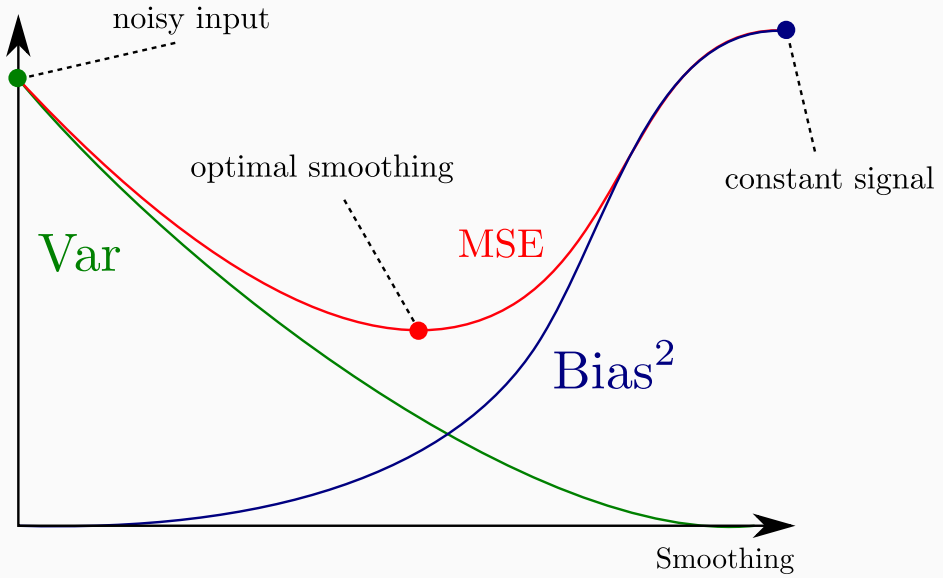
ECE 285

Image and video restoration

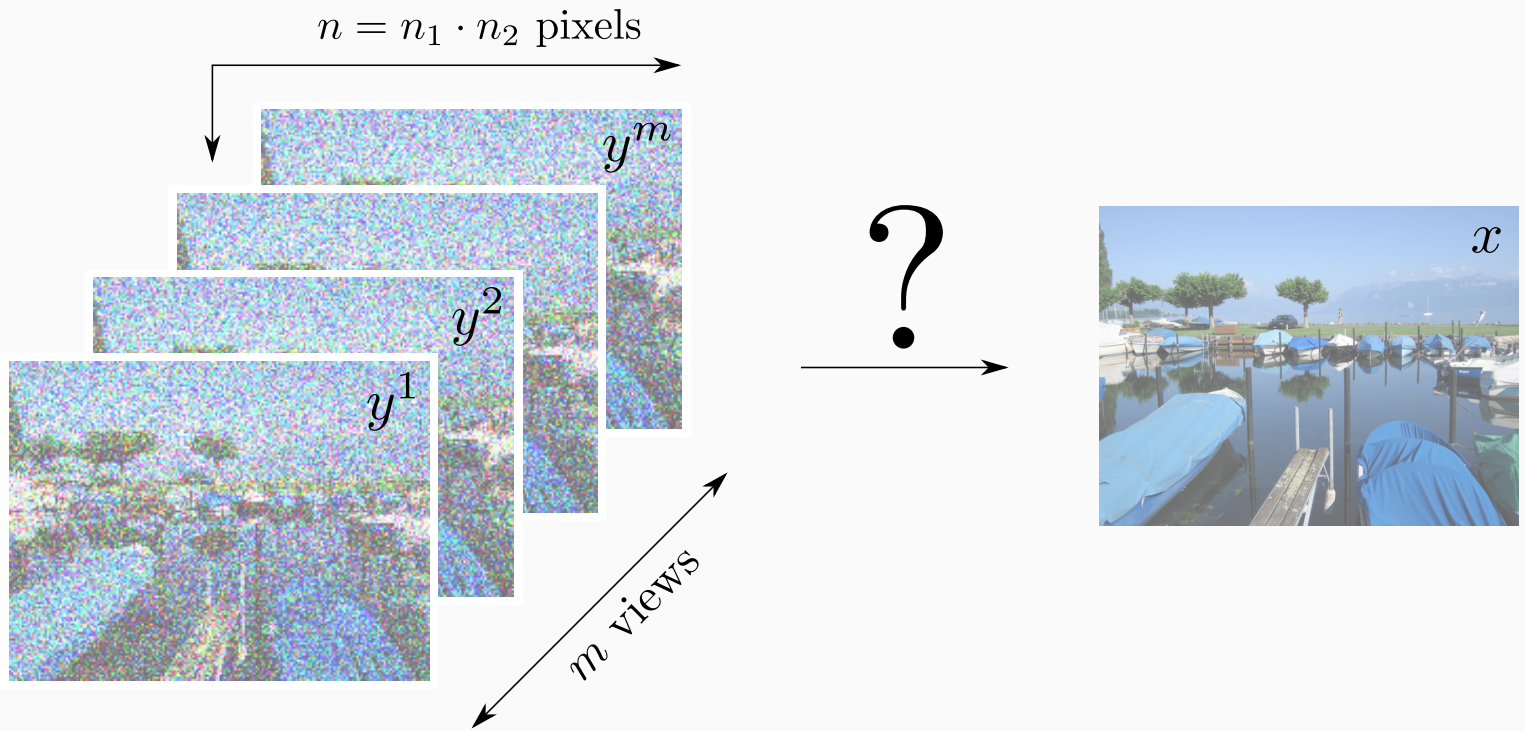
Chapter V – Bayesian methods

Charles Deledalle

May 30, 2019



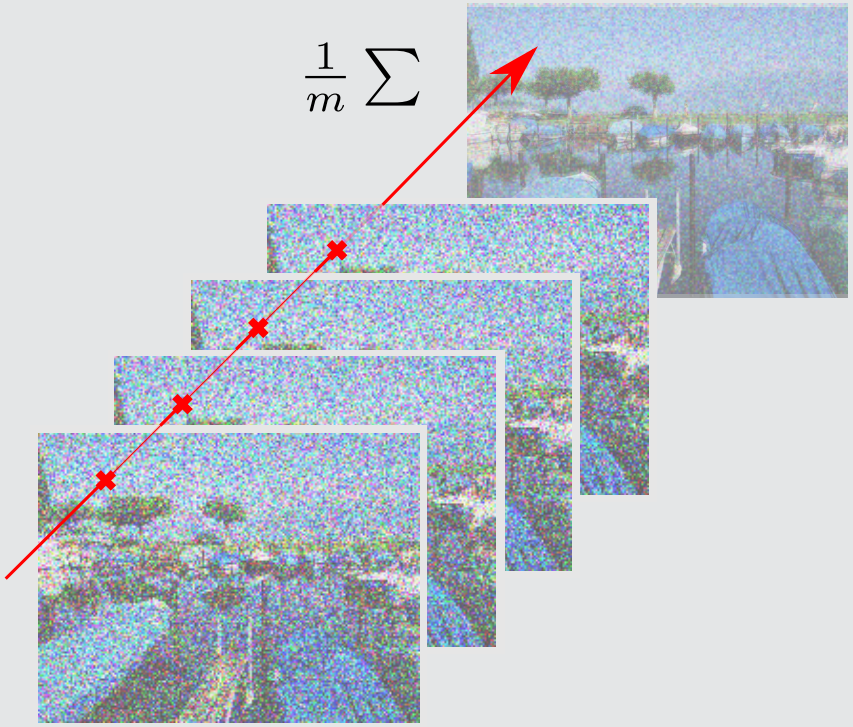
Multiview restoration – Motivation



- Given m corrupted images $y^1, \ldots, y^m$ of a same clean image x
- What is the best approach to retrieve x?

Multiview restoration – Sample mean

Average, *a.k.a.*, sample mean estimator



$$\hat{x} = \frac{1}{m}\sum_{k=1}^{m} y^k$$

Why does it perform denoising?

Multiview restoration – Sample mean

Average, a.k.a., sample mean estimator

- Assume y^k are iid (independent and identically distributed), then

$$\mathbb{E}[\hat{x}] = \mathbb{E}\left[\frac{1}{m}\sum_{k=1}^{m} y^k\right] = \frac{1}{m}\sum_{k=1}^{m}\mathbb{E}\left[y^k\right] = \mathbb{E}\left[y^k\right]$$

$\Rightarrow$ If $\mathbb{E}\left[y^k\right] = x$, the sample-mean is unbiased.

- and $\mathrm{Var}[\hat{x}] = \mathrm{Var}\left[\frac{1}{m}\sum_{k=1}^{m} y^k\right] = \frac{1}{m^2}\sum_{k=1}^{m}\mathrm{Var}\left[y^k\right] = \frac{1}{m^2}\sum_{k=1}^{m}\sigma^2 = \frac{\sigma^2}{m}$

$\Rightarrow$ Sample-mean reduces the fluctuations by $\sqrt{m}$.

Multiview restoration – Sample mean

$$\hat{x} = \frac{1}{m} \sum_{k=1}^{m} y^k$$

(Weak) Law of large numbers

$$\operatorname*{plim}_{m \to \infty} \hat{x} = x \quad \Leftrightarrow \quad \forall \varepsilon > 0, \lim_{m \to \infty} \mathbb{P}[\|\hat{x} - x\|_2 \geqslant \varepsilon] = 0$$

The sample mean $\hat{x}$ is said to be a consistent estimator of x.

As the number of views increase,
the restoration improves with high probability.

Multiview restoration – Sample mean




$\approx$


(a) y^k (Gaussian noise) (b) $m=9$ (SNR $\times 3$) (c) $m=81$ (SNR $\times 9$) (d) x

$$\mathrm{SNR} = \frac{\mathbb{E}[\hat{x}]}{\sqrt{\mathrm{Var}[\hat{x}]}} = \frac{\sqrt{m}x}{\sigma} \xrightarrow[m\to\infty]{} \infty$$

But what if the assumptions are violated?

Multiview restoration – Sample mean


(a) y^k (Impulse noise)


(b) $m = 9$


(c) $m = 81$

$\not\approx$


(e) $\hat{x}$

For impulse noise $\mathbb{E}[y^k] \neq x$

If the assumptions are violated

1. **May not converge towards x (biased estimation).**
2. May not even converge.

Multiview restoration – Sample mean


(a) y^k (Cauchy noise)


(b) $m = 9$


(c) $m = 81$

$\not\approx$


(e) $\hat{x}$

For Cauchy noise $\mathbb{E}[y^k]$ and $\mathrm{Var}[y^k]$ do not exist!

If the assumptions are violated

1. May not converge towards x (biased estimation).
2. **May not even converge.**

Multiview restoration – Sample mean


(a) y^k (Cauchy noise)


(b) $m = 9$


(c) $m = 81$

$\not\approx$


(e) $\hat{x}$

For Cauchy noise $\mathbb{E}[y^k]$ and $\mathrm{Var}[y^k]$ do not exist!

If the assumptions are violated

1. May not converge towards x (biased estimation).
2. May not even converge.

Even though: the convergence can be too slow.

Multiview restoration – Alternatives

Possible alternatives:

Samples $y^k = 4, 10, 3, 6, 2, 3, 2, 2, 4$

- Sample mean (average)

$$\frac{4+10+3+6+2+3+2+2+4}{9} = \{\mathbf{4}\}$$

- Sample median (middle one)

$$2, 2, 2, 3, \mathbf{3}, 4, 4, 6, 10$$

- Sample mode (most frequent one*s*)

$$_{3\times}\mathbf{2}, _{2\times}3, _{2\times}4, _{1\times}6, _{1\times}10$$

Estimator

Random variable Y

- Mean (expectation):

$$\mathbb{E}[Y] = \int_{-\infty}^{+\infty} yp(y;x)\,\mathrm{d}y$$

- Median:

$$\int_{-\infty}^{\mathbf{m}} p(y;x)\,\mathrm{d}y = \int_{\mathbf{m}}^{\infty} p(y;x)\,\mathrm{d}y = \frac{1}{2}$$

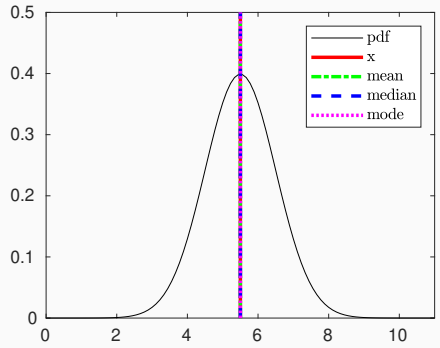
- Mode (distribution peak*s*):

$$\operatorname*{argmax}_y p(y;x)$$

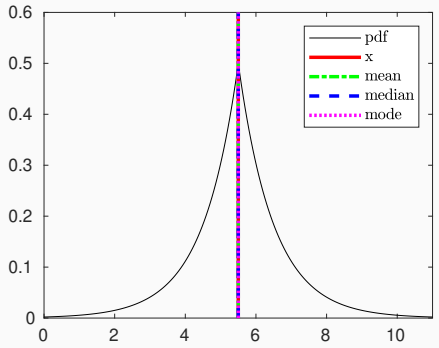
Quantity being estimated

Which one should I pick?

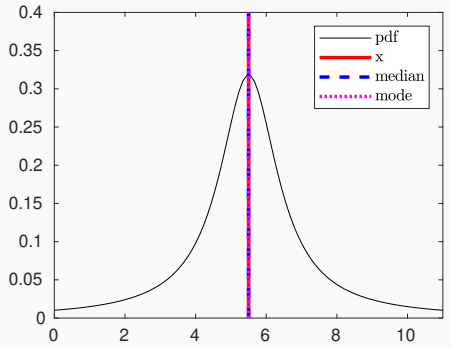
Multiview restoration – Alternatives



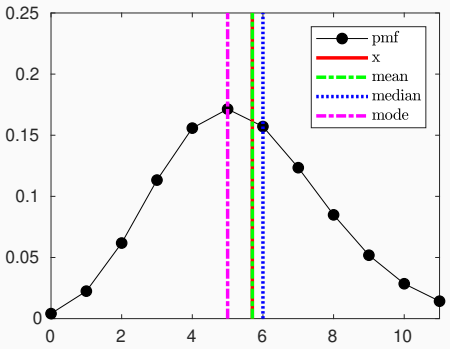
(a) Gaussian law



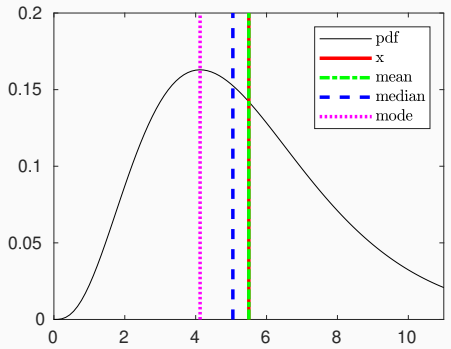
(b) Laplace law



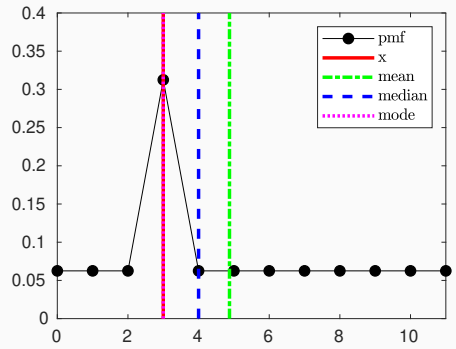
(c) Cauchy law



(d) Poisson law



(e) Gamma law



(f) Impulse law

- Mean/mode/median do not necessarily correspond to the unknown x,
- Often, function of x through a **link function**, ex: $x = g(\mathbb{E}[y])$,
- Should take into account such a link between the images y^k and x.

Method of moments (Karl Pearson, 1894)

$$\text{Assume} \begin{cases} \bullet\ y^k \text{ are iid,} \\ \bullet\ x = g(\mathbb{E}[y^k]), \text{ and} \\ \bullet\ \mathrm{Var}[y^k] \text{ is finite.} \end{cases}$$

1. Consistently estimate $\mu = \mathbb{E}[y^k]$ by sample mean

$$\hat{\mu} = \bar{y} = \frac{1}{m} \sum_{k=1}^{m} y^k$$

2. Next estimate x as

$$\hat{x} = g(\hat{\mu})$$

Properties of method of moments

- Easy to compute if we know g.
- Unbiased if g is linear.
- When g is *sufficiently regular*, asymptotically unbiased:

$$\lim_{m \to \infty} \mathbb{E}[\hat{x}] = x$$

  . . . and consistent:

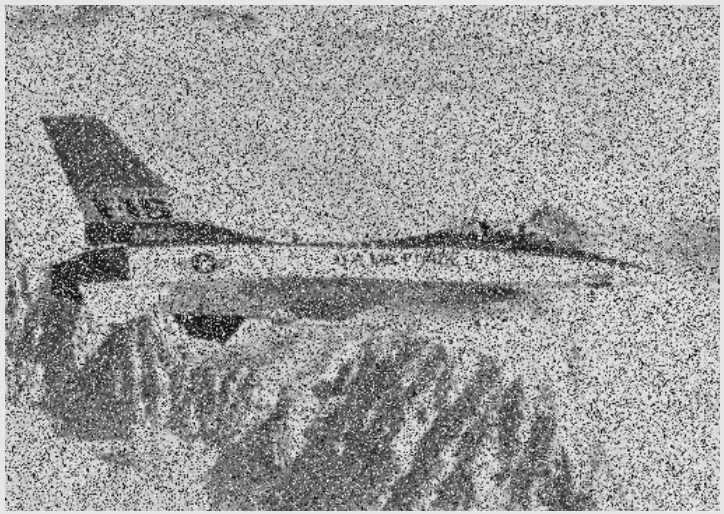
$$\operatorname*{plim}_{m \to \infty} \hat{x} = x \ .$$

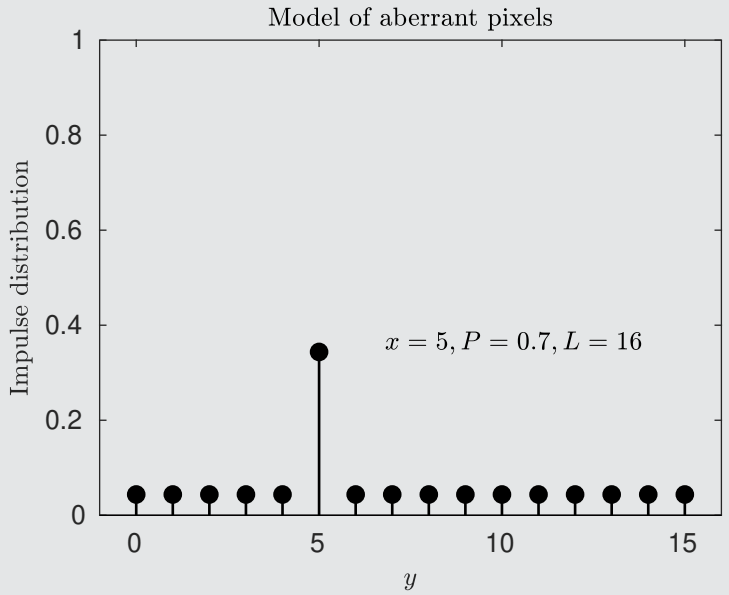
- But:
  - Biased for small m, since: $\mathbb{E}[g(\hat{\mu})] \neq g(\mathbb{E}[\hat{\mu}])$,
  - Often inefficient: slow convergence.

Example (Impulse noise (1/3))

- Consider y and x defined in $[0, L-1]$ such as

$$p(y\,;\,x) = \begin{cases} 1 - P + P/L & \text{if } y = x \\ P/L & \text{otherwise} \end{cases}$$





Example (Impulse noise (2/3))

- We have

$$\begin{aligned}
\mathbb{E}[y] &= \sum_{y=0}^{L-1} y p(y;\, x) \\
&= x(1 - P + P/L) + P/L \sum_{y \neq x} y \\
&= x(1 - P + P/L) + P/L \Big(\underbrace{\sum_{y=0}^{L-1} y}_{= \frac{1}{2}(L-1)L} - x \Big) \\
&= x(1 - P) + \frac{1}{2} P(L-1)
\end{aligned}$$

- Hence $\mathbb{E}[y] = h(x)$ with

$$h(x) = x(1 - P) + \frac{1}{2} P(L-1)$$

Example (Impulse noise (3/3))

- As $\mathbb{E}[y] = h(x)$ with

$$h(x) = x(1-P) + \frac{1}{2}P(L-1)$$

- If $P \neq 1$, h is invertible, and for $g = h^{-1}$, we have

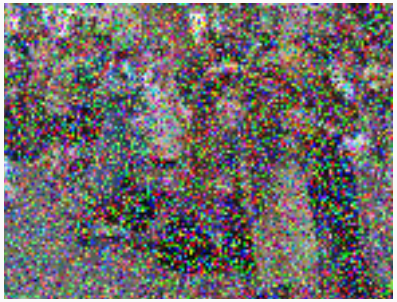
$$x = g(\mathbb{E}[y]) \quad \text{with} \quad g(\mu) = \frac{\mu - \frac{1}{2}P(L-1)}{1-P}$$

- The moment estimator is thus

$$\hat{x} = g(\bar{y}) = \frac{\frac{1}{m}\sum_{k=1}^{m} y^k - \frac{1}{2}P(L-1)}{1-P}$$

  i.e., an affine correction of the sample mean.

