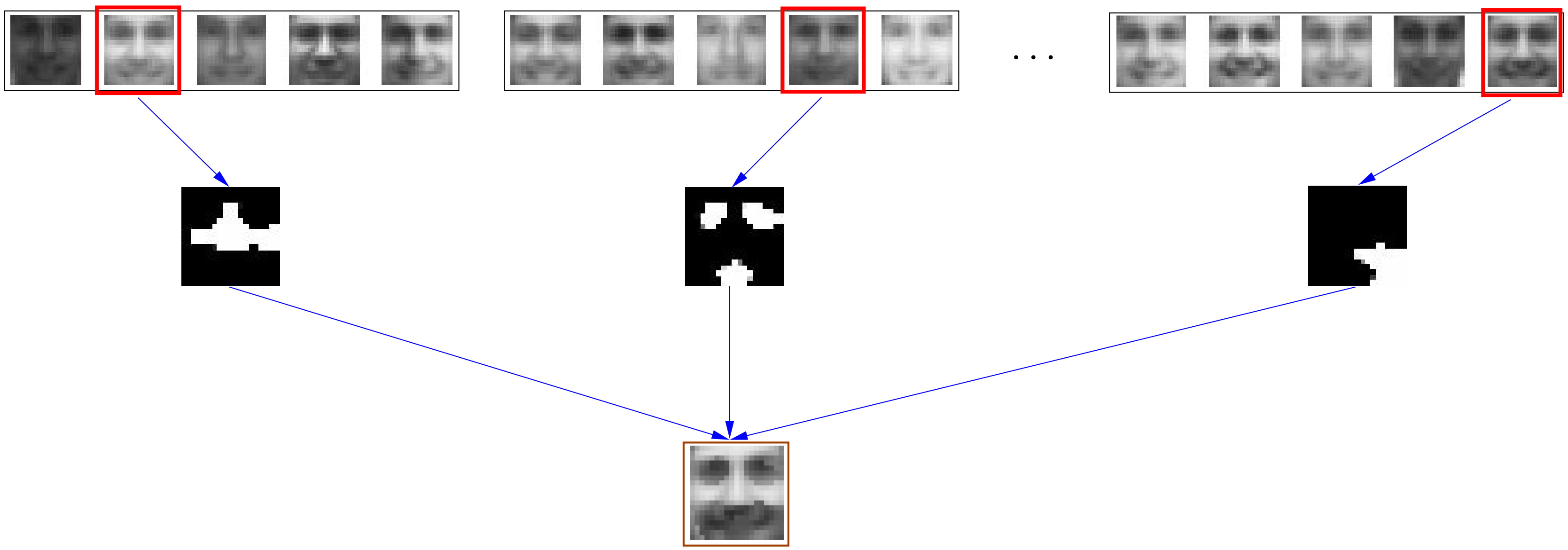
Multiple Cause Vector Quantization



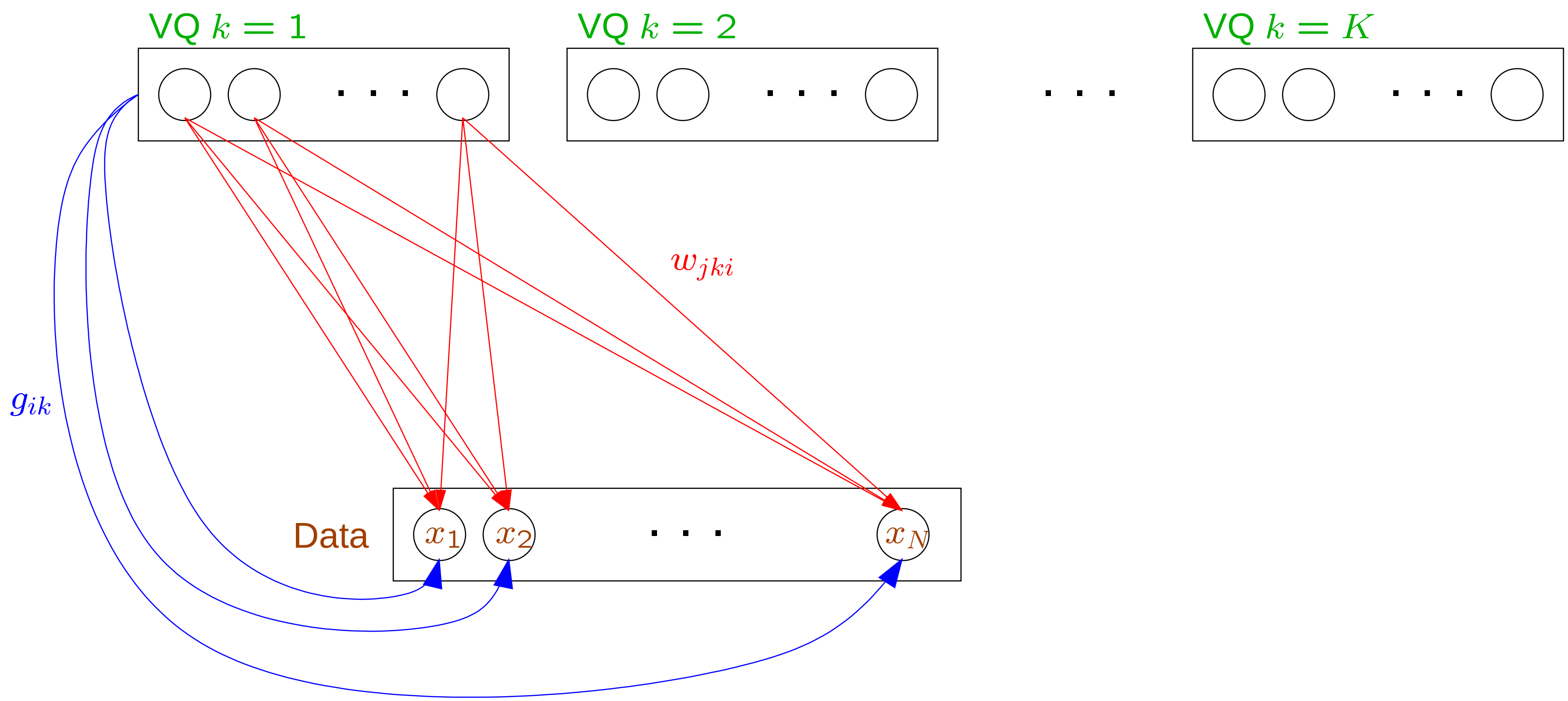
David Ross & Rich Zemel

University of Toronto

Introduction

- Goal: learn a parts-based representation of data vectors
- Motivating Assumptions:
  1. data dimensions can be separated into disjoint sets or *Causes*
  2. each cause has a small number of *States*
  3. causes take on states independently of each other
- Example: on face image data,
  causes could be *eyes*, *nose*, and *mouth*
  states could be different eye, nose and mouth shapes, respectively

Generative Model



- K Vector Quantizers (VQ's) [Causes]
