CPSC 340: Machine Learning and Data Mining

Sparse Matrix Factorization

Original version of these slides by Mark Schmidt, with modifications by Mike Gelbart.

Admin

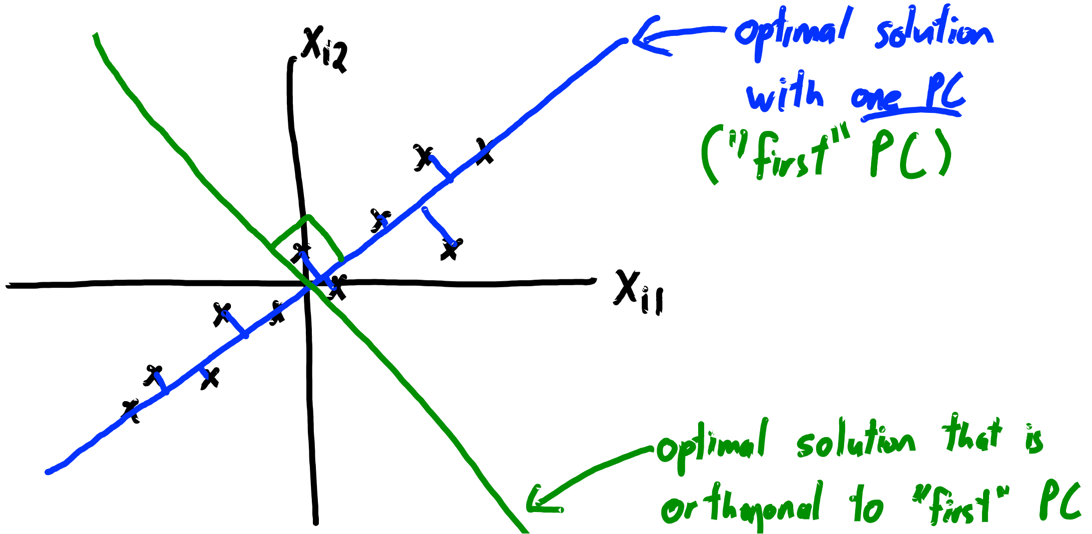
- Assignment 5:
  - Due next Friday.

Last Time: PCA with Orthogonal/Sequential Basis

- When k = 1, PCA has a scaling problem.
- When k > 1, have scaling, rotation, and label switching.
  - Standard fix: use normalized orthogonal rows W_c of 'W'.

$$\|w_c\| = 1 \quad \text{and} \quad w_c^T w_{c'} = 0 \quad \text{for } c' \neq c$$

  - And fit the rows in order:
    - First row "explains the most variance" or "reduces error the most".



Application: Face Detection

- Consider problem of face detection
- Classic methods use “eigenfaces” as basis:
  - PCA applied to images of faces.

Eigenfaces

- Collect a bunch of images of faces under different conditions:

Each row of X will be pixels in one image:

$X =$

If have 'n' images that are 'm' by 'm' then X is 'n' by m^2.

Eigenfaces

Compute <u>mean</u> μ_j of each column.

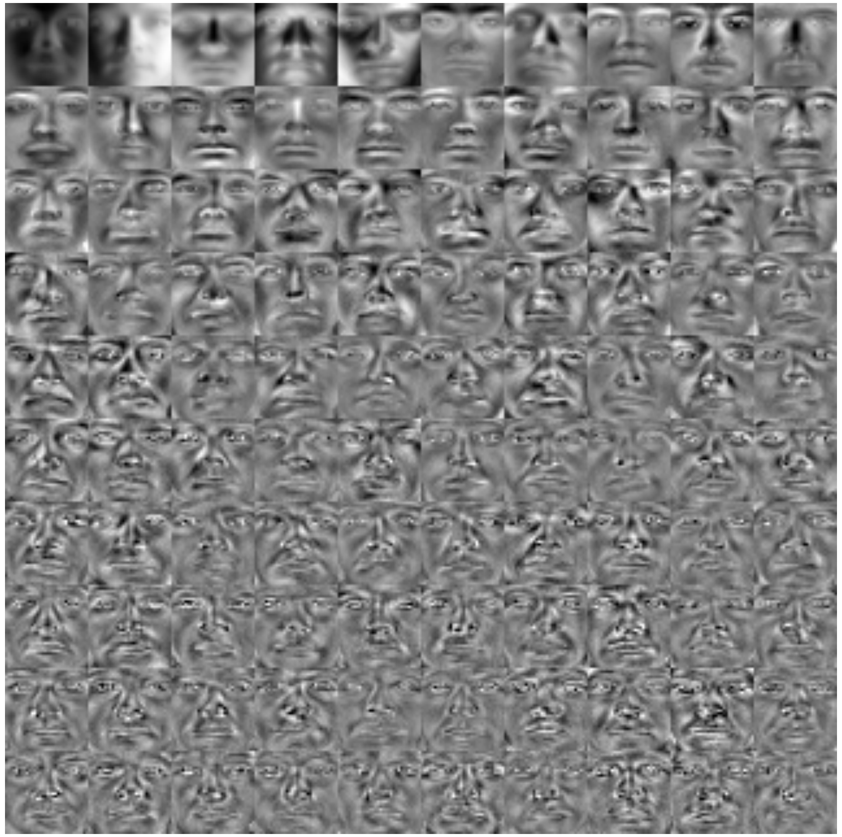
Replace each x_{ij} by $x_{ij} - \mu_j$

Each row of X will be pixels in <u>one</u> image:

$$X = \begin{bmatrix} \text{———} & x_1 - \mu & \text{———} \\ \text{———} & x_2 - \mu & \text{———} \\ & \vdots & \\ \text{———} & x_n - \mu & \text{———} \end{bmatrix}$$

Eigenfaces

Compute top 'k' PCs on centered data:



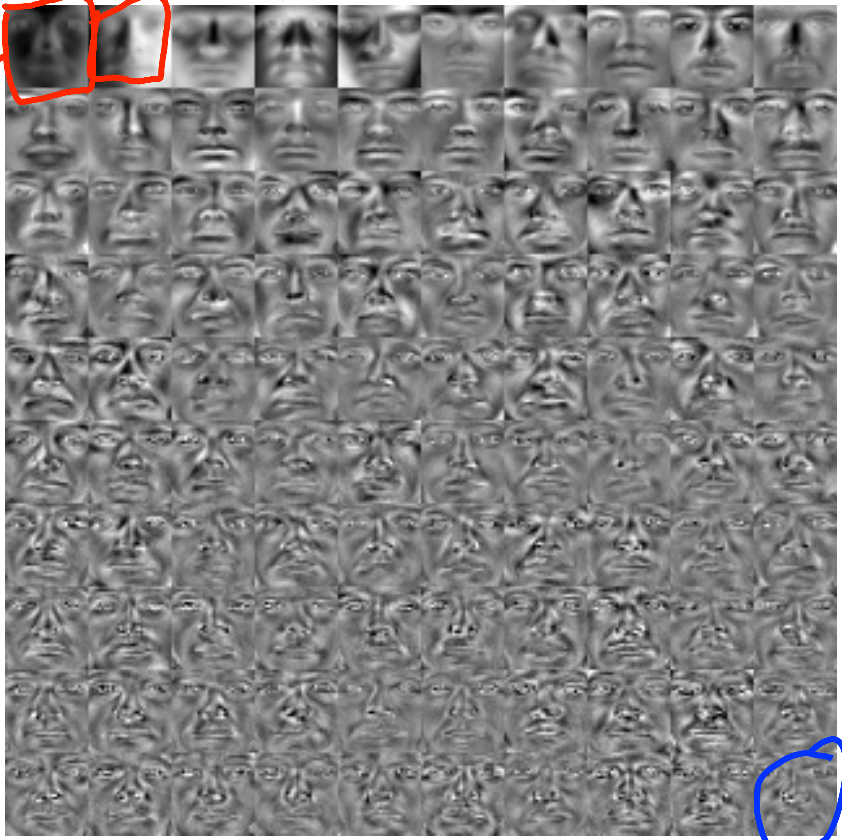
Each row of X will be pixels in <u>one</u> image:

$$X = \begin{bmatrix} \text{———} & x_1 - \mu & \text{———} \\ \text{———} & x_2 - \mu & \text{———} \\ & \vdots & \\ \text{———} & x_n - \mu & \text{———} \end{bmatrix}$$

Eigenfaces

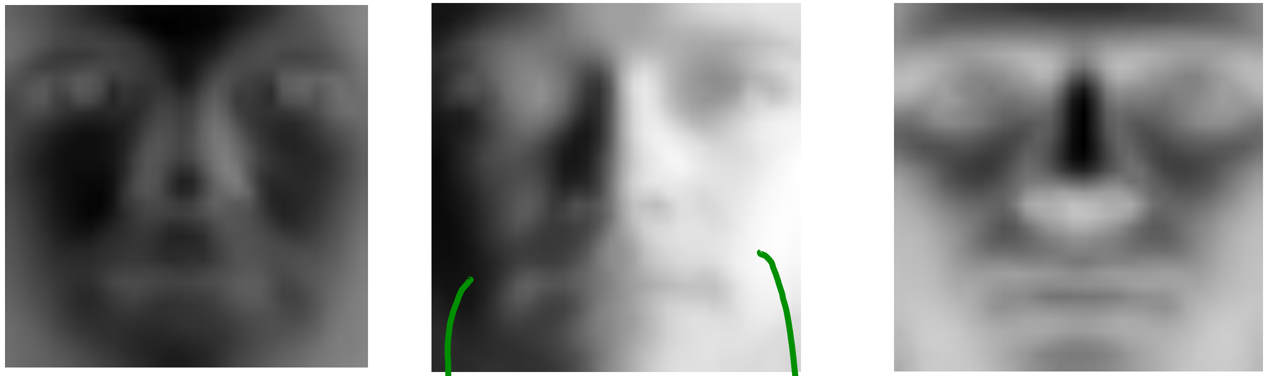
Compute top 'k' PCs on centered data:

PC2



PC1

Note that these are "signed" images.



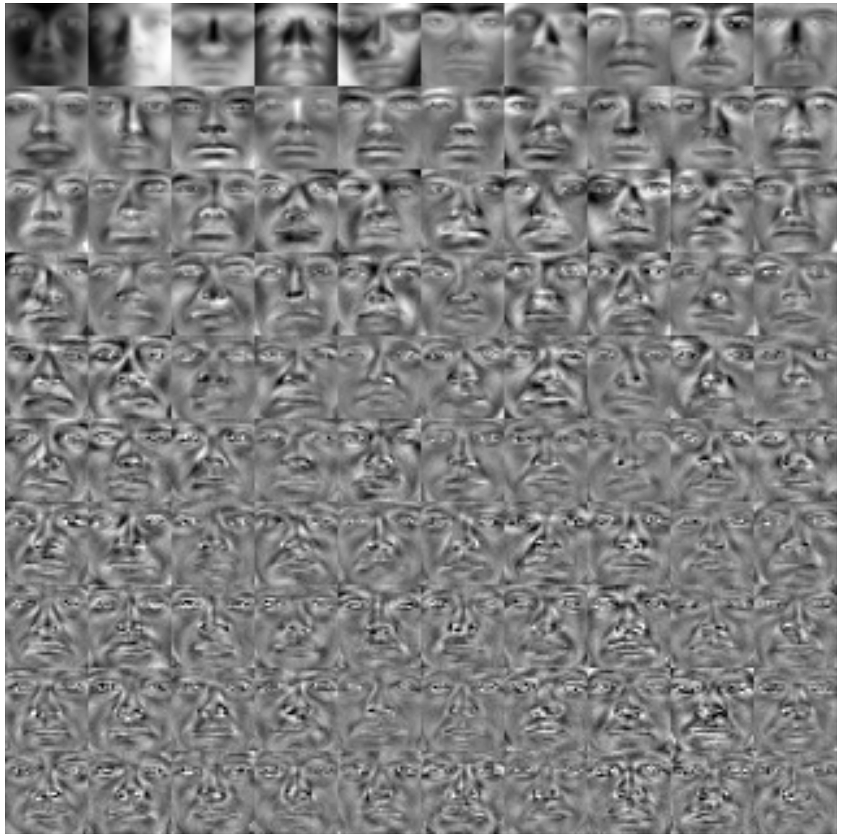
"gray" represents values close to 0.

"dark" represents negative values

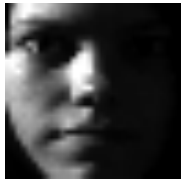
"bright" represents positive values

Eigenfaces

Compute top 'k' PCs on centered data:



"Eigenface" representation:

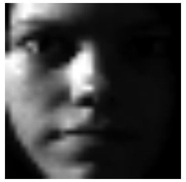
 =  $+ z_{i1}$  $+ z_{i2}$  $+ z_{i3}$  $+ \dots$

$\hat{x}_i$ μ PC1 (first row of W)   PC2   PC3

Eigenfaces

100 of the original faces:

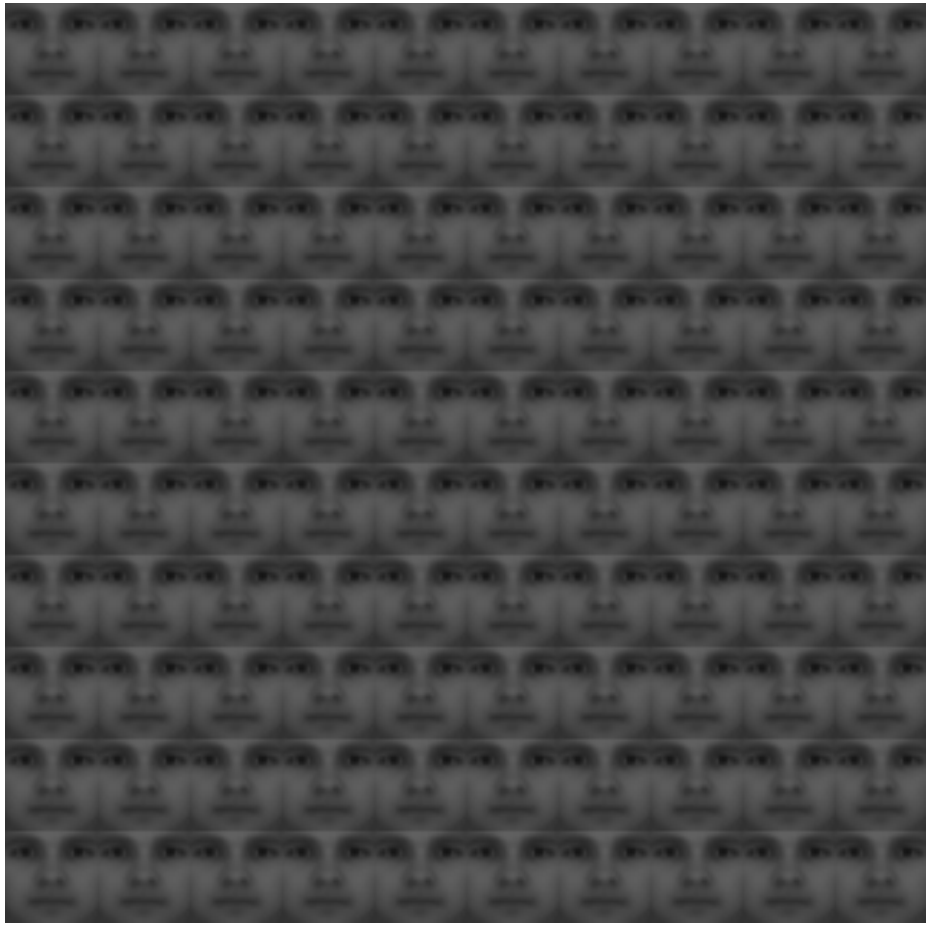
"Eigenface" representation:

 $=$  $+ z_{i1}$  $+ z_{i2}$  $+ z_{i3}$  $+ \dots$

$\hat{x}_i$ $\quad$ μ $\quad$ PC1 (first row of W) $\quad$ PC2 $\quad$ PC3

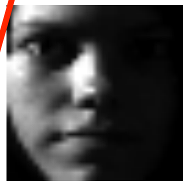
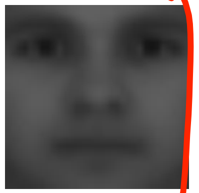
Eigenfaces

Reconstruction with $k = 0$



Variance explained: 0%

"Eigenface" representation:

 =  $+ z_{i1}$  $+ z_{i2}$  $+ z_{i3}$  $+ \dots$

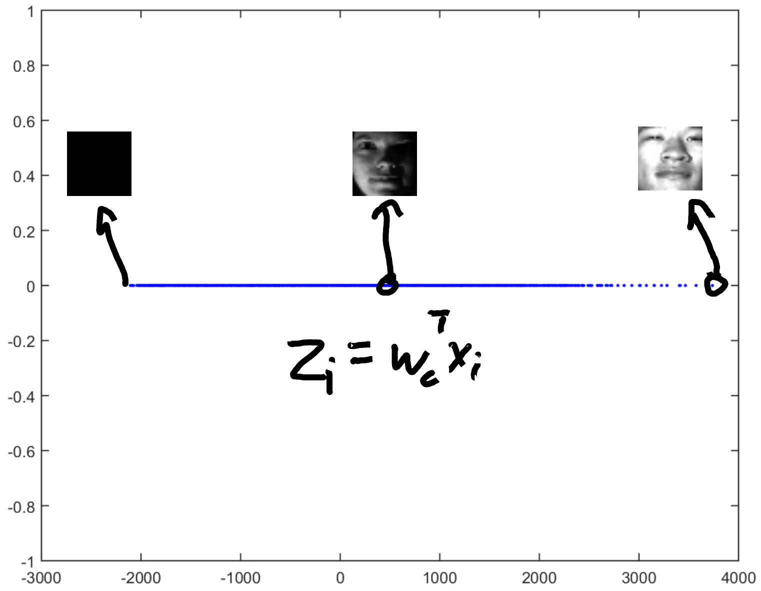
$\hat{x}_i$ — μ — PC1 (first row of W) — PC2 — PC3

Eigenfaces

Reconstruction with $k=1$

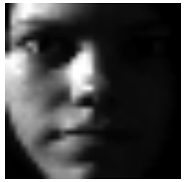
Variance explained: 34%

PCA Visualization:



$z_i = w_c^T x_i$

"Eigenface" representation:



$\hat{x}_i$

$=$

μ

$+ z_{i1}$

PC1

(first row of W)

$+ z_{i2}$

PC2

$+ z_{i3}$

PC3

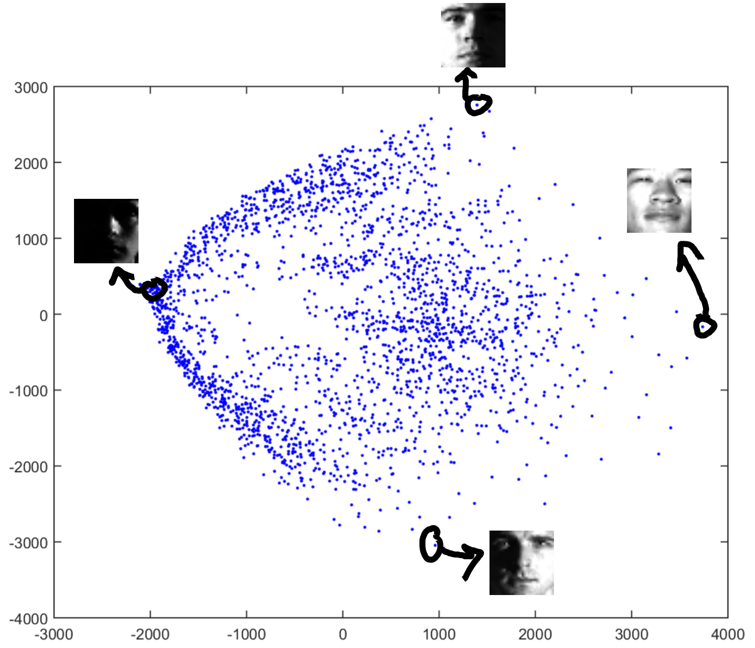
$+ \cdots$

Eigenfaces

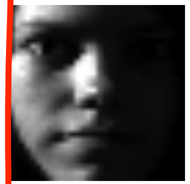
Reconstruction with $k=2$

Variance explained: 71%

PCA Visualization:



"Eigenface" representation:

 $=$  $+ z_{i1}$  $+ z_{i2}$  $+ z_{i3}$  $+ \cdots$

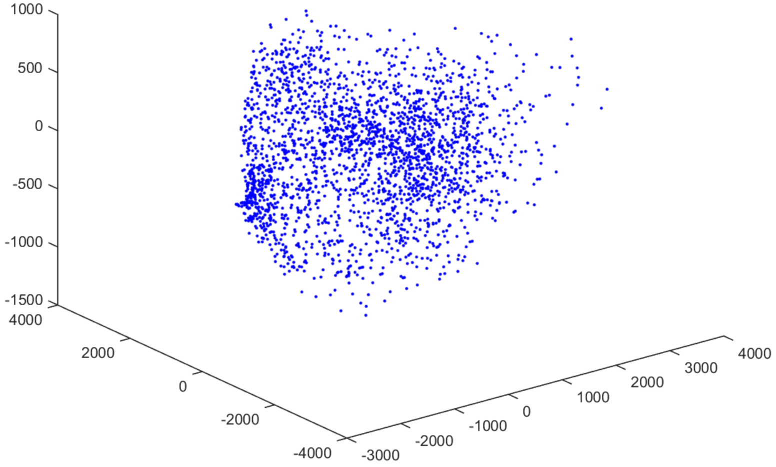
$\hat{x}_i$ μ PC1 (first row of W) PC2 PC3

Eigenfaces

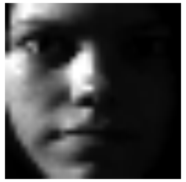
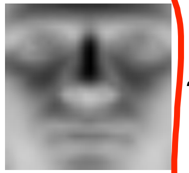
Reconstruction with $k=3$

Variance explained: 76%

PCA Visualization:



"Eigenface" representation:

 =  $+ z_{i1}$  $+ z_{i2}$  $+ z_{i3}$  $+ \dots$

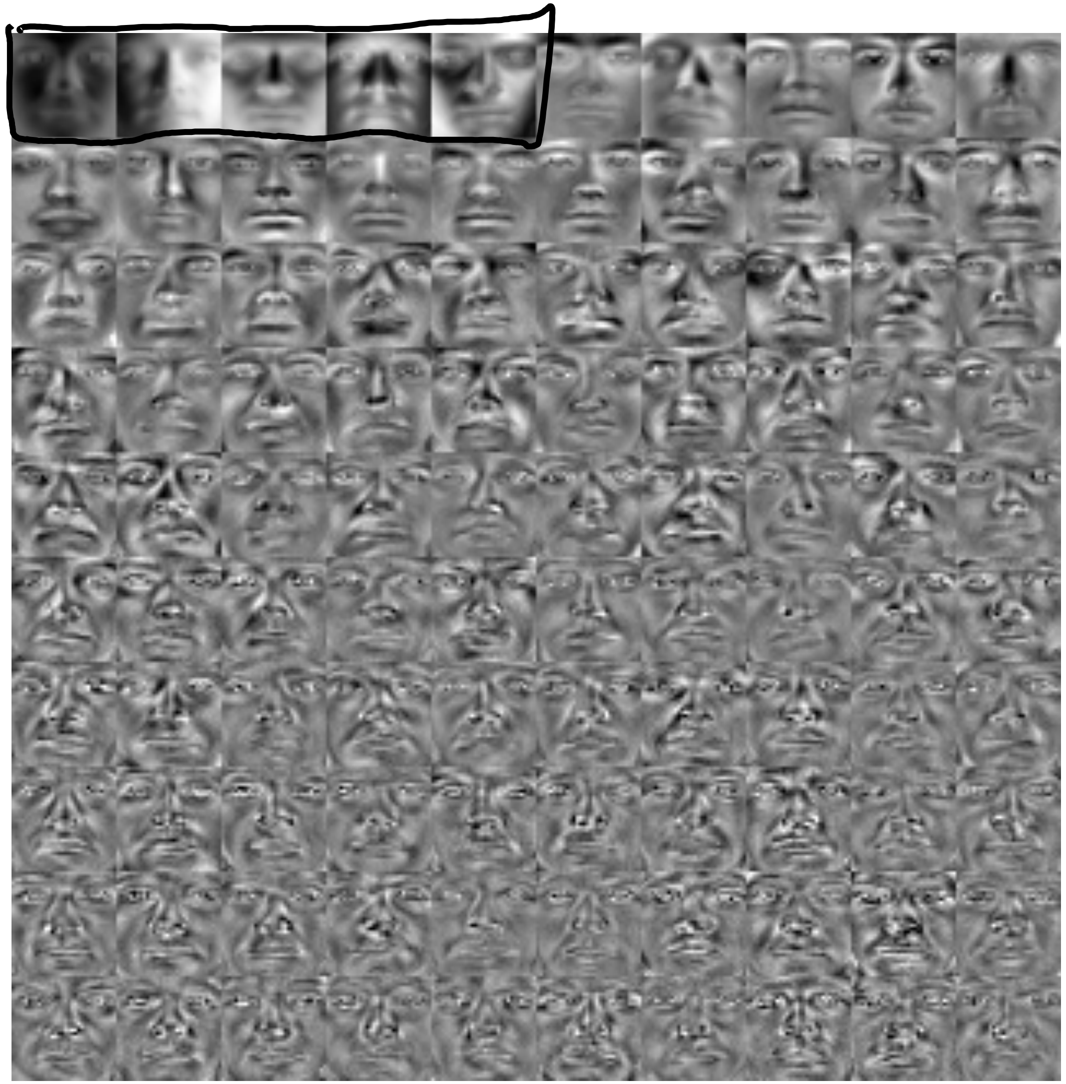
$\hat{x}_i \qquad \mu \qquad PC1 \qquad PC2 \qquad PC3$

(first row of W)