Multiview restoration – Method of moments – Results



(a) y^k (Impulse noise)

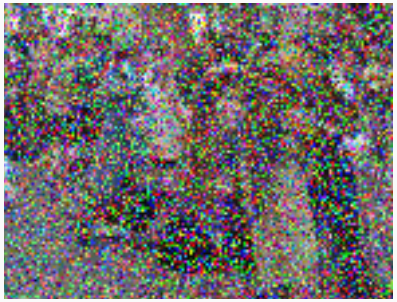
Sample mean

M. of mom.

(b) $m = 9$ (c) $m = 81$ (d) $m = 243$

- Much better than the sample mean ☺

Multiview restoration – Method of moments – Results



(a) y^k (Impulse noise)

(b) $m = 9$ (c) $m = 81$ (d) $m = 243$

- Much better than the sample mean ☺
- Clearly not as efficient as the sample mode ☹

Why?

Mean square error

Mean square error - Risk

Define optimality

- Choose a **loss-function**: $\ell(x, \hat{x})$ such that
  - $\ell(x, x) = 0$
  - $\ell(x, \hat{x}) > 0$: measure the proximity

- Write: $\hat{x} = \hat{x}(y^1, \ldots, y^m)$ with $\hat{x} : \underbrace{\mathbb{R}^n \times \ldots \times \mathbb{R}^n}_{m \text{ times}} \to \mathbb{R}^n$

- Define the **risk or expected loss** as

$$R(x, \hat{x}) = \underbrace{\int \ldots \int \ell(x, \hat{x}(y^1, \ldots, y^m))\, p(y^1, \ldots, y^m; x)\, \mathrm{d}y^1 \ldots \mathrm{d}y^m}_{\text{or in short } \mathbb{E}[\ell(x, \hat{x})]}$$

Mean square error – Definition

Mean square error

- The choice of the loss depends on the application context.
- Most common choice for linear regression problems:

square error (or ℓ_2-loss)

$$\ell(x, \hat{x}) = \|x - \hat{x}\|_2^2 = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$

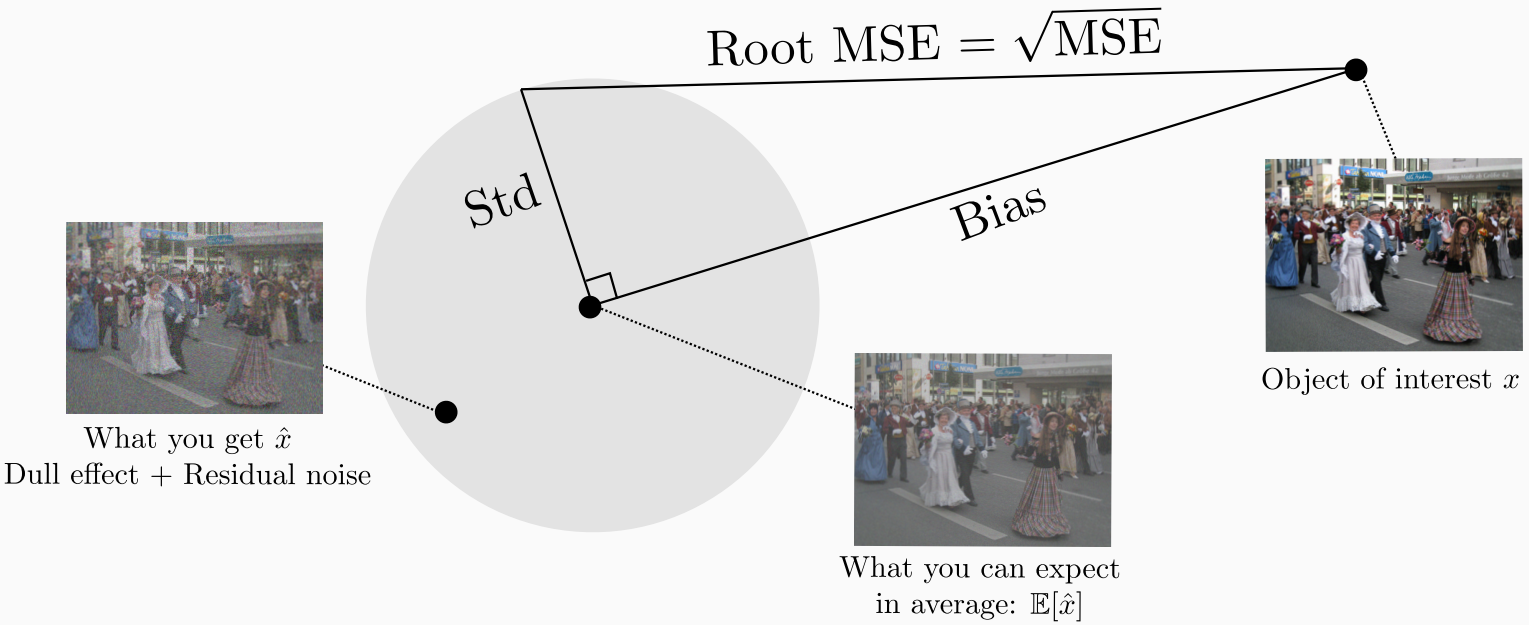
- The expected risk is thus called Mean Square Error (MSE)

$$R(x, \hat{x}) = \mathrm{MSE}(x, \hat{x}) = \mathbb{E}\|x - \hat{x}\|_2^2$$

Mean square error - Bias and Variance

Bias-variance decomposition

$$\begin{aligned}\mathrm{MSE}(x, \hat{x}) &= \sum_{i=1}^{n} (x_i - \mathbb{E}[\hat{x}_i])^2 + \sum_{i=1}^{n} \mathrm{Var}[\hat{x}_i] \\ &= \underbrace{\|x - \mathbb{E}[\hat{x}]\|^2}_{\text{Bias}^2} + \underbrace{\operatorname{tr} \mathrm{Var}[\hat{x}]}_{\text{Variance}}\end{aligned}$$

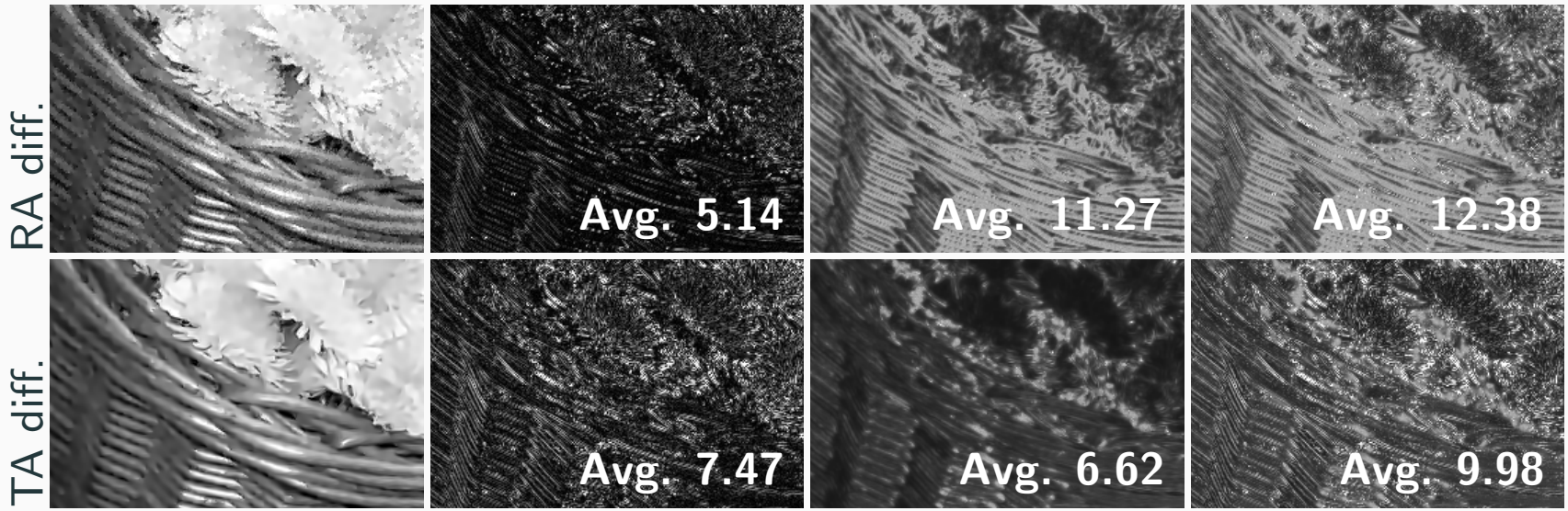


Minimizing the MSE $\equiv$ Minimizing bias^2 and variance

Mean square error – Bias and Variance

Local decomposition: bias and variance are often structure dependent

- Local map of bias: $b_i = |x_i - \mathbb{E}[\hat{x}]_i|$ $\quad$ Bias$^2 = \sum b_i^2$
- Local map of variance: $v_i = \mathrm{Var}[\hat{x}_i]$ $\quad$ Variance $= \sum v_i$



(a) Result $\hat{x}$ (b) Bias (c) Standard deviation (d) Root MSE

- residual noise $\equiv$ high estimation variance
- over-smoothing/blur $\equiv$ bias

Mean square error – Bias and Variance

$$\mathrm{MSE}(x, \hat{x}) = \underbrace{\|x - \mathbb{E}[\hat{x}]\|^2}_{\text{Bias}^2} + \underbrace{\operatorname{tr} \operatorname{Var}[\hat{x}]}_{\text{Variance}}$$

In general, the minimum MSE estimator has
non-zero bias and non-zero variance

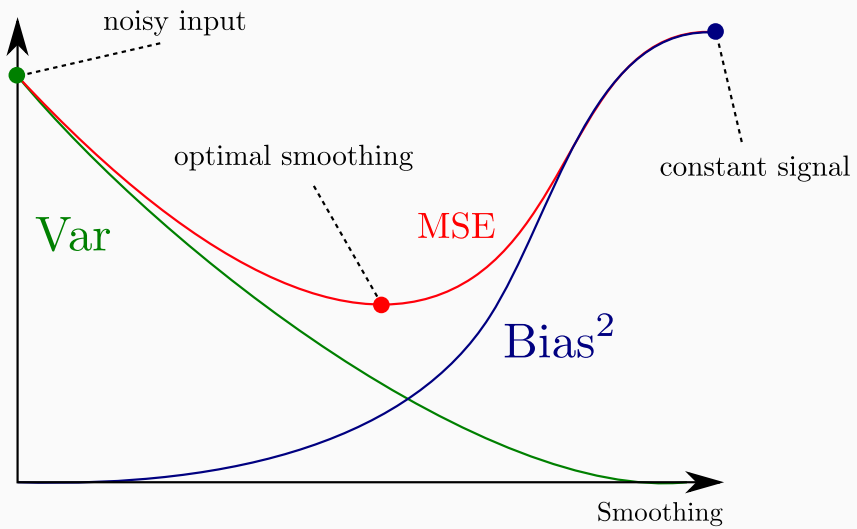


Figure 1 – Smoothing more ⇒ increasing bias while reducing variance

Minimizing the MSE = Finding a trade-off in terms of bias and variance

Mean square error – Optimal estimation

Optimize for a class of estimators

- Choose the optimal estimator as the one minimizing the expected loss

$$\hat{x}^\star = \underset{\hat{x}\in\mathcal{C}}{\operatorname{argmin}} \ \{R(x,\hat{x}) = \mathbb{E}[\ell(x,\hat{x})]\}$$

  for $\mathcal{C}$ a class of estimators.

- Without this restriction, solutions can be **unrealizable**,

$$\text{ex: trivial solution } \hat{x}(y^1,\ldots,y^m) = x$$

  i.e., the solution would depend on the unknown.

Example (Amplified sample mean (1/3))

- Assume y^k iid with $\mathbb{E}[y^k] = x$ and $\mathrm{Var}[y^k] = \sigma^2 \mathrm{Id}_n$.
- Consider the optimal estimator with respect to the MSE

$$\hat{x}^\star = \underset{\hat{x} \in \mathcal{C}}{\operatorname{argmin}} \ \left\{ \mathrm{MSE}(x, \hat{x}) = \mathbb{E}\|x - \hat{x}\|_2^2 \right\}$$

- And the class of functions that amplify the sample mean

$$\mathcal{C} = \left\{ \hat{x} : \underbrace{\mathbb{R}^n \times \ldots \times \mathbb{R}^n}_{m \text{ times}} \to \mathbb{R}^n \ ; \ \exists \alpha \in \mathbb{R}, \quad \hat{x}(y^1, \ldots, y^m) = \frac{\alpha}{m} \sum_{k=1}^{m} y^k \right\}$$

- Finding $\hat{x}^\star$ leads to find $\alpha^\star$ that minimizes the MSE.

$$\hat{x}(y^1, \dots, y^m) = \frac{\alpha}{m} \sum_{k=1}^{m} y^k$$

Example (Amplified sample mean (2/3))

- Study the first order derivative

$$\begin{aligned}
\frac{\partial \mathrm{MSE}(x, \hat{x})}{\partial \alpha} &= \frac{\partial \|x - \mathbb{E}[\hat{x}(y)]\|_2^2}{\partial \alpha} + \frac{\partial \operatorname{tr} \operatorname{Var}[\hat{x}(y)]}{\partial \alpha} \\
&= \frac{\partial \|x - \alpha x\|_2^2}{\partial \alpha} + \frac{\partial \alpha^2 \operatorname{tr} \operatorname{Var}[y]/m}{\partial \alpha} \\
&= \|x\|^2 \frac{\partial (1-\alpha)^2}{\partial \alpha} + \frac{n\sigma^2}{m} \frac{\partial \alpha^2}{\partial \alpha} \\
&= -2\|x\|^2 (1-\alpha) + \frac{2n}{m} \sigma^2 \alpha
\end{aligned}$$

- Moreover, the second order derivative is constant and positive

$$\frac{\partial^2 \mathrm{MSE}}{\partial \alpha^2} = 2\|x\|^2 + \frac{2n}{m} \sigma^2$$

Example (Amplified sample mean (3/3))

- Then the MSE is a quadratic function and its minimum is given for

$$\frac{\partial \mathrm{MSE}}{\partial \alpha} = 0 \quad \Leftrightarrow \quad -2\|x\|^2(1-\alpha) + \frac{2n}{m}\sigma^2\alpha = 0$$

$$\Leftrightarrow \quad \alpha^\star = \frac{\|x\|^2}{\|x\|^2 + \frac{n\sigma^2}{m}}$$

Define: $\mathrm{SNR}^2 = \dfrac{\|x\|_2^2}{n\sigma^2}$

$\alpha^\star = \frac{\mathrm{SNR}^2}{\mathrm{SNR}^2+1/m}$:

- large SNR, y^k has good quality, average: $\hat{x} = \bar{y}$,
- low SNR, x drawn in the noise: $\hat{x} = 0$ is safer.

This is not realizable since it depends on the unknown x.
We need an alternative to direct MSE minimization.

Unbiased estimators

Unbiased estimators – Minimum Variance Unbiased Estimator

- In the previous example

$$\frac{\partial \mathrm{MSE}(x, \hat{x})}{\partial \alpha} = \frac{\partial \mathrm{Bias}^2}{\partial \alpha} + \frac{\partial \mathrm{Variance}}{\partial \alpha}$$

$$\text{where} \quad \frac{\partial \mathrm{Bias}^2}{\partial \alpha} = -2\|x\|^2(1-\alpha) \quad \text{and} \quad \frac{\partial \mathrm{Variance}}{\partial \alpha} = \frac{2n}{m}\sigma^2\alpha$$

- The dependency on $\|x\|^2$ arises from the bias term,
- This occurs in many situations.

Constrain the estimator to be unbiased.

Find the estimator that produces the minimum variance.

This will provide the minimum MSE among all unbiased estimators.

Minimum Variance Unbiased Estimator (MVUE)

An estimator $\hat{x}^\star \in \mathcal{C}$ is the MVUE for the class of estimators $\mathcal{C}$ if

$$\underbrace{(\forall x,\ \mathbb{E}\hat{x}^\star = x)}_{\text{unbiasedness}} \quad \text{and} \quad \underbrace{\forall \tilde{x} \in \mathcal{C},\ (\forall x,\ \mathbb{E}\tilde{x} = x) \Rightarrow (\forall x,\ \operatorname{tr}\operatorname{Var}[\hat{x}^\star] \leqslant \operatorname{tr}\operatorname{Var}[\tilde{x}])}_{\text{minimum variance}}$$

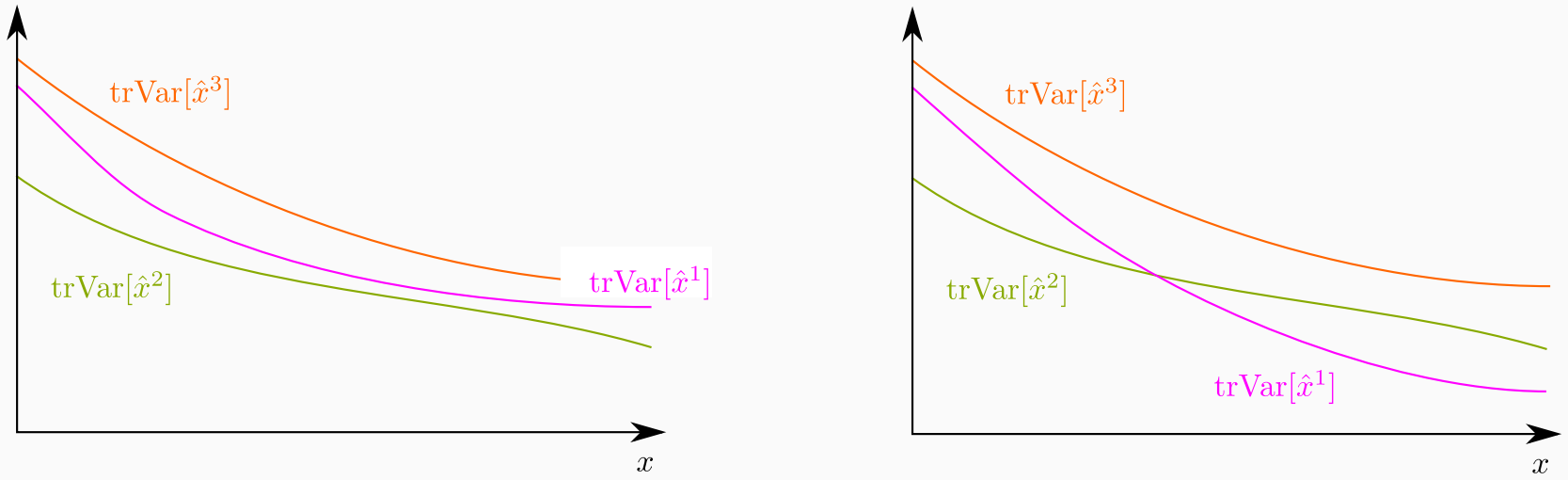
Example (Back to the amplified sample mean)

- An unbiased estimator that amplifies the sample mean should satisfy

$$\text{Bias}^2 = \|x - \mathbb{E}\left[\frac{\alpha}{m}\sum_{k=1}^{m} y^k\right]\|^2 = (1-\alpha)^2\|x\|^2 = 0 \quad \Rightarrow \quad \alpha^\star = 1$$

- Then the sample mean is the only unbiased estimator of this class.
- It is the MVUE of this class.

Unbiased estimators – Minimum Variance Unbiased Estimator



Variance of 3 unbiased estimators $\mathbb{E}[\hat{x}^1] = \mathbb{E}[\hat{x}^2] = \mathbb{E}[\hat{x}^3] = x$ as a function of x.

Quiz: which one is the MVUE on the left? on the right?

The MVUE does not always exist.

If it does exist, how to find it?

