensive. We simplify things by generating a "tiling" of the training patches on the epitome, such that each training patch is assigned to exactly one location in the epitome. (This is not a non-overlapping tiling, and there is no gaurantee of complete coverage.)

It is important to note here that any given pixel in the epitome is a component of N^2 epitome patches. Thus a patch from the training set which is "assigned" to a given location in the epitome actually affects $(3N-2)^2$ patches in the epitome image. It is this property which keeps the epitome smooth, generating patches which overlap in the "right" way. The assignment of patches from the training set to the epitome is done in a random manner at each step, according to the distance between each patch. (A patch has the highest probability of being assigned to its nearest neighbor in patch space.)

Within this framework, the KL divergence between two gaussians with means μ_1 and μ_2 with identical variances σ reduces simply to the L_2 norm:

$$\begin{aligned}
D_{KL} &= \int p_1(x) \ln \frac{p_1(x)}{p_2(x)} dx \\
&= \int \mathcal{N}(x, \mu_1, \sigma) \ln \frac{e^{-\frac{(x-\mu_1)^2}{2\sigma^2}}}{e^{-\frac{(x-\mu_2)^2}{2\sigma^2}}} dx \\
&= \int \mathcal{N}(x, \mu_1, \sigma) \frac{1}{2}((x-\mu_2)^2 - (x-\mu_1)^2) dx \\
&= \frac{1}{2} \int \mathcal{N}(x, \mu_1, \sigma)(\mu_2^2 - \mu 1^2 - 2x\mu_2 + 2x\mu_1) dx \\
&= \frac{1}{2} \left(\int \mathcal{N}\mu_2^2 dx - \int \mathcal{N}\mu_1^2 dx - 2 \int \mathcal{N}(x, \mu_1, \sigma) x\mu_2 dx + 2 \int \mathcal{N}(x, \mu_1, \sigma) x\mu_1 dx \right) \\
&= \frac{1}{2} \left(\mu_2^2 - \mu_1^2 - 2E(x)\mu_2 + 2E(x)\mu_1 \right) \\
&= \frac{1}{2} \left(\mu_2^2 - \mu_1^2 - 2\mu_1\mu_2 + 2\mu_1^2 \right) \\
&= \frac{1}{2} \left(\mu_2^2 - 2\mu_1\mu_2 + \mu_1^2 \right) \\
&= \frac{1}{2} (\mu_1 - \mu_2)^2
\end{aligned} \tag{2}$$

Recall that μ_1 and μ_2 are simply the values of our two patches. Thus, when considering a certain pixel p, each training patch $\mathbf{J}$ which is assigned to an epitome patch $\mathbf{K}$ which overlaps p contributes $\| \mathbf{J}, \mathbf{K} \|_2$ (where $\mathbf{K}$ is the patch using the candidate value for p) to the score. (This actually calculates a value inverted from the normal sense of "score": larger is worse in this case.)

2 Results

Figure 1 shows a 256×256 epitome generated from a satellite photo (1600×1024) of the Earth. Figure 2 shows an epitome generated from 15 cropped images (typical size 160×110) out of the Yale Face Database[3].

3 Discussion

I am fairly pleased with the results of this project. The Gibbs sampler is a viable alternative to Frey's EM technique, and produces a greater variety of interesting images. EM minimizes the energy of the result, such that a given input should really only produce one possible output (as long as there are no convergence issues). The Gibbs sampler actually samples from the space of images given by a distribution according to their energy, so will produce a new result every time.

Both techniques have the same overall time complexity: $O(ENP)$, where E is the size of the epitome, P is the size of training patches, and N is the number of training patches. $N \times P$ is determined by the epitome size; we want enough training data to "cover" the epitome a few times over. So the time complexity is essentially $O(E^2)$. Generating larger images gets expensive rather quickly, which is why the examples are fairly small. I implemented an optimized version in which the distance function is coded in assembly with MMX and 3DNow! instructions. This runs about $10\times$ faster than the original C version did, but still takes several hours to generate a 256×256 image on a 1.4GHz Athlon. This slowness is why I was unable to pursue some of the additional applications I had considered, such as generating larger "PhotoMosaic" type images.

The size of the training patches determines the scale on which features from the training set will arise in the epitome. Larger training patches mean that larger recognizable blocks will appear in the output. However, since the training set and training time are finite, larger blocks are more difficult to fit and result in a chunky appearance in the epitome. Smaller blocks yield a smoother result but may not capture the desired features. A good compromise is to use blocks at multiple scales. The larger blocks get overall features to be recognizable, and the smaller blocks fill in the details and smooth the edges. The content of the training images makes a big difference in the results. If images are too heterogeneous, it becomes difficult to recombine them - there aren't enough examples of transitions from one object or region to another. This can be seen to a certain extent in figure 2, due to the small number of faces used to generate this epitome.

Convergence of the Gibbs sampler is very sensitive to the temperature schedule. Cooling too rapidly can cause "cracks" in the image - areas of the epitome which are overlapped by no training patch. Control of this seems to come from experience and trial and error.

Source code, images, and a video of the convergence steps can be found at
http://www.toblave.org/epitome.

Figure 2: 15 images from the Yale Faces Database and their epitome. The epitome is somewhat chunky due to the small number of images used.

References

[1] B. Frey and N. Jogic. Learning the “epitome” of an image. *University of Toronto Technical Report PSI 2003-56*, 2003.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B.*, 39:1–38, 1977.

[3] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.