Eigenfaces

Reconstruction with $k=5$

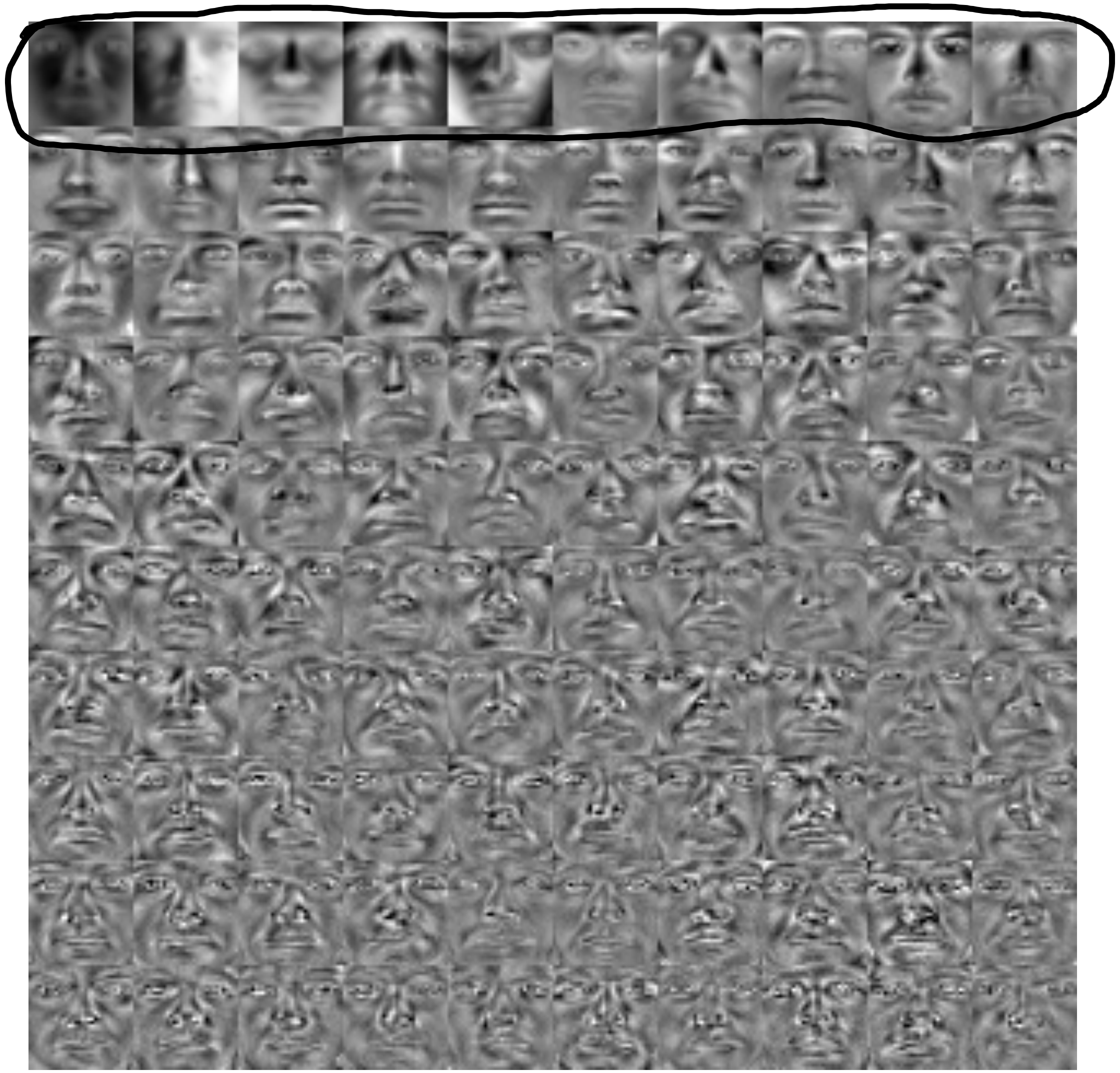
Variance explained: 80%



Eigenfaces

Reconstruction with k= 10

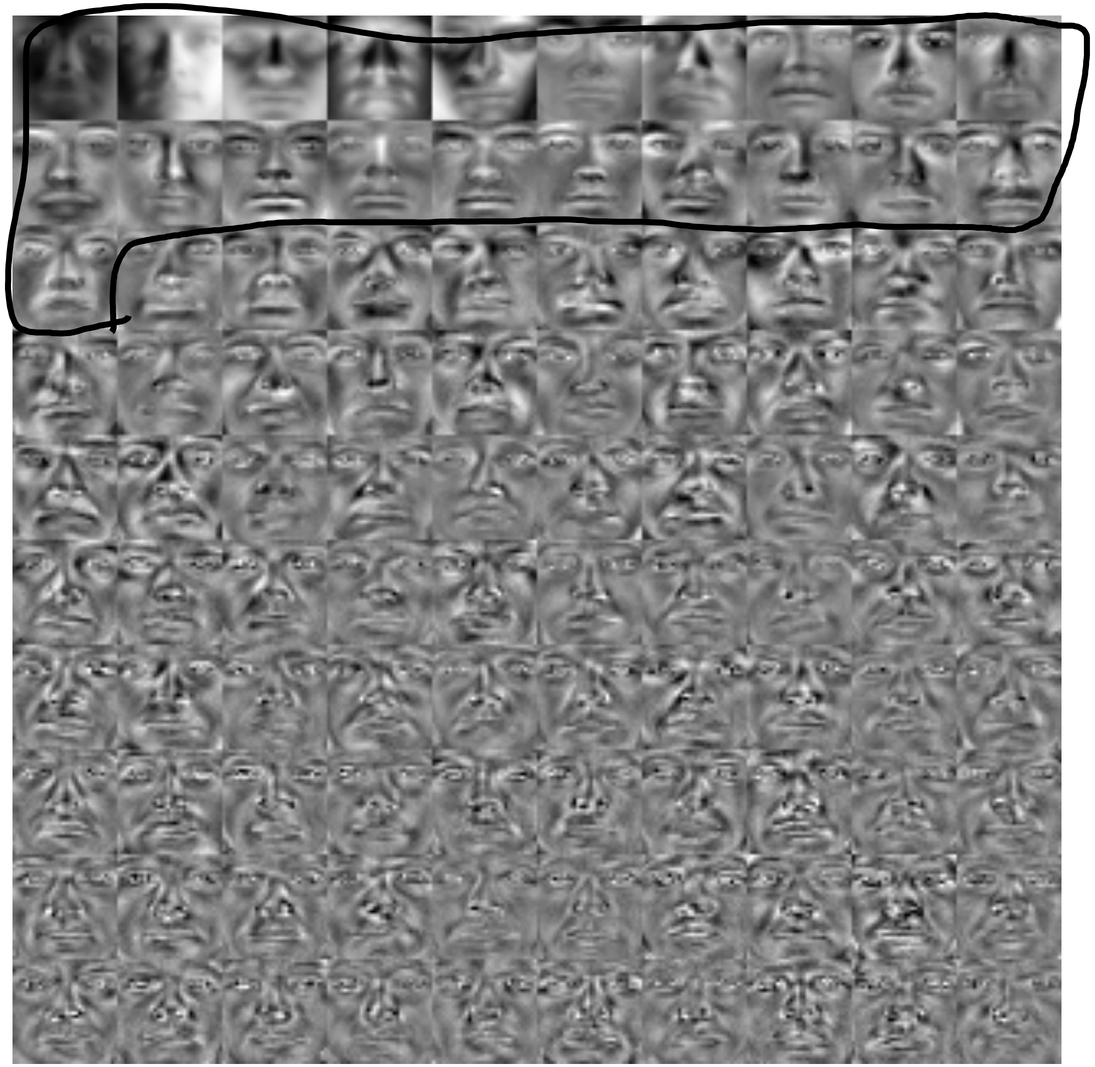
Variance explained: 85%



Eigenfaces

Reconstruction with $k=21$

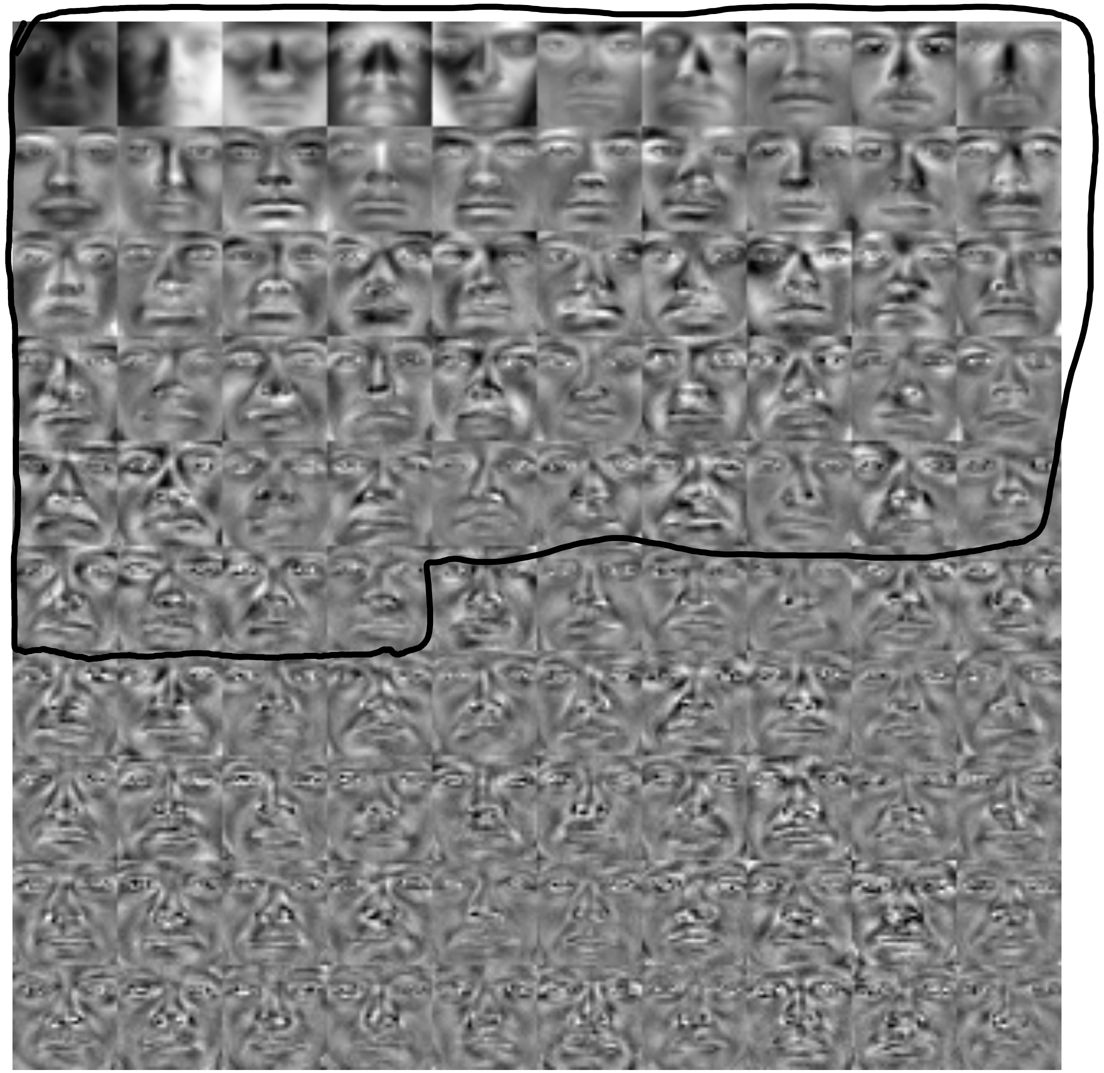
Variance explained: 90%



Eigenfaces

Reconstruction with k=54

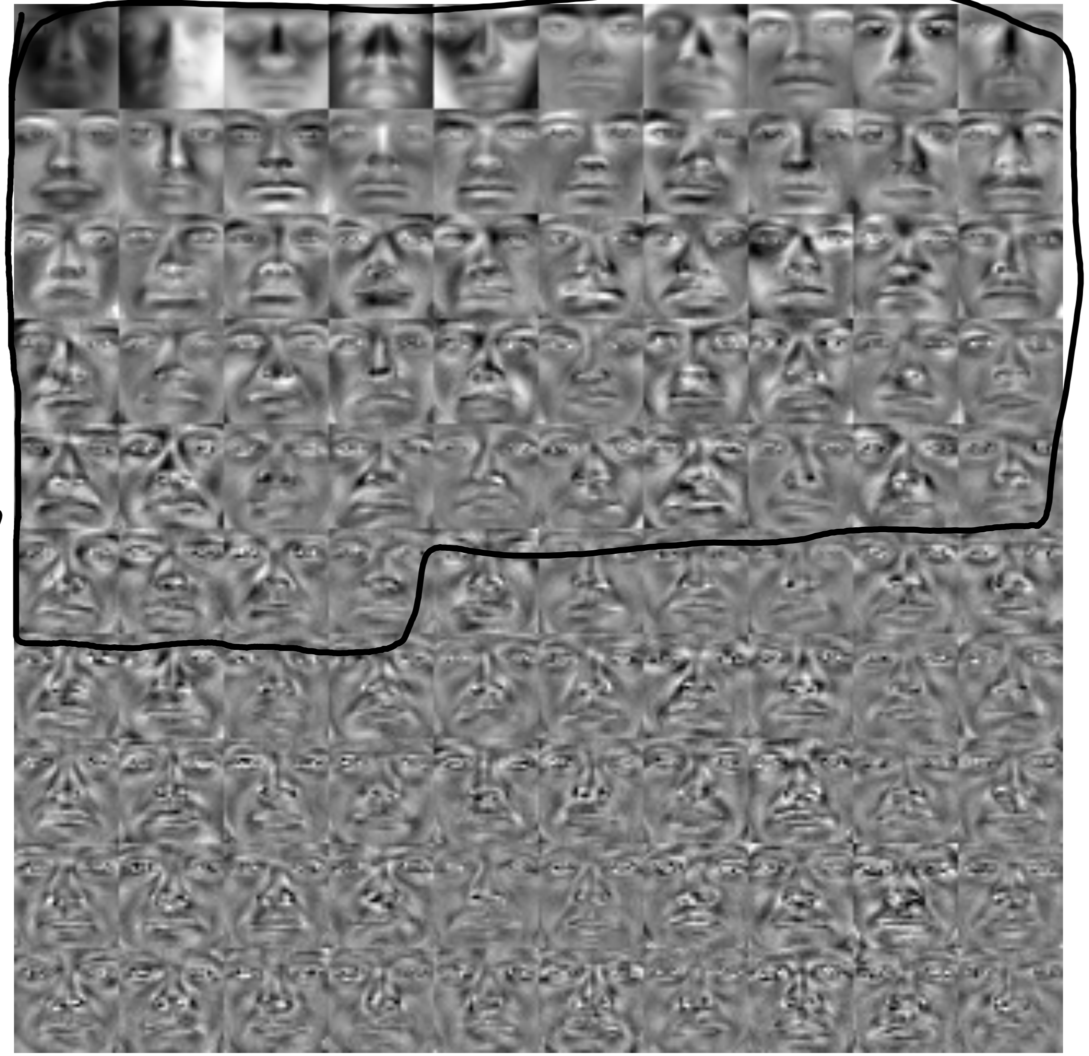
Variance explained: 95%



Eigenfaces

Original Images again:

Plus these "eigenfaces" and the mean.



We can replace 1024 x_i values by 54 z_i values

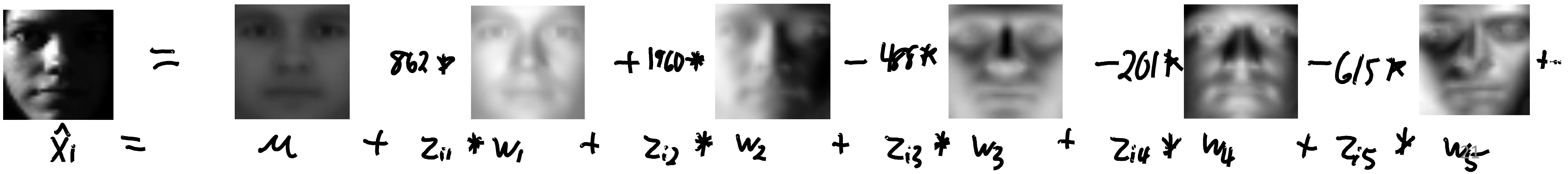
VQ vs. PCA vs. NMF

- But how *should* we represent faces?
  - Vector quantization (k-means).
    - Replace face by the average face in a cluster.
    - Can’t distinguish between people in the same cluster (only ‘k’ possible faces).

$$\hat{x}_i = z_{i1} * w_1 + z_{i2} * w_2 + z_{i3} * w_3 + z_{i4} * w_4 + z_{i5} * w_5 + z_{i6} * w_6$$

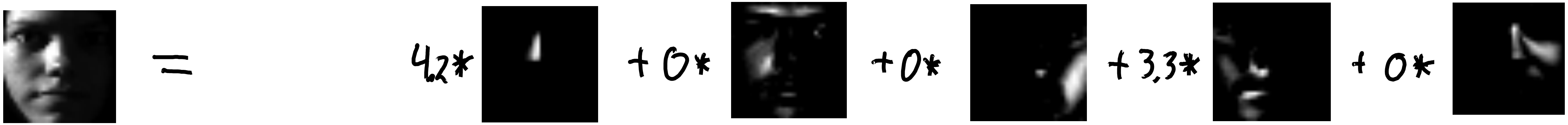
VQ vs. PCA vs. NMF

- But how *should* we represent faces?
  - Vector quantization (k-means).
  - PCA (orthogonal basis).
    - Global average plus linear combination of “eigenfaces”.
    - But “eigenfaces” are not intuitive ingredients for faces.
      - PCA tends to use positive/negative cancelling bases.



