CO902
Probabilistic and statistical inference

Lab 8

Tom Nichols
Department of Statistics &
Warwick Manufacturing Group

t.e.nichols@warwick.ac.uk

Multivariate GMM

- Applies in natural way to higher dimensional Gaussians
- Model: $$p(\mathbf{x}) = \sum_{j=1}^{k} \pi_j \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$
- Parameters: $$\{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j\} \quad j = 1..k \qquad \sum_{j=1}^{k} \pi_j = 1$$
- Likelihood: $$p(\mathbf{x}_1 \ldots \mathbf{x}_n | \{\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j, \hat{\pi}_j\}) = \prod_{i=1}^{n} \sum_{j=1}^{k} \hat{\pi}_j \mathcal{N}(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)$$

Multivariate GMM: Practicalities

- Problem 1 – Singularities
  - Can maximize the likelihood (i.e. make it ∞) with 1 class per observation with zero variance!
  - Good implementations will avoid this
- Problem 2 – What k?
  - Information-theoretic criterion to avoid ML's over-fitting
    - Akaike Information Criterion (AIC)
      - optimised log likelihood - M, M = number of estimated parameters
    - Bayesian Information Criterion (BIC)
      - optimised log likelihood - M ln(N) / 2, N = number of observations

AIC vs BIC

- Both are based on asymptotic approximations
- AIC
  - Approximation based on relative distance between the true and fitted likelihood function
  - Motivated by over-all accuracy of the distribution
- BIC
  - Approximation based on posterior probability of a given model being the "true" model
  - Motivated by getting the "right" model order
- AIC tends to pick bigger models, BIC smaller models
  - BIC solutions may be easier to interpret; AIC maybe more accurate for prediction
- Practical warning
  - Many authors (& Matlab) define them as "smaller better"
    - AIC = $- \log l(\theta;x) + M$
    - BIC = $- \log l(\theta;x) + M \ln(N) / 2$

PCA Reminder (1)

- For d × n data matrix $\mathbf{X}$, PCA finds $\mathbf{U}$ such that
  $$\mathbf{Y} = \mathbf{U}' \mathbf{X}$$
  has maximal variance
- $\mathbf{U}$ is a set m of length-d column vectors
  - $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)$
  - m = min(d,n-1)
    - Matlab will give you more than m, but they correspond to zero eigenvalues
- Moreover, the first $d^* \leq m$ of $\mathbf{U}$ give the maximal-variance d^*-dimensional
  $$\mathbf{Y}^* = \mathbf{U}^{*\prime} \mathbf{X}$$
  - $\mathbf{U}^* = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{d^*})$
  - In Matlabese… Ustar = U(:,1:dstar)

PCA Reminder (2)

- To move back from 'reduced' d*-dimensional space to full d-dim space, premultiply by $\mathbf{U}$
  - E.g. if GMM finds a d*-dimensional mean $\boldsymbol{\mu}_k$

    $\mathbf{U}^* \mu_k$

    is the d-dimensional representation of that mean

“Classification” with GMM

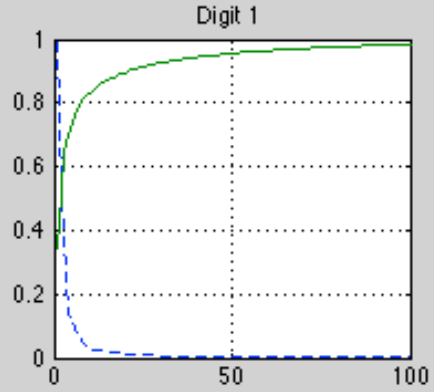
- Once a GMM is fit, each observations can be assigned to the class that is most likley to have generated it
  - Precisely, it is the class that maximizes the posterior probability of class k given x…

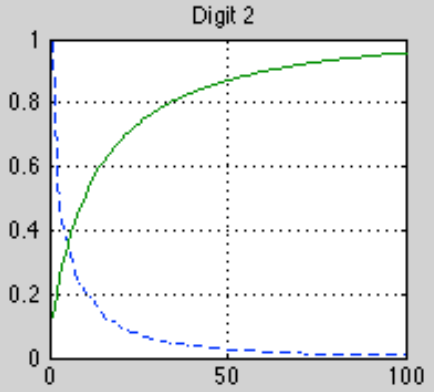
$$P(Z = k|X = x) \propto p(x|Z = k)\, p(Z = k) = \mathcal{N}(x|\mu_k, \Sigma_k)\pi_k$$

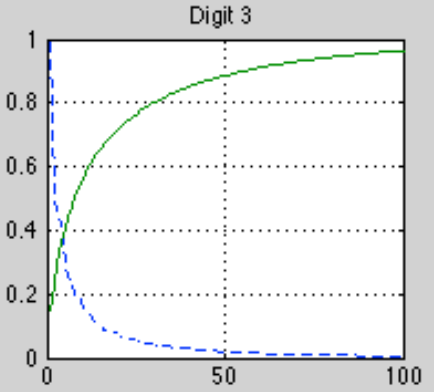
  - That is, it is *not* the class k that minimizes the Mahalanobis distance between x & μ_k!
  - It is the class that maximizes $\mathcal{N}(x|\mu_k, \Sigma_k)\pi_k$
    - The joint likelihood of X & latent class variable Z

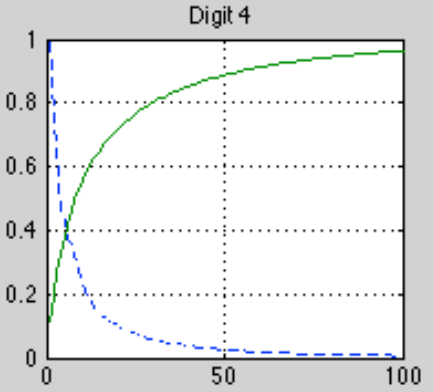
Lab “Solutions”

