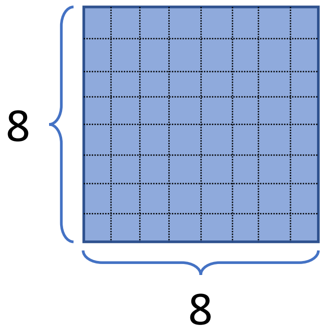
What can happen in an 8x8 image window?



Theoretically, 256^{64} possible images

But, which ones happen?

How to represent images?

- Basis Functions / Fourier Series
- Overcomplete bases, sparse coding
- Learning bases: (i) PCA, (ii) Sparsity, (iii) Matched Filters
- Shift invariance: Mini-epitomes, Active Patches (next lecture)

Representing images in terms of basis function

Classic: Orthogonal set of basis functions

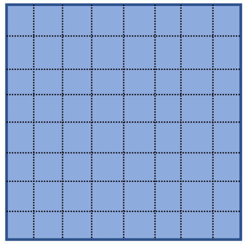
$$\{b_i(x): i = 1, \dots, N\}$$

where $\sum_x \{b_i(x)\}^2 = 1$

$$\sum_x b_i(x) b_j(x) = 0, \text{ if } i \neq j$$

or $\int dx \{b_i(x)\}^2 = 1$

$$\int dx\, b_i(x) b_j(x) = 0, \text{ if } i \neq j$$



$\mathcal{D}$

8x8 patch

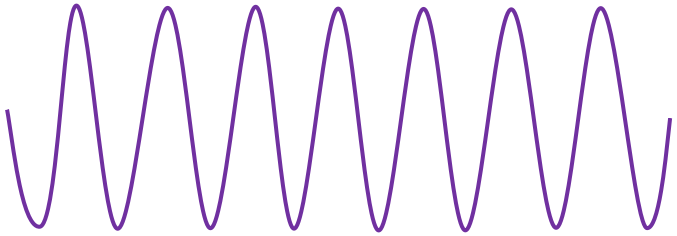
Examples

- Sinusoids / Fourier Analysis
- Haar Bases
- Impulse Function

Note: the number of orthogonal basis functions is equal to the dimension of the space (e.g., 64 for an 8x8 image), because they have to span the space. This limits the number of orthogonal basis functions is limited.

JPEG Coding

Choose basis function to be sinusoids



Represent image by $I(x) = \sum_i \alpha_i b_i(x)$

because the bases are orthonormal, we can solve to get

$$\alpha_i = \sum_x I(x) b_i(x) \quad (\text{or} \int dx \cdots)$$

An image is represented by the coefficients $\{\alpha_i\}$

We can also approximate the image by minimizing

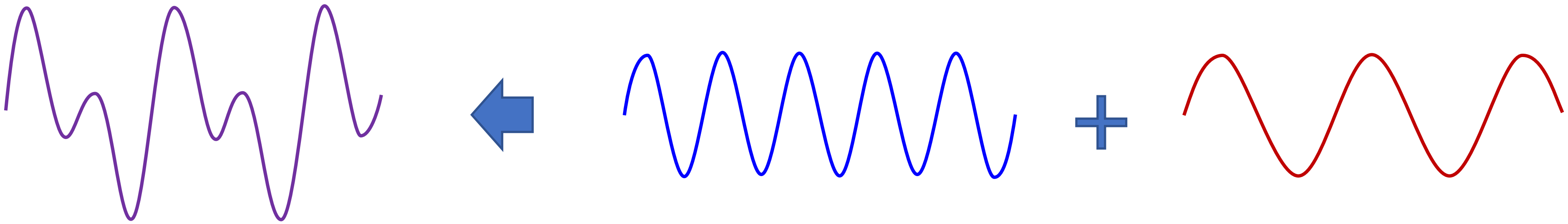
a cost function $\sum_x \left| I(x) - \sum_i \alpha_i b_i(x) \right|^2$

And keeping the terms where the α's are large, setting the others to be 0.

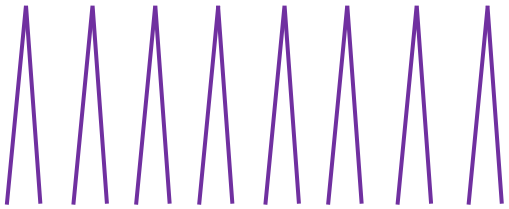
← This gives standard image format of JPEG if we use sinusoids

Sinusoids / Fourier Theory work well if the image can be approximated well by a set of sinusoids: i.e. only a small number of non-zero coefficients.

E.G.

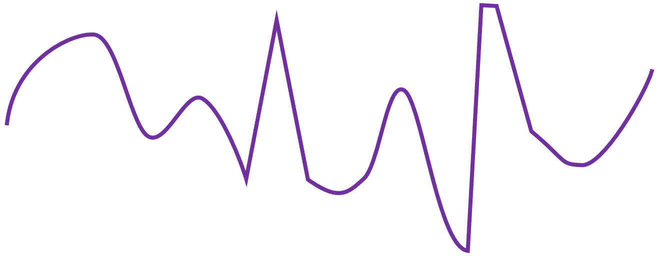


But an image like this:



is better approximated by a set of impulse functions (i.e. much fewer impulse functions are needed).