VQ vs. PCA vs. NMF

- But how *should* we represent faces?
  - Vector quantization (k-means).
  - PCA (orthogonal basis).
  - NMF (non-negative matrix factorization):
    - Instead of orthogonality/ordering in W, require W and Z to be non-negativity.
    - Example of "sparse coding":
      - The z_i are sparse so each face is coded by a small number of neurons.
      - The w_c are sparse so neurons tend to be "parts" of the object.



$= 4.2 * \ + 0 * \ + 0 * \ + 3.3 * \ + 0 * \ + \dots$

$\hat{x}_i = z_{i1} * w_1 + z_{i2} * w_2 + z_{i3} * w_3 + z_{i4} * w_4 + z_{i5} * w_5$

Warm-up to NMF: Non-Negative Least Squares

- Consider our usual least squares problem:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^Tx_i - y_i)^2$$

- But assume y_i and elements of x_i are non-negative:
  - Could be sizes ('height', 'milk', 'km') or counts ('vicodin', 'likes', 'retweets').
- Assume we want elements of 'w' to be non-negative, too:
  - Maybe no sensible interpretation to negative weights.

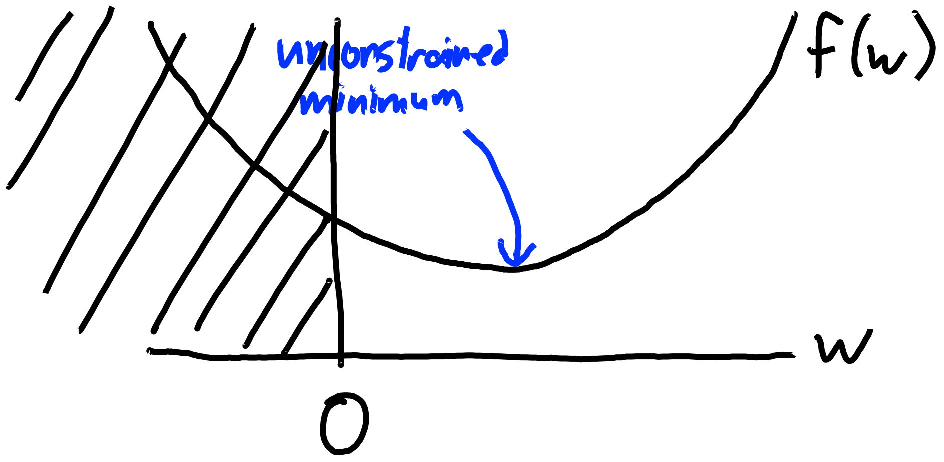
- Non-negativity leads to sparsity...

Sparsity and Non-Negative Least Squares

- Consider 1D non-negative least squares objective:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w x_i - y_i)^2 \quad \text{with} \quad w > 0$$

- Plotting the (constrained) objective function:



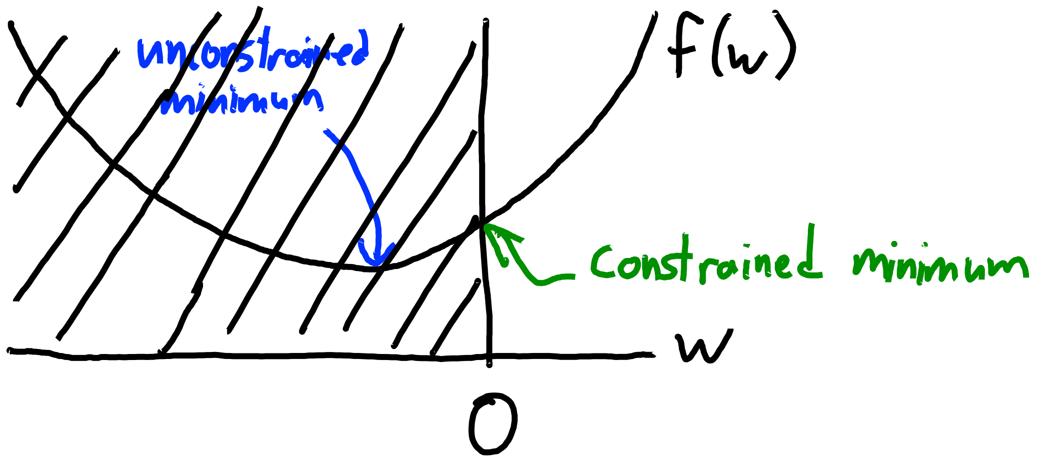
- In this case, non-negative solution is least squares solution.

Sparsity and Non-Negative Least Squares

- Consider 1D non-negative least squares objective:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w x_i - y_i)^2 \quad \text{with} \quad w \geq 0$$

- Plotting the (constrained) objective function:



- In this case, non-negative solution is w = 0.

Sparsity and Non-Negativity

- Similar to L1-regularization, non-negativity leads to sparsity.
  - Also regularizes: w_j are smaller since can't "cancel" out negative values.
- How can we minimize f(w) with non-negative constraints?
  - Naive approach: solve least squares problem, set negative w_j to 0.

    Compute $w = (X^T X) \backslash (X^T y)$

    Set $w_j = \max\{0, w_j\}$

  - This is correct when d = 1.
  - Doesn't make sense when d ≥ 2.
    - Consider two collinear or almost collinear features, with w_1=10 and w_2=-10
    - Setting w_1=w_2=0 might be OK, but setting w_1=10 and w_2=0 is wrong.

Sparsity and Non-Negativity

- Similar to L1-regularization, non-negativity leads to sparsity.
  - Also regularizes: w_j are smaller since can't "cancel" out negative values.
- How can we minimize f(w) with non-negative constraints?
  - A correct approach is projected gradient algorithm:
    - Run a gradient descent iteration:

$$w^{t+1/2} = w^t - \alpha^t \nabla f(w^t)$$

    - After each step, set negative values to 0.

$$w_j^{t+1} = \max\{0, w_j^{t+1/2}\}$$

    - Repeat.

Sparsity and Non-Negativity

- Similar to L1-regularization, non-negativity leads to sparsity.
  - Also regularizes: w_j are smaller since can't "cancel" out negative values.
- How can we minimize f(w) with non-negative constraints?
  - A correct approach is projected gradient algorithm:

$$w^{t+1/2} = w^t - \alpha^t \nabla f(w^t) \qquad\qquad w_j^{t+1} = \max\{0, w_j^{t+1/2}\}$$

  - Similar properties to gradient descent:
    - Guaranteed decrease of 'f' if α_t is small enough.
    - Reaches local minimum under weak assumptions (global minimum for convex 'f').
      - Least squares objective is still convex when restricted to non-negative variables.
    - Generalizations allow things like L1-regularization instead of non-negativity.

(findMinL1)

Projected-Gradient for NMF

- Back to the non-negative matrix factorization (NMF) objective:

$$f(W,Z) = \sum_{i=1}^{n}\sum_{j=1}^{d}((w^j)^T z_i - x_{ij})^2 \quad \text{with } w_{cj} \geq 0 \text{ and } z_{ij} \geq 0$$

  – Different ways to use projected gradient:
    - Alternate between projected gradient steps on 'W' and on 'Z'.
    - Or run projected gradient on both at once.
    - Or sample a random 'i' and 'j' and do stochastic projected gradient.

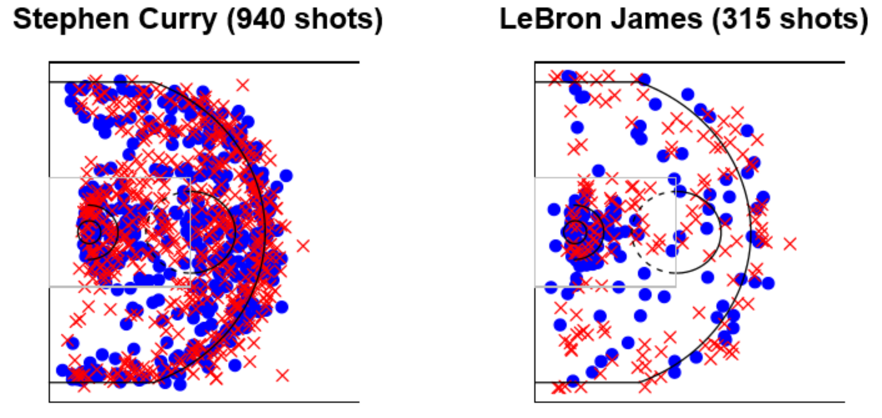
Set $z_i^{t+1} = z_i^t - \alpha^t \nabla_{z_i} f(W,Z)$ and $(w^j)^{t+1} = (w^j)^t - \alpha^t \nabla_{w^j} f(W,Z)$ for selected i and j

$$\underbrace{\text{(keep other values of } W \text{ and } Z \text{ fixed)}}_{\text{Then set negative values to 0.}}$$

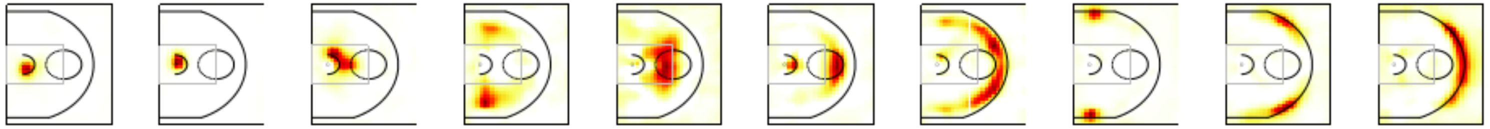
  – Non-convex and (unlike PCA) is sensitive to initialization.
    - Hard to find the global optimum.
    - Typically use random initialization.

Application: Sports Analytics

- NBA shot charts:



- NMF (using “KL divergence” loss with k=10 and smoothed data).
  - Negative values would not make sense here.



LeBron James	0.21	0.16	0.12	0.09	0.04	0.07	0.00	0.07	0.08	0.17
Brook Lopez	0.06	0.27	0.43	0.09	0.01	0.03	0.08	0.03	0.00	0.01
Tyson Chandler	0.26	0.65	0.03	0.00	0.01	0.02	0.01	0.01	0.02	0.01
Marc Gasol	0.19	0.02	0.17	0.01	0.33	0.25	0.00	0.01	0.00	0.03
Tony Parker	0.12	0.22	0.17	0.07	0.21	0.07	0.08	0.06	0.00	0.00
Kyrie Irving	0.13	0.10	0.09	0.13	0.16	0.02	0.13	0.00	0.10	0.14
Stephen Curry	0.08	0.03	0.07	0.01	0.10	0.08	0.22	0.05	0.10	0.24
James Harden	0.34	0.00	0.11	0.00	0.03	0.02	0.13	0.00	0.11	0.26
Steve Novak	0.00	0.01	0.00	0.02	0.00	0.00	0.01	0.27	0.35	0.34

http://jmlr.org/proceedings/papers/v32/miller14.pdf

Application: Topic Modeling

- You have ‘n’ documents, ‘d’ bag-of-word features, want to find “topics”
- You can use NMF for this!
  - Interpretation of W: k topics, each with a selection of words
  - Interpretation of Z: each movie is a mixture of the k topics
- NMF makes much more sense than PCA
  - Each topic involves a small number of words
  - Each document has a small number of topics
- PCA would not make sense
  - you could have negative inclusion of a topic for a document
  - Topics can have negative words
  - all documents are a mixture of every possible topic and all topics involve every possible word
- So here we like both the sparsity and the non-negativity

Regularized Matrix Factorization

- More recently people have considered L2-regularized PCA:

$$f(W,Z) = \frac{1}{2}\|ZW - X\|_F^2 + \frac{\lambda_1}{2}\|W\|_F^2 + \frac{\lambda_2}{2}\|Z\|_F^2$$

- Replaces normalization/orthogonality/sequential-fitting.
  - But requires regularization parameters λ_1 and λ_2.

- Need to regularize W and Z because of scaling problem:
  - If you only regularize 'W' it doesn't do anything:
    - I could take unregularized solution, replace W by αW for a tiny α to shrink $\|W\|_F$ as much as I want, then multiply Z by $(1/\alpha)$ to get same solution.
  - Similarly, if you only regularize 'Z' it doesn't do anything.

Sparse Matrix Factorization

- Instead of non-negativity, we could use L1-regularization:

$$f(W,Z) = \frac{1}{2}\|ZW - X\|_F^2 + \frac{\lambda_1}{2}\sum_{i=1}^{n}\|z_i\|_1 + \frac{\lambda_2}{2}\sum_{j=1}^{d}\|w_j\|_1$$

  - Called sparse coding (L1 on 'Z') or sparse dictionary learning (L1 on 'W').
  - sklearn's SparsePCA is L1 and 'W' and L2 on 'Z'
- Disadvantage of using L1-regularization over non-negativity:
  - Sparsity controlled by λ_1 and λ_2 (so you need to set these)
- Advantage of using L1-regularization:
  - Sparsity controlled by λ_1 and λ_2 (so you can control amount of sparsity)
  - Also, negative coefficients often make sense.

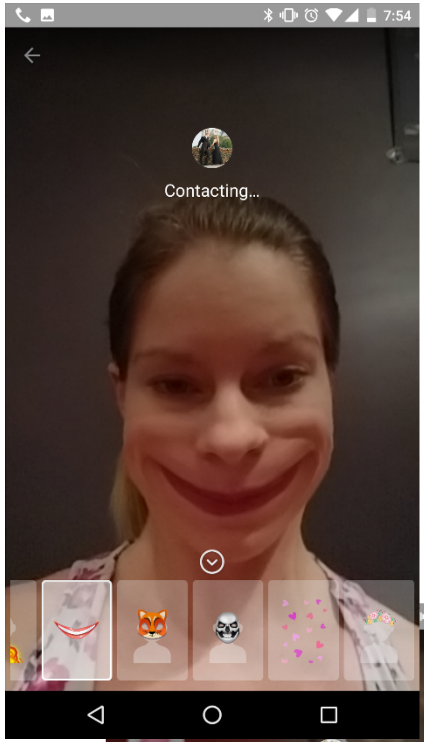
Sparsity: what is it good for?