Unbiased estimators – Cramér-Rao bound

Cramér-Rao lower bound ($\approx$ 1945)

- Provided the bound exists, any **realizable unbiased estimator** satisfies

$$\mathrm{Var}[\hat{x}] \geqslant \mathcal{I}^{-1} \quad \text{where} \quad \mathcal{I}_{i,j} = \mathbb{E}\left[\frac{\partial^2 - \log p(y^1, \dots, y^m; x)}{\partial x_i \partial x_j} \bigg|_x \right]$$

- $\mathrm{Var}[\hat{x}] \geqslant \mathcal{I}^{-1}$ means $\mathrm{Var}[\hat{x}] - \mathcal{I}^{-1}$ is symmetric positive definite,
- $y^1, \dots, y^m \mapsto p(y^1, \dots, y^m; x)$ is the law of the observations,
- $x \mapsto p(y^1, \dots, y^m; x)$ is the likelihood of the unknown,
- $\mathcal{I}$ the Fisher information matrix: expected Hessian of the log-likelihood,
- $\mathcal{I}$ measures the organization/entropy/simplicity of the problem,
- Ex: small noise $\to$ likelihood peaky $\to$ large Hessian $\to$ small bound.

Unbiased estimators – Cramér-Rao bound

Consequence: $\mathrm{Var}[\hat{x}] = \mathcal{I}^{-1} \quad \Rightarrow \quad \hat{x}$ is the MVUE

If you find an unbiased estimator that reaches the Cramér-Rao bound, then it is the MVUE.

⚠ **The MVUE may not reach the Cramér-Rao bound, in this case, no estimator reaches the bound.**

Unbiased estimators – Efficiency

Efficiency

- The ratio $0 \leqslant \dfrac{\operatorname{tr} \mathcal{I}^{-1}}{\operatorname{tr} \operatorname{Var}[\hat{x}]} \leqslant 1$ is called efficiency of the estimator,
- Measures by how much the estimator is close to the Cramér-Rao bound,
- An unbiased estimator with efficiency 100% is said to be efficient,
- If an efficient estimator exists, it is the MVUE,
- The MVUE is not necessarily efficient.

If we cannot build an efficient estimator, we may build one whose efficiency converges to 100% with the number of observations m.

Unbiased estimators – Maximum Likelihood Estimator

Maximum likelihood estimators (MLE) **[Fisher, 1913 and others]**

- Define the MLE, for y^k iid, as one of the global maxima of the likelihood

$$\hat{x} \in \operatorname*{argmax}_x p(y^1, \dots, y^m; x) = \operatorname*{argmax}_x \prod_{k=1}^{m} p(y^k; x)$$

- Then $\hat{x}$ is asymptotically unbiased, asymptotically efficient and consistent

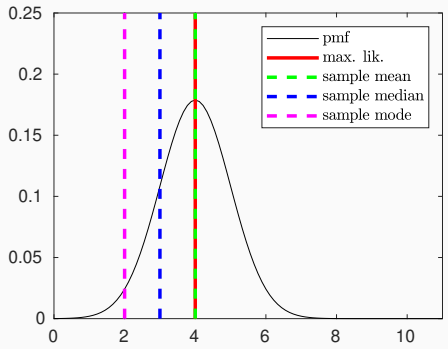
$$\lim_{m\to\infty} \mathbb{E}[\hat{x}] = x, \quad \lim_{m\to\infty} \operatorname{Var}[\hat{x}] = 0, \quad \operatorname*{plim}_{m\to\infty} \hat{x} = x \quad \text{and} \quad \lim_{m\to\infty} \frac{\operatorname{tr} \mathcal{I}^{-1}}{\operatorname{tr} \operatorname{Var}[\hat{x}]} = 1$$

- If an efficient estimator exists, it is the MLE and, then, the MVUE.

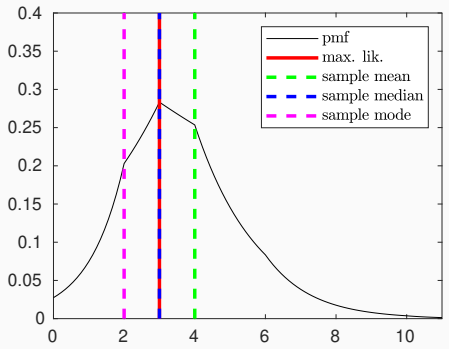
- Otherwise, the MVUE might be more efficient than the MLE.

MLE: look for the image that best explains the observations.

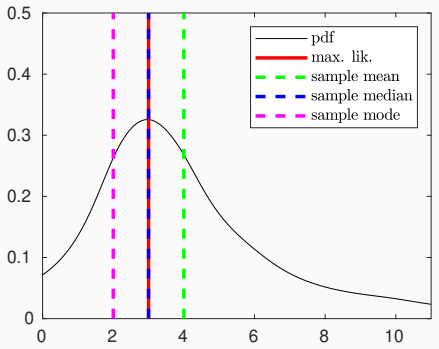
Unbiased estimators – Maximum Likelihood Estimator



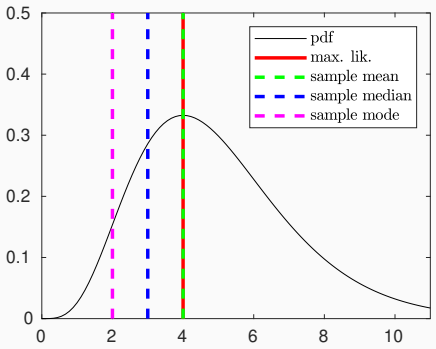
(a) Gaussian likelihood



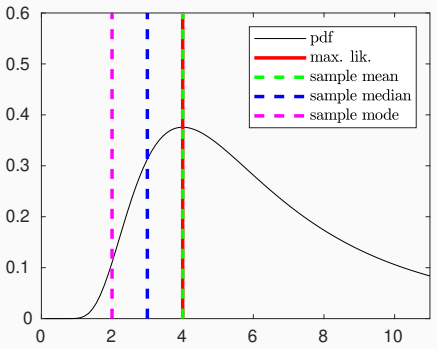
(b) Laplace likelihood



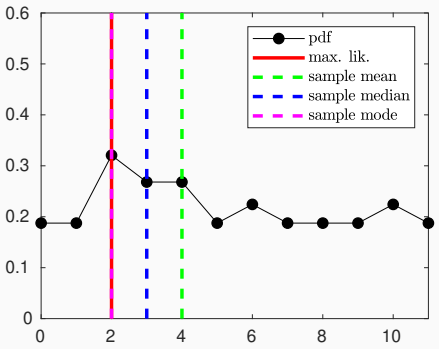
(c) Cauchy likelihood



(d) Poisson likelihood



(e) Gamma likelihood



(f) Impulse likelihood

- These curves are not laws but likelihoods: $x \mapsto p(y^1, \ldots, y^m; x)$.
- The considered samples are $y^k = 4, 10, 3, 6, 2, 3, 2, 2, 4$.
- MLE can be the sample mean, median, mode, or something else.

How to find the MLE for y^k iid?

- Rewrite the MLE as a **minimization problem**

$$\hat{x}^\star \in \operatorname*{argmax}_x \prod_{k=1}^{m} p(y^k; x) = \operatorname*{argmin}_x \underbrace{\sum_{k=1}^{m} -\log p(y^k; x)}_{\hat{\ell}(x)}$$

- Why $-\log$?
  - Strictly decreasing function: same locations for the optima,
  - Sums are easier to manipulate than products,
  - $\prod p(y^k; x)$ very small: safer to manipulate -16 than 10^{-16}.

Unbiased estimators – Maximum Likelihood Estimator

How to find the MLE for y^k iid?

- If $\hat{\ell}$ differentiable: find $\hat{x}^\star$ that **cancels the gradient**
  - Twice diff. at $\hat{x}^\star$ and $\hat{\ell}''(\hat{x}^\star) > 0$: a local minimum
  - $\hat{\ell}$ convex: a global minimum

$$\hat{\ell}(\alpha x + \beta y) \leqslant \alpha \hat{\ell}(x) + \beta \hat{\ell}(y)$$

  - Twice diff. everywhere and $\hat{\ell}''(x) \geqslant 0$: $\hat{\ell}$ convex $\Rightarrow$ a global minimum

- Often analytical solutions, otherwise use **numerical solvers**
  (ex: Gradient descent, Newton-Raphson, Expectation-Maximization)

In multivariate setting, $\hat{\ell}''(x) > 0$ means
the Hessian is symmetric positive definite at x.

Example (Poisson noise (1/3))

- Consider samples $y^1, \dots, y^m \in \mathbb{N}$ iid versions of $x > 0$ such that

$$p(y; x) = \frac{x^y e^{-x}}{y!}$$

- We have

$$\hat{\ell}(x) = \sum_{k=1}^{m} -\log p(y^k; x) = \sum_{k=1}^{m} -y^k \log x + x - \log y^k!$$

Example (Poisson noise (1/3))

- Consider samples $y^1, \dots, y^m \in \mathbb{N}$ iid versions of $x > 0$ such that

$$p(y; x) = \frac{x^y e^{-x}}{y!}$$

- We have

$$\hat{\ell}(x) = \sum_{k=1}^{m} -\log p(y^k; x) = \sum_{k=1}^{m} -y^k \log x + x - \log y^k!$$

- It follows that $\hat{\ell}$ is twice differentiable and

$$\hat{\ell}'(x) = \sum_{k=1}^{m} \left(-\frac{y^k}{x} + 1 \right) = m - \frac{1}{x} \sum_{k=1}^{m} y^k \quad \text{and} \quad \hat{\ell}''(x) = \frac{1}{x^2} \sum_{k=1}^{m} y^k$$

Example (Poisson noise (1/3))

- Consider samples $y^1, \dots, y^m \in \mathbb{N}$ iid versions of $x > 0$ such that

$$p(y; x) = \frac{x^y e^{-x}}{y!}$$

- We have

$$\hat{\ell}(x) = \sum_{k=1}^{m} -\log p(y^k; x) = \sum_{k=1}^{m} -y^k \log x + x - \log y^k!$$

- It follows that $\hat{\ell}$ is twice differentiable and

$$\hat{\ell}'(x) = \sum_{k=1}^{m} \left(-\frac{y^k}{x} + 1 \right) = m - \frac{1}{x} \sum_{k=1}^{m} y^k \quad \text{and} \quad \hat{\ell}''(x) = \frac{1}{x^2} \sum_{k=1}^{m} y^k$$

- For all x, $\hat{\ell}''(x) > 0$, then $\hat{\ell}''$ is convex and thus $\hat{x}^\star$ satisfies

$$m - \frac{1}{\hat{x}^\star} \sum_{k=1}^{m} y^k = 0 \quad \Leftrightarrow \quad \hat{x}^\star = \frac{1}{m} \sum_{k=1}^{m} y^k$$

Example (Poisson noise (2/3))

- The MLE for Poisson noise is unique and is the sample mean,
- Recall that for Poisson noise $\mathbb{E}[y^k] = x$ and $\mathrm{Var}[y^k] = x$,
- It follows that the variance of the MLE is

$$\mathrm{Var}[\hat{x}^\star] = \frac{1}{m^2}\sum_{k=1}^{m}\mathrm{Var}[y^k] = \frac{1}{m^2}\sum_{k=1}^{m} x = \frac{x}{m}$$

Example (Poisson noise (2/3))

- The MLE for Poisson noise is unique and is the sample mean,
- Recall that for Poisson noise $\mathbb{E}[y^k] = x$ and $\mathrm{Var}[y^k] = x$,
- It follows that the variance of the MLE is

$$\mathrm{Var}[\hat{x}^\star] = \frac{1}{m^2}\sum_{k=1}^{m}\mathrm{Var}[y^k] = \frac{1}{m^2}\sum_{k=1}^{m} x = \frac{x}{m}$$

- Besides, the Fisher information is

$$\mathcal{I} = \mathbb{E}[\hat{\ell}''(x)] = \mathbb{E}\left[\frac{1}{x^2}\sum_{k=1}^{m} y^k\right] = \frac{1}{x^2}\sum_{k=1}^{m}\mathbb{E}[y^k] = \frac{1}{x^2}\sum_{k=1}^{m} x = \frac{m}{x}$$

Example (Poisson noise (2/3))

- The MLE for Poisson noise is unique and is the sample mean,
- Recall that for Poisson noise $\mathbb{E}[y^k] = x$ and $\mathrm{Var}[y^k] = x$,
- It follows that the variance of the MLE is

$$\mathrm{Var}[\hat{x}^\star] = \frac{1}{m^2}\sum_{k=1}^{m}\mathrm{Var}[y^k] = \frac{1}{m^2}\sum_{k=1}^{m} x = \frac{x}{m}$$

- Besides, the Fisher information is

$$\mathcal{I} = \mathbb{E}[\hat{\ell}''(x)] = \mathbb{E}\left[\frac{1}{x^2}\sum_{k=1}^{m} y^k\right] = \frac{1}{x^2}\sum_{k=1}^{m}\mathbb{E}[y^k] = \frac{1}{x^2}\sum_{k=1}^{m} x = \frac{m}{x}$$

- Hence

$$\mathrm{Var}[\hat{x}^\star] = \mathcal{I}^{-1}$$

- The MLE reaches the Cramér-Rao, even non-asymptotically.
- It is 100% efficient for all m, it's the MVUE.

Unbiased estimators – Maximum Likelihood Estimator

Law	MLE	Comments	MVUE
Gaussian	Sample mean	100% efficient for all m	$\surd$
		(sample median $\approx 64\%$*)*	
Poisson	Sample mean	100% efficient for all m	$\surd$
Gamma	Sample mean	100% efficient for all m	$\surd$
Cauchy	No closed form	*(sample median* $\approx 81\%$*)*	
		(24%*-trimmed mean* $\approx 88\%$*)*	
Laplacian	Sample median	asymptotically efficient with m	
Impulse	Sample mode	no CR bound	

Remarks: robust estimators

- Even though the sample mean is often the MLE, the sample median (which always converges) or the trimmed mean are often preferred when the noise distribution is unknown or known approximately.
  (their efficiency often drops slower under mis-specified noise models)
- More robust to outliers, when some samples are not iid.

Least square estimator

Least square – Best Linear Unbiased Estimator

Motivation:
- The MVUE not always exist,
- Can be difficult to find.

Idea:
- Restrict the estimator to be linear with respect to y,
- Restrict the estimator to be unbiased,
- Find the best one (*i.e.*, with minimum variance)

Definition (BLUE: Best linear unbiased estimator)

- Consider $y^k \in \mathbb{R}^n$ and $x \in \mathbb{R}^p$,
- BLUE is the MVUE for the class of linear estimators

$$\mathcal{C} = \left\{ \hat{x} \; ; \; \exists \boldsymbol{A}_1, \dots, \boldsymbol{A}_m \in \mathbb{R}^{p \times n}, \quad \hat{x}(y^1, \dots, y^m) = \sum_{k=1}^{m} \boldsymbol{A}_k y^k \right\}$$

Least square – Best Linear Unbiased Estimator

Theorem (Gauss-Markov theorem)

- *Assume*
  - $\boldsymbol{H} \in \mathbb{R}^{n \times p}$ *with rank* $p \leqslant n$ *(i.e., over-determined)*
  - $\mathbb{E}[y^k] = \boldsymbol{H} x$
  - $\mathrm{Var}[y^k] = \boldsymbol{\Sigma}$
- *Then, the BLUE is*

$$\hat{x}^\star = (\boldsymbol{H}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^* \boldsymbol{\Sigma}^{-1} \bar{y} \quad \textit{where} \quad \bar{y} = \frac{1}{m} \sum_{k=1}^{m} y^k \ .$$

- BLUE always exists for over-determined linear regression problems,
- Can be used even though we do not know precisely $p(y^1, \ldots, y^m; x)$ as:

It only requires the noise to be zero-mean,
and knowing its covariance matrix.

- $\boldsymbol{H}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{H}$ is always sdp $\Rightarrow$ can be inverted by conjugate gradient.

Least square – Best Linear Unbiased Estimator

MLE with Gaussian noise = BLUE = LSE

- Assume
  - $\boldsymbol{H} \in \mathbb{R}^{n \times p}$ with rank $p \leqslant n$ (i.e., over-determined),
  - $y^k \sim \mathcal{N}(\boldsymbol{H}x, \boldsymbol{\Sigma})$ and independent.

- Then, the MLE is unique and is the **least square estimator** (LSE)

$$\begin{aligned} x^\star &= \operatorname*{argmin}_x \sum_{k=1}^{m} \|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{H}x - y^k)\|_2^2 \\ &= (\boldsymbol{H}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^*\boldsymbol{\Sigma}^{-1}\bar{y} \end{aligned}$$

- It is also the BLUE and the MVUE.

Imposing linearity and unbiasedness
(sub-optimal in general)

$\equiv$

Imposing unbiasedness and assuming Gaussianity.
(optimal under this assumption)

Least square – Least square for super-resolution

```
p1, p2 = x.shape[:2]
sigma  = 60/255
m      = 20

# Subsampling: each line is the average of two consecutive ones
H      = lambda x: (x[0::2, :] + x[1::2, :]) / 2
y      = [ H(x) + sigma * np.random.randn(int(p1 / 2), p2, 3) for k in range(m) ]

# Adjoint of H: each line is duplicated and divided by two
Ha     = lambda x: x[[int(i/2) for i in range(p1)], :] / 2

# Least square solution with cg (Note: non optimal)
ybar   = np.mean(y, axis=0)
xblue  = im.cg(lambda x: Ha(H(x)), Ha(ybar))
```

$$\boldsymbol{\Sigma} = \sigma^2 \mathrm{Id}$$
$$\hat{x}^\star = (\boldsymbol{H}^* \boldsymbol{H})^{-1} \boldsymbol{H}^* \bar{y}$$

(a) $y^k = \boldsymbol{H}x + w^k$

(b) $\hat{x}^\star$, $m = 1$

(c) $\hat{x}^\star$, $m = 4$

(d) $\hat{x}^\star$, $m = 20$

Oops, we have not checked that H^*H was invertible.

In image processing tasks

- H is almost always under-determined $\Rightarrow$ H^*H is non-invertible.

Examples

- Low-pass: sets high frequencies to zero, thus non-invertible,
- Radon transform: sets some frequencies to zero, thus non-invertible,
- Inpainting: sets some pixels to zero, thus non-invertible,
- Super-resolution: $p > n$ the problem is under-determined.

If H is non-invertible, BLUE does not exist.
Realizable estimators cannot be unbiased.
They cannot guess, not even in average, what was lost.

So, what was `im.cg` computing in the previous example?

Least-square estimator and normal equation

- `im.cg` finds one of the **infinite solutions** of the least square problem

$$x^\star \in \operatorname*{argmin}_x \|\boldsymbol{H}x - \bar{y}\|_2^2$$

- They are characterized by the **normal equation**

$$\boldsymbol{H}^*\boldsymbol{H}x^\star = \boldsymbol{H}^*\bar{y}$$

- If initialized to zero, `im.cg` finds the one with minimum norm $\|\hat{x}^\star\|_2$.
- This solution reads as $\hat{x}^\star = \boldsymbol{H}^+\bar{y}$

  where $\boldsymbol{H}^+ \in \mathbb{R}^{p\times n}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{H}$.

Moore-Penrose pseudo-inverse

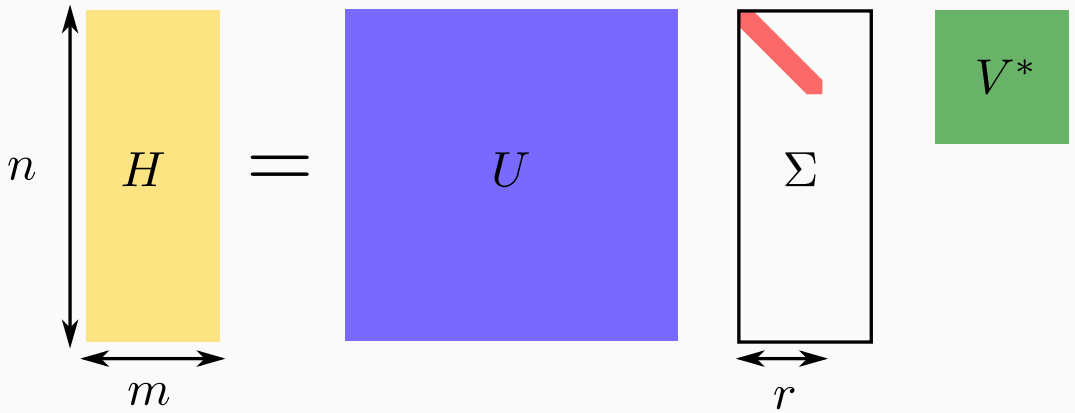
- The Moore-Penrose pseudo-inverse is the unique matrix satisfying
  1. $\boldsymbol{H}\boldsymbol{H}^{+}\boldsymbol{H} = \boldsymbol{H}$
  2. $\boldsymbol{H}^{+}\boldsymbol{H}\boldsymbol{H}^{+} = \boldsymbol{H}^{+}$
  3. $(\boldsymbol{H}\boldsymbol{H}^{+})^{*} = \boldsymbol{H}\boldsymbol{H}^{+}$
  4. $(\boldsymbol{H}^{+}\boldsymbol{H})^{*} = \boldsymbol{H}^{+}\boldsymbol{H}$
- The Moore-Penrose pseudo-inverse always exists.
- If $\boldsymbol{H}$ is square and invertible: $\boldsymbol{H}^{+} = \boldsymbol{H}^{-1}$
- $\boldsymbol{H}^{+}$ also satisfy: $\boldsymbol{H}^{+} = (\boldsymbol{H}^{*}\boldsymbol{H})^{+}\boldsymbol{H}^{*} = \boldsymbol{H}^{*}(\boldsymbol{H}\boldsymbol{H}^{*})^{+}$
- If $\boldsymbol{H}$ has full rank: $\boldsymbol{H}^{+} = (\boldsymbol{H}^{*}\boldsymbol{H})^{-1}\boldsymbol{H}^{*}$, we recover Gauss-Markov thm.