- J Vectors per VQ [States]
- C Training Examples $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^C\} \subseteq \mathbb{R}^N$

To Generate an Example $\mathbf{x}^c$

1. stochastically select one state of each VQ to be active

   - selection vector $\mathbf{s}^c \in \{0, 1\}^{JK}$,
     $s^c_{jk} = 1 \;\Leftrightarrow\;$ state j of VQ k is active

2. stochastically select one VQ for each data dimension

   - selection matrix $\mathbf{R} \in \{0, 1\}^{N \times K}$,
     $r_{ik} = 1 \;\Leftrightarrow\;$ VQ k is relevant for x^c_i

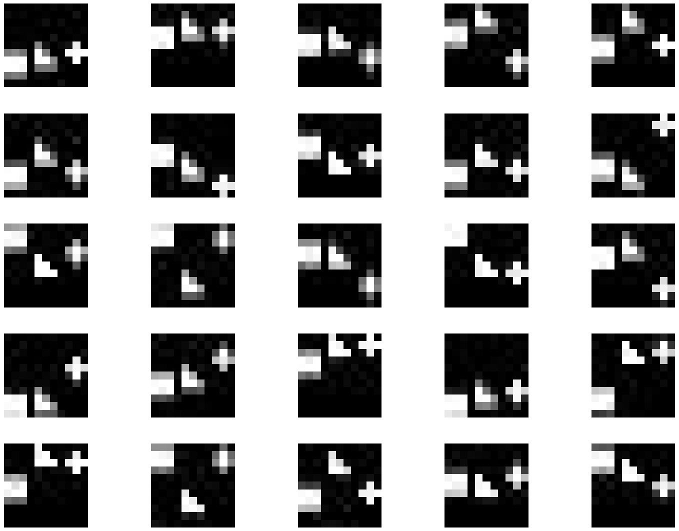
3. the value assigned to x^c_i is the weight of the active state from the relevant VQ