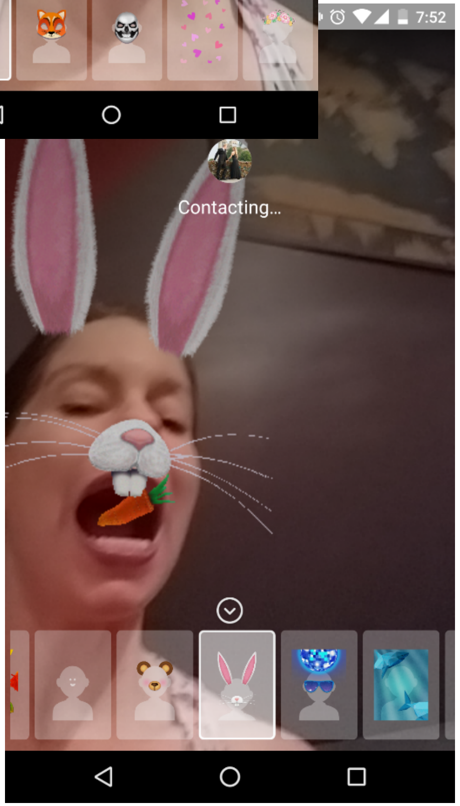
- Sparsity: a vector/matrix with a bunch of zeros (often our ‘w’)
- Can be achieved in several ways:
  - Explicit feature selection, L1 regularization, non-negativity constraints
- Intuition: we want something “explained by a few factors”
  - NMF leads to sparse Z and W, whereas PCA does not.
- There can be big computational gains
  - We said earlier than SVM+kernels are fast because of the small number of support vectors. This has to do with “sparsity in the dual” (see CPSC 406)
- There are biological motivations
  - We believe there is “sparse coding” in the brain (few neurons in a pattern)
  - This might also mean more energy efficiency (both in the brain and in our tech)

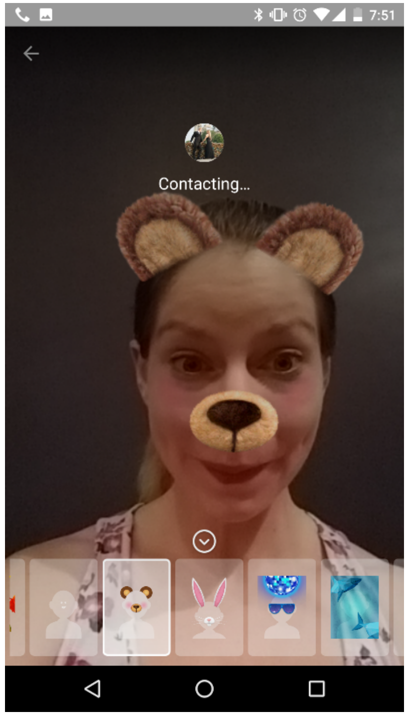
Summary

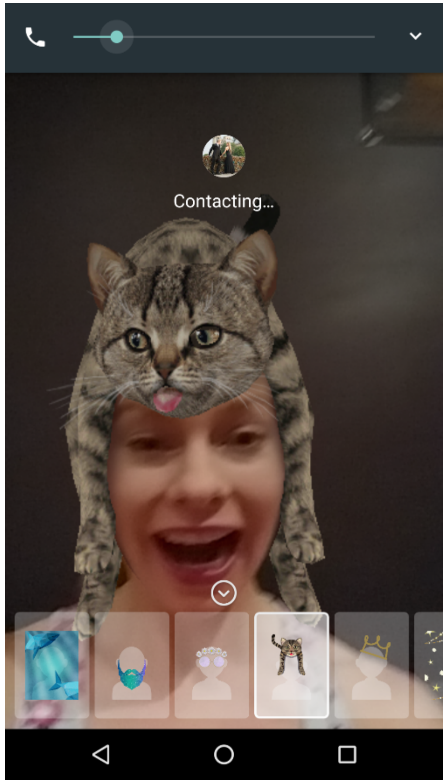
- Non-negative matrix factorization leads to sparse 'W' and 'Z'.
- Non-negativity constraints lead to sparse solution.
  - Projected gradient adds constraints to gradient descent.
- L1-regularization leads to other sparse latent-factor models.

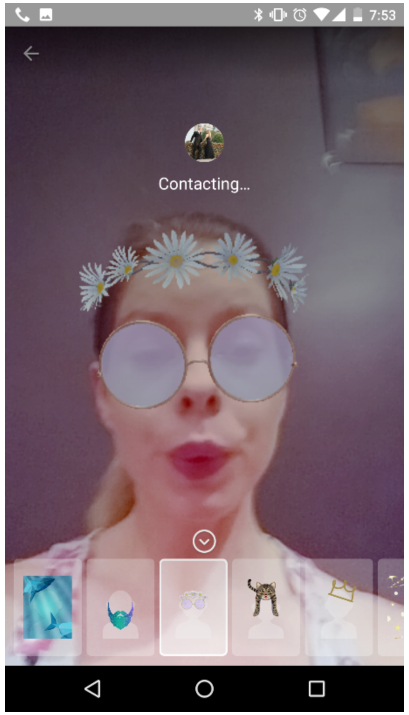
Application: Face Detection

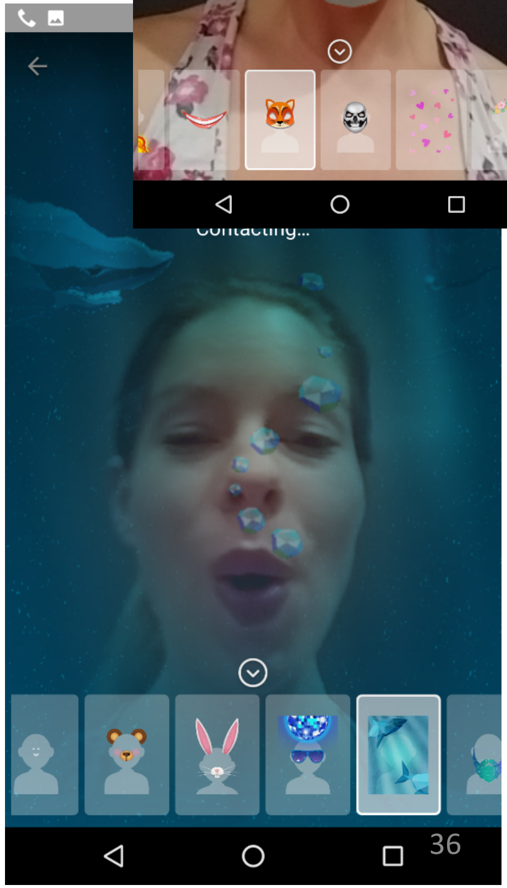


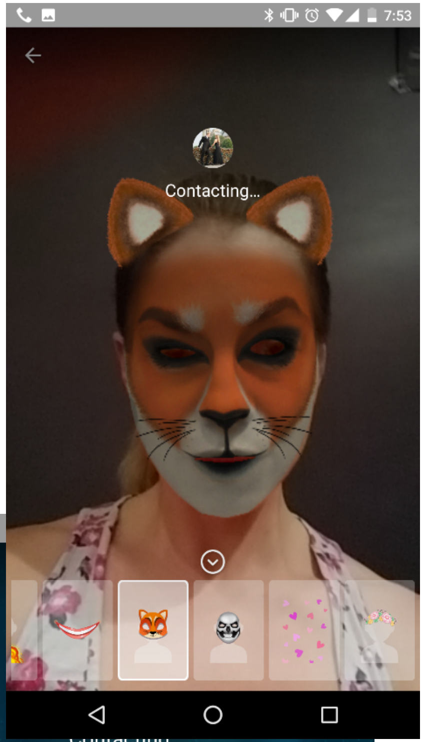






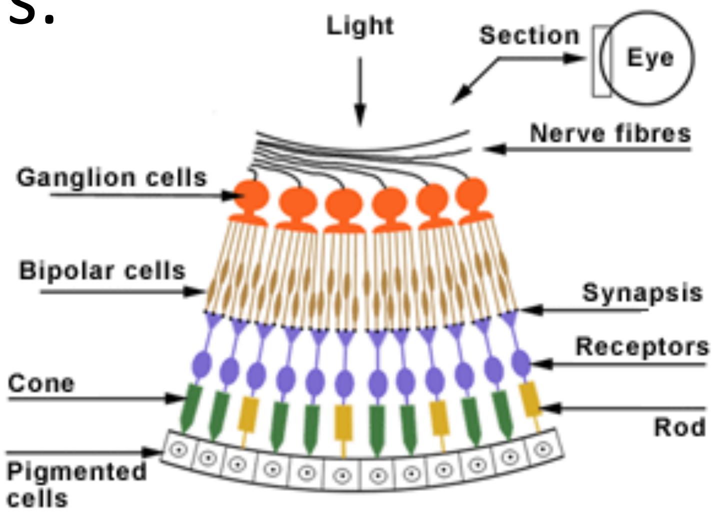




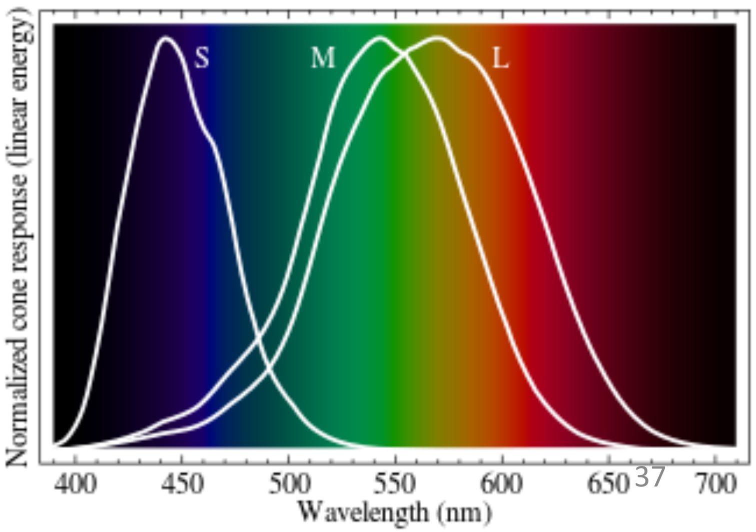


Colour Opponency in the Human Eye

- Classic model of the eye is with 4 photoreceptors:
  - Rods (more sensitive to brightness).
  - L-Cones (most sensitive to red).
  - M-Cones (most sensitive to green).
  - S-Cones (most sensitive to blue).



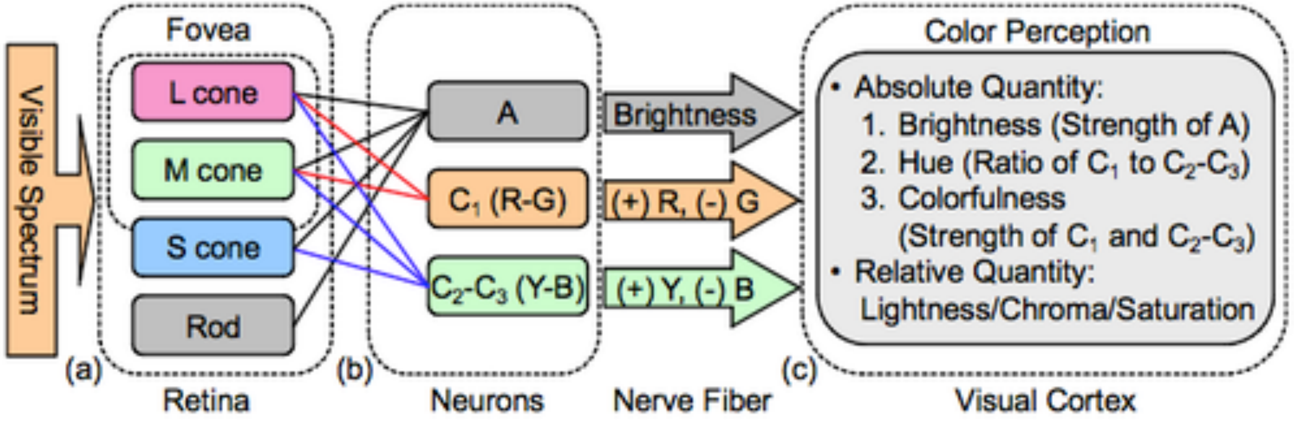
- Two problems with this system:
  - Not orthogonal.
    - High correlation in particular between red/green.
  - We have 4 receptors for 3 colours.

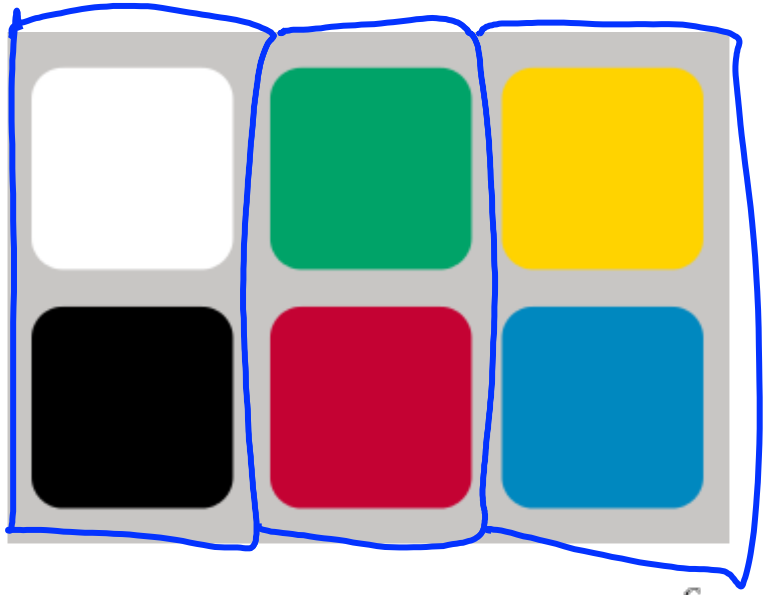


http://oneminuteastronomer.com/astro-course-day-5/
https://en.wikipedia.org/wiki/Color_visio

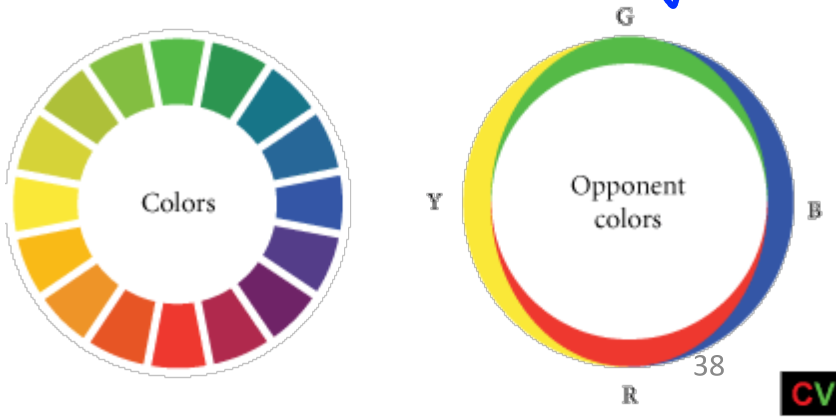
Colour Opponency in the Human Eye

- Bipolar and ganglion cells seem to code using “opponent colors”:
  - 3-variable orthogonal basis:





- This is similar to PCA (d = 4, k = 3).