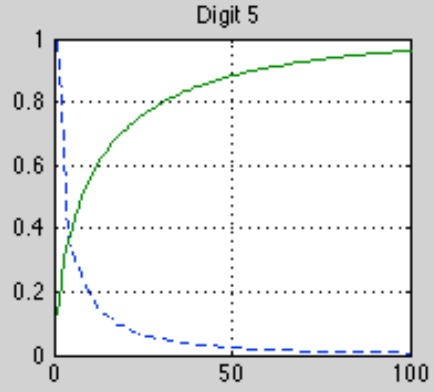
PCA Eigenspectrum

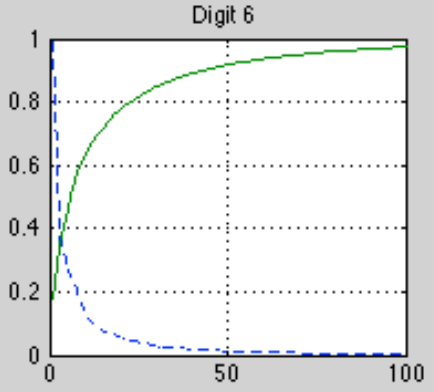


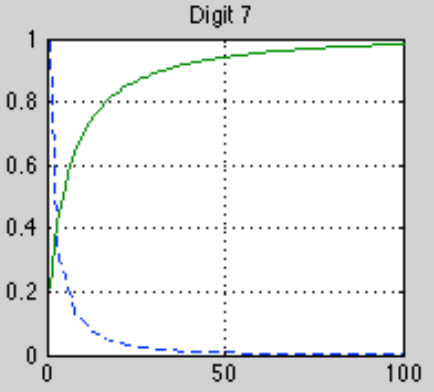


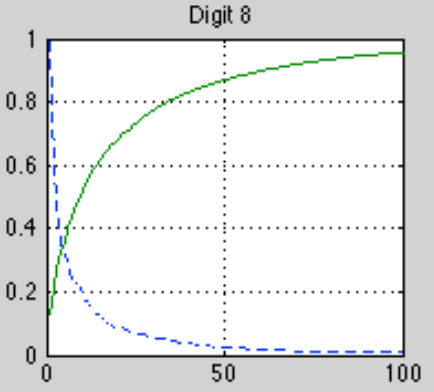


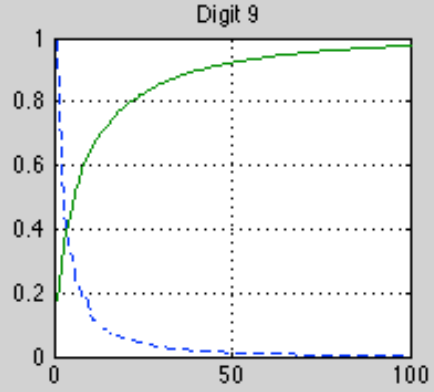