$$x^c_i = \sum_{k, j \in k} s^c_{jk}\, r_{ik}\, w_{jki}$$

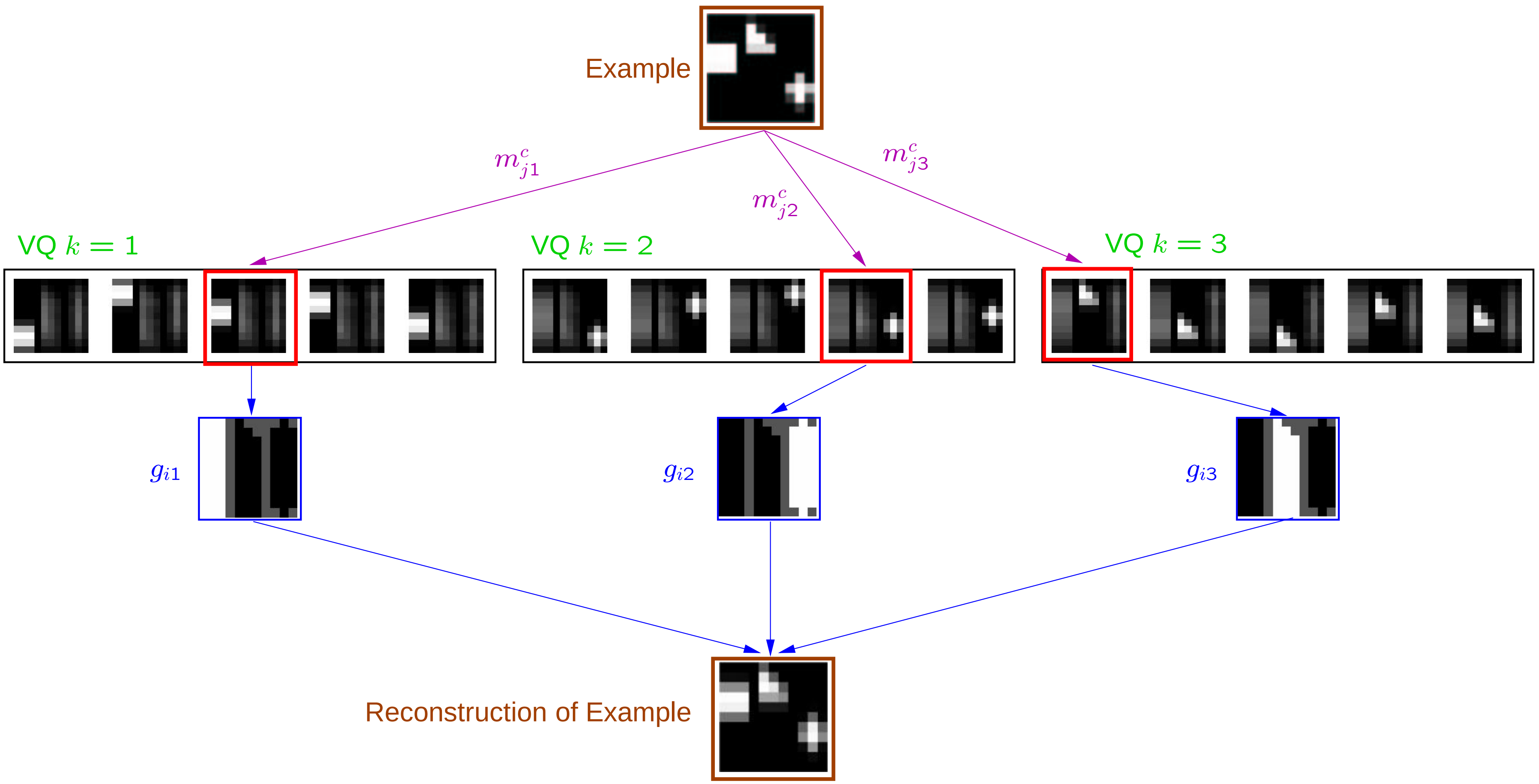
Shapes Data

- 1000 randomly generated gray-scale images
- each contains three shapes
- each shape has a fixed horizontal position, but variable vertical position
- vertical position of each shape randomly and independently selected, according to a uniform distribution

Data Examples



Shapes Data - Learned Model

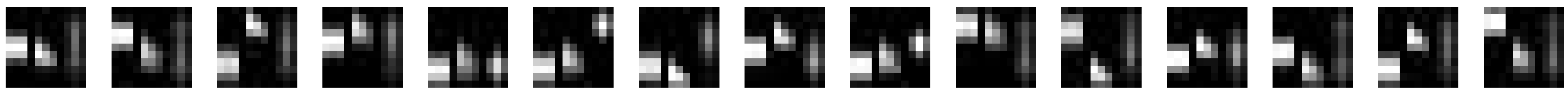


Related Approaches

like MCVQ, the following approaches represent data as a linear combination of ‘basis’ vectors