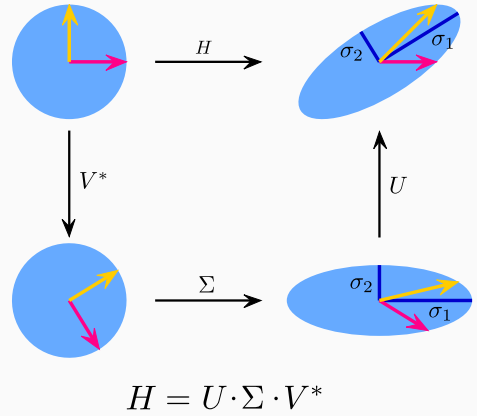
Small detour to Singular Value Decompositions (SVD)

- Any matrix $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ admits a Singular Value Decomposition (SVD) as

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \quad \text{with} \quad \begin{cases} \bullet\ \boldsymbol{U} \in \mathbb{C}^{n\times n},\ \boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{U}^*\boldsymbol{U} = \mathrm{Id}_n \\ \bullet\ \boldsymbol{V} \in \mathbb{C}^{m\times m},\ \boldsymbol{V}^*\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^* = \mathrm{Id}_m \\ \bullet\ \boldsymbol{\Sigma} \in \mathbb{R}^{n\times m} \text{ a diagonal matrix.} \end{cases}$$

- $\sigma_i = \boldsymbol{\Sigma}_{ii} > 0$: called singular values (often sorted in decreasing order),
- Rank $r \leqslant \min(n, m)$: number of non-zero singular values.





$H = U \cdot \Sigma \cdot V^*$

Least square – Moore-Penrose pseudo-inverse

SVD, image and null space

- If the singular values are sorted in decreasing order

$$\begin{aligned}\mathrm{Im}[\boldsymbol{H}] &= \{y \in \mathbb{R}^n \ ; \ \exists x \in \mathbb{R}^m, \ y = \boldsymbol{H}x\} \\ &= \mathrm{Span}(\{u_i \in \mathbb{R}^n \ ; \ i \in [1 \dots r]\}) \qquad \text{(what can be observed)}\end{aligned}$$

$$\begin{aligned}\mathrm{Ker}[\boldsymbol{H}] &= \{x \in \mathbb{R}^m \ ; \ \boldsymbol{H}x = 0\} \\ &= \mathrm{Span}(\{v_i \in \mathbb{R}^m \ ; \ i \in [r+1 \dots m]\}) \qquad \text{(what is lost)}\end{aligned}$$

where u_i are the columns of $\boldsymbol{U}$ and v_i are the columns of $\boldsymbol{V}$

Null-space: set of zero-frequencies, set of missing pixels, ...

Least square – Moore-Penrose pseudo-inverse

SVD and Moore-Penrose pseudo-inverse

- Let $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ be its SVD, the Moore-Penrose pseudo inverse is

$$\boldsymbol{H}^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^* \quad \text{where} \quad \sigma_i^+ = \begin{cases} \frac{1}{\sigma_i} & \text{if } \sigma_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

- For deconvolution: SVD $\cong$ eigendecomposition $\cong$ Fourier decomposition
  $\Rightarrow$ inversion of the non-zero frequencies.

- Difficulty: $\frac{1}{\sigma_i}$ can be very large (ill-conditioned matrix)
  $\Rightarrow$ numerical issues (refer to Assignment 6).

Compared to other least square solutions, the Moore-Penrose pseudo-inverse does not create new content in $\mathrm{Ker}[\boldsymbol{H}]$, *i.e.*, where the information was lost.

It is unbiased only on $\mathrm{Ker}[\boldsymbol{H}]^\perp$.

Least square – Pseudo inverse for deconvolution

```
n1, n2 = x.shape[:2]
sigma = 2/255
m = 20

# Deconvolution problem setting
nu  = im.kernel('gaussian', tau=2, s1=20, s2=20)
lbd = im.kernel2fft(nu, n1, n2)
H   = lambda x: im.convolvefft(x, lbd)
y   = [ H(x) + sigma * np.random.randn(n1, n2, 3) for i in range(m) ]

# Numerical Fourier approximation of the pseudo inverse
lbd_pinv = 1 / lbd
lbd_pinv[np.abs(lbd) < 1e-2] = 0
ybar = np.mean(y, axis=0)
x_pinv = im.convolvefft(ybar, lbd_pinv)
```

In practice, the threshold 1e−2 is difficult to choose

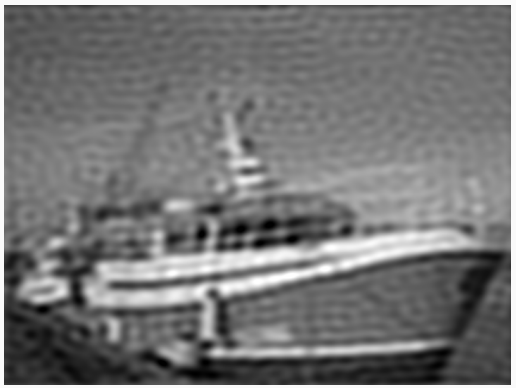
(a) y^k (b) $m = 1$ (c) $m = 10$ (d) $m = 20$

Least square and unbiased estimator issues

The M.-P. pseudo-inverse does not create new content on $\mathrm{Ker}[\boldsymbol{H}]$, it cannot recover missing information ☹

Moreover, unbiased estimators amplify and colorize noise.

$\longrightarrow$
PINV/LSE/MLE/MVUE



- In practice, the number m of samples/frames/views is small,
- The asymptotic behavior when $m \to \infty$ is far from being reached,
- In fact, in our contexts of interest, we often have $m = 1$.

Bayesian approach

Bayesian approach – Motivations

Why unbiased estimators are not working in our context?

Because they attempt to be optimal
whatever the underlying image x.

Bayesian answer:
Forget about unbiasedness and be optimal
only for the class of images x that looks like clean images.

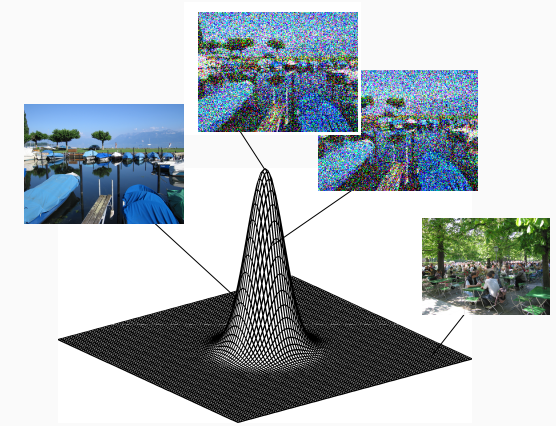
Expected behavior:
If x looks like a clean image: small bias, small variance.
If x does not look like a clean image: large bias and/or large variance.

Bayesian approach – Random vector models

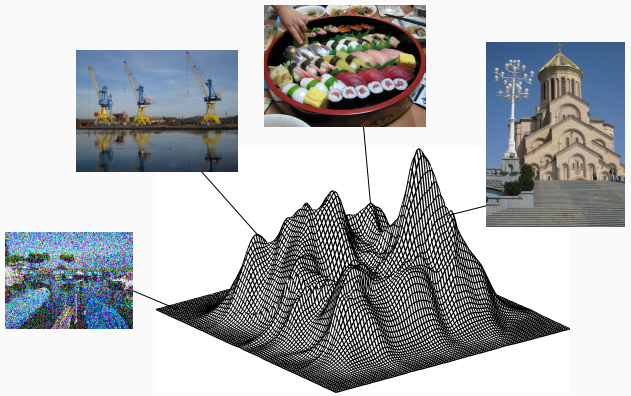
Observed image $y \in \mathbb{R}^n$ (n pixels): random vector with density

$$p(y) = \int p(y, x)\, \mathrm{d}x = \int p(y|x)p(x)\, \mathrm{d}x \qquad \text{(Marginalization)}$$

where $x \in \mathbb{R}^n$ is also a random vector **modeling clean images**.



Likelihood $x \mapsto p(y \,|\, x)$



Prior $x \mapsto p(x)$

- $p(y|x)$ degradation model – law of $y|x$ – likelihood of $x|y$
- $p(x)$ **prior distribution** of x
- $p(y)$ marginal distribution of y

Bayesian approach – How to choose the likelihood?

Modeling the likelihood $p(y|x)$ **(relatively easy)**

- **Based on the knowledge of the acquisition process**
- Linear additive model: $y = \boldsymbol{H}x + w$
- Multiplicative noise: $y = x \times w$
- Poisson noise: $p(y|x) = \dfrac{x^y e^{-x}}{y!}$
- White noise: $\mathbb{E}[w] = 0$ and $\mathrm{Var}[w]$ diagonal

Bayesian approach – How to choose the prior?

Modeling the prior $p(x)$ **(hard)**

- **Based on your prior knowledge of the underlying signal**
- Are clean images piece-wise smooth? simple? observable on the web? ...

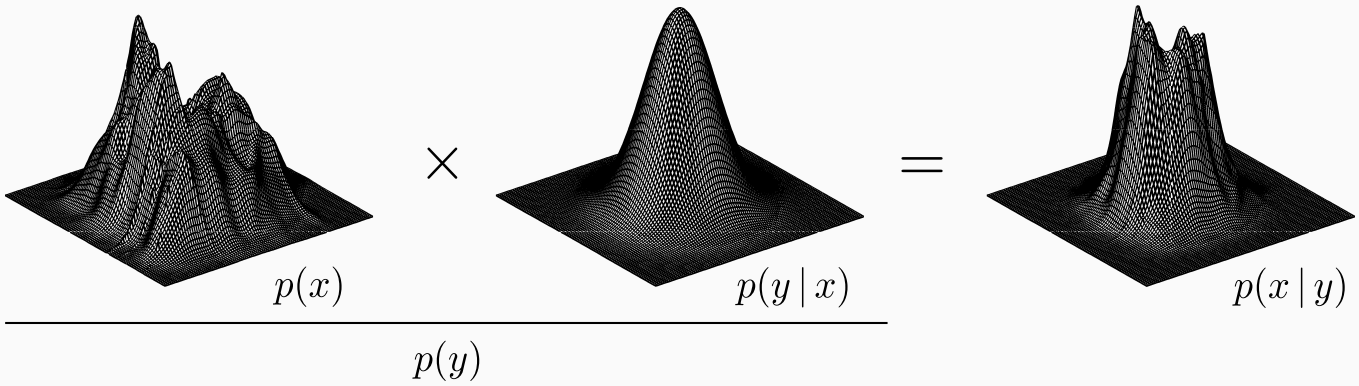
$x =$  $?$

Try to cover as many cases as possible without covering bad images.
The prior should help at separating signal and noise.

Bayesian approach

- Model the likelihood: $p(y \mid x)$,
- Choose a prior: $p(x)$,
- Estimate the posterior distribution based on **Bayes rule**:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \quad \textit{i.e,} \quad \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Marginal}}$$



Posterior mean

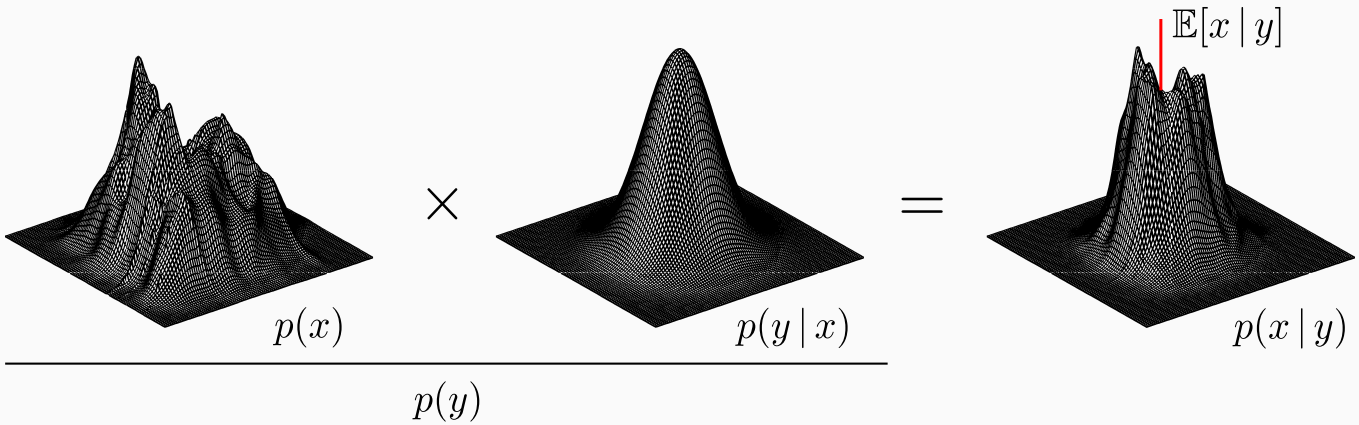
- Compute the mean of the posterior, or **posterior mean**

$$x^\star = \mathbb{E}[x|y] = \int x p(x \,|\, y) \, \mathrm{d}x$$

- Potentially, compute the posterior variance

$$\mathrm{Var}[x|y] = \int (x - \mathbb{E}[x|y])(x - \mathbb{E}[x|y])^* p(x \,|\, y) \, \mathrm{d}x$$

to build regions of confidence, and account for uncertainty.

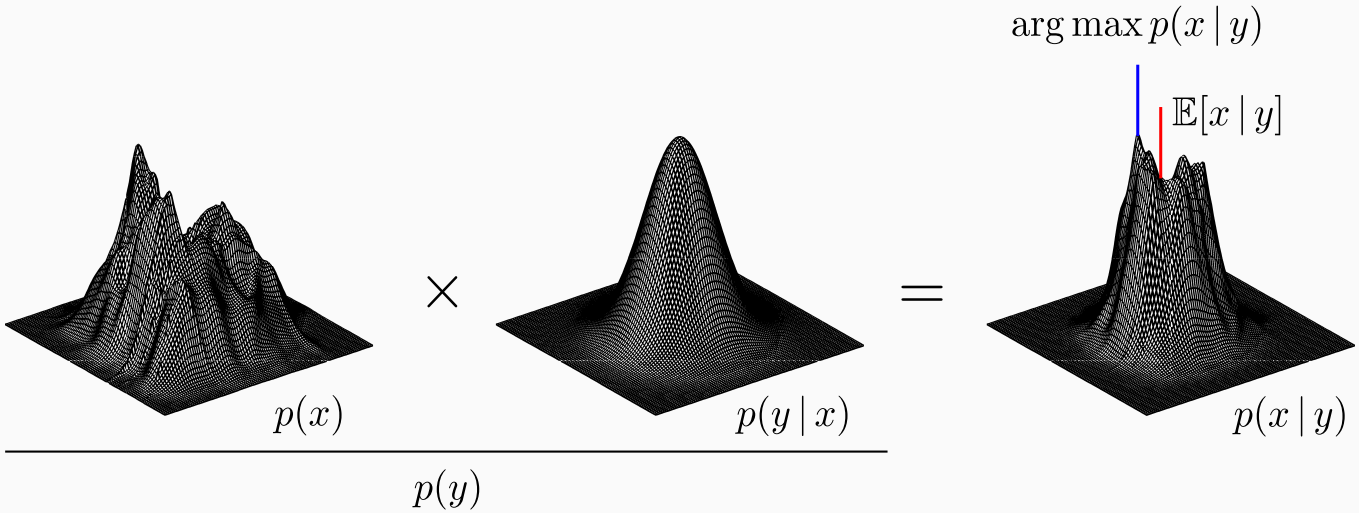


or

Maximum a Posteriori (MAP)

- Take a mode of the posterior, or **Maximum a Posteriori**

$$x^\star \in \operatorname*{argmax}_x p(x \mid y)$$



Posterior mean estimator

Posterior mean

Posterior mean depends only on the likelihood and the prior

$$\begin{aligned}\hat{x}(y) = \mathbb{E}[x \mid y] &= \int x p(x|y)\,\mathrm{d}x \\ &= \frac{\int x p(y|x) p(x)\,\mathrm{d}x}{p(y)} \\ &= \frac{\int x p(y|x) p(x)\,\mathrm{d}x}{\int p(y,x)\,\mathrm{d}x} \\ &= \frac{\int x p(y|x) p(x)\,\mathrm{d}x}{\int p(y|x) p(x)\,\mathrm{d}x}\end{aligned}$$

Posterior mean is always realizable.

Posterior mean

Bayesian MSE

As x is random, the mean square error (MSE) is defined as a double integral

$$\mathrm{MSE}(\hat{x}) = \mathbb{E}[\|x - \hat{x}\|_2^2] = \iint \|x - \hat{x}(y)\|_2^2 \, p(y, x) \, \mathrm{d}y \, \mathrm{d}x$$

Theorem (Optimality of the posterior mean)

The Minimum (Bayesian) MSE estimator (MMSE) is unique and given by the posterior mean

$$\hat{x}^\star = \operatorname*{argmin}_{\hat{x}} \mathrm{MSE}(\hat{x}) = \mathbb{E}[x \mid y] = \int x p(x \mid y) \, \mathrm{d}x$$

The posterior mean $\hat{x}$ is the estimator that is as close as possible to x, on average for likely clean images x and their corrupted versions y.

Posterior mean – Proof

Lemma – Conditional expectation.

$$
\begin{aligned}
\mathbb{E}[f(x)] &= \int f(x)p(x)\,\mathrm{d}x \\
&= \iint f(x)p(x,y)\,\mathrm{d}x\,\mathrm{d}y \\
&= \iint f(x)p(x|y)p(y)\,\mathrm{d}x\,\mathrm{d}y \\
&= \int \left[\int f(x)p(x|y)\,\mathrm{d}x\right] p(y)\,\mathrm{d}y \\
&= \int \mathbb{E}[f(x)\,|\,y]\,p(y)\,\mathrm{d}y \\
&= \mathbb{E}\left\{\mathbb{E}[f(x)|y]\right\}
\end{aligned}
$$

□

Posterior mean – Proof

Proof.