And an image like this:



Is badly modeled by either

Over-complete Bases

We can represent the image by an over-complete set of bases.

E.G. use all the sinusoids and all the impulse functions. Represent the image by a combination of sinusoids and impulses. In the 1990's mathematicians invented wavelets, which is another way to get an over-complete set of basis functions.

But now we have a problem.

There will be many ways to represent the image in form

$$I(x) = \sum_i \alpha_i b_i(x)$$

because we can represent it equally well by sinusoids only, or by impulse function only, or by combinations of each.

Sparsity <u>L1-Sparsity:</u>

Resolve this problem by imposing a penalty on, or regularizing, the alphas. Determine the α's by minimizing

$$E[\alpha] = \sum_x \left\{ I(x) - \sum_i \alpha_i b_i(x) \right\}^2 + \underbrace{\lambda \sum_i \overset{\text{L1-norm}}{\boxed{|\alpha_i|}}}_{\text{regularization}}$$

Note: $E[\alpha]$ is a convex function (the L1-norm is convex)

- There are efficient algorithms to estimate $\hat{\alpha} = \arg\min E[\alpha]$
- Solution: $I(x) = \sum_i \hat{\alpha}_i b_i(x)$

By a "miracle" (later in lecture), many of the α's will be zero

Extreme Sparsity: Matched Filters

Typo: replace second“arg min” by arg max on line “Set…”

Set of basis function: $\{b_i(x)\}$

Represent each image by one basis function only

$$E[\alpha] = \sum_x \left| I(x) - \sum_i \alpha_i b_i(x) \right|^2 \quad \text{with constant only one} \quad \alpha_i \neq 0$$

Algorithm estimate $\hat{\alpha} = \arg\min E[\alpha]$

Set $\hat{\alpha}_i = \arg\min \sum_x |I(x) - \alpha_i b_i(x)|^2 = \arg\min \sum_x I(x) b_i(x)$ ⬅ $\sum \{b_i(x)\}^2 = 1$

Choose $\hat{i} = \min_i \sum_x |I(x) - \hat{\alpha}_i b_i(x)|^2$ ➡ Set $\alpha_{\hat{i}} = \hat{\alpha}_i$

$\alpha_j = 0$ otherwise

But this needs an enormous number of basis functions. How many? See mini-epitomes.

Comments

We described three ways to represent images using basis functions

- Classical: e.g. Fourier Theory / Harr Basis
- L1-Sparsity
- Matched Filters

} Both, overcomplete

But what bases to use?

- We can use the bases, like sinusoids (20th century math)
- Or we can learn them from data (21th century math)

Typo: replace S by delta

Learning the bases

Let’s start with the classical approach

Bases are orthogonal → $\sum_x b_i(x) b_j(x) = S_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$ (Kronecker Delta)

Dataset of images: $\{I^\mu(x) : \mu \in \Lambda\}$

Energy Function $E[b,\alpha] = \frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} \sum_x \left\{ I^\mu(x) - \sum_i \alpha_i^\mu b_i(x) \right\}^2$

Note: basis functions are the same for all images

the coefficients α_i^μ vary between images

Minimize

$$E[b,\alpha] = \frac{1}{|\Lambda|}\sum_{\mu\in\Lambda}\sum_{x}\left\{I^{\mu}(x) - \sum_{i}\alpha_i^{\mu} b_i(x)\right\}^2$$

w.r.t. (b,α)

This is simply Principal Component Analysis (PCA)

Provided we extract the means from the images

$$I^{\mu}(x) \to I^{\mu}(x) - \frac{1}{|\Lambda|}\sum_{\mu\in\Lambda} I^{\mu}(x)$$

so that $\sum_{\mu} I^{\mu}(x) = 0$

(after subtraction)

Solution: Singular Value Decomposition (SVD) implies that

The basis function $b_i(x)$ are the eigenvectors of the correlation matrix

$$K(x,y) = \frac{1}{|\Lambda|}\sum_{\mu\in\Lambda} I^{\mu}(x) I^{\mu}(y)$$

The coefficients $\alpha_i^{\mu} = \sum_x b_i(x) I^{\mu}(x)$ (as before)

We can restrict the number of basis function by only use those eigenvectors whose eigenvalues are above a threshold T

$$\Rightarrow \sum_y K(x,y) b_i(y) = \lambda_i b_i(x), \quad \text{keep } b_i(x) \text{ if } \lambda_i > T$$

What are the eigenvectors of image patches?

Claim If the image patches are randomly drawn from real images, then the eigenvectors are sinusoids?