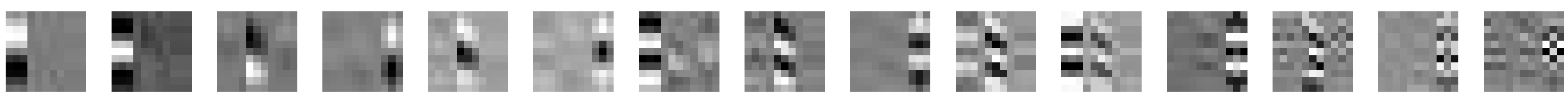
Vector Quantization:

basis comprised of data templates - each example represented by nearest template (soft version: example is affine combination of templates)



Principal Component Analysis:

basis vectors are eigenvectors of data covariance matrix - example represented by arbitrary linear combination of bases

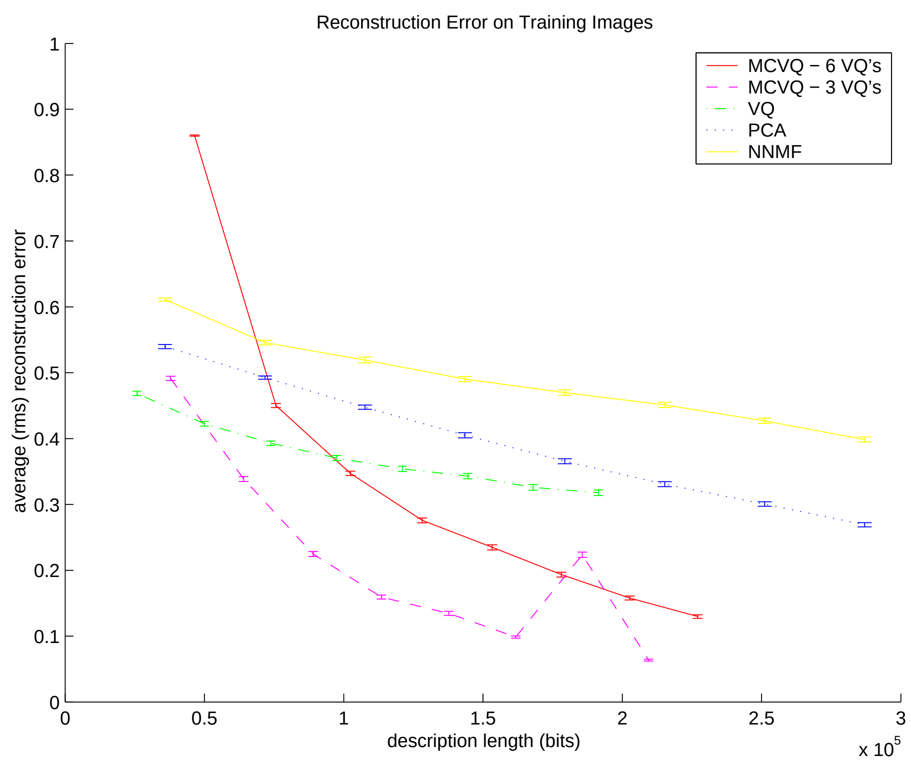


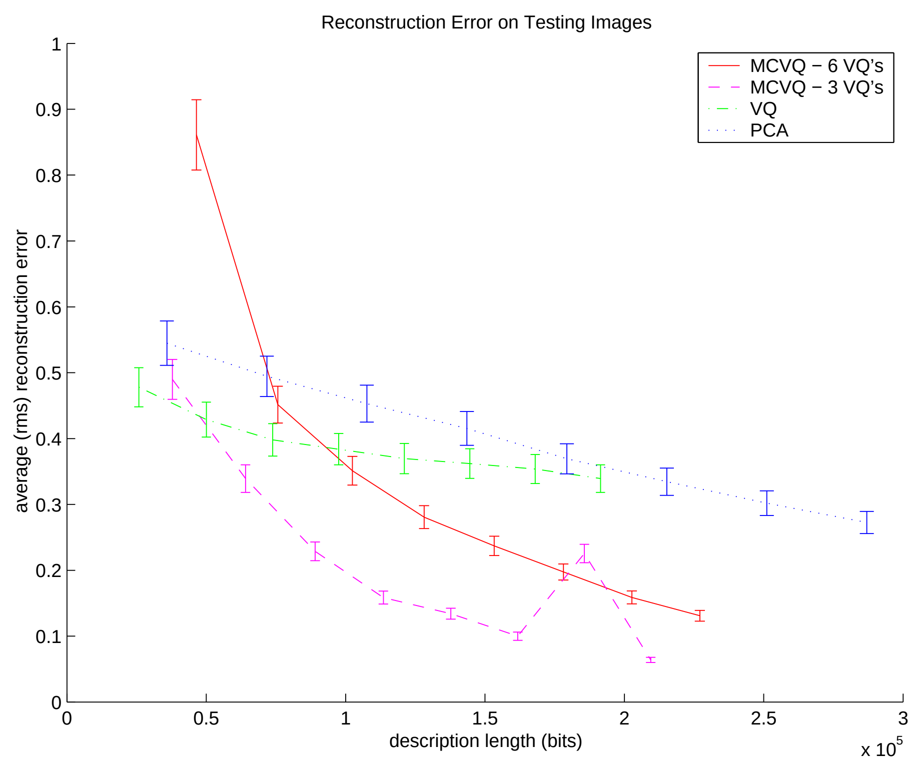
Non-Negative Matrix Factorization: (D. Lee & S. Seung)

example represented by non-negative linear combination of non-negative basis vectors

Shapes Data - Reconstruction

- learned model used to represent & reconstruct example images
- average root-mean-squared error was calculated for the training set (left) and an independent testing set (right)
- compared using description length (# of bits used to represent model + # of bits to encode all training examples using the model)





Learning & Inference

probability of a single example

$$P(\mathbf{x}^c|\mathbf{s}^c, \mathbf{r}, \mathbf{W}) = \prod_{i,k,j\in k} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{1}{2\sigma^2}\, s_{jk}^c\, r_{ik}\, (x_i^c - w_{jki})^2 \Big)$$

define the prior probability of selecting each state and VQ to be

$m_{jk}^c = E[s_{jk}^c]$ and $g_{ik} = E[r_{ik}]$

