Supplementary Material - From Learning Models of Natural Image Patches to Whole Image Restoration

Daniel Zoran and Yair Weiss

1 Model Likelihood Calculations

In the paper we compare the log likelihoods for four different models. Here we provide the details on how these models were trained and how the likelihood values were calculated. For all models we remove the DC component of each patch independently, both during training and evaluation. Models were trained on patches randomly sampled from the training set of the Berkeley Segmentation Database. Unless otherwise noted, we trained on 50,000 8×8 patches, and evaluated log likelihoods on 10,000 patches, sampled from the test set of the Berkeley Database.

1.1 Independent Pixels

This model assumes that each pixel in the patch is independently drawn from a Gaussian distribution. We learn a variance parameter σ_i^2 for each of the N pixels in the patch. The log likelihood of a given patch under this model is:

$$\log p(\mathbf{x}) = \sum_i \log(N(x_i; 0, \sigma_i^2))$$

where $N(x; \mu, \sigma^2)$ is the density of the Gaussian distribution with mean μ and variance σ^2.

1.2 Multivariate Gaussian

This model assumes that each patch is drawn from an N dimensional Gaussian with a learned covariance $\mathbf{\Sigma}$. We learn the covariance from the training set, the mean is assumed to be $\mathbf{0}$. With this model, the likelihood of a given patch is:

$$\log p(\mathbf{x}) = \log N(\mathbf{x}; \mathbf{0}, \mathbf{\Sigma})$$

1.3 PCA with Sparse Marginals

Since natural images are very non-Gaussian, we learn a sparse marginal density function $f(x)$ from the marginal distribution of PCA filter responses. This function was chosen to be a Gaussian Scale Mixture such that:

$$f(x) = \sum_k \pi_k N(x; 0, \sigma_k^2)$$

where we learn the parameters π_k and σ_k^2 from the data using EM. We assume that the joint distribution is factorial, that is, the filter responses are independent. In order to calculate the likelihood, we first learn the PCA basis from the data covariance data Σ, that is the eigenvectors matrix $\mathbf{V}$ and diagonal eigenvalues matrix $\mathbf{D}$. Then, likelihood of a given patch $\mathbf{x}$ is:

$$\begin{aligned} p(\mathbf{x}) &= \sum_i \log f(z_i) + \log |\mathbf{D}^{-0.5}\mathbf{V}^T| \\ \mathbf{z} &= \mathbf{D}^{-0.5}\mathbf{V}^T\mathbf{x} \end{aligned}$$

1.4 ICA

This model is identical to the previous model, only we learn a rotation matrix $\mathbf{W}$ over the whitened training set such that log likelihood of the rotated patches is maximized under a factorial model, that is:

$$\begin{aligned} \mathbf{W} &= \arg\max_{\mathbf{W}} \sum_i \log f(y_i) \\ \mathbf{y} &= \mathbf{Wz} \end{aligned}$$

We optimize the matrix $\mathbf{W}$ using gradient ascent on the likelihood. Using the model, the likelihood is calculated as above, only using the rotated patches $\mathbf{y}$. This is a simple form of doing ICA.