http://oneminuteastronomer.com/astro-course-day-5/
https://en.wikipedia.org/wiki/Color_visio
http://5sensesnews.blogspot.ca/

Colour Opponency Representation

For this pixel, eye gets 4 signals

Can represent 4 original values with these 3 z_i values and matrix 'W'

$= w_1$

First row of W (First PC)

Analogous to means in k-means.

brightness

$+ w_2$

Second row (4×1)

red/green

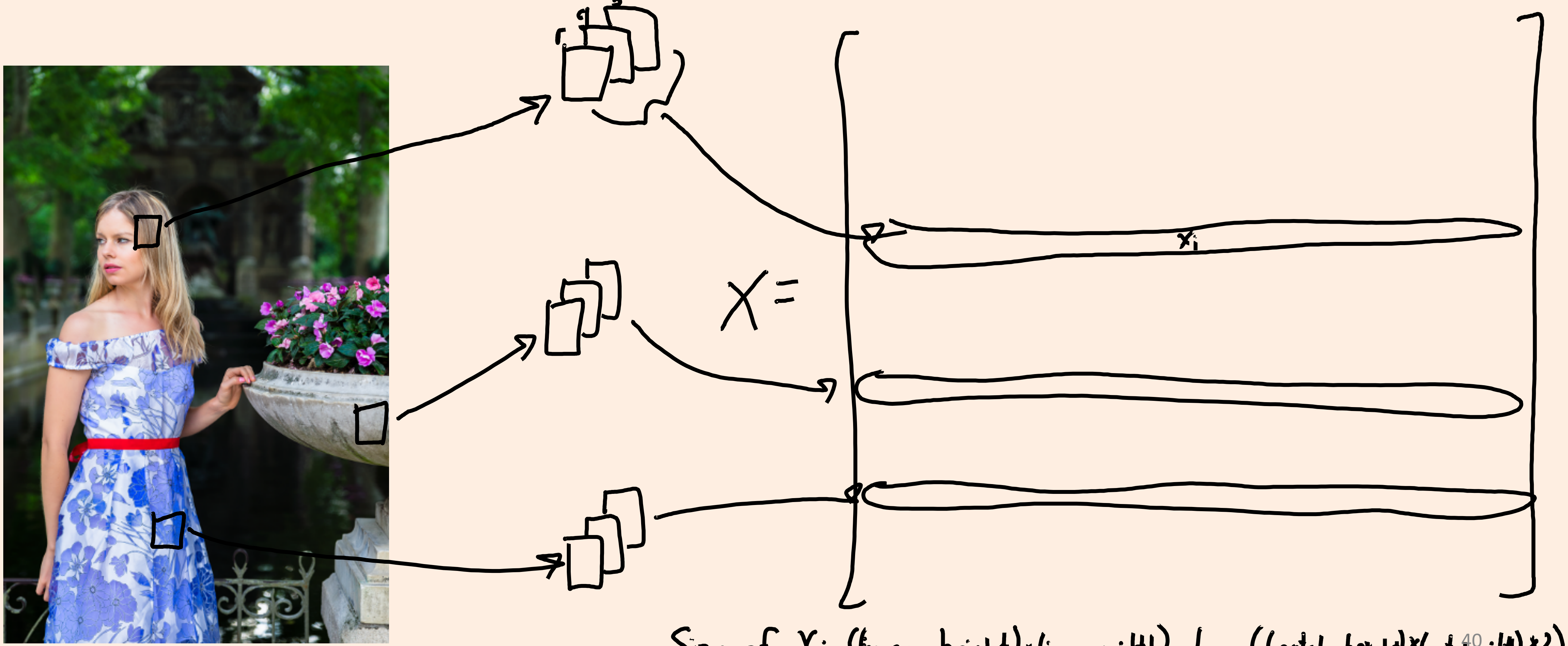
$+ w_3$

Third row (4×1)

blue/yellow

Latent-Factor Models for Image Patches

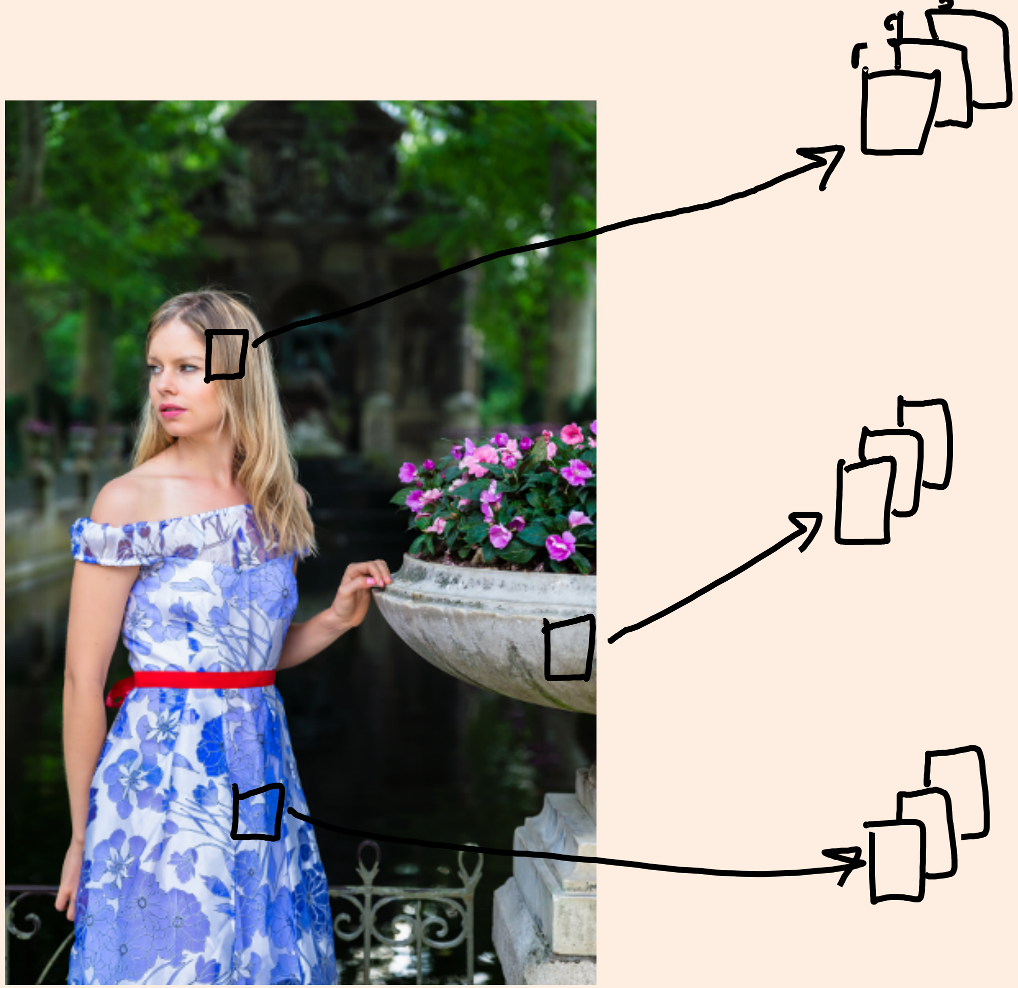
- Consider building latent-factors for general image patches:



Size of X: (image height)*(image width) by ((patch height)*(patch width)*3)

Latent-Factor Models for Image Patches

- Consider building latent-factors for general image patches:



Typical pre-processing:

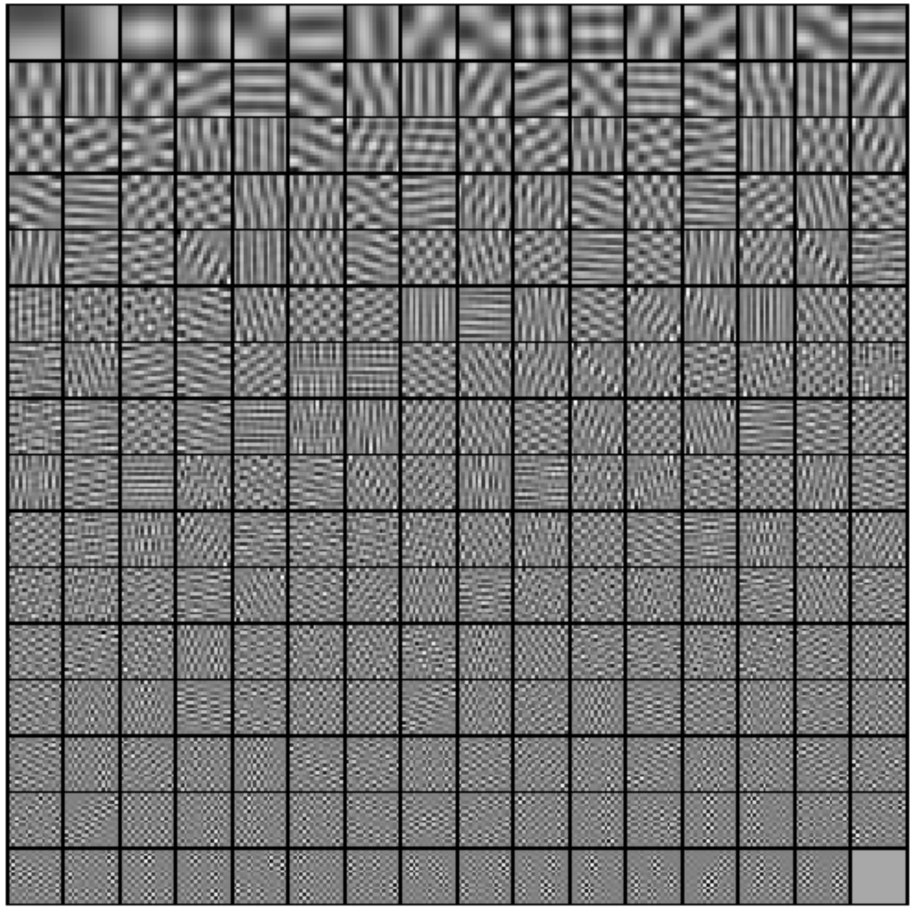
1. Usual variable centering
2. "Whiten" patches.

(remove correlations)

Application: Image Restoration

http://www.jmlr.org/papers/volume11/mairal10a/mairal10a.pdf

Latent-Factor Models for Image Patches

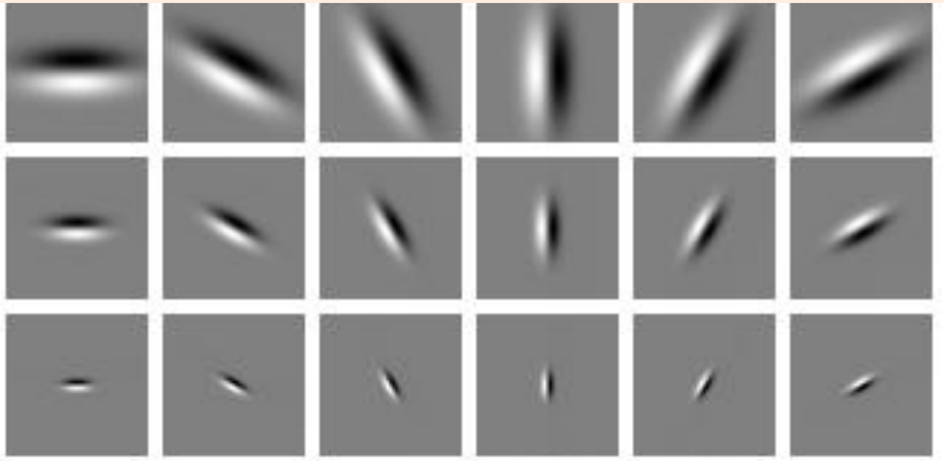


(b) Principal components.

Orthogonal bases don't seem right:

- Few PCs do almost everything.
- Most PCs do almost nothing.

We believe "simple cells" in visual cortex use:

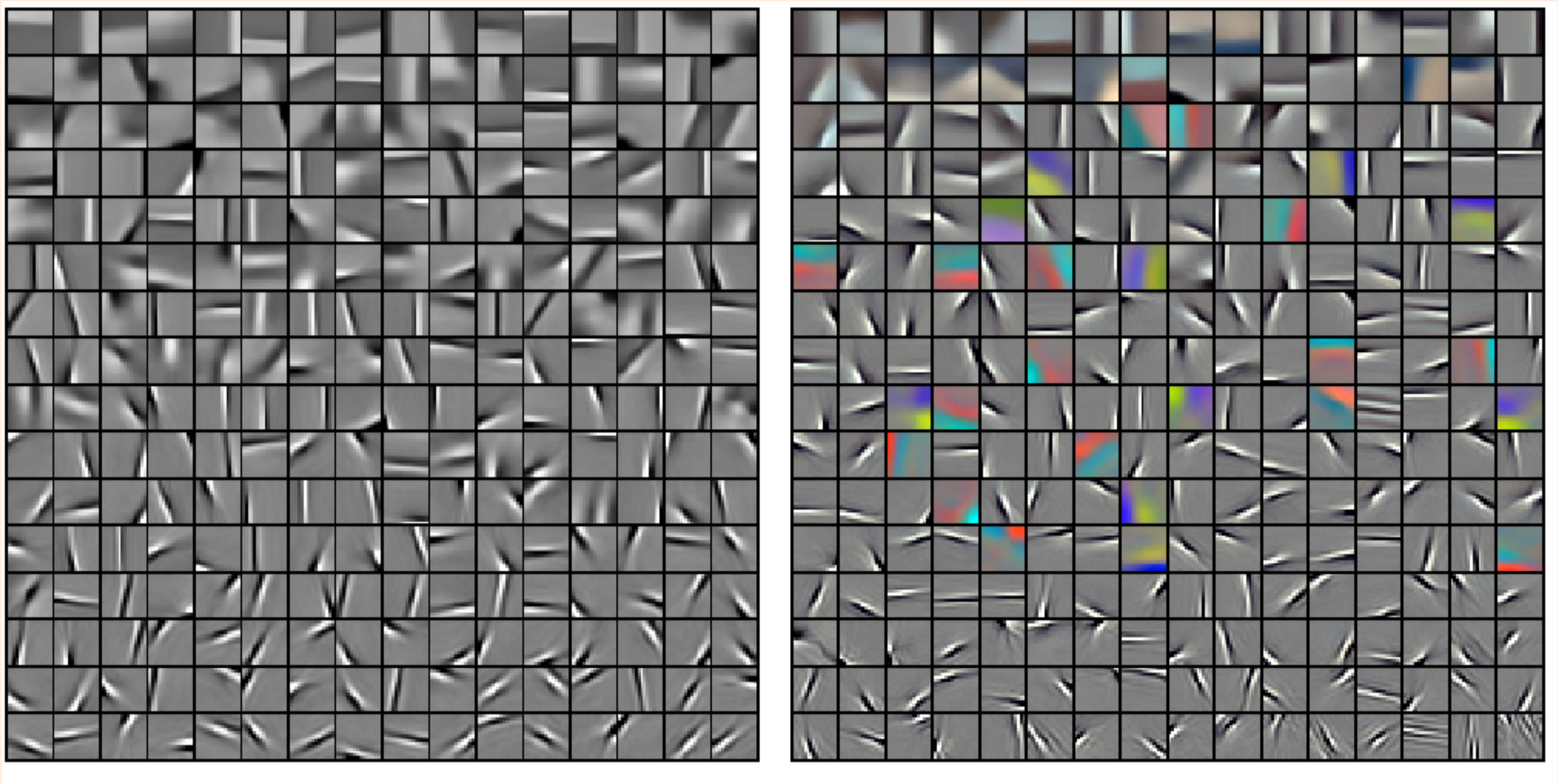


'Gabor' filters

http://lear.inrialpes.fr/people/mairal/resources/pdf/review_sparse_arxiv.pdf
http://stackoverflow.com/questions/16059462/comparing-textures-with-opencv-and-gabor-filters