$$P(\mathbf{s}^c, \mathbf{R}) = \prod_{k,j\in k} (m_{jk}^c)^{s_{jk}^c} \quad \prod_{i,k} (g_{ik})^{r_{ik}}$$

state and VQ selections ($\{\mathbf{s}^c\}$ and $\mathbf{R}$) are latent variables

if instead $\{\mathbf{s}^c\}$ and $\mathbf{R}$ are observed, get complete likelihood of training data:

$$
\begin{aligned}
\mathcal{L} &= \prod_c P(\mathbf{x}^c \mid \mathbf{s}^c, \mathbf{R}, \mathbf{W}, \mathbf{M}, \mathbf{G})\; P(\mathbf{s}^c, \mathbf{R} \mid \mathbf{W}, \mathbf{M}, \mathbf{G}) \\
&= \prod_{c,i,k,j\in k} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{1}{2\sigma^2} s^c_{jk} r_{ik} (x^c_i - w_{jki})^2\Big) (m^c_{jk})^{s^c_{jk}} (g_{ik})^{r_{ik}}
\end{aligned}
$$

Use EM algorithm to find a model that maximizes the expected complete data log-likelihood, or, equivalently, minimizes cost function

$$
\begin{aligned}
C &= -E[\log \mathcal{L}]_{\mathbf{s}^c, \mathbf{R}} \\
&= \frac{1}{2\sigma^2} \sum_{c,k,j,i} m^c_{jk}\, g_{ik}\, (x^c_i - w_{jki})^2 - \sum_{c,j,k} m^c_{jk} \log m^c_{jk} - \sum_{i,k} g_{ik} \log g_{ik}
\end{aligned}
$$

Intuition: choose one VQ per pixel, one state per VQ that matches input

Update Rules

E – Step (inference)

$$m^c_{jk} = \frac{\exp\left(-\frac{1}{2\sigma^2}\sum_i g_{ik}(x^c_i - w_{jki})^2\right)}{\sum_{\nu=1}^{J}\exp\left(-\frac{1}{2\sigma^2}\sum_i g_{ik}(x^c_i - w_{\nu ki})^2\right)}$$