Why? Because images are shift-invariant

$$K(x,y) = F(x-y)$$

The correlation function depends only on the different $(x\text{-}y)$

Eigenvectors: $\sum_y F(x-y)e(y) = \lambda e(x)$

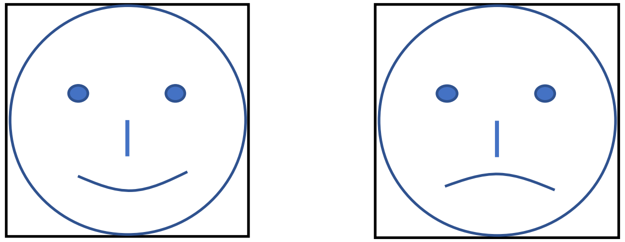
Sinusoids ➔ proof: apply the convolution theorem

So PCA doesn’t help much

You know you will get sinusoids before you look at the images

It is different if we align the images

For example, if we have images of faces and center them in the image patch, then the bases will not be sinusoids (Pentland & Turk)

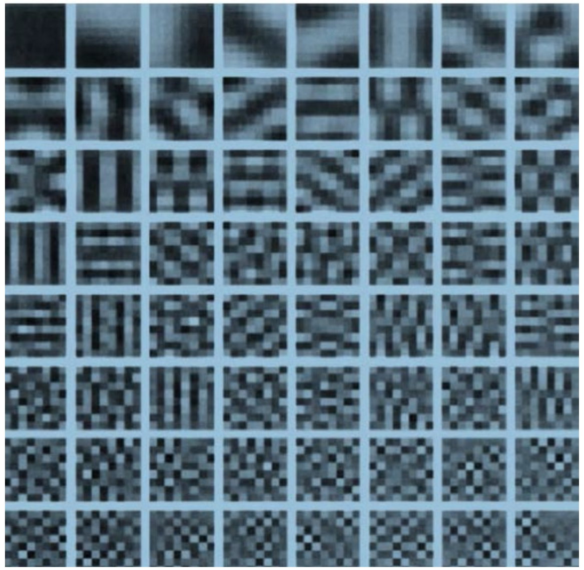


The alignment means that we remove shift-invariance

But it is not possible to align general images

PCA for generic images.

- Due to shift-invariance, the eigenvectors of generic images are sinusoid functions.

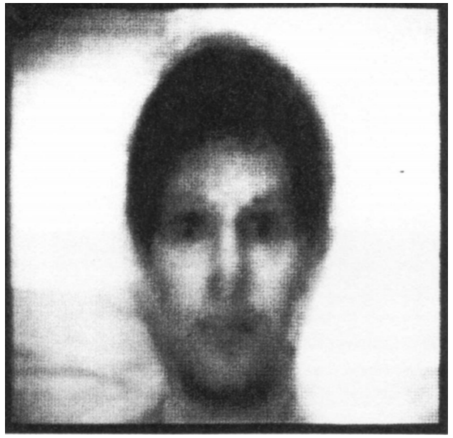


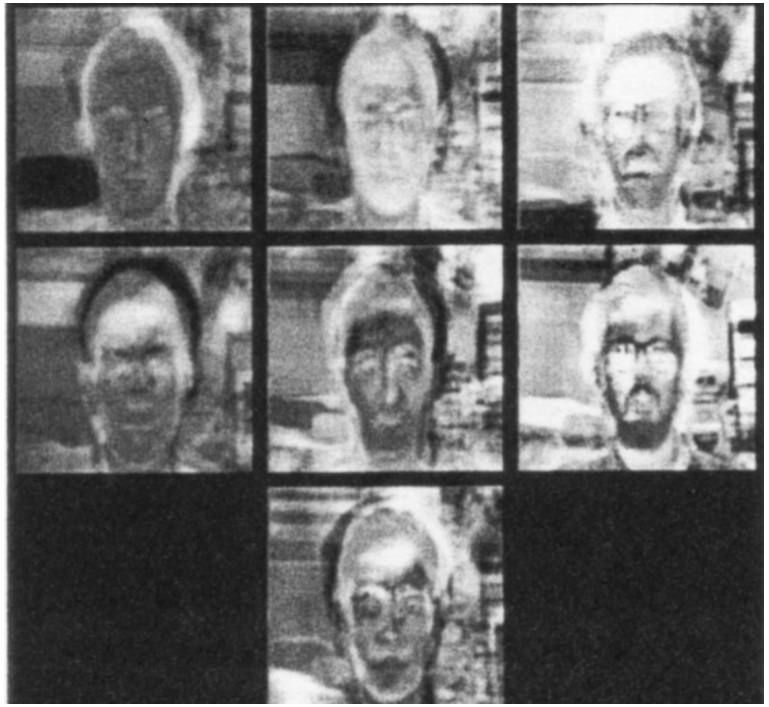
PCA for Faces (1)