The previous Lemma leads to

$$\begin{aligned}\mathbb{E}[\|x - \hat{x}(y)\|_2^2] &= \mathbb{E}\left\{\mathbb{E}[\|x - \hat{x}(y)\|_2^2 \mid y]\right\} \\ &= \int \mathbb{E}[\|x - \hat{x}(y)\|_2^2 \mid y]\, p(y)\, \mathrm{d}y\end{aligned}$$

This quantity is minimal if, for all y, we have

$$\hat{x}(y) \in \operatorname*{argmin}_z \mathbb{E}[\|x - z\|_2^2 \mid y]$$

Given y, we have to minimize with respect to z the quantity

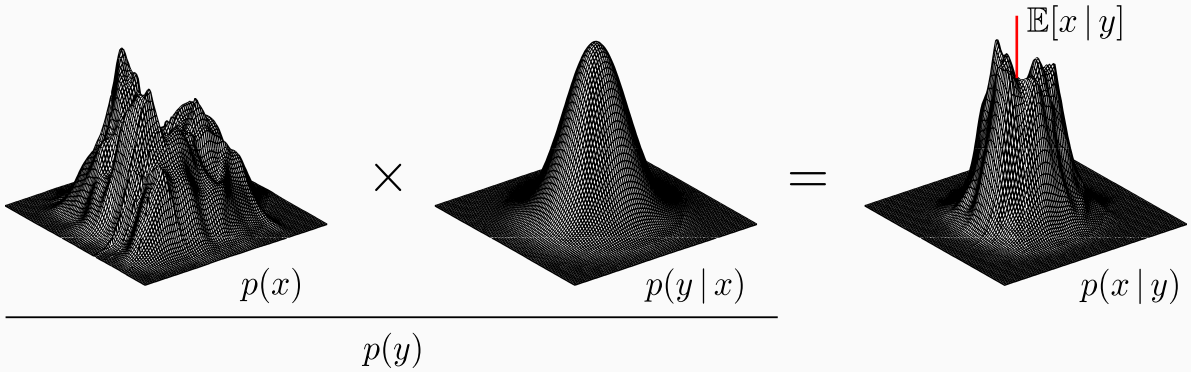
$$\mathbb{E}[\|x - z\|_2^2 \mid y] = \mathbb{E}[\|x\|^2 \mid y] + \|z\|^2 - 2\langle z, \mathbb{E}[x \mid y]\rangle$$

The first order optimality condition gives

$$2z^\star - 2\mathbb{E}[x \mid y] = 0 \quad \Leftrightarrow \quad z^\star = \mathbb{E}[x \mid y]$$

□

Posterior mean



- Optimal in the MMSE sense ☺
- In general, difficult integration problem ☹
- Explicit solutions in few cases
  (see, conjugate priors)

$$\mathbb{E}[x \mid y] = \frac{\int x p(y|x) p(x) \, \mathrm{d}x}{\int p(y|x) p(x) \, \mathrm{d}x}$$

If no explicit solutions, several workarounds:

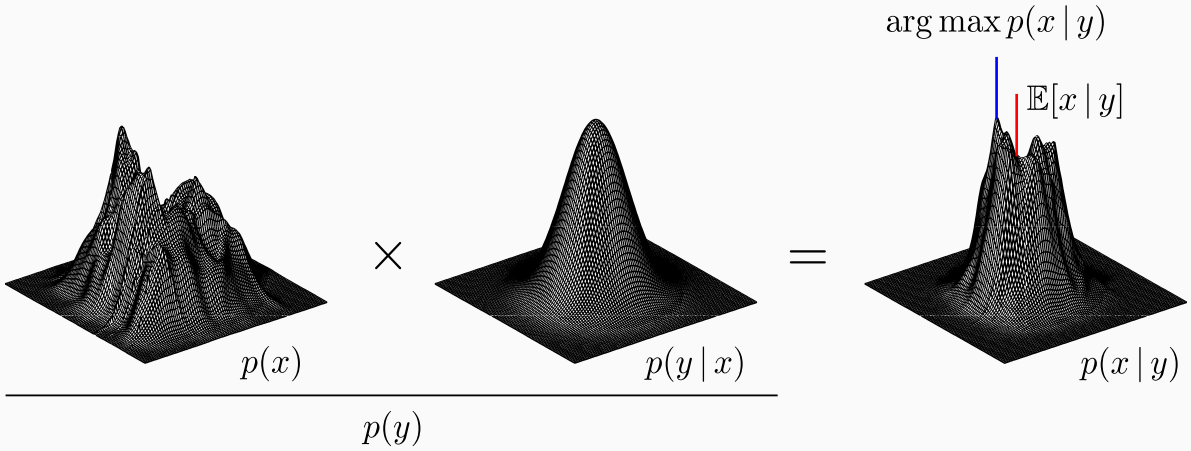
1. LMMSE estimator: Restrict to linear estimators.
2. Wiener estimator: Restrict to LTI estimators.
3. Monte-Carlo estimator: Estimate $\mathbb{E}[x \mid y]$ from a data-set.
4. MCMC/Metropolis Hastings: Otherwise.

Maximum A Posteriori

Maximum A Posteriori (MAP) estimator

Forget about the optimality of the MMSE, and

instead of taking the posterior mean, take the posterior mode.



Maximum A Posteriori (MAP) estimator

$$\hat{x}(y) \in \operatorname*{argmax}_x p(x \mid y) = \operatorname*{argmax}_x \frac{p(y|x)p(x)}{p(y)} = \operatorname*{argmax}_x p(y|x)p(x)$$

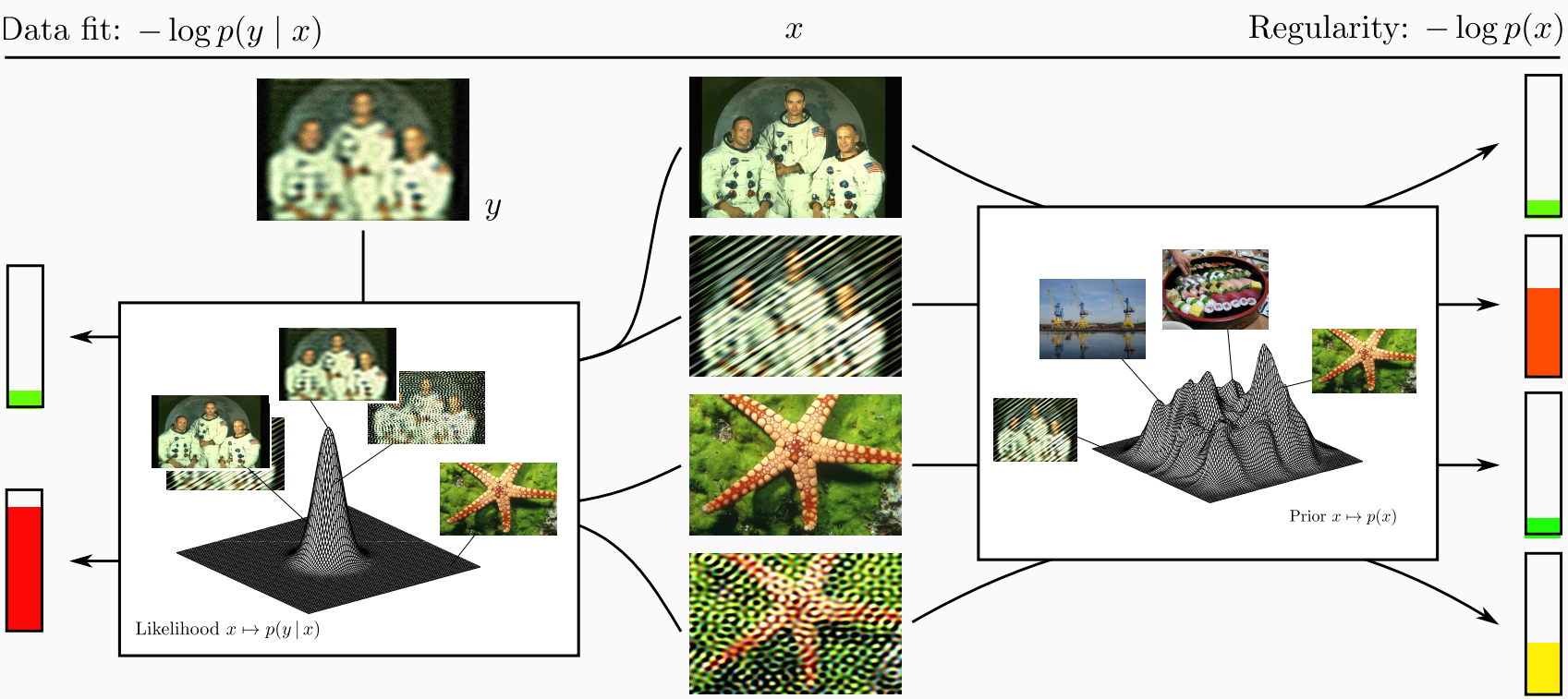
As for the posterior mean,

the MAP depends only on the likelihood and the prior.

Maximum A Posteriori (MAP) estimator

- The MAP is connected with **variational methods**:

$$\hat{x}(y) \in \operatorname*{argmax}_x \underbrace{p(y|x)}_{\text{Likelihood}} \underbrace{p(x)}_{\text{prior}} = \operatorname*{argmin}_x \underbrace{-\log p(y|x)}_{\text{Data fit}} \underbrace{-\log p(x)}_{\text{Regularisation}}$$



Maximum A Posteriori (MAP) estimator

MMSE

$$\frac{\int x p(y|x)p(x)\,\mathrm{d}x}{\int p(y|x)p(x)\,\mathrm{d}x}$$

Integration problem

vs

MAP

$$\underset{x}{\operatorname{argmax}}\; p(y|x)p(x)$$

Optimization problem

- Integration can be **intractable** and/or leads to **long computation time**.
- Optimization is often **simpler and faster** (does not mean straightforward).
- We will see several examples later.

But first, **when do both estimators coincide?**

Linear Minimum Mean Square Error

LMMSE estimator – Optimal linear filtering

Linear Minimum Mean Square Error (LMMSE)

- Let $y \in \mathbb{R}^n$ and $x \in \mathbb{R}^p$ be two random vectors such that

$$\mathbb{E}[y \,|\, x] = \boldsymbol{H}x \quad \text{and} \quad \mathrm{Var}[y \,|\, x] = \boldsymbol{\Sigma}$$
$$\mathbb{E}[x] = \mu \quad \text{and} \quad \mathrm{Var}[x] = \boldsymbol{L}$$

- Consider the class of affine estimators

$$\mathcal{C} = \left\{ \hat{x} \; ; \; \exists \boldsymbol{A} \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p, \quad \hat{x}(y) = \boldsymbol{A}y + b \right\}$$

- The LMMSE estimator is

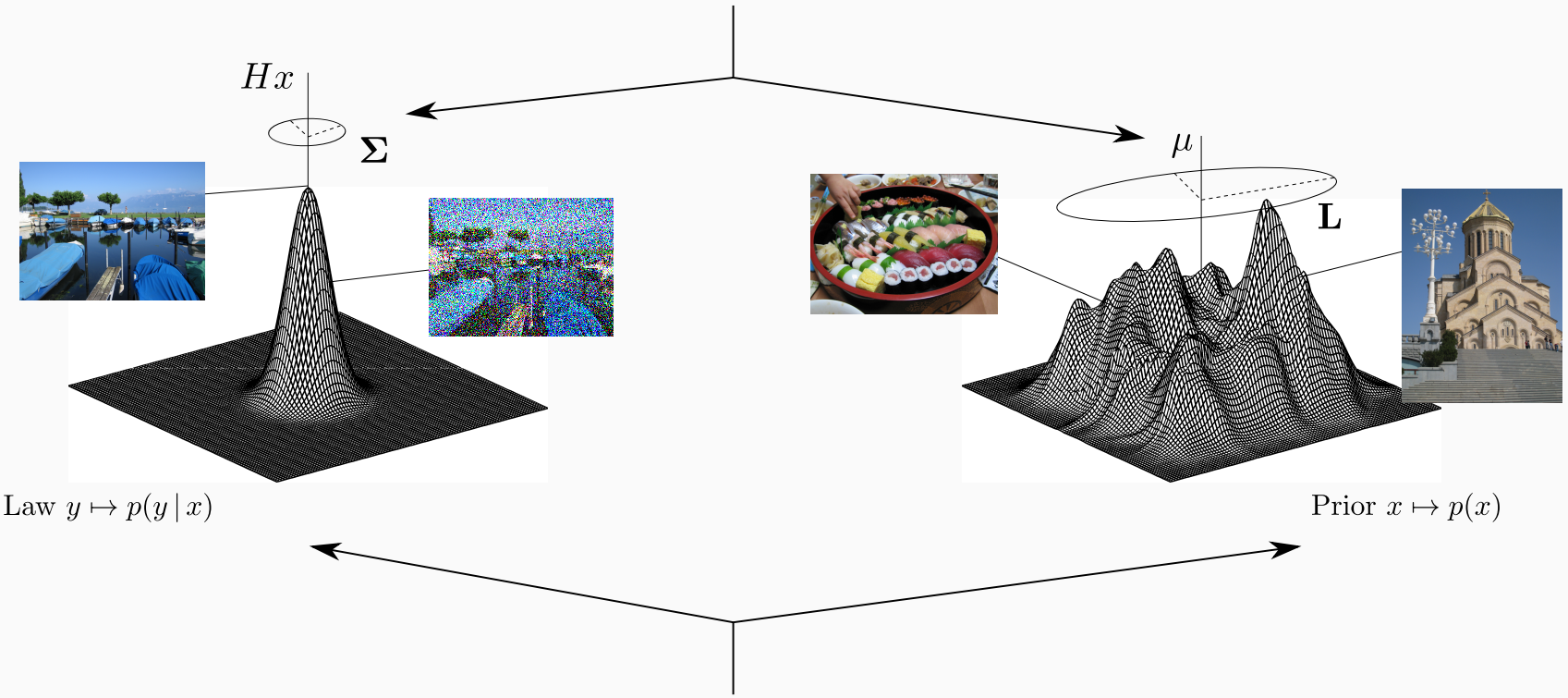
$$\begin{aligned} \hat{x}^\star &= \underset{\hat{x} \in \mathcal{C}}{\operatorname{argmin}} \, \mathbb{E}[\|x - \hat{x}\|_2^2] \\ &= \mu + \boldsymbol{L}\boldsymbol{H}^*(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^* + \boldsymbol{\Sigma})^{-1}(y - \boldsymbol{H}\mu) \end{aligned}$$

- Unlike BLUE, $\boldsymbol{H}$ can be under-determined here because $\boldsymbol{L}$ and $\boldsymbol{\Sigma}$ are always sdp, hence invertible.

LMMSE estimator – Optimal linear filtering

As for the BLUE, the LMMSE depends only on the means and variances

If these are known, the best **linear** estimator is realizable, it's the LMMSE



Even though, these are unknown

LMMSE estimator – Optimal linear filtering

MAP with Gaussian models = LMMSE = Penalized Least Square

- Let $y \in \mathbb{R}^n$ and $x \in \mathbb{R}^p$ be independent and such that

$$y|x \sim \mathcal{N}(\boldsymbol{H}x, \boldsymbol{\Sigma})$$
$$x \sim \mathcal{N}(\mu, \boldsymbol{L})$$

- Then, the MAP is a **penalized least square estimator** (PLSE)

$$\begin{aligned}
\hat{x}^\star &= \operatorname*{argmin}_{\hat{x}} \; -\log p(y|\hat{x}) - \log p(\hat{x}) \\
&= \operatorname*{argmin}_{\hat{x}} \; \underbrace{\|\boldsymbol{\Sigma}^{-1/2}(y - \boldsymbol{H}\hat{x})\|_2^2}_{\text{Data fit}} + \underbrace{\|\boldsymbol{L}^{-1/2}(\hat{x} - \mu)\|_2^2}_{\text{Penalization}} \\
&= \mu + (\boldsymbol{H}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{H} + \boldsymbol{L}^{-1})^{-1}\boldsymbol{H}^*\boldsymbol{\Sigma}^{-1}(y - \boldsymbol{H}\mu) && \text{(Null gradient)} \\
&= \mu + \boldsymbol{L}\boldsymbol{H}^*(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^* + \boldsymbol{\Sigma})^{-1}(y - \boldsymbol{H}\mu) && \text{(Woodbury)}
\end{aligned}$$

- It is also the MMSE and the LMMSE.

LMMSE under white noise

- For simplicity add the restriction that the noise is white

$$\mathrm{Var}[y \,|\, x] = \boldsymbol{\Sigma} = \sigma^2 \mathrm{Id}$$

- In this case the LMMSE has a simplified expression

$$\begin{aligned}\hat{x}^\star &= \operatorname*{argmin}_{\hat{x}} \frac{1}{\sigma^2}\|y - \boldsymbol{H}\hat{x}\|_2^2 + \|\boldsymbol{L}^{-1/2}(\hat{x} - \mu)\|_2^2 \\ &= \mu + (\boldsymbol{H}^*\boldsymbol{H} + \sigma^2\boldsymbol{L})^{-1}\boldsymbol{H}^*(y - \boldsymbol{H}\mu)\end{aligned}$$

- $\boldsymbol{H}^*\boldsymbol{H} + \sigma^2\boldsymbol{L}^{-1}$ always invertible and symmetric positive definite

(you can always use conjugate gradient ☺)

LMMSE estimator – Optimal linear filtering

$$\hat{x}^\star = \mu + (\boldsymbol{H}^*\boldsymbol{H} + \sigma^2\boldsymbol{L}^{-1})^{-1}\boldsymbol{H}^*(y - \boldsymbol{H}\mu)$$

LMMSE vs BLUE

- Define $\mathrm{SNR}^2 = \dfrac{\operatorname{tr}\boldsymbol{L}}{n\sigma^2} = \dfrac{\text{Uncertainty on the signal}}{\text{Variation of the noise}}$
- If $\boldsymbol{H}$ over-determined, the LMMSE tends to the BLUE (least square)

$$\begin{aligned}
\lim_{\mathrm{SNR}\to\infty} \hat{x}^\star &= \mu + (\boldsymbol{H}^*\boldsymbol{H})^{-1}\boldsymbol{H}^*(y - \boldsymbol{H}\mu) \\
&= \mu + (\boldsymbol{H}^*\boldsymbol{H})^{-1}\boldsymbol{H}^*y - \underbrace{(\boldsymbol{H}^*\boldsymbol{H})^{-1}\boldsymbol{H}^*\boldsymbol{H}\mu}_{\mu} \\
&= (\boldsymbol{H}^*\boldsymbol{H})^{-1}\boldsymbol{H}^*y \\
&= \boldsymbol{H}^+y
\end{aligned}$$

As you try to be uniformly optimal for all x (maximal uncertainty) you will fail at restoring y, all the more as you have noise.

LMMSE estimator – Optimal linear filtering

LMMSE for denoising

- If moreover $\boldsymbol{H} = \mathrm{Id}$ (denoising problem), this simplifies as

$$\begin{aligned}\hat{x}^\star &= \operatorname*{argmin}_{\hat{x}} \frac{1}{\sigma^2}\|y - \hat{x}\|_2^2 + \|\boldsymbol{L}^{-1/2}(\hat{x} - \mu)\|_2^2 \\ &= \mu + \boldsymbol{L}(\boldsymbol{L} + \sigma^2 \mathrm{Id})^{-1}(y - \mu)\end{aligned}$$

Let us pick some μ and L and try this formula for denoising.

Example (Lets pick a naive model)

- Consider $y = x + w \in \mathbb{R}^n$ with x and w independent such that

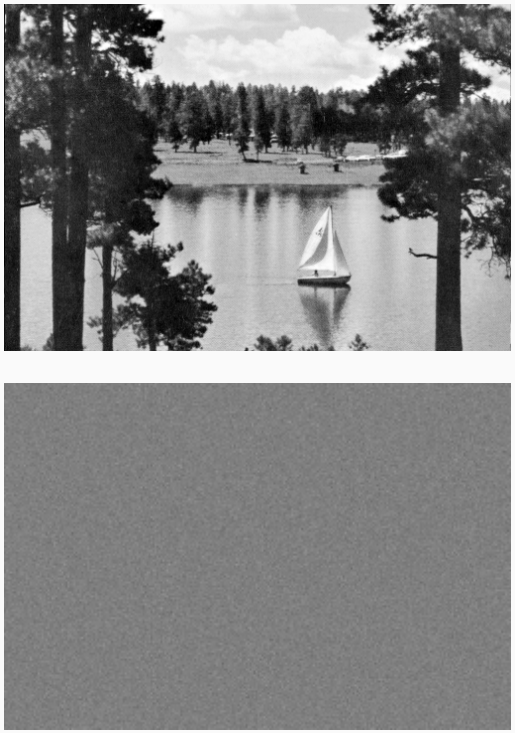
$$\begin{aligned} \mathbb{E}[y \,|\, x] = x \quad &\text{and} \quad \mathrm{Var}[y \,|\, x] = \sigma^2 \mathrm{Id}_n, \\ \mathbb{E}[x] = 0 \quad &\text{and} \quad \mathrm{Var}[x] = \boldsymbol{L} = \lambda^2 \mathrm{Id}_n. \end{aligned}$$

- Then the LMMSE filter reads as

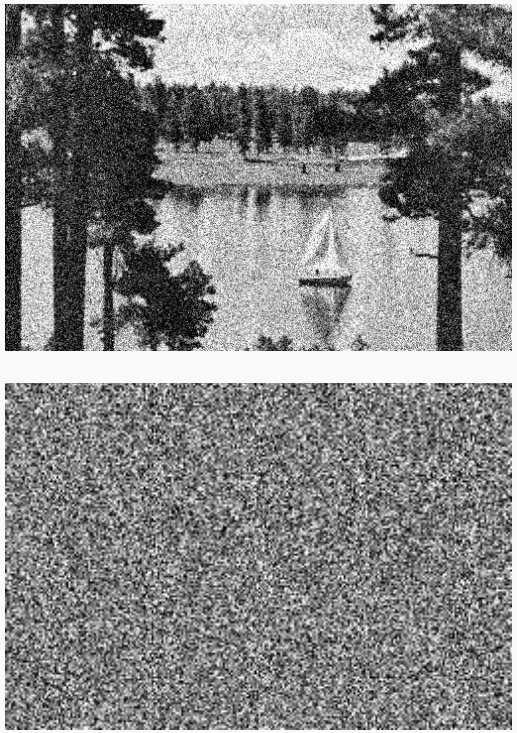
$$\begin{aligned} \hat{x}^\star &= \mu + \boldsymbol{L}(\boldsymbol{L} + \sigma^2 \mathrm{Id})^{-1}(y - \mu) \\ &= 0 + \lambda^2(\lambda^2 \mathrm{Id} + \sigma^2 \mathrm{Id})^{-1}(y - 0) \\ &= \frac{\lambda^2}{\lambda^2 + \sigma^2} y \\ &= \frac{\mathrm{SNR}^2}{\mathrm{SNR}^2 + 1} y \quad \text{with} \quad \mathrm{SNR}^2 = \frac{\lambda^2}{\sigma^2} \end{aligned}$$

- It's similar to the optimal amplified estimator except this one is realizable.