M – Step

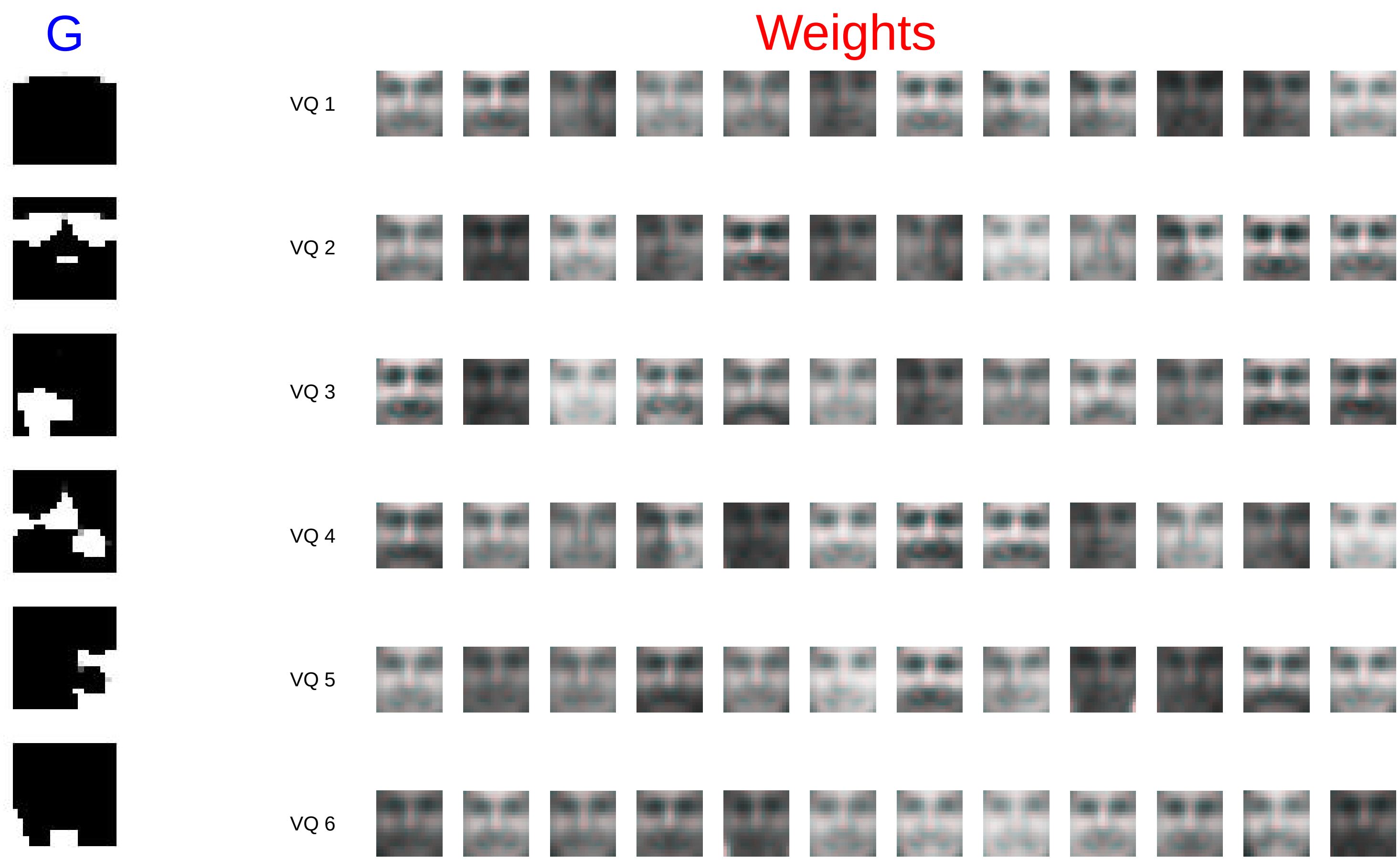
$$w_{jki} = \frac{\sum_c m^c_{jk}\, x^c_i}{\sum_c m^c_{jk}}$$

$$g_{ik} = \frac{\exp\left(-\frac{1}{2\sigma^2}\sum_{c,j} m^c_{jk}(x^c_i - w_{jki})^2\right)}{\sum_{\beta=1}^{K}\exp\left(-\frac{1}{2\sigma^2}\sum_{c,j} m^c_{j\beta}(x^c_i - w_{j\beta i})^2\right)}$$

