Summary of "Non-Parametric Bayesian Dictionary Learning"

Matt Dickenson `mcd31`
STA571/CS590.01 Due: 17 March, 2014

Zhou et al (2009) introduce a method for learning dictionaries without training images. This method has several advantages. Noise variance can be non-stationary (often true in applied cases), and need not be known for learning. The method can also perform sequential inference, which allows it to scale to large images. A beta process prior is used, allowing the elements of the dictionary and their importance to be inferred directly, rather than set arbitrarily by the researcher as in other methods we have seen in this course. Components of the image can also have spatial relationships that are inferred. The method can be implemented with Gibbs sampling or variational inference, and general enough to be widely applicable, and has better performance when compared with several other current image analysis tools.

Dictionary learning is used when we have a model in which documents x are realizations of portions of a dictionary $\mathbb{D}$ with weights α and additive noise ϵ so that $x = \mathbb{D}\alpha + \epsilon$. For any given x, α is sparse and so only a few columns of $\mathbb{D}$ are observed. In many models $\mathbb{D}$ is assumed rather than inferred. Dictionary learning infers $\mathbb{D}$ from the data, rather than assuming it. Because $\mathbb{D}$ is potentially infintie, a beta process prior is placed on its columns. $\mathbb{D}$ and α can be thought of as a lower dimensional way to represent de-noised x's.

Dictionary learning has several useful applications in addition to dimensionality reduction. If the full model can be inferred, we can also reconstruct the de-noised x's, which is useful for visual images. Similarly, the method can be used to fill in missing pixels in visual images, an application the authors refer to as "inpainting." As with other dimensionality reduction techniques, dictionary learning can also be used for the compression of large files such as images or verbal documents. Because the method can be run in sequence, it can compress very large data using a single dictionary, which should grow in memory at a rate much slower than that of the input.