Latent-Factor Models for Image Patches

- Results from a sparse (non-orthogonal) latent factor model:

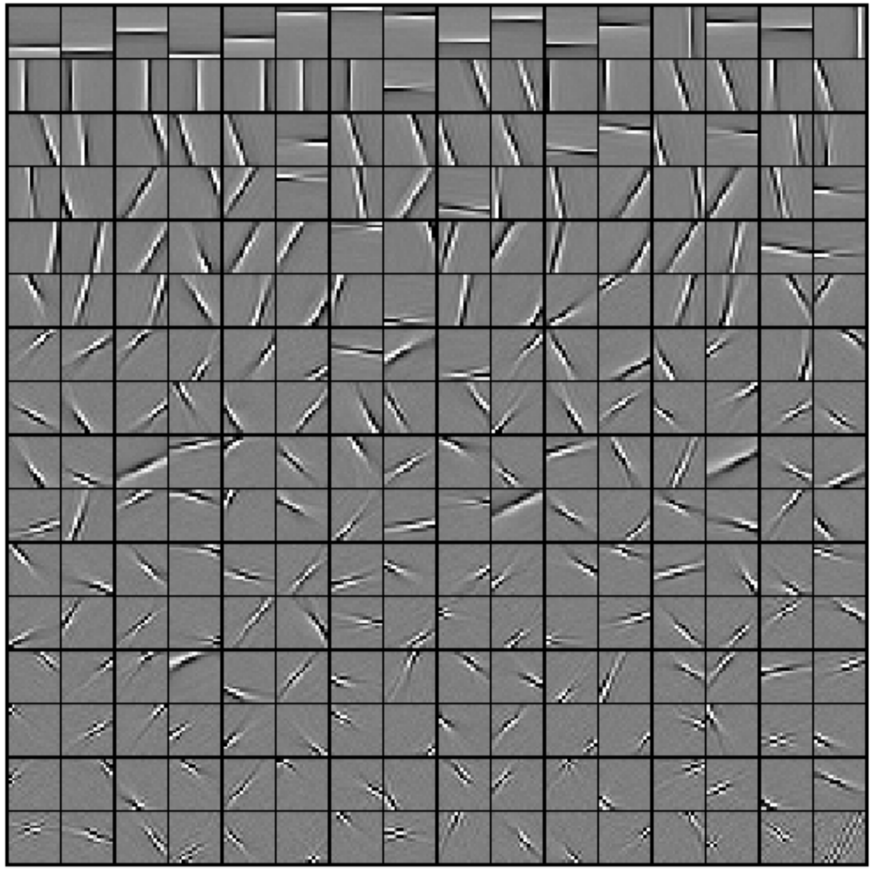


(a) With centering - gray. (b) With centering - RGB.

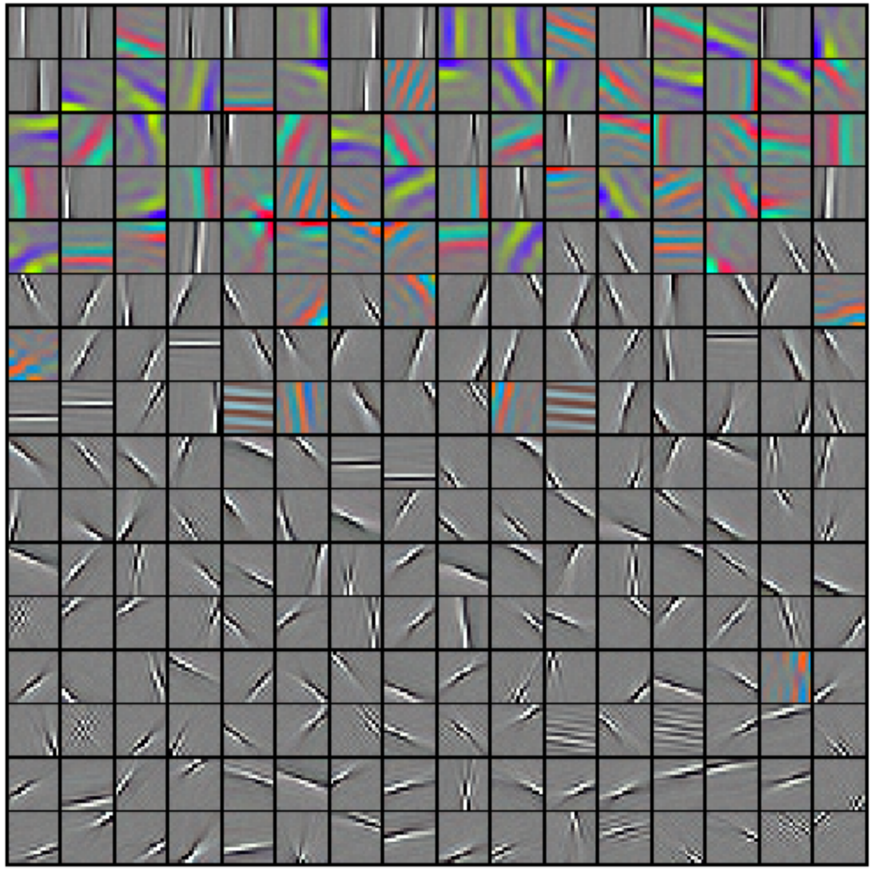
http://lear.inrialpes.fr/people/mairal/resources/pdf/review_sparse_arxiv.pdf

Latent-Factor Models for Image Patches

- Results from a “sparse” (non-orthogonal) latent-factor model:



(c) With whitening - gray.



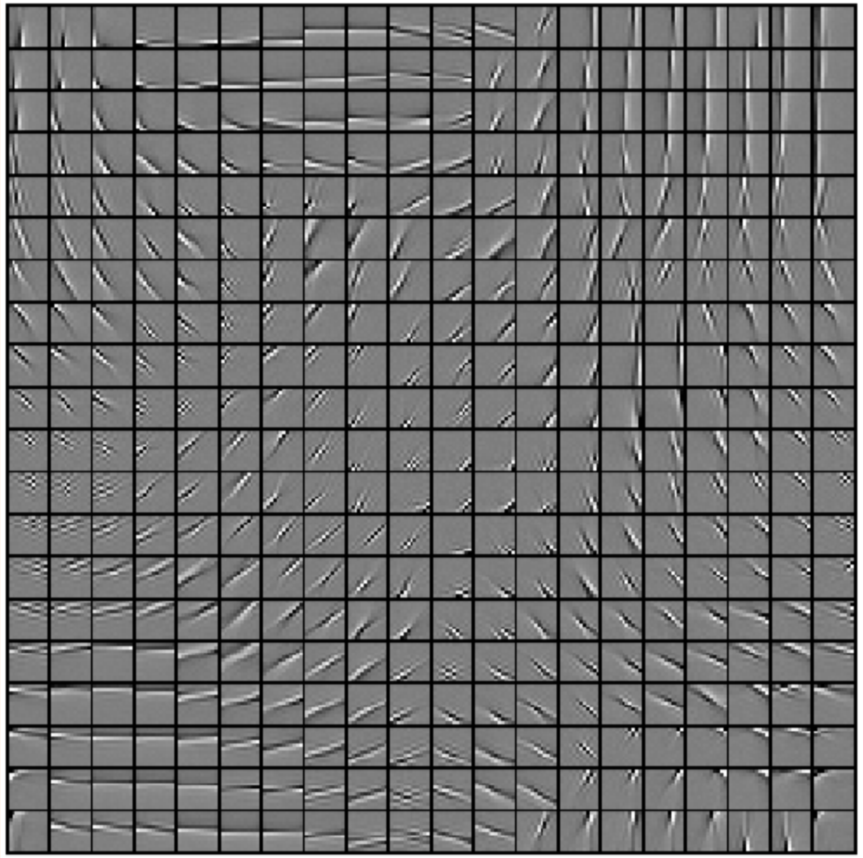
(d) With whitening - RGB.

"colour opponency"

http://lear.inrialpes.fr/people/mairal/resources/pdf/review_sparse_arxiv.pdf

Recent Work: Structured Sparsity

- Basis learned with a variant of "structured sparsity":



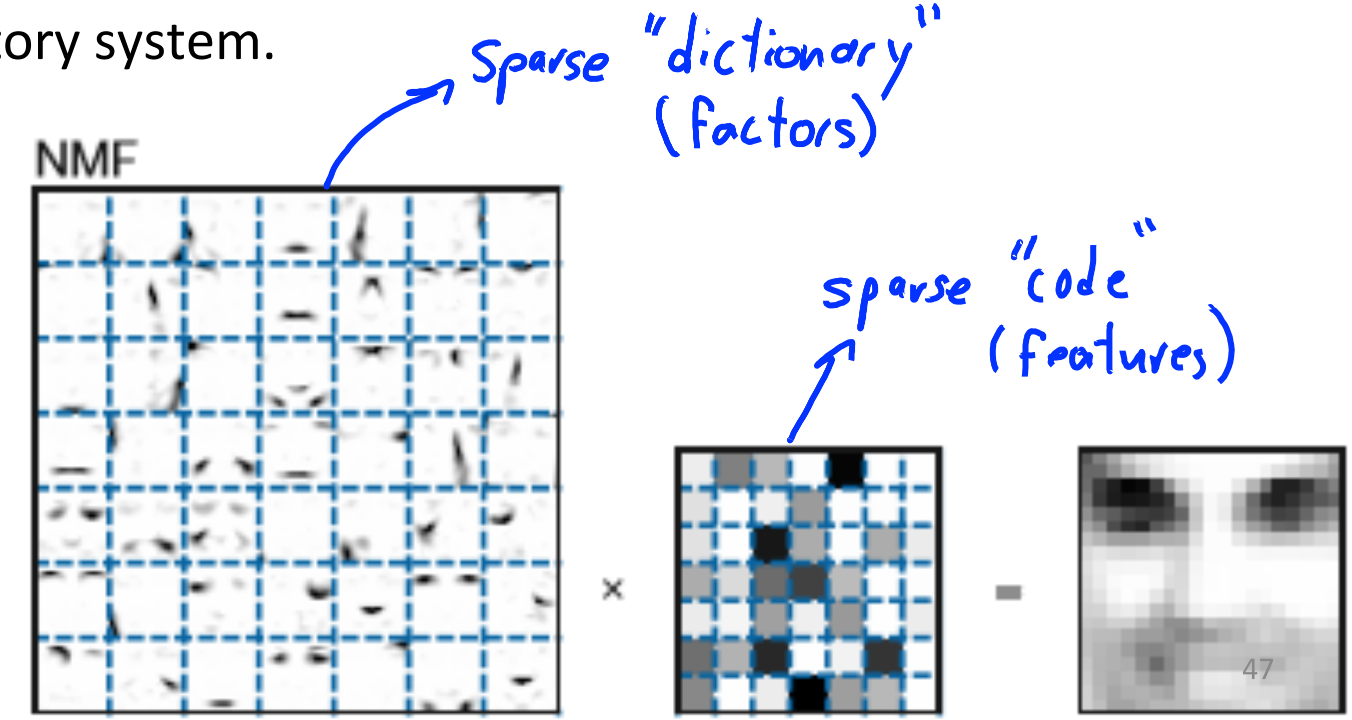
(b) With 4 × 4 neighborhood.

Similar to "cortical columns" theory in visual cortex.

http://lear.inrialpes.fr/people/mairal/resources/pdf/review_sparse_arxiv.pdf

Representing Faces

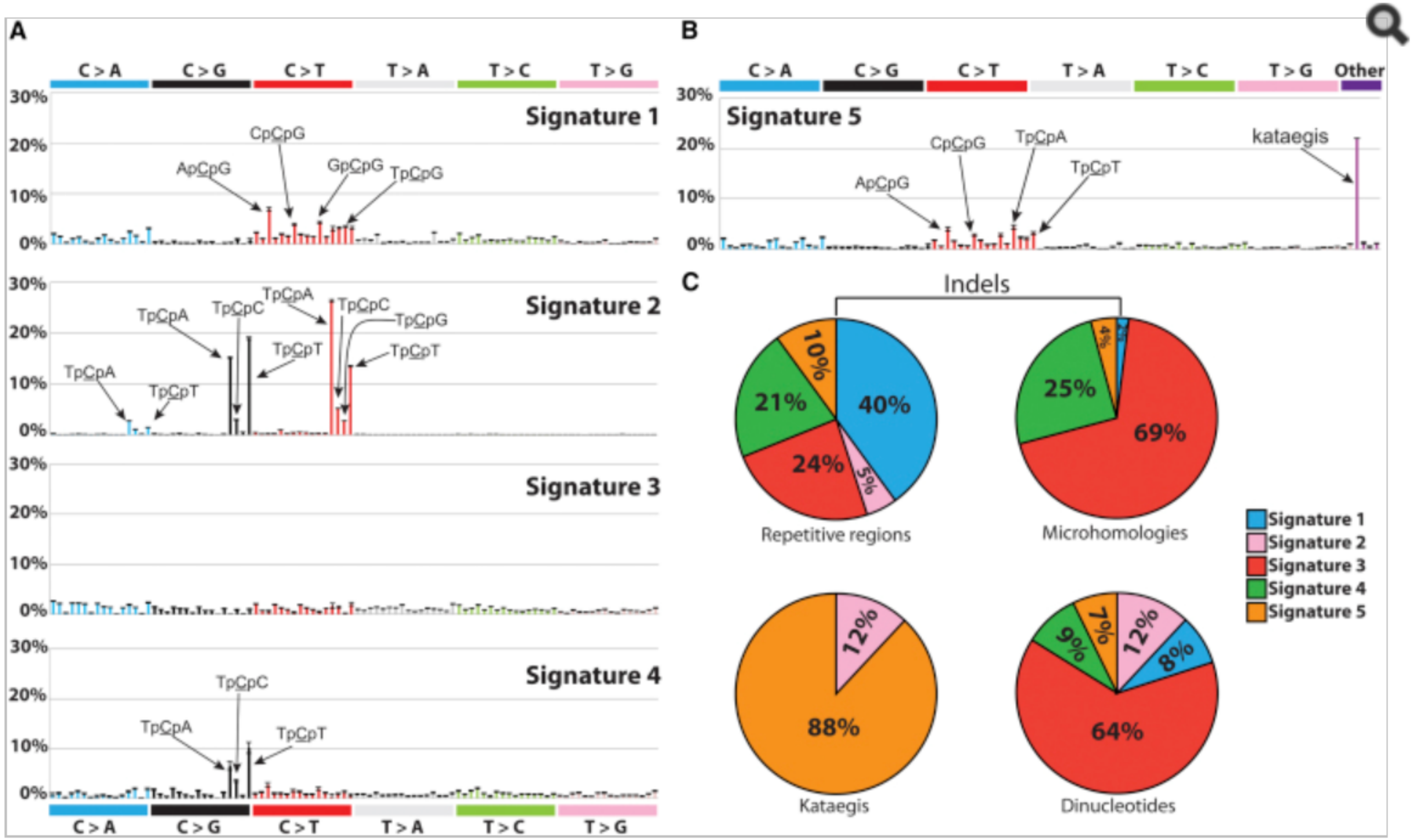
- Why sparse coding?
  - "Parts" are intuitive, and brains seem to use sparse representation.
  - Energy efficiency if using sparse code.
  - Increase number of concepts you can memorize?
    - Some evidence in fruit fly olfactory system.