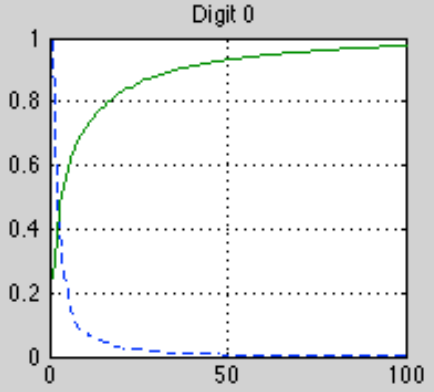




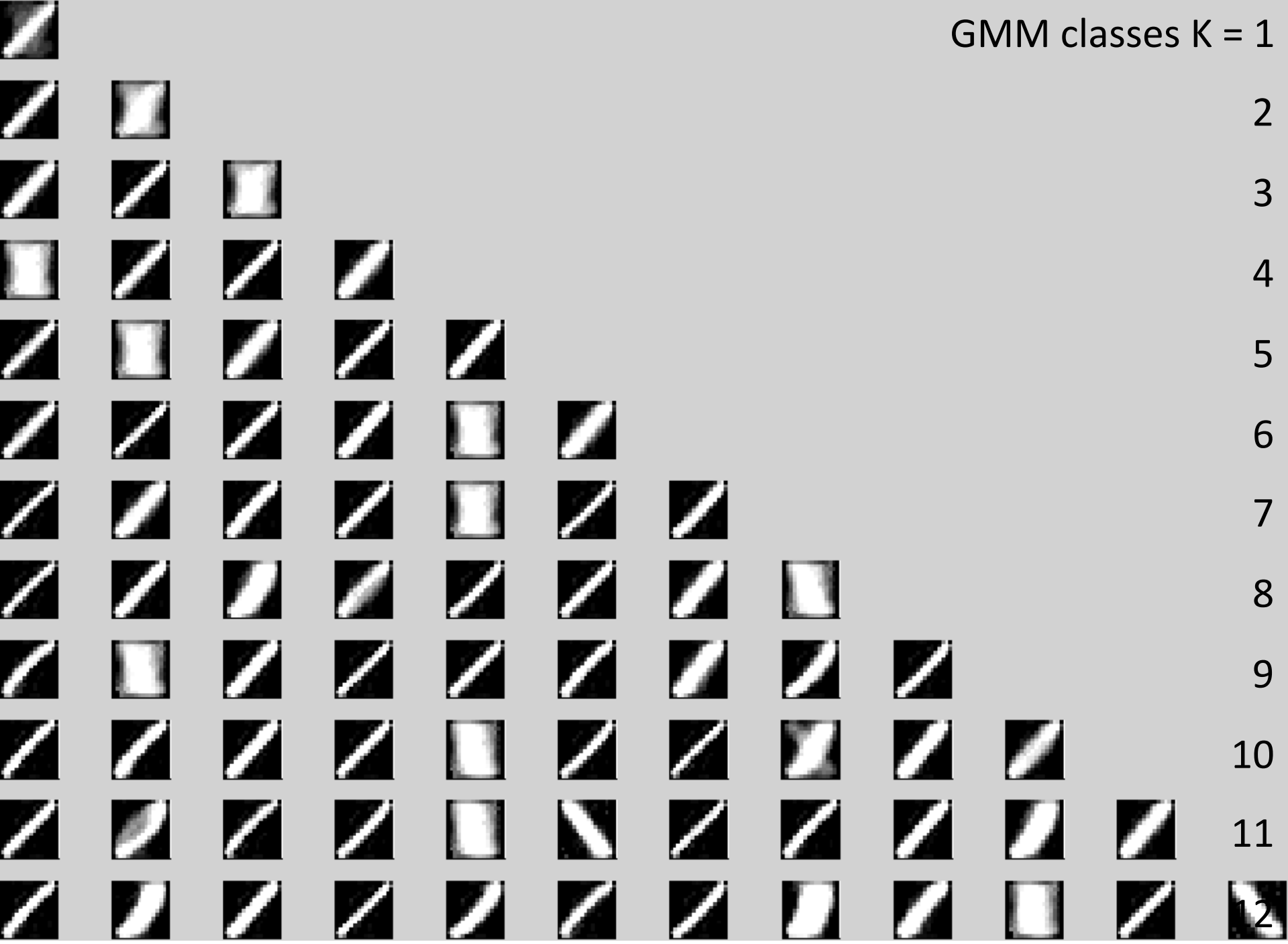


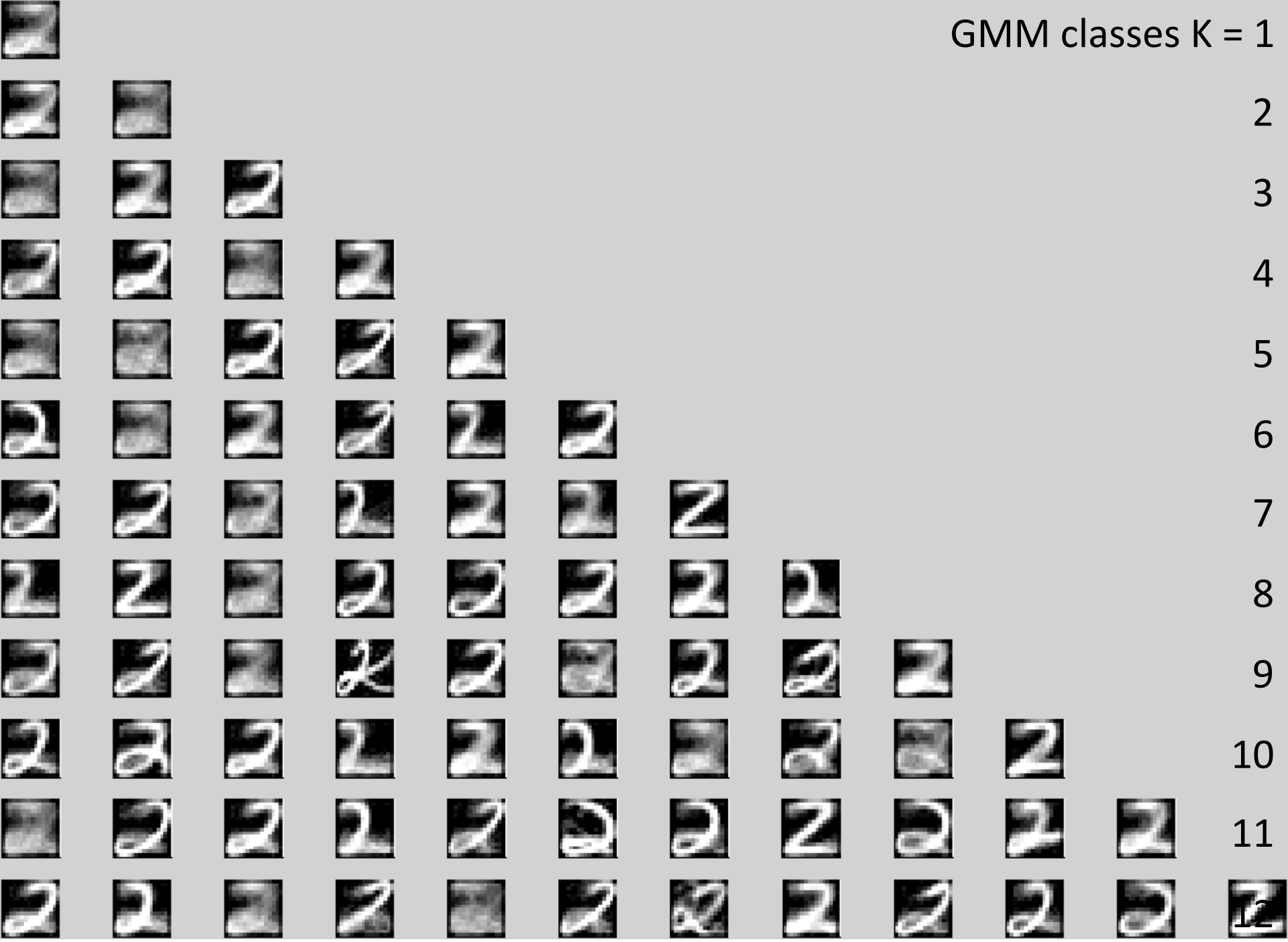


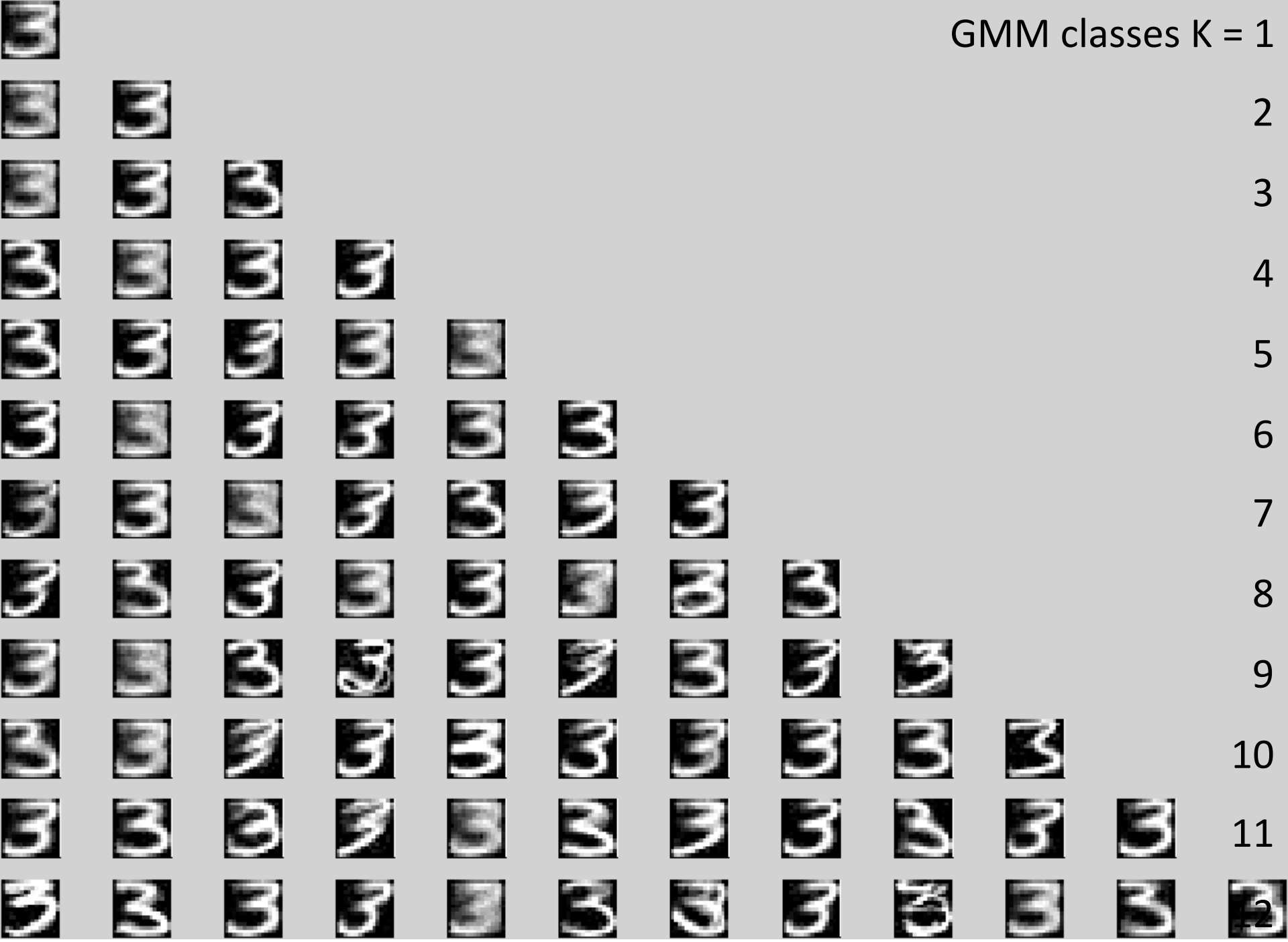




- Some digits need a lot more components to represent the variation well