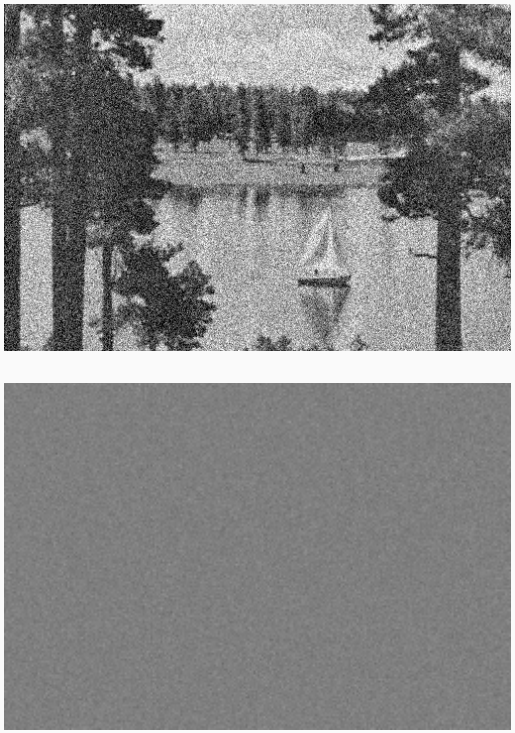
LMMSE estimator – Naive model – Limitations



(a) x (unknown)



(b) y (observation)

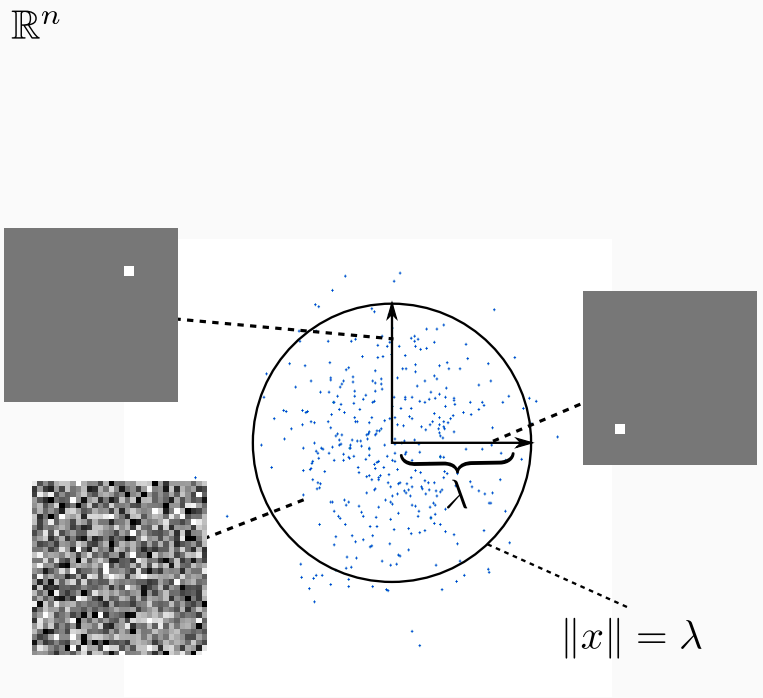


(c) $\hat{x}^\star$ (estimate)

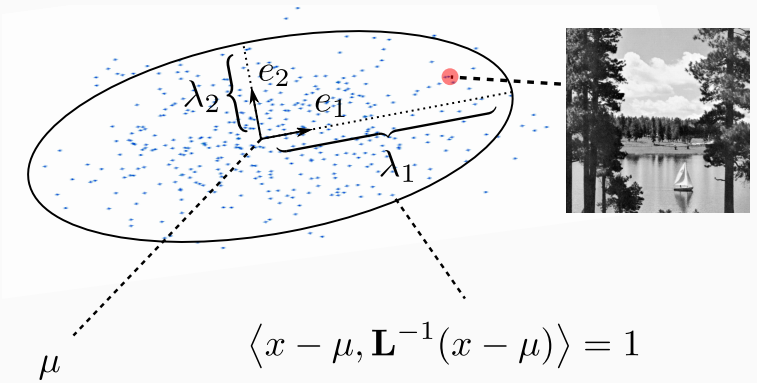
Limitations of this naive model

- Amplifying the noisy input will never allow us removing noise,
- Would work only if all clean images were like pure white noise.

LMMSE estimator – Naive model – Limitations



(a) We assumed x to be likely somewhere here.



(b) but it was somewhere there.

Limitations of this naive model

- Images contain structures that must be captured by μ and $\boldsymbol{L}$,
- Goal: define/find the ellipsoid localizing x with high probability,
- How: make use of the eigendecomposition: $\boldsymbol{L} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^*$.

LMMSE estimator – Filtering in the eigenspace

$$\hat{x}^\star = \mu + \boldsymbol{L}(\boldsymbol{L} + \sigma^2 \mathrm{Id})^{-1}(y - \mu)$$

- Consider the eigendecomposition: $\boldsymbol{L} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^*$, then

$$\boldsymbol{L}(\boldsymbol{L} + \sigma^2 \mathrm{Id}_n)^{-1} = \boldsymbol{E} \begin{pmatrix} \frac{\lambda_1^2}{\lambda_1^2+\sigma^2} & & & 0 \\ & \frac{\lambda_2^2}{\lambda_2^2+\sigma^2} & & \\ & & \ddots & \\ 0 & & & \frac{\lambda_n^2}{\lambda_n^2+\sigma^2} \end{pmatrix} \boldsymbol{E}^*$$

$\Rightarrow$ The LMMSE filter can be re-written as

$$\hat{x}^\star = \underbrace{\mu + \boldsymbol{E}\hat{z}}_{\text{Come back}} \quad \text{where} \quad \underbrace{\hat{z}_i = \frac{\lambda_i^2}{\lambda_i^2 + \sigma^2} z_i}_{\text{shrinkage}} \quad \text{and} \quad z = \underbrace{\boldsymbol{E}^*(y - \mu)}_{\text{Change of origin and basis}}$$

- Shrinkage adapts to the SNR in each eigendirection: $\mathrm{SNR}_i^2 = \lambda_i^2/\sigma^2$.

What representation (μ and $\boldsymbol{E}$)? What shrinkage ($\boldsymbol{\Lambda}$)?

Wiener filtering

Wiener filtering – Whitening via the Fourier transform

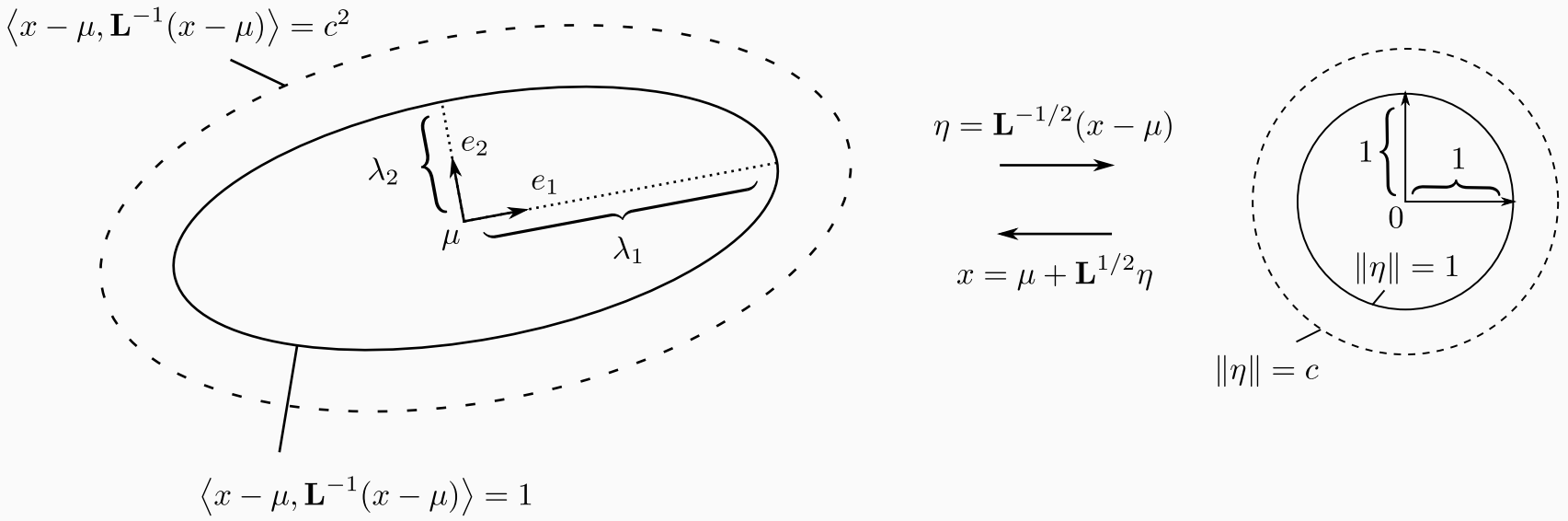
Whitening

Let $\boldsymbol{L}^{-1/2} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{E}^*$. Let x be a random variable. Then

$$\mathbb{E}[x] = \mu \quad \text{and} \quad \mathrm{Var}[x] = \boldsymbol{L}$$

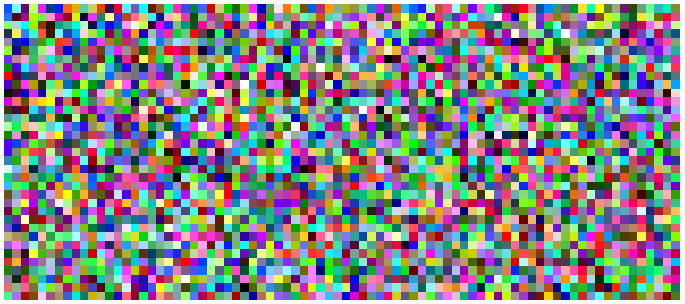
if and only if

$$\eta = \boldsymbol{L}^{-1/2}(x-\mu) \quad \text{with} \quad \mathbb{E}[\eta] = 0 \quad \text{and} \quad \mathrm{Var}[\eta] = \mathrm{Id}_n$$



Wiener filtering – Whitening via the Fourier transform

$$\boldsymbol{L}^{-1/2}(x-\mu) \longrightarrow$$



Modeling the two first moments of x

$\equiv$

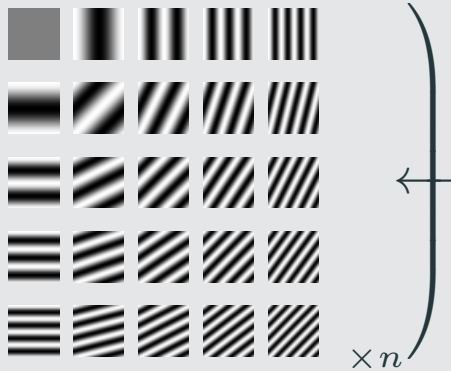
Find the affine transform that makes it look like white noise

What kind of transform can make typical images white?

Whitening using DFT

- Assume Fourier coefficients to be decorrelated,
- Choose $\mu = 0$ and $\boldsymbol{L} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^{-1}$ with

$$\boldsymbol{E} = \frac{1}{\sqrt{n}} \underbrace{\left(\begin{array}{c} \\ \\ \\ \\ \\ \end{array} \right)}_{\text{DFT: } \boldsymbol{F}} \times n$$

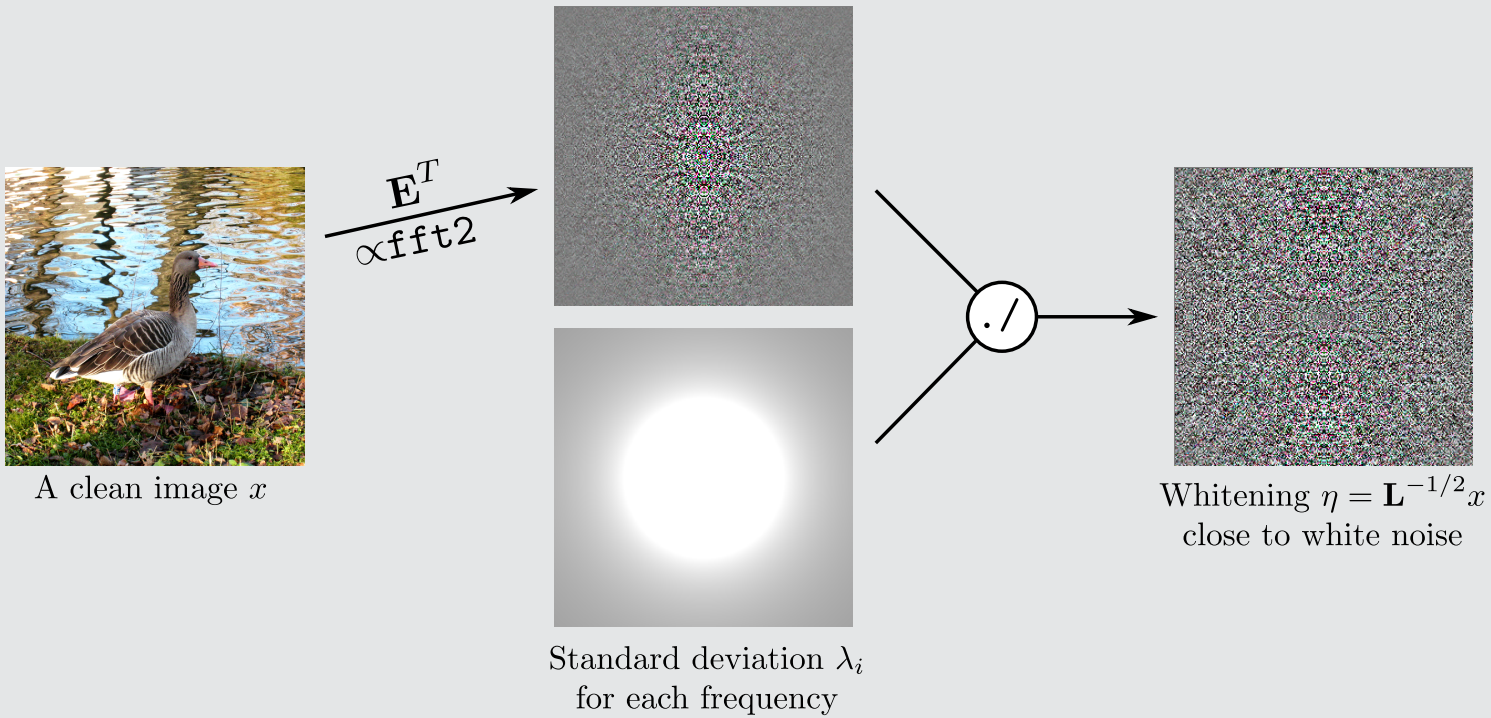


← Columns form the Fourier basis

$$\boldsymbol{E}^{-1} = \sqrt{n}\boldsymbol{F}^{-1}$$

$\boldsymbol{\Lambda}$ = diagonal, real and positive ($\boldsymbol{L}$ is sdp)

Whitening using DFT



- $\boldsymbol{E}$ and μ fixed: $\quad \boldsymbol{L} = \mathrm{Cov}[x] \quad \Leftrightarrow \quad \boldsymbol{\Lambda} = \mathrm{Var}[\boldsymbol{E}^*(x - \mu)]$
- In our case: $\quad \lambda_i^2 = n^{-1} \times \mathbb{E}[|(\boldsymbol{F}x)_i|^2] =$ mean power spectral density
  $= $ variance for each frequency

Wiener filtering – Whitening via the Fourier transform

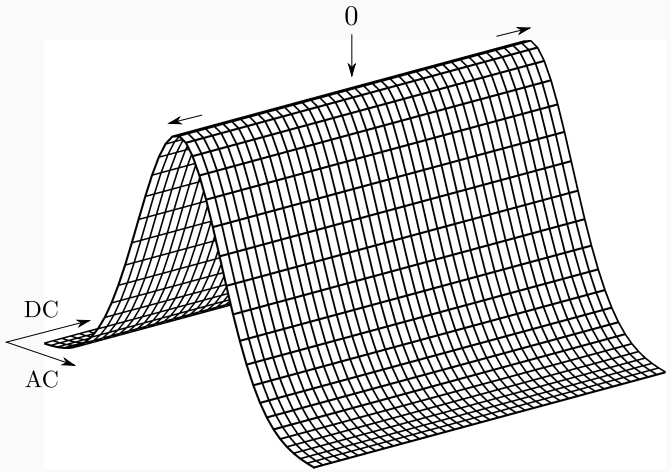
Mean power spectral density

$$S_i = \mathbb{E}[|(\boldsymbol{F}x)_i|^2]$$

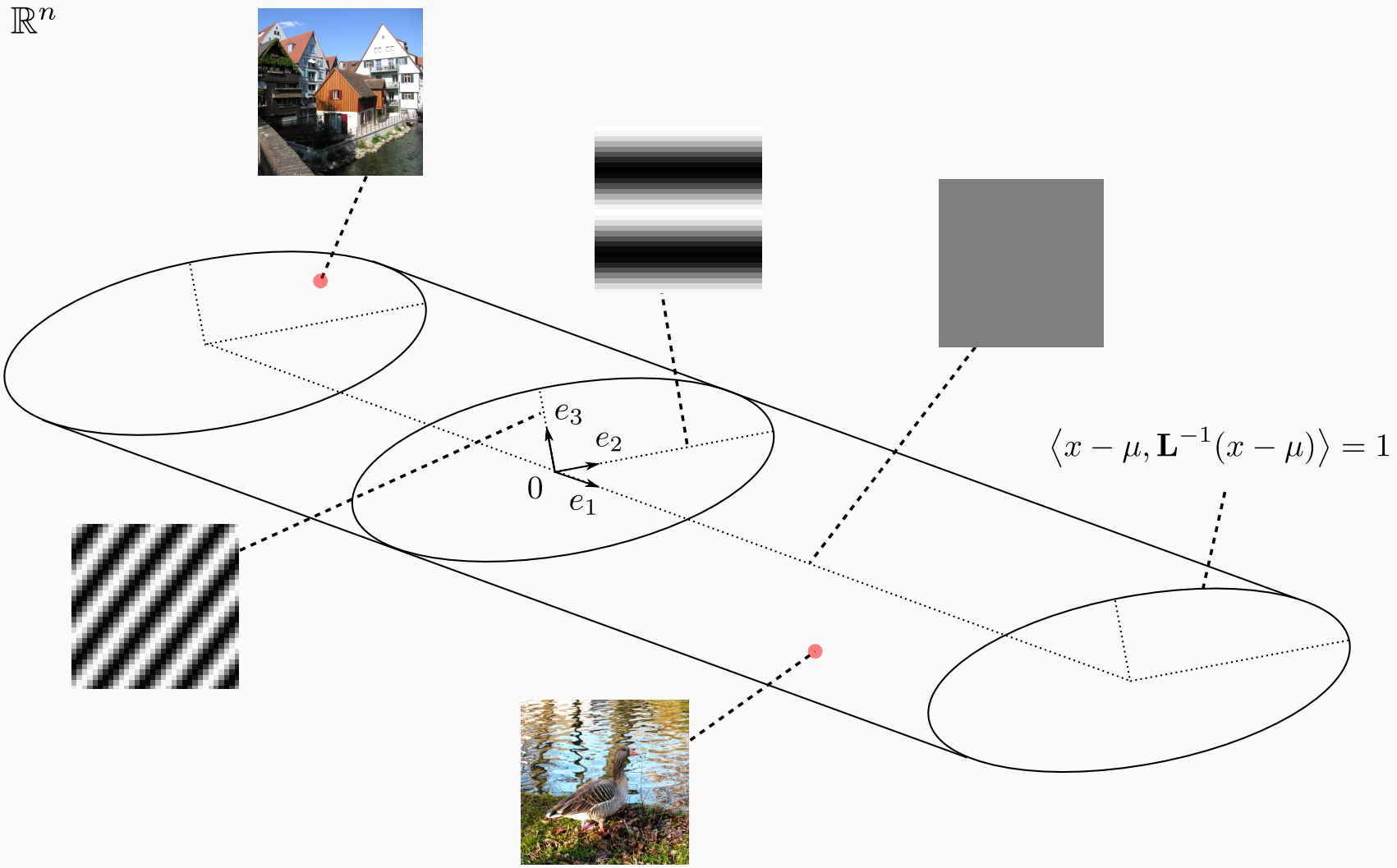
- Estimate it from a collection of clean images x_1, x_2, ...
- Use a model to reduce its degree of freedom [Van der Schaafa *et al.*, 1996]

$$S_i = ne^{\beta}\sqrt{\left(\frac{u_i}{n_1}\right)^2 + \left(\frac{v_i}{n_2}\right)^2}^{\alpha}.$$

- Estimate α and β by least square in log-log (see assignment).
- Arbitrary zero frequency (DC component): its variance goes to infinity.