Performance on Face Images

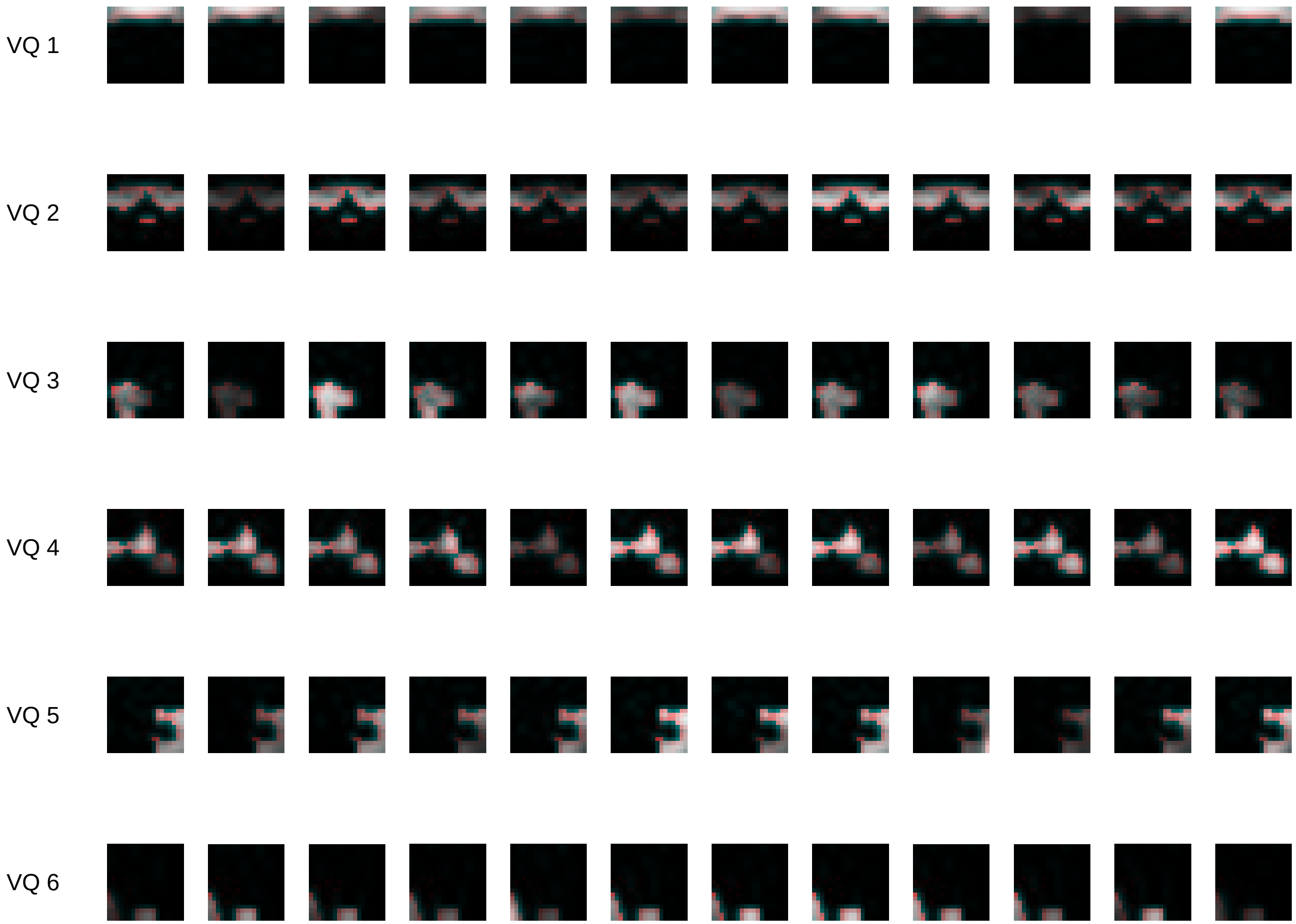
Training Set * : 2429 gray-scale images of faces, each 19×19 pixels

Model Parameters: 6 VQ's, 12 states per VQ



*CBCL Face Database # 1; MIT Center For Biological and Computation Learning
http://www.ai.mit.edu/projects/cbcl