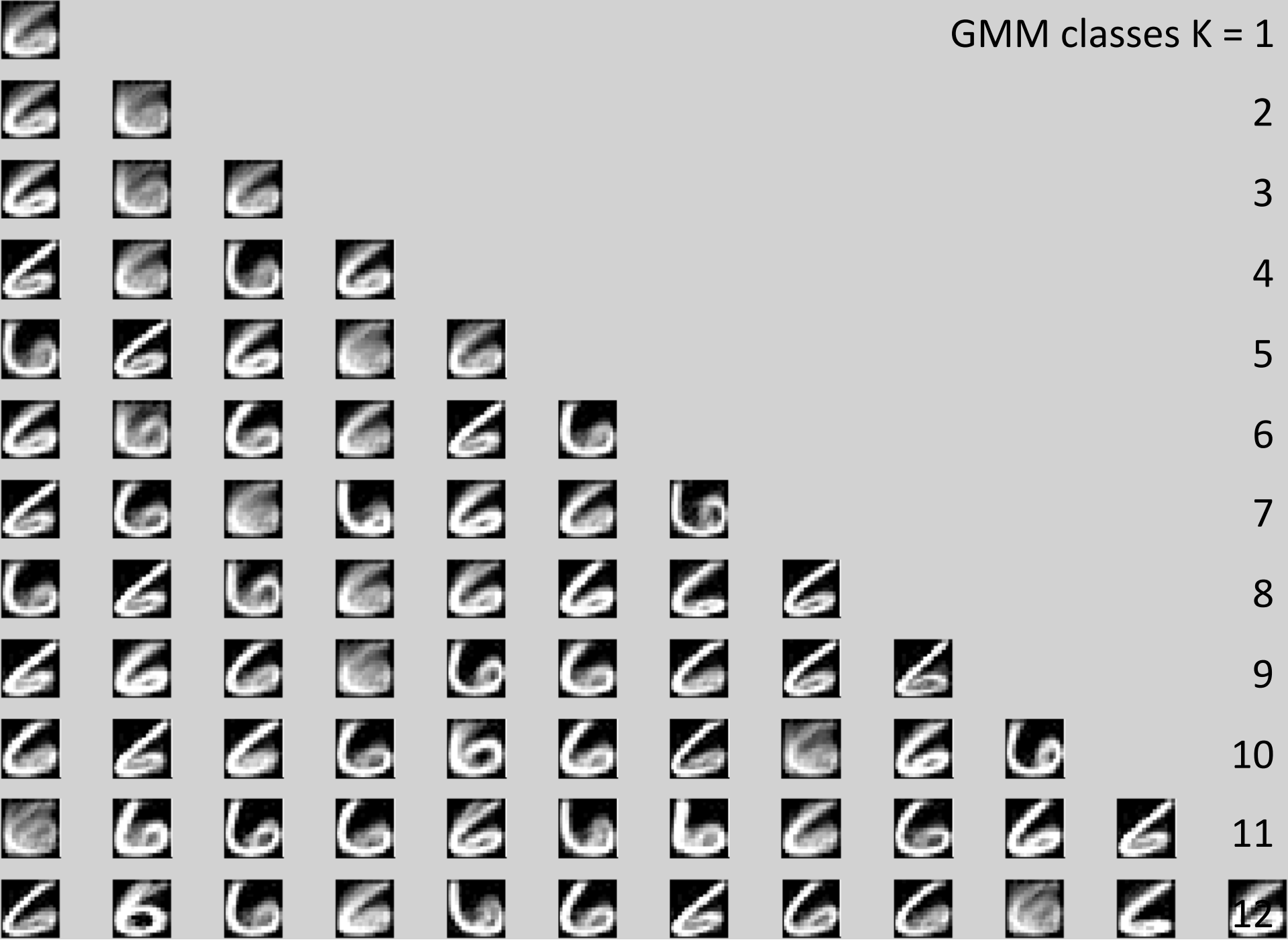

GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12

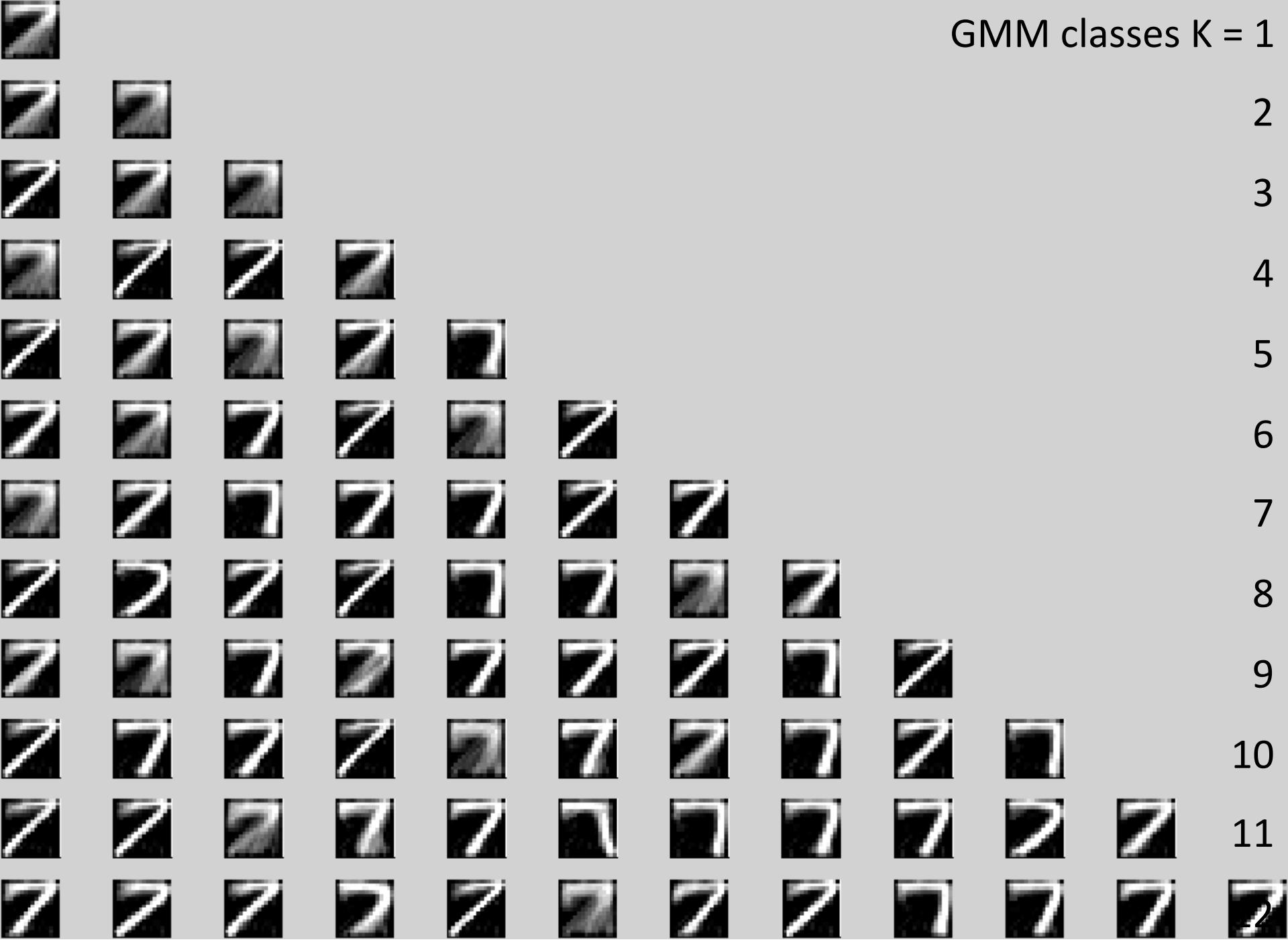

GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12