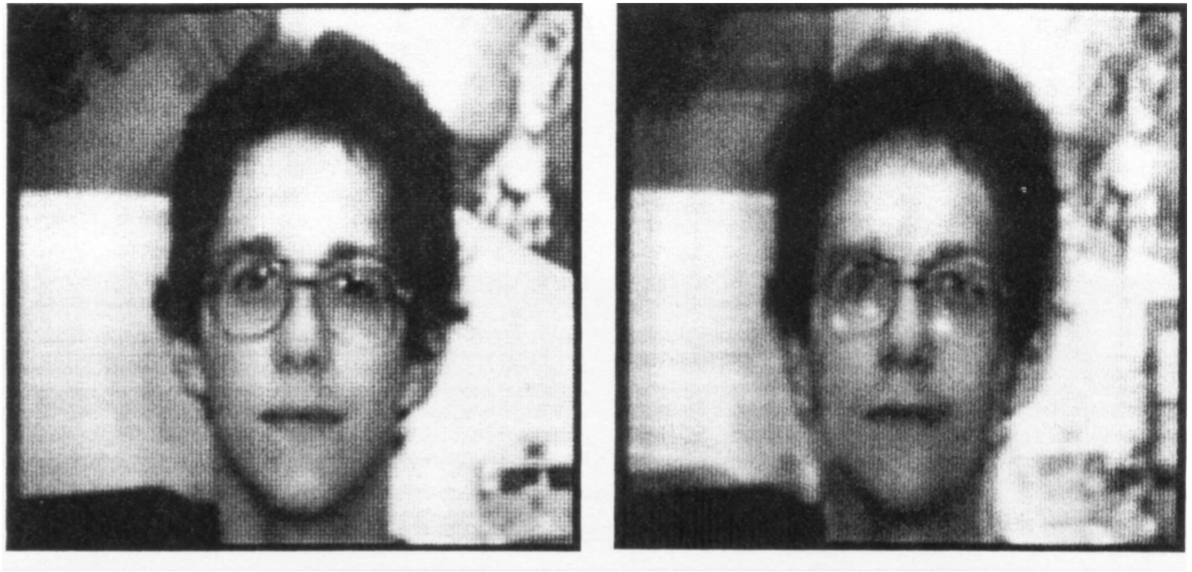
- The Faces are aligned, so there is no shift-invariance.
- PCA on faces (Pentland and Turk 1991).
- Dataset of faces:

PCA for Faces (2)

- Mean Face (left). Top 7 eigenvectors (center). Reconstruction (right).
- Number of eigenvalues: M=7, or M=14.







Now try sparsity – Olshausen & Field, 1996

$$E[b,\alpha]=\frac{1}{|\Lambda|}\sum_{\mu\in\Lambda}\sum_{x}\left\{I^{\mu}(x)-\sum_{i}\alpha_i^{\mu}b_i(x)\right\}^2+\lambda\sum_{\mu\in\Lambda}\sum_{i}\left|\alpha_i^{\mu}\right|$$

constraint: $\sum\{b_i(x)\}^2=1$

Minimize E w.r.t. (b,α)

Note: $E[b,\alpha]$ is convex in α if b is fixed (sparsity)

$E[b,\alpha]$ is convex in b if α is fixed

Alternative Algorithm

- Initialize b’s
- Minimize w.r.t a and b alternatively
- Guaranteed to converge to local minima

code available online

Olshausen & Field, 1996

Applied these to natural images (See examples)

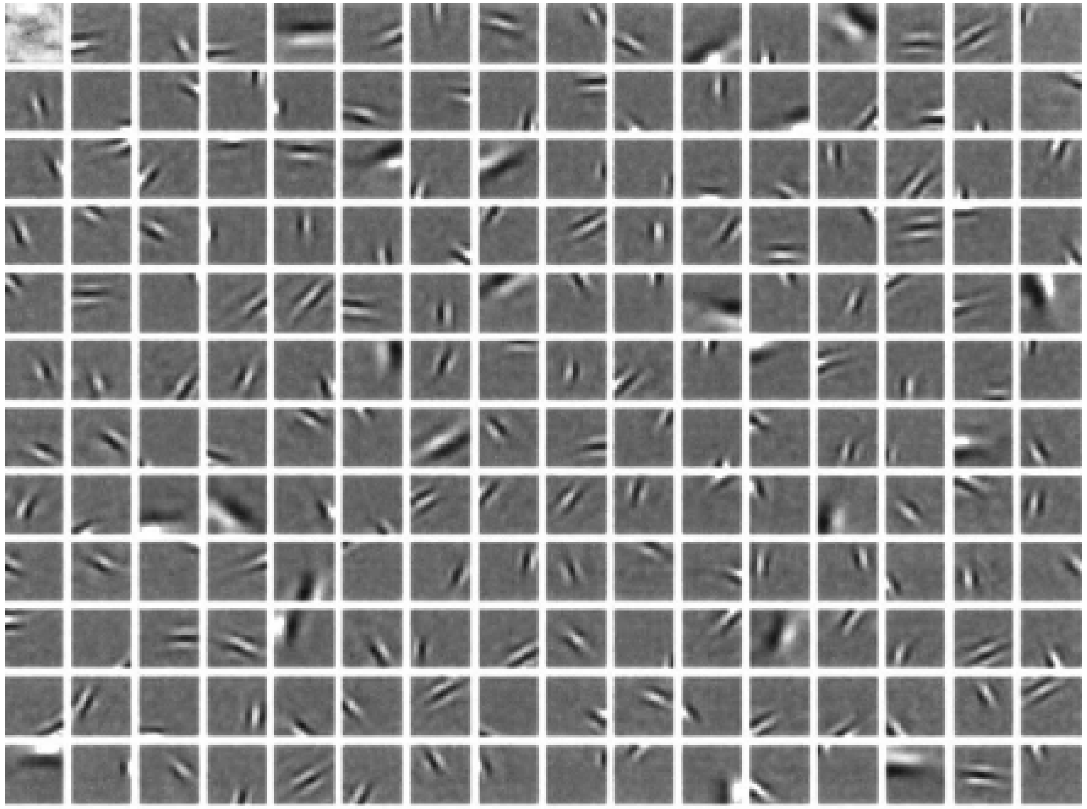
This gives more interesting bases than PCA