http://www.columbia.edu/~jwp2128/Teaching/W4721/papers/nmf_nature.pdf

Application: Cancer "Signatures"

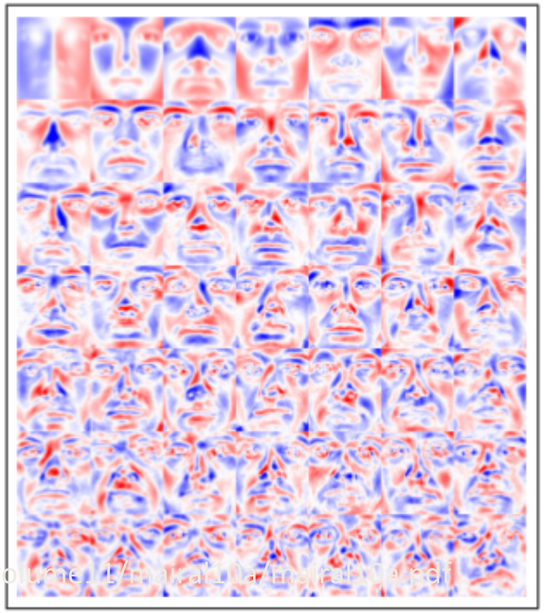
- What are common sets of mutations in different cancers?
  - May lead to new treatment options.



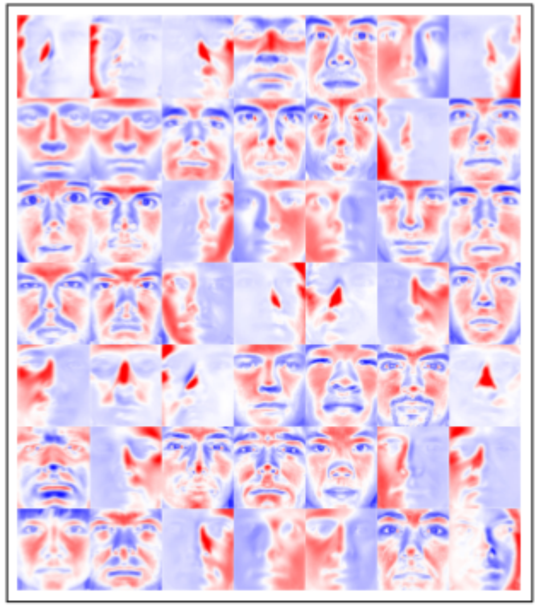
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3588146/

Regularized Matrix Factorization

- For many PCA applications, ordering orthogonal PCs makes sense.
  - Latent factors are independent of each other.
  - We definitely want this for visualization.
- In other cases, ordering orthogonal PCs doesn't make sense.
  - We might not expect a natural "ordering".



Usual orthogonal eigenfaces



PCA with non-orthogonal basis

http://www.jmlr.org/papers/volume11/mairal10a/mairal10a.pdf

Sparse Matrix Factorization

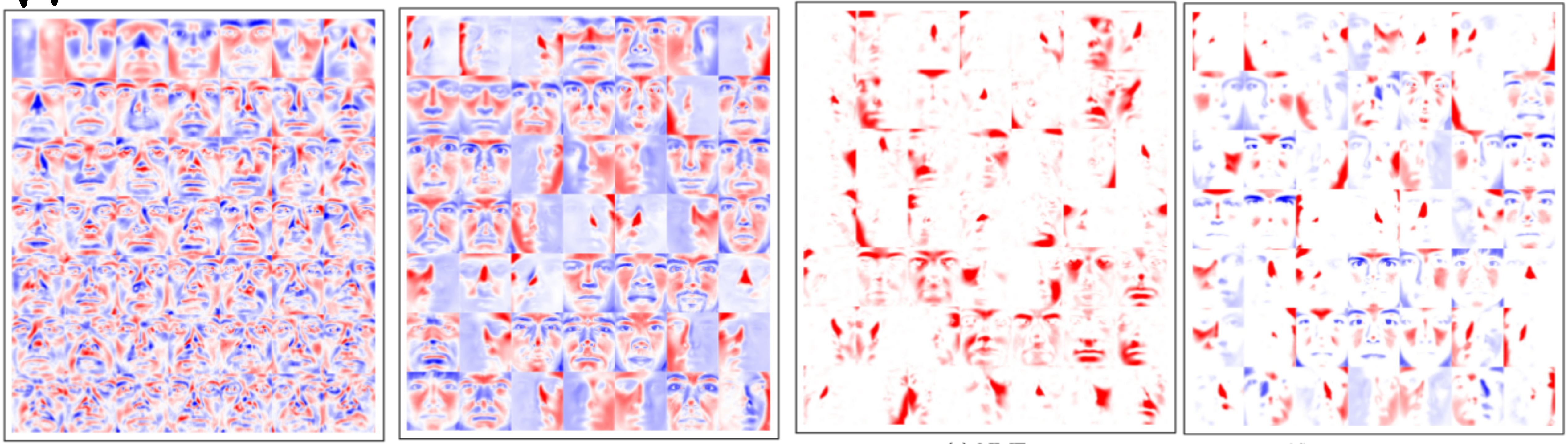
- Instead of non-negativity, we could use L1-regularization:

$$f(W,Z) = \frac{1}{2}\|ZW - X\|_F^2 + \frac{\lambda_1}{2}\sum_{i=1}^{n}\|z_i\|_1 + \frac{\lambda_2}{2}\sum_{j=1}^{d}\|w_j\|_1$$

  – Called sparse coding (L1 on 'Z') or sparse dictionary learning (L1 on 'W').
- Many variations exist:
  – Mixing L2-regularization and L1-regularization.
    - Or normalizing 'W' (in L2-norm or L1-norm) and regularizing 'Z'.
  – K-SVD constrains each z_i to have at most 'k' non-zeroes:
    - K-means is special case where k = 1.
    - PCA is special case where k = d.

Matrix Factorization with L1-Regularization

blue: negative
red: positive



(a) PCA

(e) Dictionary Learning

(c) NMF

(d) SPCA, $\tau = 30\%$

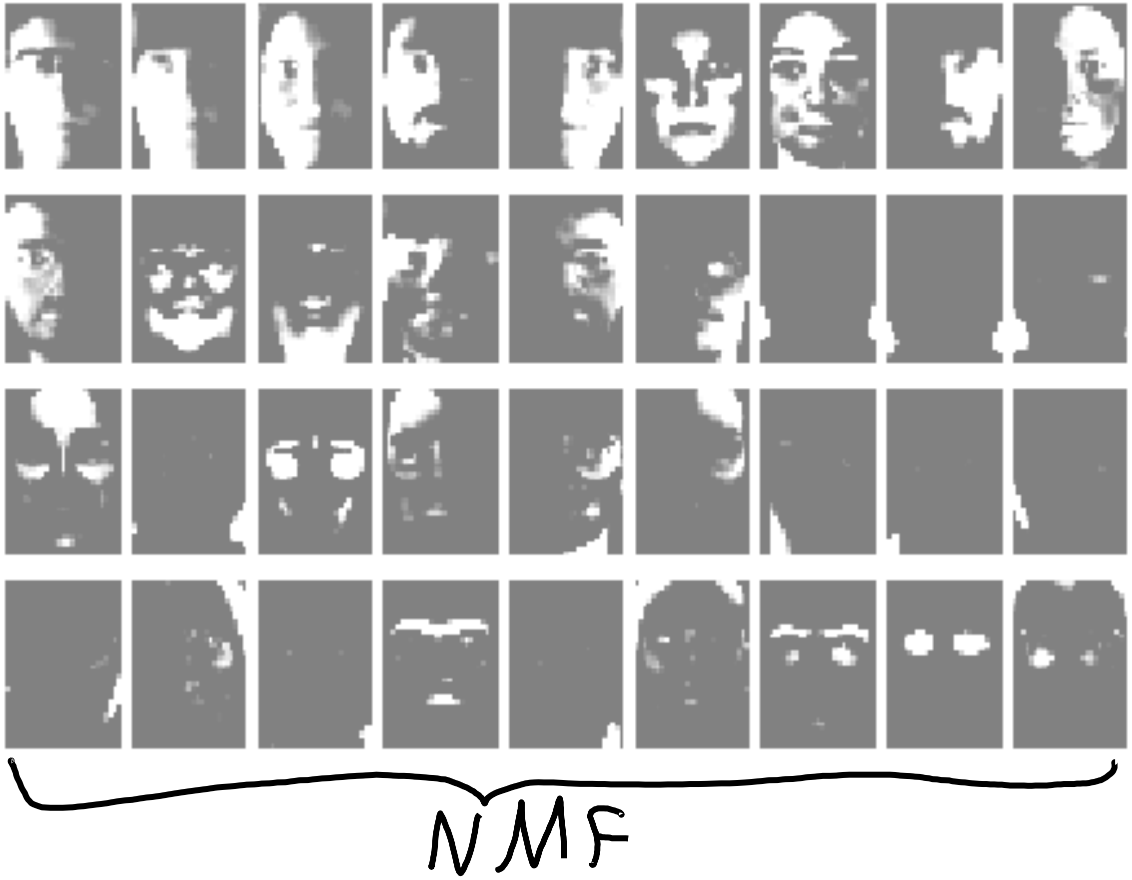
PCA without orthogonality

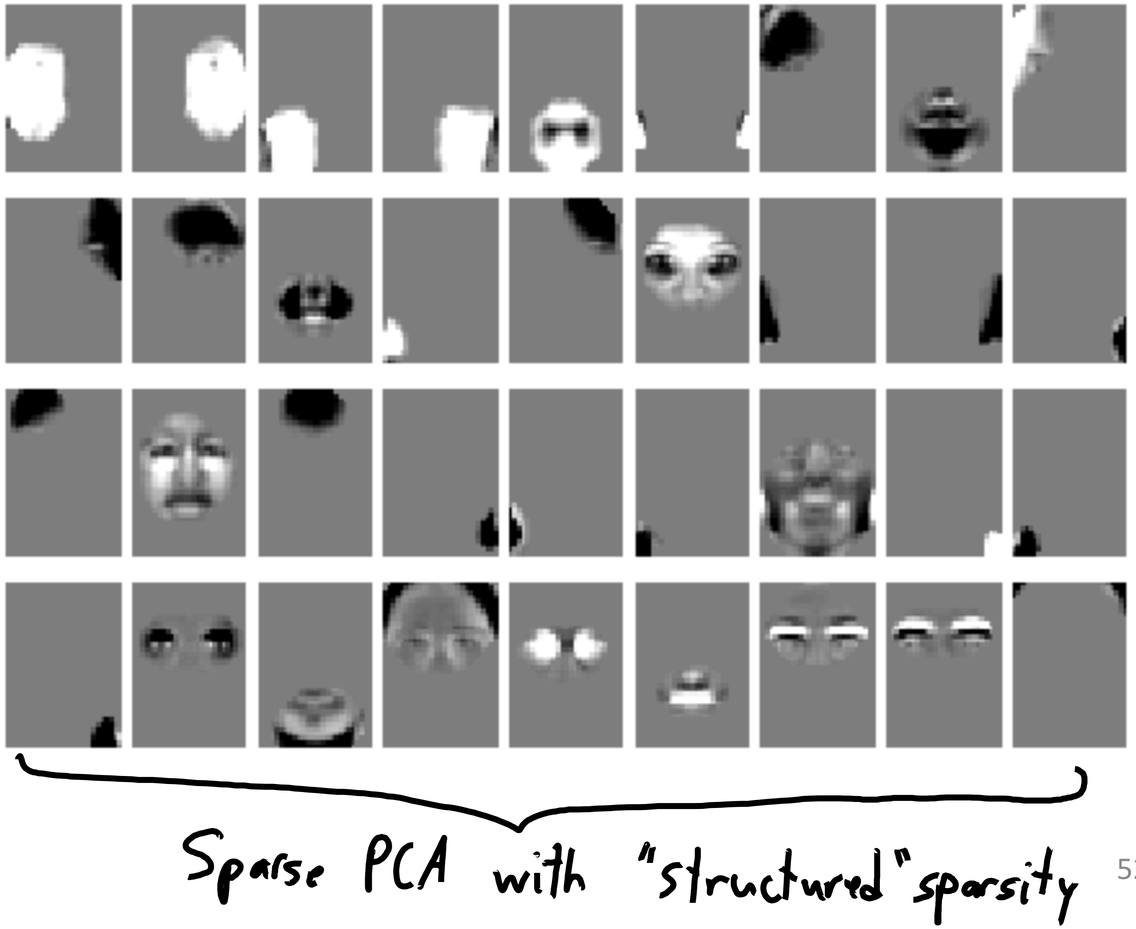
sparsity due to non-negativity

Sparsity due to L_1-regularization

Recent Work: Structured Sparsity

- "Structured sparsity" considers dependencies in sparsity patterns.
  - Can enforce that "parts" are convex regions.





http://jmlr.org/proceedings/papers/v9/jenatton10a/jenatton10a.pdf