Weights masked by G

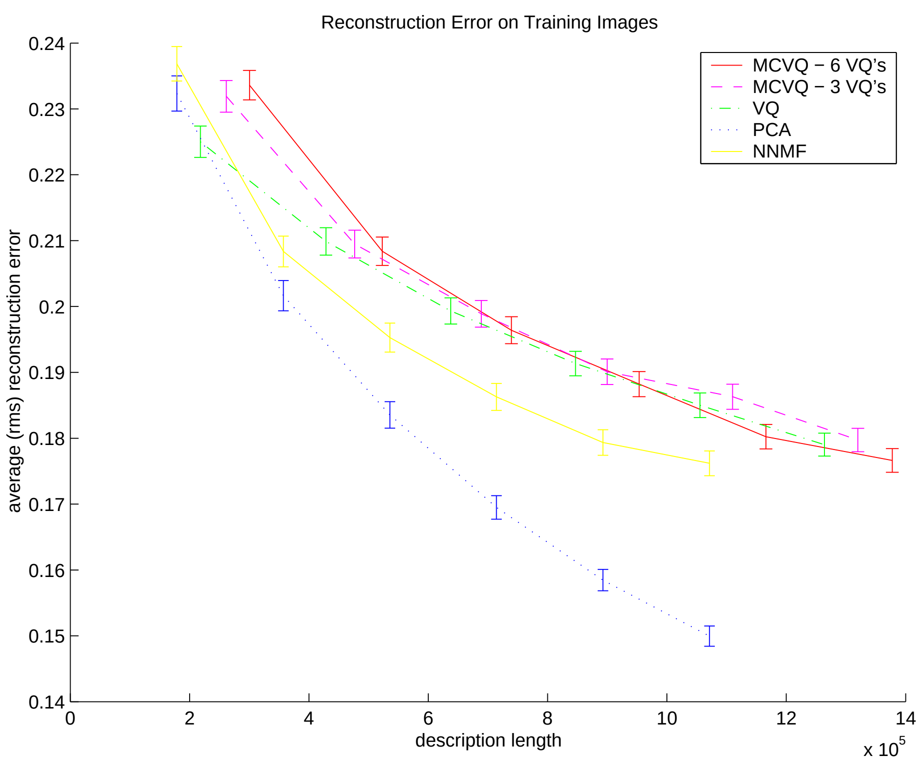


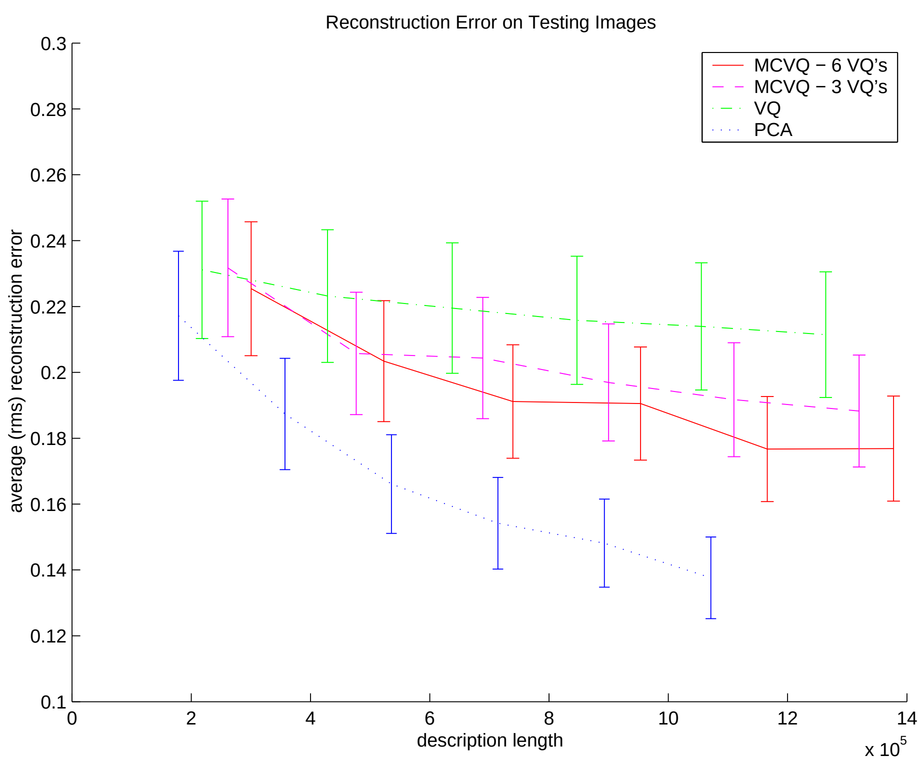
Example Reconstructions

- original on left, reconstruction on right

Comparison of Reconstruction Error

- testing set contained 472 images
- compared using description length (# of bits used to represent model + # of bits to encode all training examples under the model)
- i.e. for PCA: $RNB + CRB$, for MCVQ: $K(J+1)NB + CK \log_2 J$
  R=#components; N=input dims; B=bits/float; C=#cases





Summary & Current Directions

- a generative model for data composed of independent *causes*
- learns a parts-based segmentation of images, and a range of states for each part
- competitive performance when summarizing and reconstructing data
- inherent feature selection provides low-dimensional representation for further processing

we are currently exploring:

1. applications to text classification
2. collaborative filtering
3. Bayesian learning for model selection