Note: Deep Neural Networks obtain similar bases. So does Independent Component Analysis (ICA).

They look similar to Gabor functions – sinusoids multiplied by Gaussians.

Sparse Representations of Generic Images

- Olshausen and Field.



The Miracle of Sparsity

Sparsity represents an input y by

$$\hat{\alpha} = \arg\min \left\{ \left| y - \sum_i \alpha_i b_i \right|^2 + \lambda \sum_i |\alpha_i| \right\}$$

The miracle: many $\hat{\alpha}_i$ will be zero ➡ Why?

This won’t happen if we replaced $\sum_i |\alpha_i|$ (L^1-loss) by $\sum_i \alpha_i^2$ (L^2-loss)

(Easy to see, with L2-loss you can compute $\hat{\alpha}$ analytically)

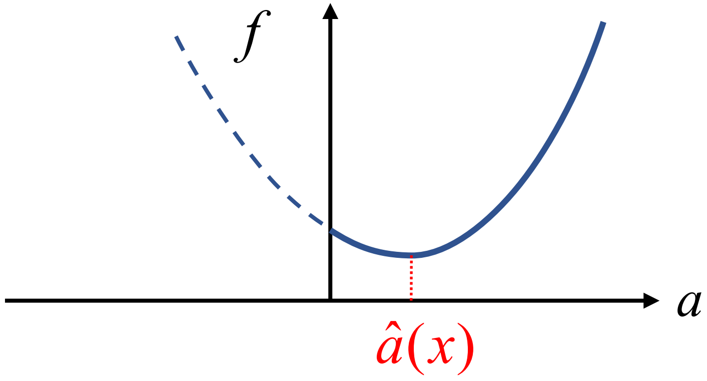
Why the miracle? 1D case

Let $f(a;x) = (x-a)^2 + \lambda\,|\,a\,|$

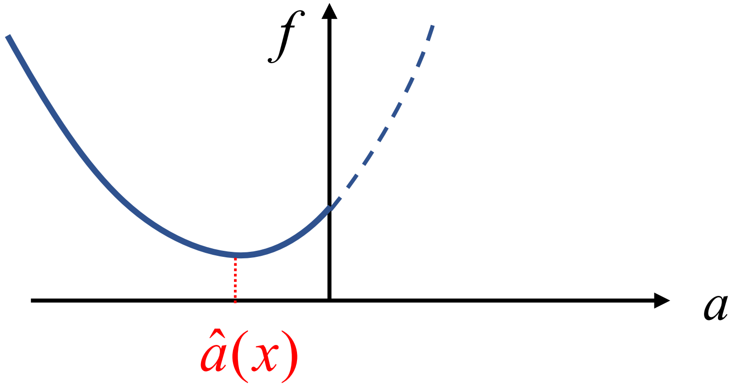
Claim
$$\hat{a}(x) = x - \lambda/2, \text{ if } x \ge \lambda/2$$
$$\hat{a}(x) = x + \lambda/2, \text{ if } x \le -\lambda/2$$
$$\hat{a}(x) = 0, \quad\quad , \text{ if } |\,x\,| \le \lambda/2$$