Wiener filtering – Whitening via the Fourier transform



Prior localization for clean images

Wiener filter for denoising

- In denoising, this LMMSE is called **Wiener filter** and reads

$$\hat{x}^\star = \underbrace{\boldsymbol{F}^{-1}\hat{z}}_{\text{iDFT}} \quad \text{where} \quad \underbrace{\hat{z}_i = \frac{\lambda_i^2}{\lambda_i^2 + \sigma^2} z_i}_{\text{shrink each frequency}} \quad \text{and} \quad z = \underbrace{\boldsymbol{F}y}_{\text{DFT}}$$

- Using $\lambda_0 \to \infty$: DC is unchanged.

(a) $y = x + w$ (b) z (c) $\frac{\lambda_i^2}{\lambda_i^2+\sigma^2}$ (d) $\hat{z}$ (e) $\hat{x}$

Wiener filtering – Denoising

Results of Wiener filtering in denoising

(a) y (b) z (c) $\frac{\lambda_i^2}{\lambda_i^2+\sigma^2}$ (d) $\hat{z}$ (e) $\hat{x}$

```
y    = x + sigma * np.random.randn(x.shape)

z    = nf.fft2(y, axes=(0, 1))
zhat = lbd**2 / (lbd**2+ sigma**2) * z
xhat = np.real(nf.ifft2(zhat, axes=(0, 1)))
```

$$z = \sqrt{n}^{-1} \boldsymbol{F} y$$

$$\hat{z}_i = \frac{\lambda_i^2}{\lambda_i^2 + \sigma^2} z_i$$

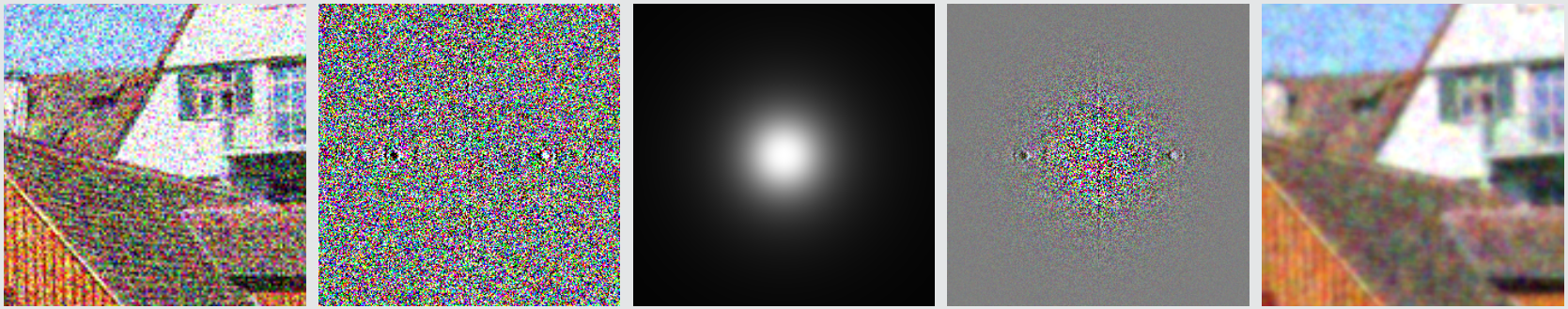
$$\hat{x} = \sqrt{n} \boldsymbol{F}^{-1} \hat{z}$$

The technique auto-adapts to the noise level ☺

Results are blurry, too smooth ☹

Wiener filtering – Denoising

Results of Wiener filtering in denoising



(a) y (b) z (c) $\frac{\lambda_i^2}{\lambda_i^2+\sigma^2}$ (d) $\hat{z}$ (e) $\hat{x}$

```
y    = x + sigma * np.random.randn(x.shape)

z    = nf.fft2(y, axes=(0, 1))
zhat = lbd**2 / (lbd**2+ sigma**2) * z
xhat = np.real(nf.ifft2(zhat, axes=(0, 1)))
```

$$z = \sqrt{n}^{-1} \boldsymbol{F} y$$

$$\hat{z}_i = \frac{\lambda_i^2}{\lambda_i^2 + \sigma^2} z_i$$

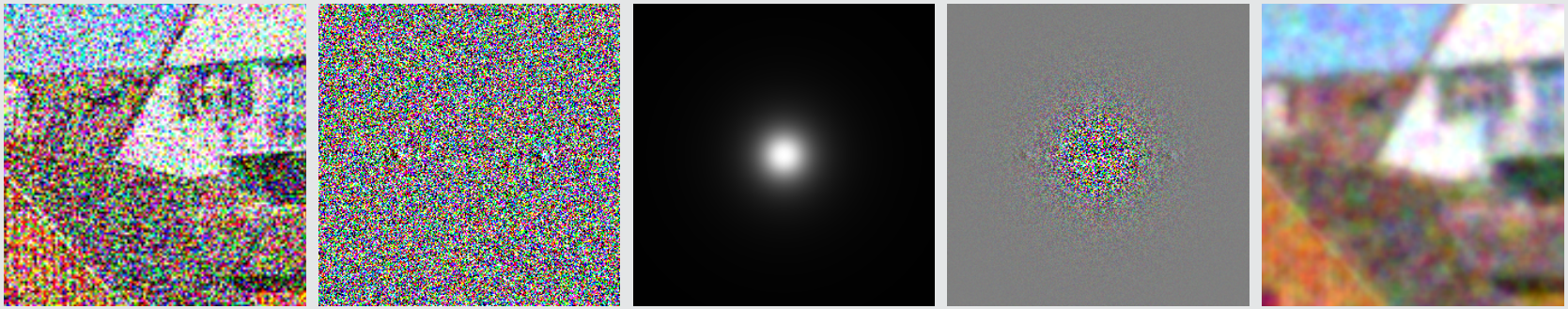
$$\hat{x} = \sqrt{n} \boldsymbol{F}^{-1} \hat{z}$$

The technique auto-adapts to the noise level ☺

Results are blurry, too smooth ☹

Wiener filtering – Denoising

Results of Wiener filtering in denoising



(a) y (b) z (c) $\frac{\lambda_i^2}{\lambda_i^2+\sigma^2}$ (d) $\hat{z}$ (e) $\hat{x}$

```
y    = x + sigma * np.random.randn(x.shape)

z    = nf.fft2(y, axes=(0, 1))
zhat = lbd**2 / (lbd**2+ sigma**2) * z
xhat = np.real(nf.ifft2(zhat, axes=(0, 1)))
```

$$z = \sqrt{n}^{-1} \boldsymbol{F} y$$

$$\hat{z}_i = \frac{\lambda_i^2}{\lambda_i^2 + \sigma^2} z_i$$

$$\hat{x} = \sqrt{n} \boldsymbol{F}^{-1} \hat{z}$$

The technique auto-adapts to the noise level ☺

Results are blurry, too smooth ☹

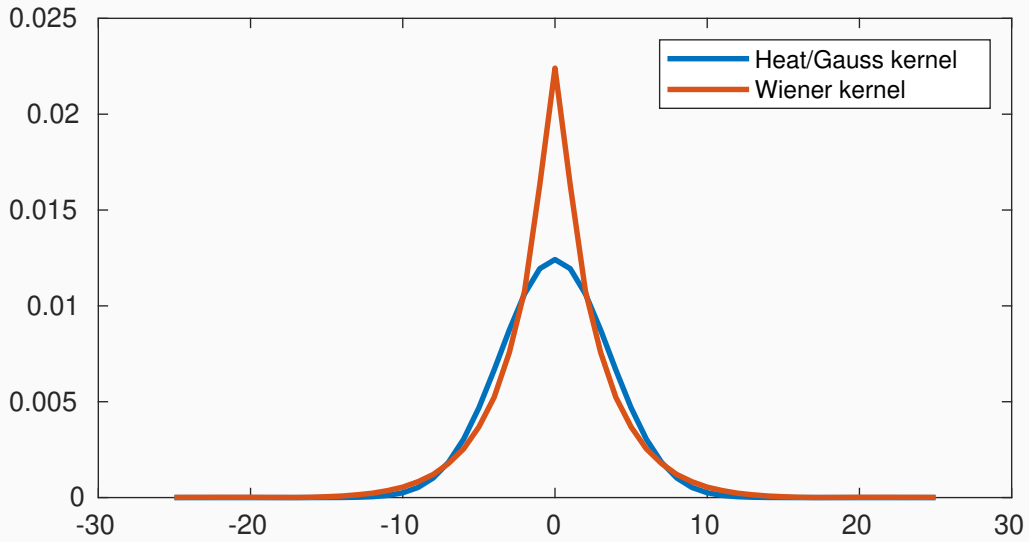
Wiener filtering – Limits and connections to LTI filters

Wiener filtering for denoising

$$\underbrace{\hat{z}_i = \frac{\lambda_i^2}{\lambda_i^2 + \sigma^2} z_i}_{\text{elementwise product}} \quad \Leftrightarrow \quad \underbrace{\hat{x} = \nu * y}_{\text{convolution} \equiv \text{moving average}}$$

- Wiener filter: **optimal LTI filter** in the Bayesian MMSE sense,

$\Rightarrow$ It is a low-pass filter, *i.e.*, a weighted average.



Wiener deconvolution

- For deconvolution, the LMMSE filter reads as

$$\hat{x}^\star = \mu + (\boldsymbol{H}^*\boldsymbol{H} + \sigma^2\boldsymbol{L}^{-1})^{-1}\boldsymbol{H}^*(y - \mu)$$

  with $\boldsymbol{H}$ a circulant matrix: $\boldsymbol{H} = \boldsymbol{F}^{-1}\boldsymbol{\Omega}\boldsymbol{F}$ with $\boldsymbol{\Omega} = \mathrm{diag}(\omega_1, \ldots, \omega_n)$.

- We get that

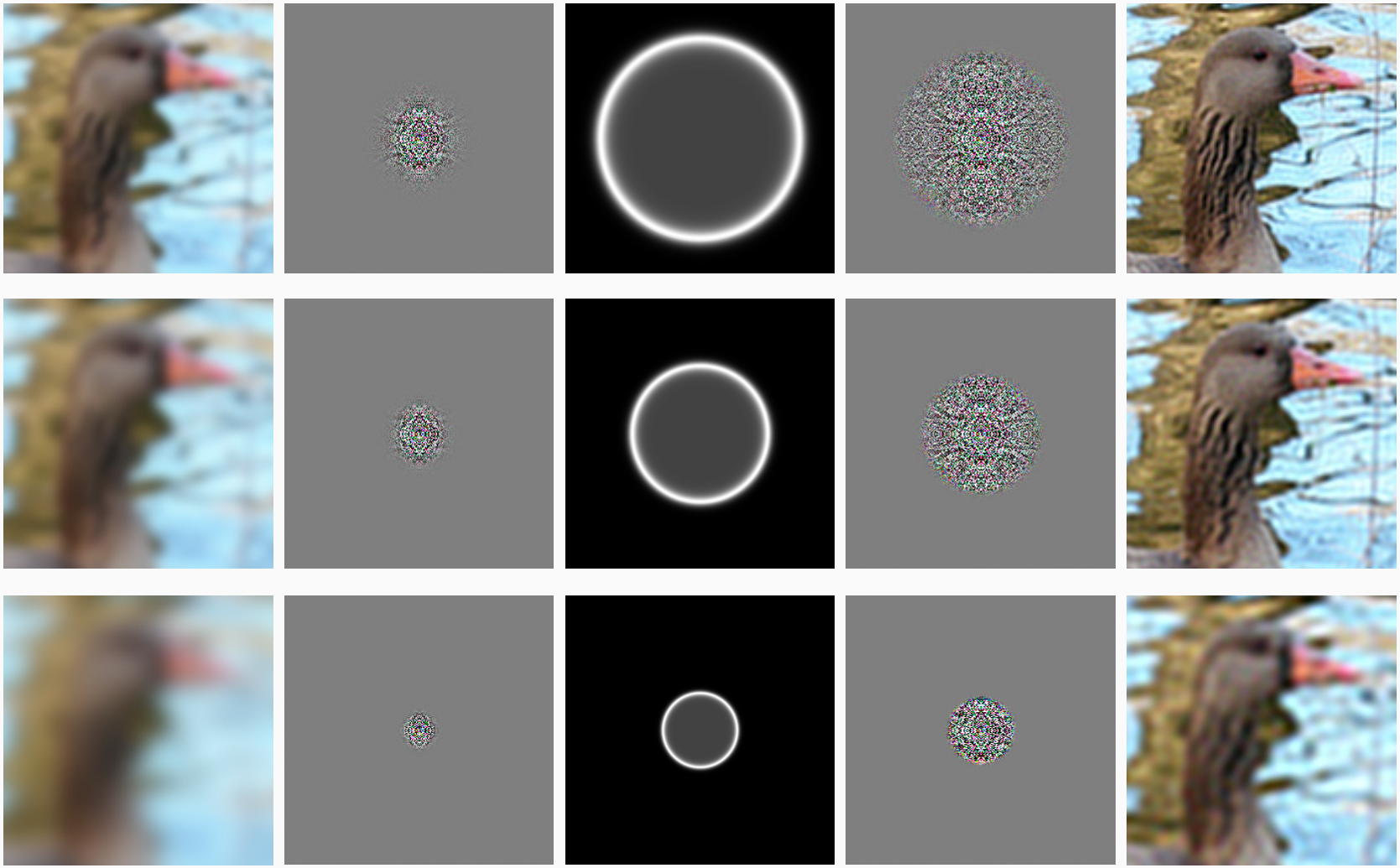
$$\begin{aligned}\hat{x}^\star &= (\boldsymbol{F}^{-1}\boldsymbol{\Omega}^*\boldsymbol{F}\boldsymbol{F}^{-1}\boldsymbol{\Omega}\boldsymbol{F} + \sigma^2\boldsymbol{F}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^{-1}\boldsymbol{\Omega}^*\boldsymbol{F}y \\ &= \boldsymbol{F}^{-1}(\boldsymbol{\Omega}^*\boldsymbol{\Omega} + \sigma^2\boldsymbol{\Lambda}^{-1})^{-1}\boldsymbol{\Omega}^*\boldsymbol{F}y\end{aligned}$$

- Or equivalently

$$\hat{x}^\star = \boldsymbol{F}^{-1}\hat{z} \quad \text{where} \quad \hat{z}_i = \frac{\omega_i^*}{|\omega_i|^2 + \sigma^2/\lambda_i^2} z_i \quad \text{and} \quad z = \boldsymbol{F}y$$

- Filter adapts to the SNR for each frequency.

Wiener filtering – Application to deconvolution

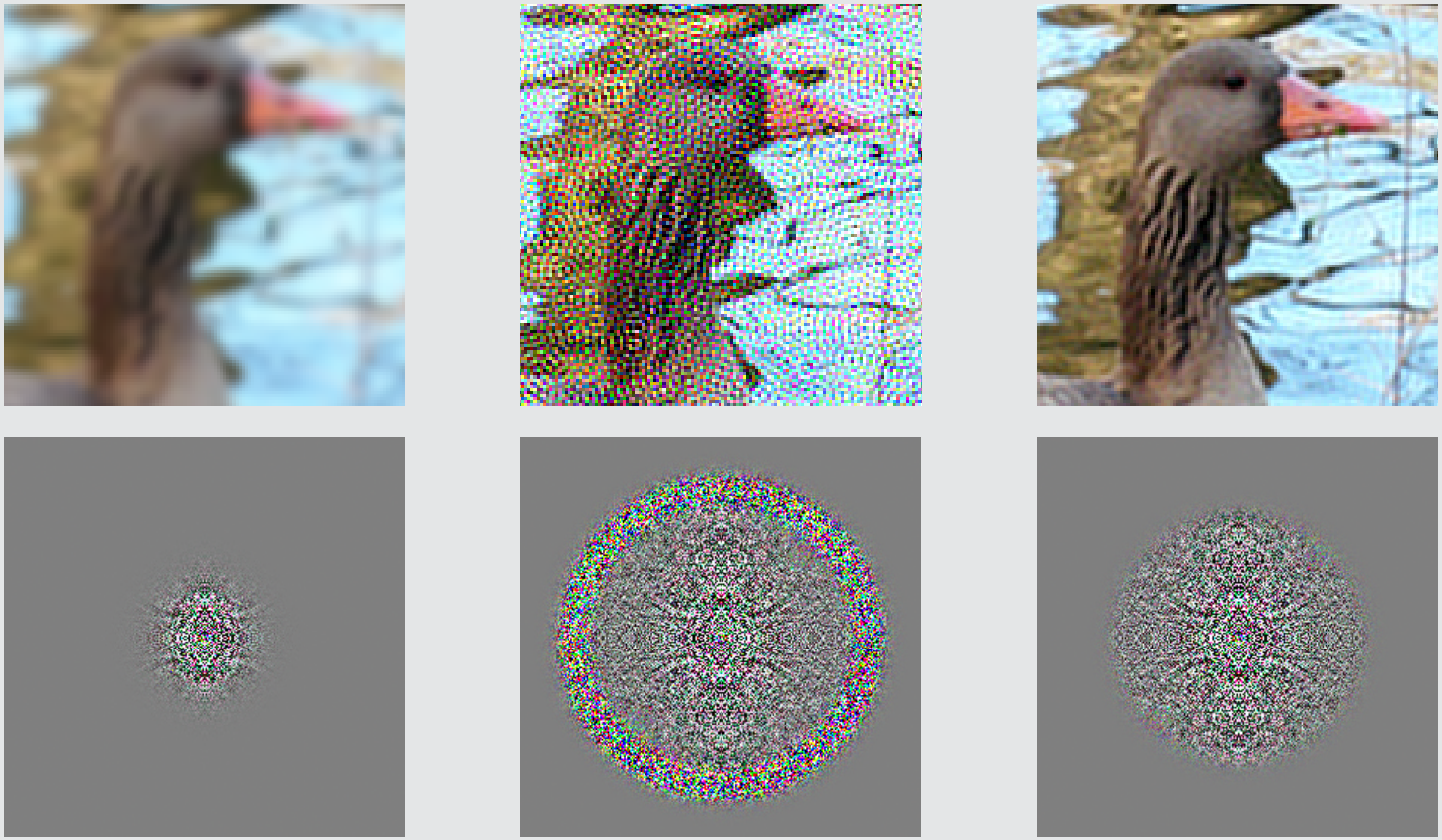


(a) y (b) z (c) $\frac{\omega_i^*}{|\omega_i|^2+\sigma^2/\lambda_i^2}$ (d) $\hat{z}$ (e) $\hat{x}$

Wiener deconvolution: optimal spectral sharpening.