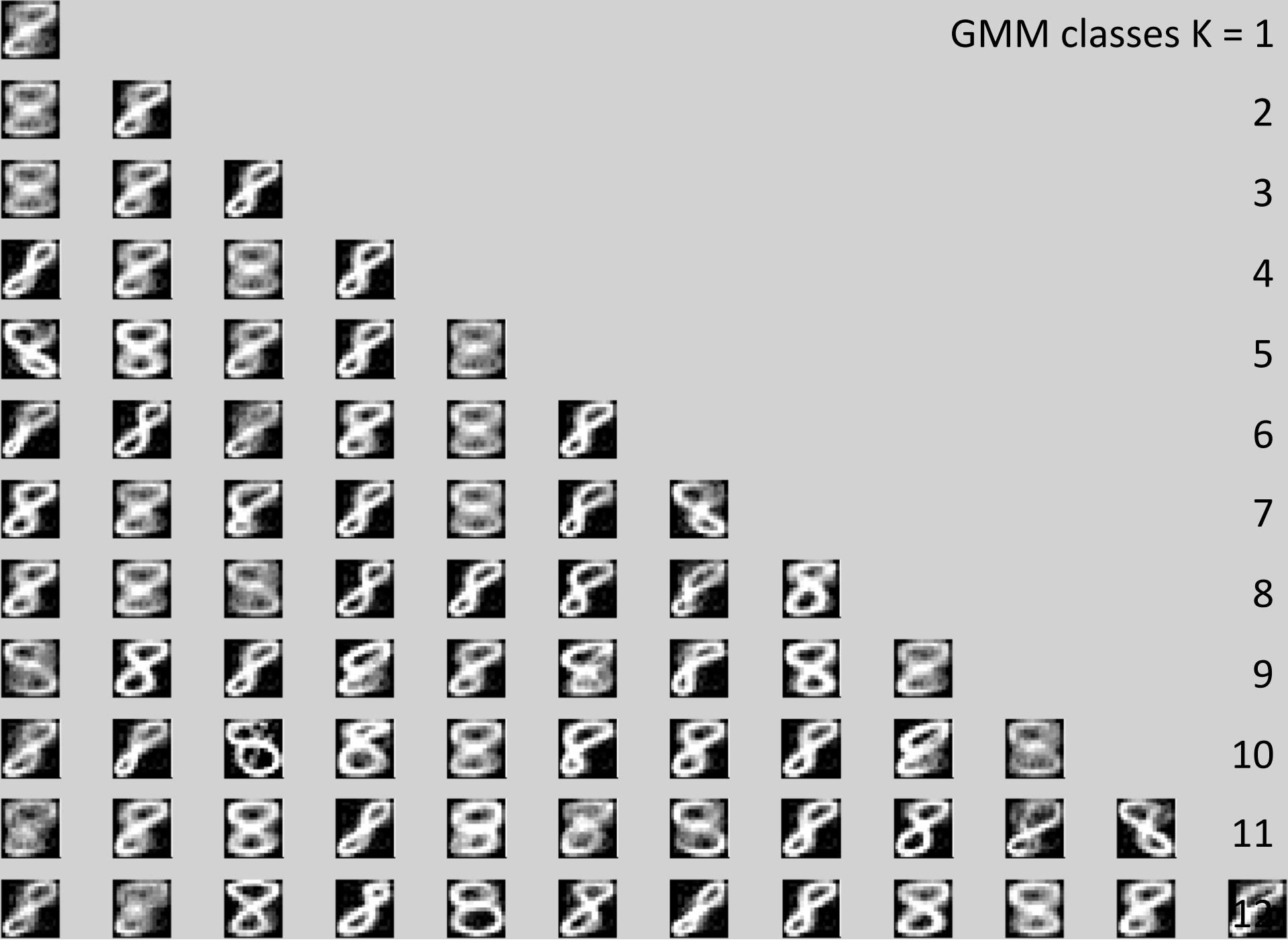

GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12

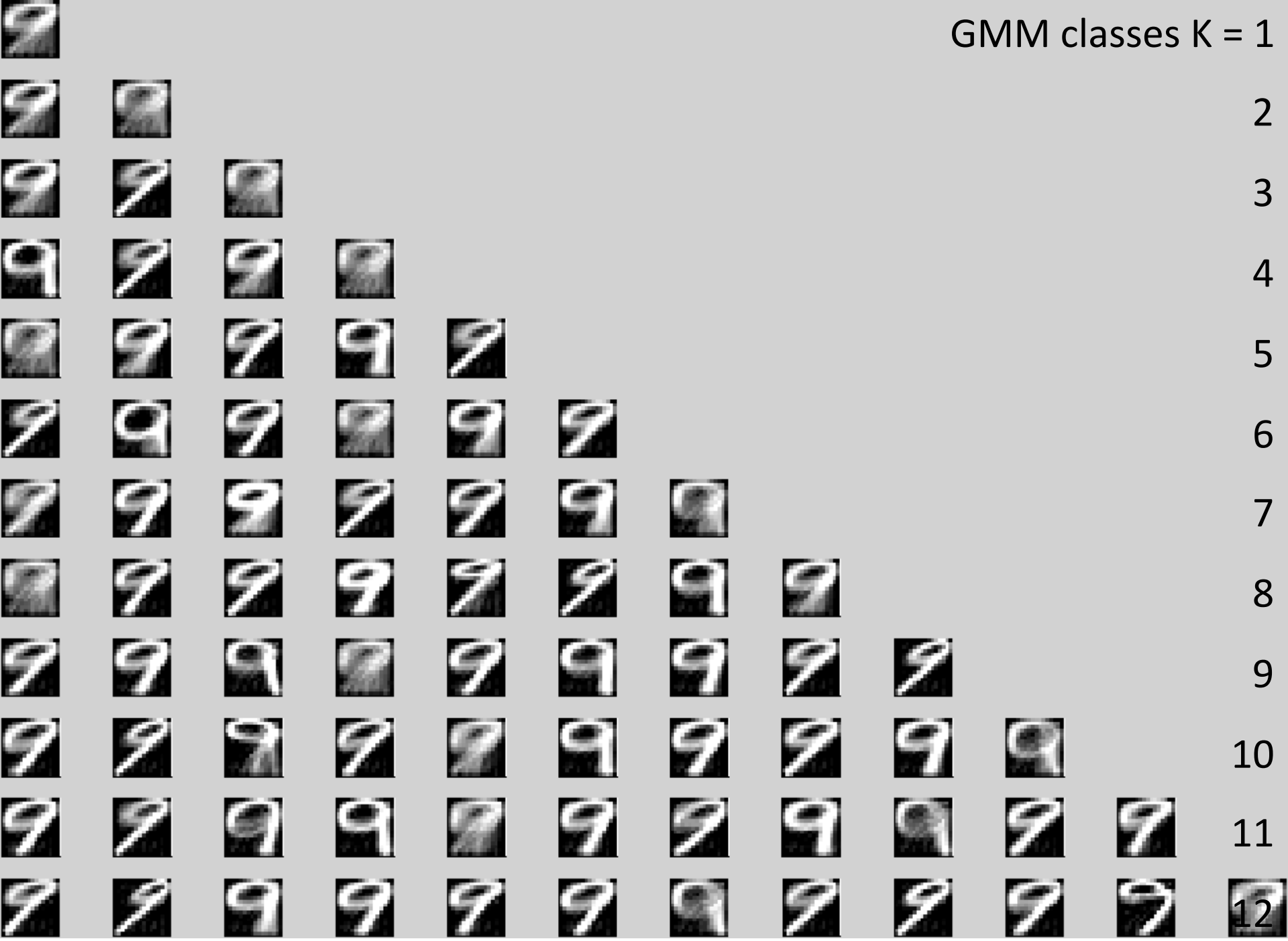

GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
15

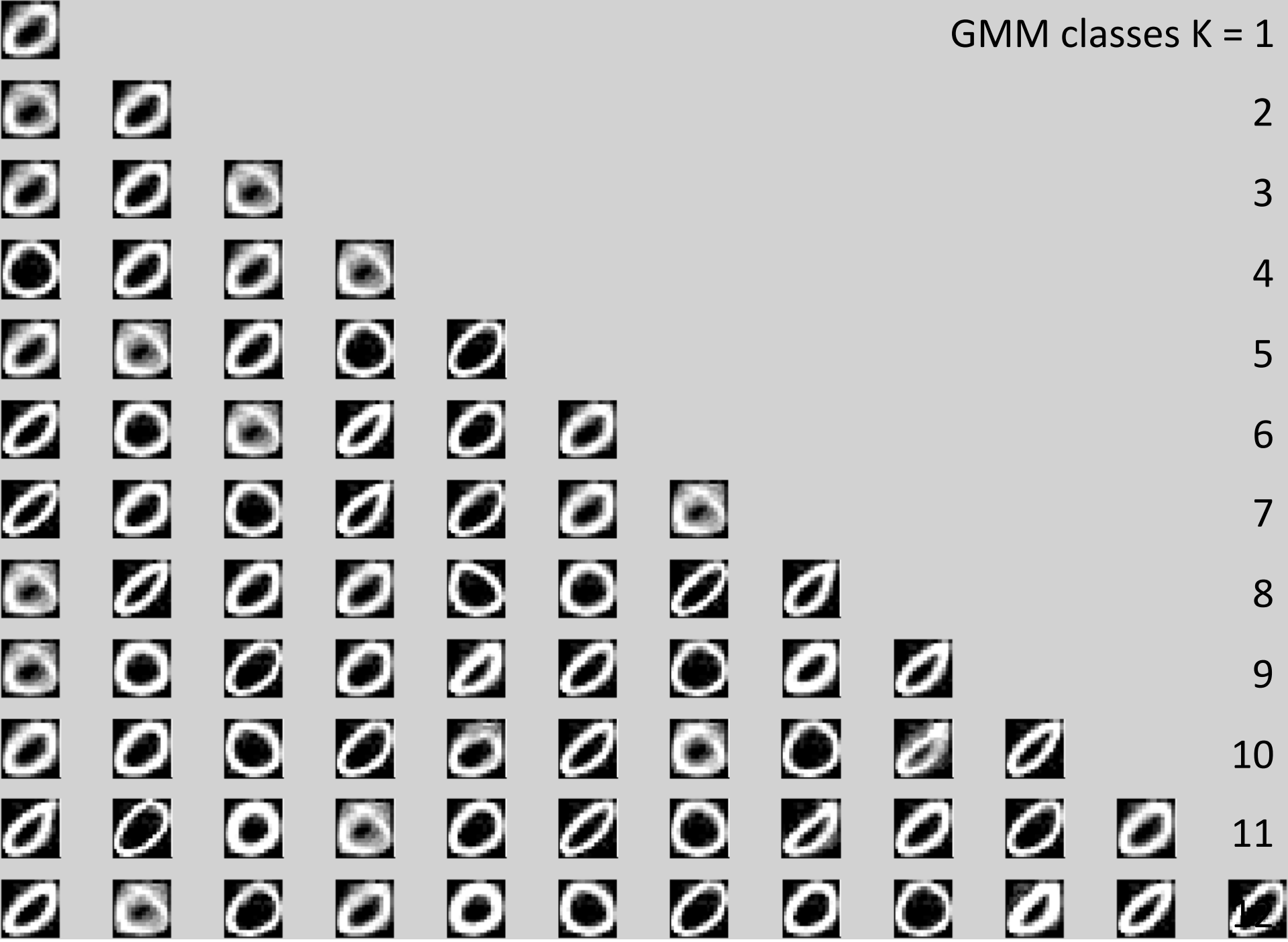

GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12


GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12

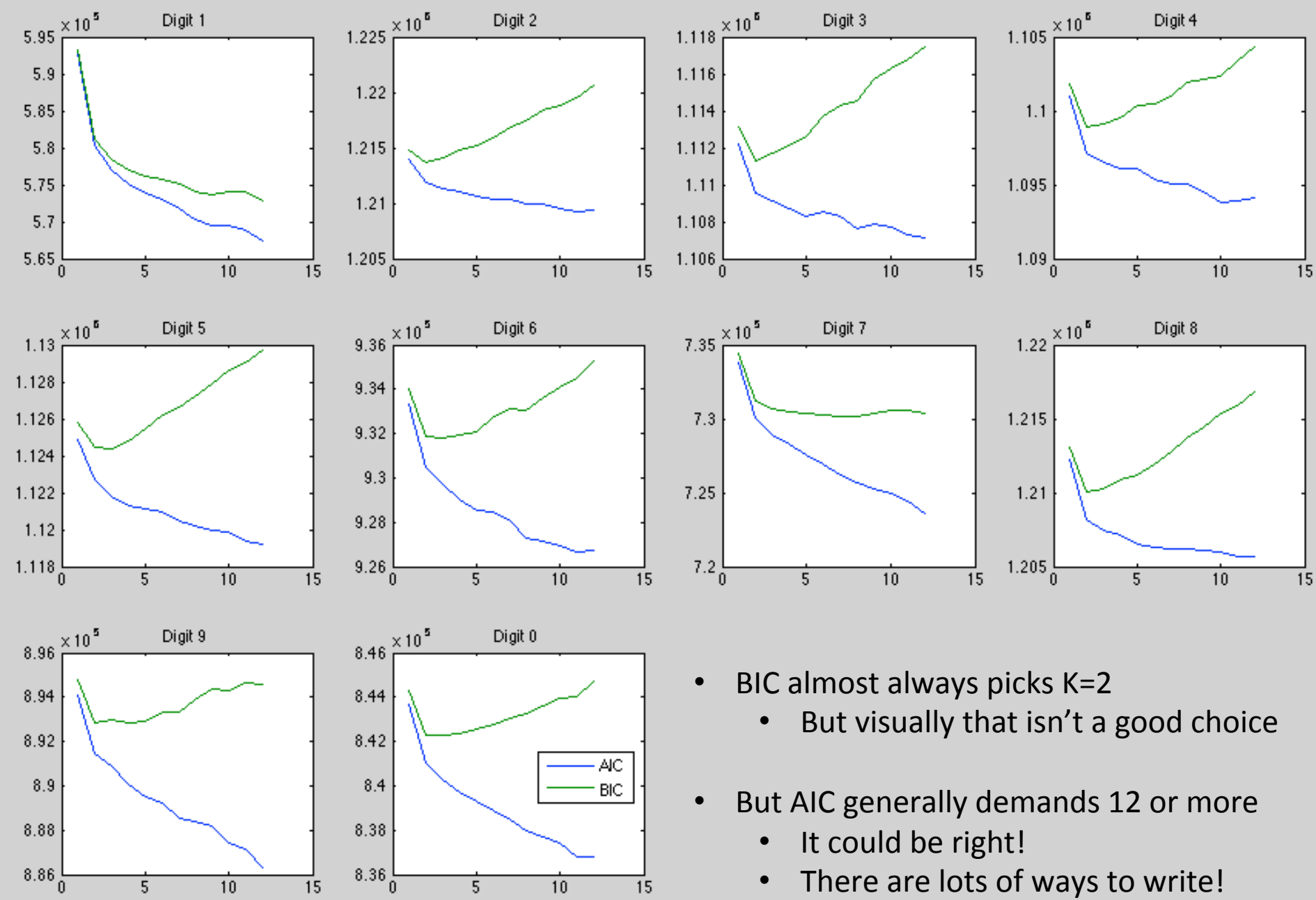

GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
2


GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12


GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12


GMM classes K = 1
2
3
4
5
6
7
8
9
10
11
12

AIC vs BIC for selection of K



- BIC almost always picks K=2
  - But visually that isn't a good choice
- But AIC generally demands 12 or more
  - It could be right!
  - There are lots of ways to write!

Finally… compare to eigenvectors from PCA… GMM $\boldsymbol{\mu}_k$'s much more interpretable!