here $\hat{a}(x) = \arg\min f(a;x)$

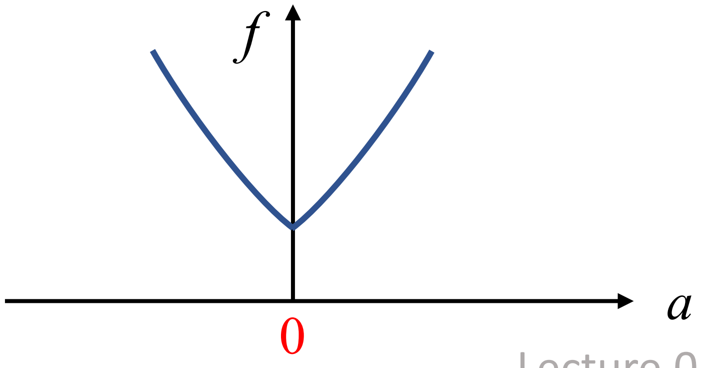
if $x \ge \lambda/2$



if $x \le -\lambda/2$



if $|x| \le \lambda/2$



Can check analytically

If $a \geq 0$ $\quad f_+(a;x) = (x-a)^2 + \lambda a$

$$\frac{df_+}{da} = -2(x-a) + \lambda$$

minima at $\hat{a} = x - \lambda / 2$

but $\hat{a} \geq 0 \Rightarrow x \geq \lambda / 2$

Similarly, If $a \leq 0$ $\quad f_-(a;x) = (x-a)^2 - \lambda a$

$$\frac{df_-}{da} = -2(x-a) - \lambda$$

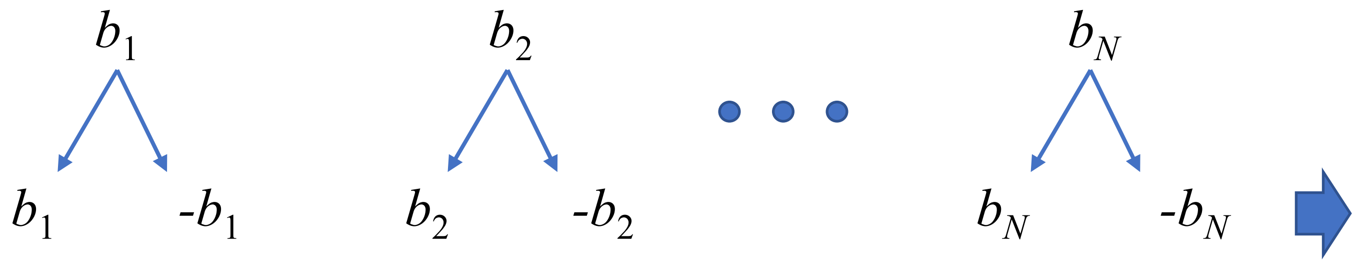
minima at $\hat{a} = x + \lambda / 2$

but $\hat{a} \leq 0 \Rightarrow x \geq -\lambda / 2$

In higher dimensions

Reformulate the problem in terms of convex hulls

First, duplicate each basis function



b_1 b_1 $-b_1$

b_2 b_2 $-b_2$

b_N b_N $-b_N$

new 2*N* bases: $\bar{b}_i$

Then we can express $\sum_{i=1}^{N} \alpha_i b_i = \sum_{i=1}^{2N} \bar{\alpha}_i \bar{b}_i$ with $\bar{\alpha}_i \geq 0$

Trick
$$\begin{aligned} \alpha_i b_i &= \alpha_i b_i, & \text{if } \alpha_i \geq 0 \\ &= (-\alpha_i)(-b_i), & \text{if } \alpha_i < 0 \end{aligned}$$

In higher dimensions

Typo: Then should be followed by "y=" and not By "y –".

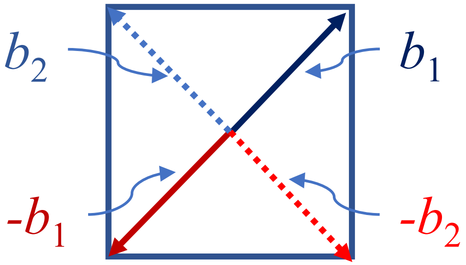
Now consider encoding an input y

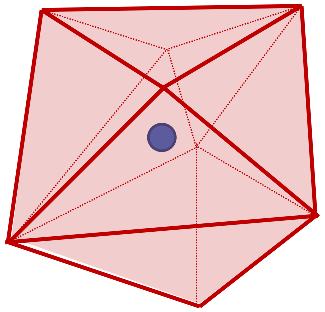
$$\overset{\square}{\alpha} = \arg\min\left\{\left|y - \sum_i \bar{\alpha}_i b_i\right|^2 + \lambda \sum_i \bar{\alpha}_i\right\}, \quad \text{s.t. } \bar{\alpha}_i \geq 0$$

Let $\displaystyle\sum_{i=1}^{2N} \bar{\alpha}_i = \alpha$

Then $\displaystyle\left\{y : \left\| y - \sum_i \bar{\alpha}_i \bar{b}_i \right\| \quad s.t. \sum_i \bar{\alpha}_i = \alpha \right\}$ specifies the convex hull of the $\{\bar{b}_i\}$ with radius α

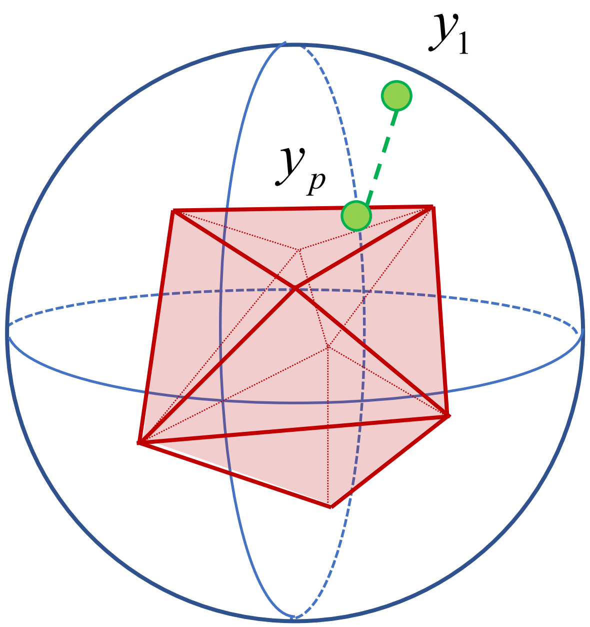
E.G.





In higher dimensions

Consider an input data y, w.l.o.g. $|y| = 1$ ⬅ Lies on a sphere



Hence, solving for $\bar{\alpha}_i$ corresponds to finding the closest point y_p on the convex hull

Sparsity ➔find closest point on convex hull while penalizing the radius α of the convex hull

Hence, y is projected to a point y_p on the boundary of the convex hull

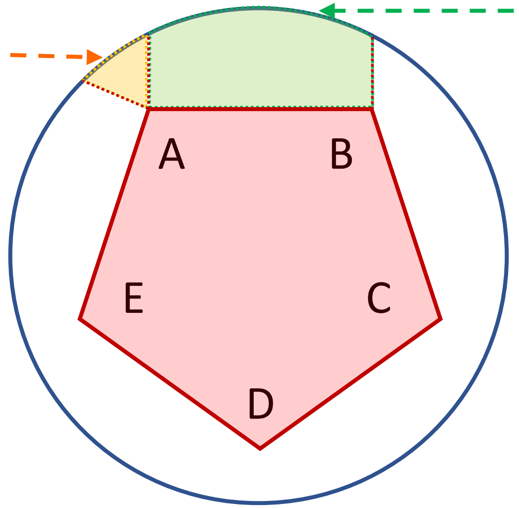
In higher dimensions

Increasing the size of λ

Corresponds w increasing the penalty for the radius of the convex hull

Hence causing the radius to get smaller

Where do point project?



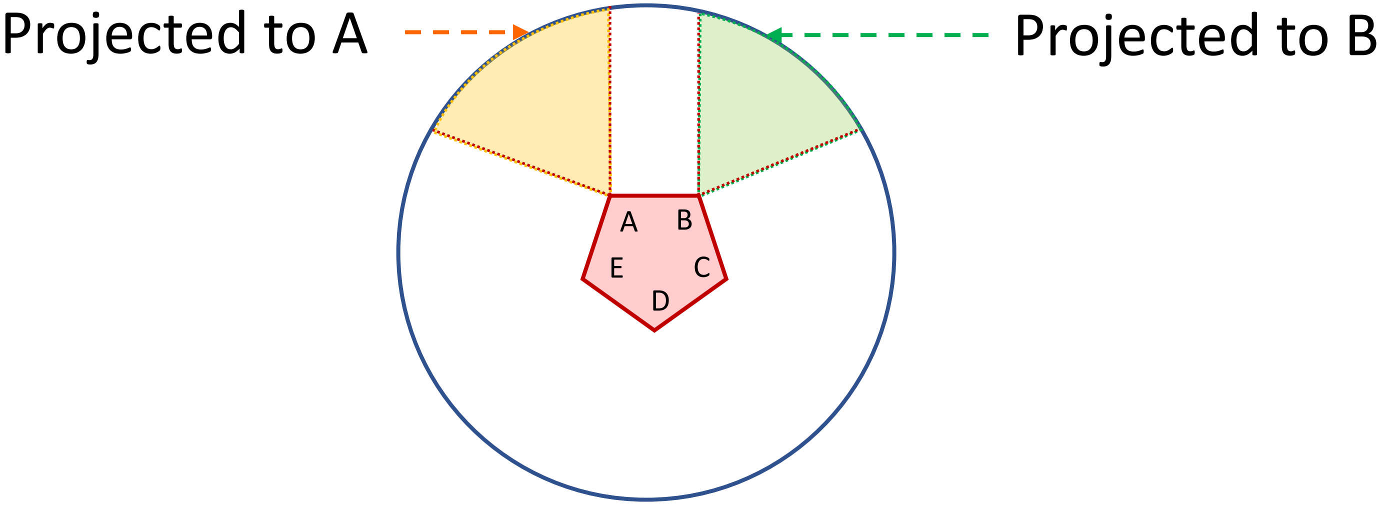
Projected to basis A

Projected to bases A&B (zero coefficients for C, D, and E)

This shows that many bases will have zero coefficients

In higher dimensions, Increasing the size of λ

As λ gets bigger, the convex hull gets smaller and increasingly bases have non-zero coefficients



This gives geometric intuition into the miracle of sparsity

Final Alternative Matched Filters $\sum \{b_i(x)\}^2 = 1$

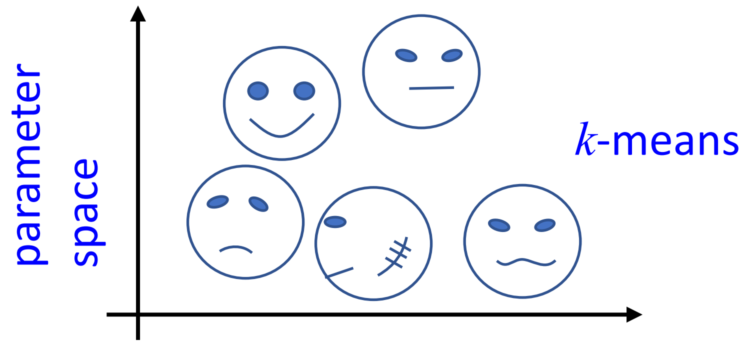
Minimize $E[b,\alpha] = \frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} \sum_x \left\{ I^\mu(x) - \sum_i \alpha_i^\mu b_i(x) \right\}^2$

with constraint that only one α_i^μ is non-zero for each μ

How to minimize?

- Convert this to k-means clustering

Requires normalizing each image $I^\mu(x) \to \frac{I^\mu(x)}{\sqrt{\sum_x \{I^\mu(x)\}^2}}$ so that $\sum_x \{I^\mu(x)\}^2 = 1$



- Implies that the best $\alpha_i^\mu = 1$

Extensions

All the previous methods have problem with shift-invariance

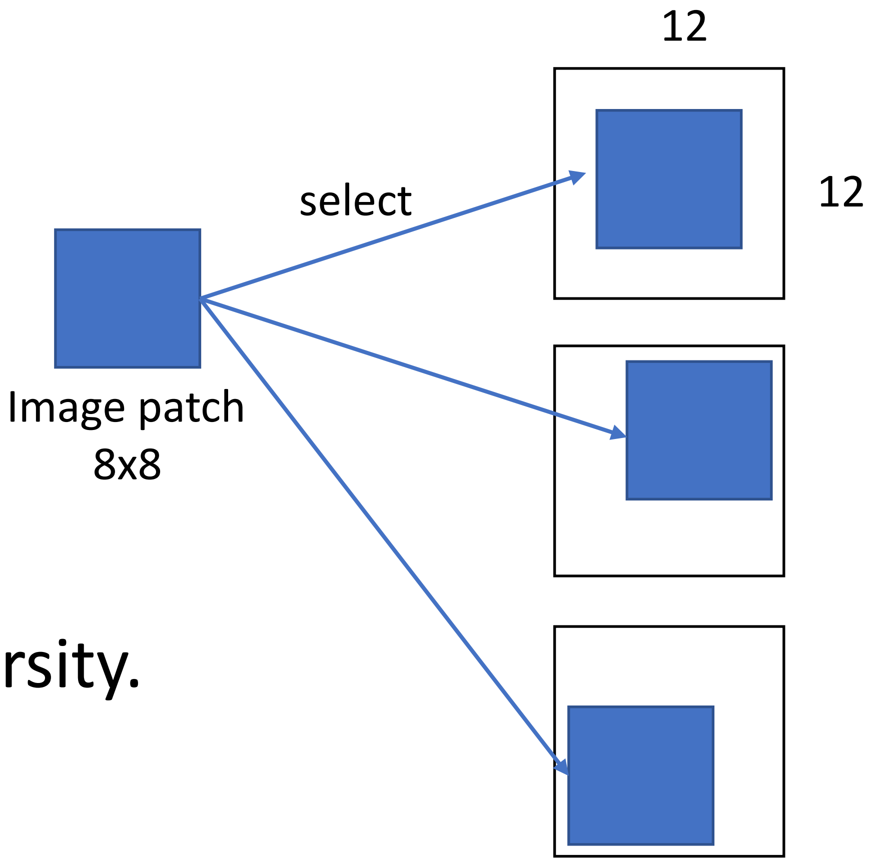
The basis function are encoding the space as well as the image patterns → See PowerPoints

One solution ➡ Mini-Epitomes

(G. Papandreou, L.-C. Chen, A.L. Yuille, 2014)

Extensions

Mini-Epitomes



This is like an extension of L0-sparsity.

But with smarter patches

➔ See next lecture

Can be learnt by the EM algorithm: extending k-means (next lecture)

Extensions

One result: A small set of mini-epitomes.

- 128 is able to represent most image patches in 10,000 images with good accuracy.
- So the number of possible image patches may not be too enormous.

Another approach: Active Patches (J. Mao, J. Zhu, A.L. Yuille, 2014)

- Allow the patch to be deformed when it matches the image

➔ See next lecture

Why Image Patches?

Helps capture what locally happens in images

Can rediscover edges by examining the bases learnt from images
(by matched filters or mini-epitomes)

- Can be used for image processing applications

  (i) Image denoising, (ii) Super-resolution (state-of-the-art)

- Can be used for high-level vision tasks (later in course)