Wiener filtering – Application to deconvolution

Wiener versus pseudo inverse

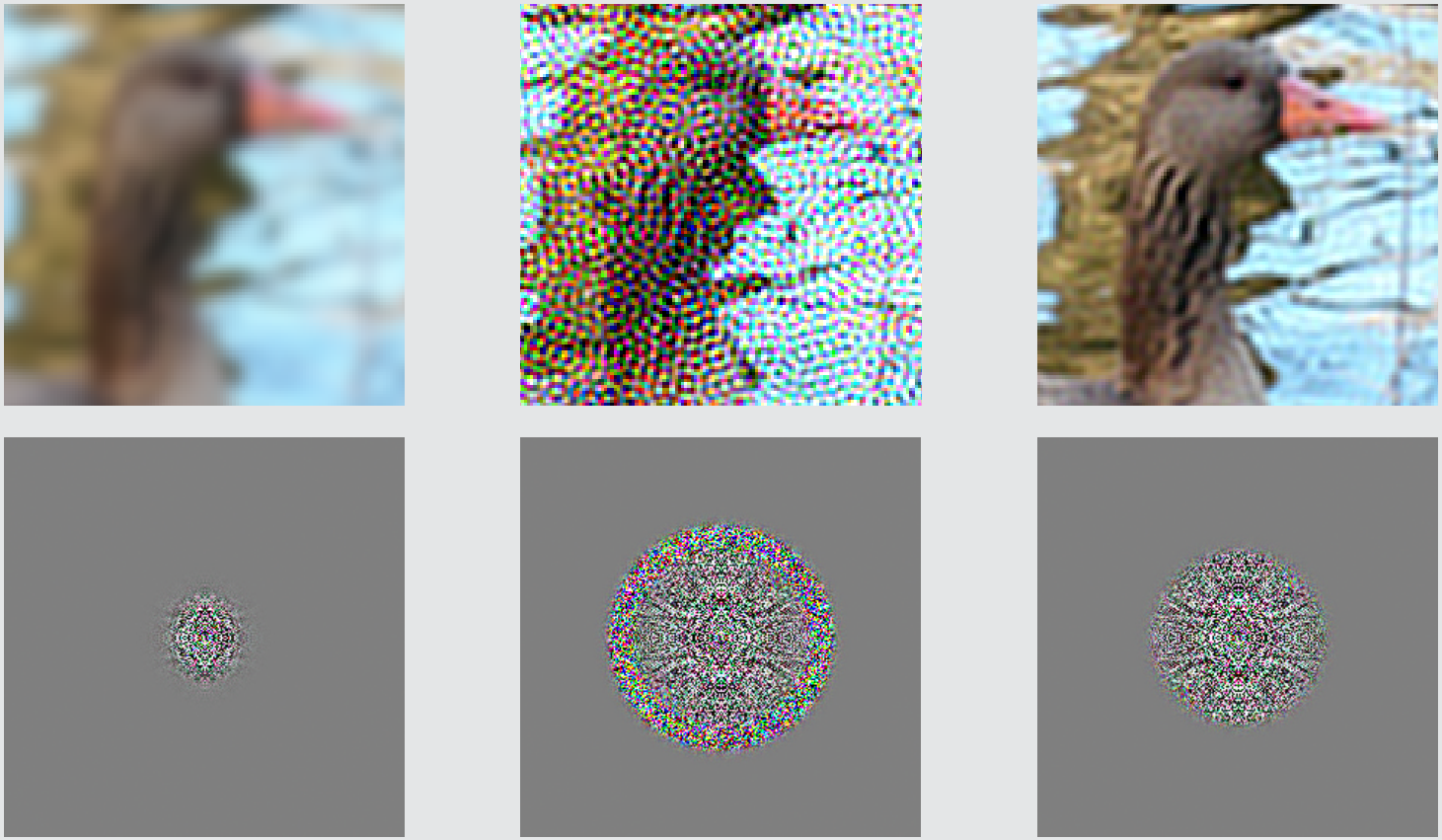


(a) Observation y (b) PINV $\boldsymbol{H}^{+}y$ (c) Wiener $\hat{x}^{\star}$

Invert attenuated frequencies while preventing from amplifying the noise.

Wiener filtering – Application to deconvolution

Wiener versus pseudo inverse



(a) Observation y (b) PINV $\boldsymbol{H}^{+}y$ (c) Wiener $\hat{x}^{\star}$

Invert attenuated frequencies while preventing from amplifying the noise.

Learning the LMMSE filter with PCA

LMMSE + Learning with PCA

Can we instead learn the ellipsoid (μ and L)?

Use an external data-set

- Let $x_1, \dots, x_k$ be a collection of images.
- Estimate $\mu = \mathbb{E}[x_k]$ and $\boldsymbol{L} = \mathrm{Var}[x_k]$ from the samples

$$\mu = \frac{1}{K}\sum_{k=1}^{K} x_k \quad \text{and} \quad \boldsymbol{L} = \frac{1}{K-1}\sum_{k=1}^{K}(x_k - \mu)(x_k - \mu)^*$$

- Problem: computing $\boldsymbol{L}$ requires to store n^2 values,
- For an image of size $n = 256 \times 256$

$$\mathbf{n^2 = 4,294,967,296} \quad (16\text{Gb in single precision}) \ \text{☹}$$

LMMSE + Learning with PCA

Use SVD

- Let $\tilde{x}_k = x_k - \mu$ and $\tilde{\boldsymbol{X}} = \frac{1}{\sqrt{K-1}} \begin{pmatrix} \tilde{x}_1 & \dots & \tilde{x}_K \end{pmatrix}$, we have

$$\boldsymbol{L} = \frac{1}{K-1} \sum_{k=1}^{K} (x_k - \mu)(x_k - \mu)^* = \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^*$$

- The SVD of $\tilde{\boldsymbol{X}}$ reads $\tilde{\boldsymbol{X}} = \boldsymbol{E} \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^*$ for some $\boldsymbol{V}^* = \boldsymbol{V}^{-1}$.

$$\text{Proof:} \quad \boldsymbol{L} = \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^* = \boldsymbol{E} \boldsymbol{\Lambda}^{1/2} \boldsymbol{V}^* \boldsymbol{V} \boldsymbol{\Lambda}^{1/2} \boldsymbol{E}^* = \boldsymbol{E} \boldsymbol{\Lambda} \boldsymbol{E}^*$$

- The SVD decomposition of $\tilde{\boldsymbol{X}}$ gives $\boldsymbol{E}$ and $\boldsymbol{\Lambda}$.
- No need to build $\tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^*$, but still $\boldsymbol{E}$ is a $n \times n$ matrix (16 Gb) ☹

Low rank property

- $\tilde{\boldsymbol{X}}$ is a $n \times K$ matrix $(K < n)$, whose columns are zero on average:

$$\frac{1}{K}\sum_{k=1}^{K}\tilde{x}^k = \frac{1}{K}\sum_{k=1}^{K}(x^k - \mu) = \frac{1}{K}\sum_{k=1}^{K}x^k - \mu = 0$$

- One of the columns is a linear combination of the others

$$\sum_{k=1}^{K}\tilde{x}^k = 0 \quad \Leftrightarrow \quad \tilde{x}^1 = -\sum_{k=2}^{K}\tilde{x}^k$$

- The family $\{\tilde{x}^k\}$ has $K-1$ independent vectors.
- The rank of $\tilde{\boldsymbol{X}}$ is $r = K - 1$.

How does that help?

LMMSE + Learning with PCA

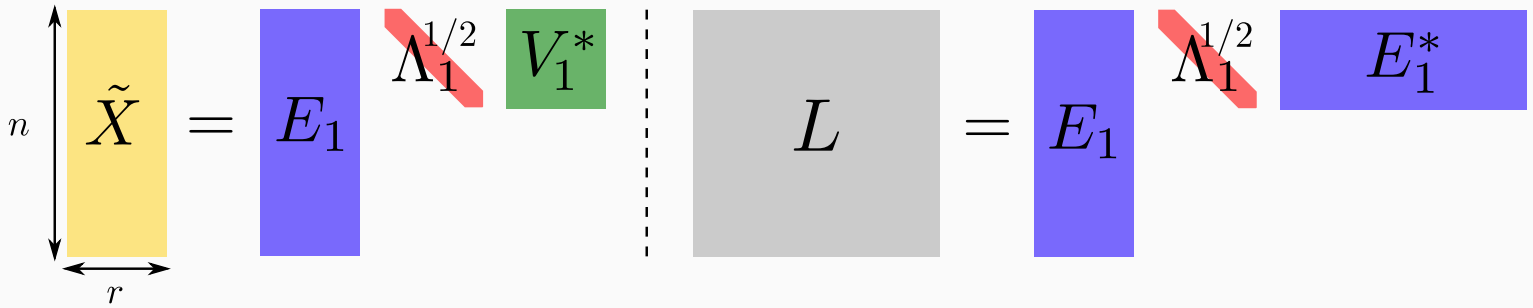
Reduced SVD

- $\tilde{\boldsymbol{X}}$ has only r non-zero singular values

$$\tilde{\boldsymbol{X}} = \boldsymbol{E}\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^* = \begin{pmatrix} \boldsymbol{E}_1 & \boldsymbol{E}_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_1^{1/2} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{V}_1^* \\ \boldsymbol{V}_0^* \end{pmatrix} = \boldsymbol{E}_1\boldsymbol{\Lambda}_1^{1/2}\boldsymbol{V}_1^*$$

- $\boldsymbol{L}$ has only r non-zero eigenvalues and depends only on E_1 and $\boldsymbol{\Lambda}_1$

$$\boldsymbol{L} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^* = \begin{pmatrix} \boldsymbol{E}_1 & \boldsymbol{E}_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{E}_1^* \\ \boldsymbol{E}_0^* \end{pmatrix} = \boldsymbol{E}_1\boldsymbol{\Lambda}_1\boldsymbol{E}_1^*$$



Require to store only $n \times r + r$ values

(r vectors $e_i \in \mathbb{R}^n$ + r values $\lambda_i \in \mathbb{R}^+$)

Principle component analysis (PCA)

- Assume the rank r being $K-1$ or even lower

$$\boldsymbol{L} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^* = \boldsymbol{E}_1\boldsymbol{\Lambda}_1\boldsymbol{E}_1^* = \sum_{i=1}^{r} \lambda_i^2 e_i e_i^*$$

$\Rightarrow$ Clean images are a linear combination of a few r images

$$x = \mu + \boldsymbol{L}^{1/2}\eta = \mu + \sum_{i=1}^{r} \eta_i \lambda_i e_i$$

with controlled weights $\mathbb{E}[\eta_i] = 0$ and $\mathrm{Var}[\eta_i] = 1$.

- The r directions e_i are called principle direction.
- Best way to capture most of variability with r dimensions only,

  i.e., to **cover most of the clean images with limited memory.**

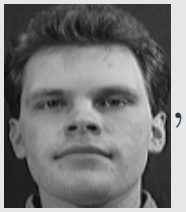
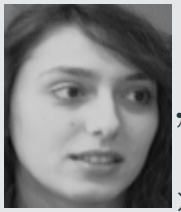
Example (Face dataset with additive white Gaussian noise)

- AT&T Database of Faces

  `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`

- 40 subjects, 10 images per subject (400 images in total)
- Gray-scale images of size $92 \times 112 = 10304$ pixels
- 39 subjects for the x_k and 1 subject for $y = x + w$

$$\boldsymbol{X} = (x_1, \ldots, x_K)$$

$$= \Big(\text{  }, \text{  }, \text{  }, \text{  }, \text{  }, \text{  }, \ldots \Big)_{\times 390}$$

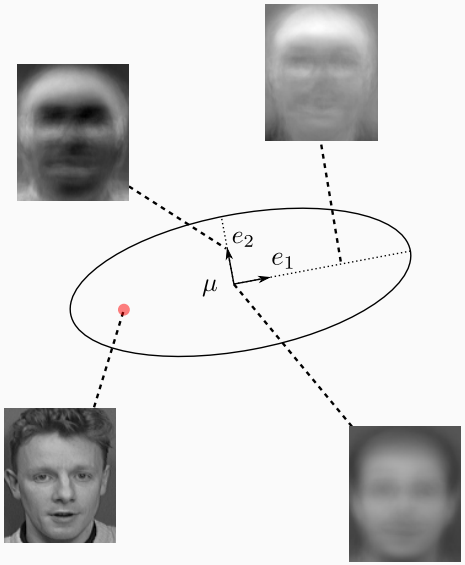
$$y = \text{  } = \text{  } + \text{  }, \quad w \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}_n)$$

LMMSE + Learning with PCA – Example

$$\boldsymbol{X} = \left(\quad , \quad , \quad , \quad , \quad , \quad , \quad \cdots \right)_{\times 390}$$

```
import numpy.linalg as nl

# Learning step (K = 390 files)
fs = [ 'filetrain1.png', ... ]
X  = np.zeros((n, K))
for i in range(K):
    X[:, i] = plt.imread(fs{i}).reshape(-1,)
mu = X.mean(axis=1)
X  = X - mu
X  = X / np.sqrt(K-1)
E, L, _ = nl.svd(X, full_matrices=False)
```



$$\boldsymbol{E} = \left(\quad , \quad , \quad , \quad , \quad , \quad , \quad \cdots \right)_{\times 390} \qquad \mu =$$

LMMSE + Learning with PCA – Filtering

```
# Denoising step
sig  = 40
x    = plt.imread('filetest.png').reshape(-1,)
y    = x + sig * np.random.randn(size(x))
z    = E.T.dot(y - mu)
hatz = L**2 / (L**2 + sig**2) * z
hatx = mu + E.dot(hatz)
```

$$z = \boldsymbol{E}^*(y - \mu)$$

$$\hat{z}_i = \frac{\lambda_i^2}{\lambda_i^2 + \sigma^2} z_i$$

$$\hat{x} = \mu + \boldsymbol{E}\hat{z}$$

(a) x (unknown)

(b) y (observation)

(c) $\hat{x}$ (estimate)

Was there a bug somewhere?

LMMSE + Learning with PCA – Checking the model

Did the model learn correctly on the training samples?

Testing sample

Training sample

(a) x (unknown) (b) y (observation) (c) $\hat{x}$ (estimate)

Yes. But it cannot generalize to new samples (over-fitting).

LMMSE + Learning with PCA – Checking the model

How to assess the model quality?

- Generate samples from $\mathcal{N}(\mu, \boldsymbol{L})$
- Judge if representative of targeted images.

How to generate samples from $\mathcal{N}(\mu, \boldsymbol{L})$?

$$\begin{cases} \eta & \sim \mathcal{N}(0, \mathrm{Id}_n) \\ \tilde{x} & = \mu + \boldsymbol{E}\boldsymbol{\Lambda}^{1/2}\eta \end{cases} \quad \Rightarrow \quad \tilde{x} \sim \mathcal{N}(\mu, \boldsymbol{L})$$

The model does not generate realistic faces.

LMMSE + Learning with PCA – Limits

Should we have used more than $K = 390$ training images?

Accuracy problem

- $\mu \in \mathbb{R}^n$: n degrees of freedom
- $\boldsymbol{L} \in \mathbb{R}^{n \times n}$: $(n^2 + n)/2$ degrees of freedom

For the estimation to be accurate: $n\sqrt{K} \gg$ degrees of freedom!

- For a size $n = 256 \times 256$: $\Rightarrow \boldsymbol{K \gg 1,073,840,130}$ ☹
- For a size $n = 8 \times 8$: $\Rightarrow \boldsymbol{K \gg 1,122}$ ☺

Scaling/Memory problem (assume 1Gb available)

- Storing $\boldsymbol{E}$ and $\boldsymbol{\Lambda}$ require $4(n \times K + n)$ bytes in single precision
- For $n = 256 \times 256$: $\Rightarrow \boldsymbol{K < 4,096}$ ☹
- For $n = 8 \times 8$: $\Rightarrow \boldsymbol{K < 4,194,304}$ ☺

Except if $n = 8 \times 8$ this method does not work...

Non-Local Bayes

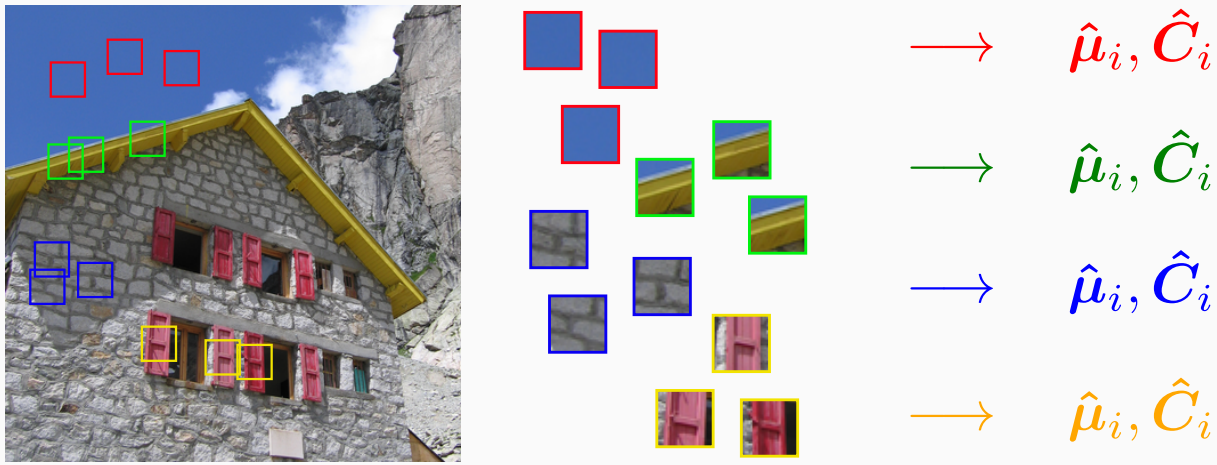
Non-Local Bayes [Lebrun, Buades, Morel, 2013]

- Apply LMMSE on patches by mimicking the (block) Non-Local means,
- Instead of taking only the average of similar patches

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{Z}\sum_j w_{i,j}\boldsymbol{y}_j \quad \text{and} \quad Z = \sum_j w_{i,j}$$

- Estimate also the **covariance matrix** of the patches in the stack

$$\hat{\mathbf{C}}_i = \frac{1}{Z}\sum_j w_{i,j}(\boldsymbol{y}_j - \hat{\boldsymbol{\mu}}_i)(\boldsymbol{y}_j - \hat{\boldsymbol{\mu}}_i)^*$$



Non-Local Bayes [Lebrun, Buades, Morel, 2013]

- Assuming the noise to be additive Gaussian, we have

$$\mathbb{E}[\hat{\mathbf{C}}_i] = \boldsymbol{L}_i + \boldsymbol{\Sigma}_i$$

- which provides us a local estimate for $\boldsymbol{L}_i$:

$$\hat{\boldsymbol{L}}_i = \hat{\mathbf{C}}_i - \boldsymbol{\Sigma}_i$$

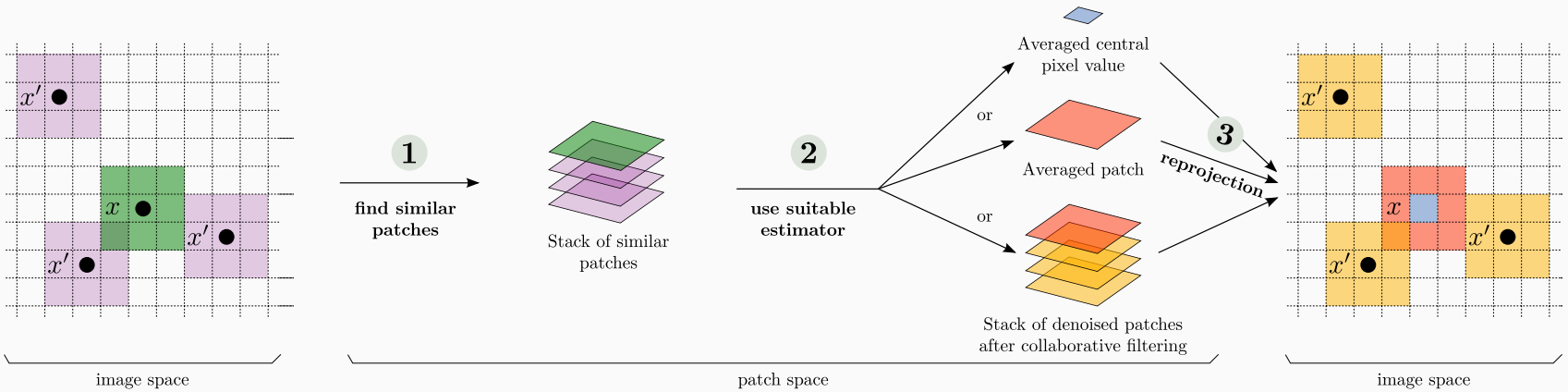
- $\hat{\boldsymbol{L}}_i$ may have negative eigenvalues, set them to zero (often required for small stacks or low SNR).

Non-Local Bayes [Lebrun, Buades, Morel, 2013]

- Plug these estimators in the LMMSE to denoise each patch of the stack

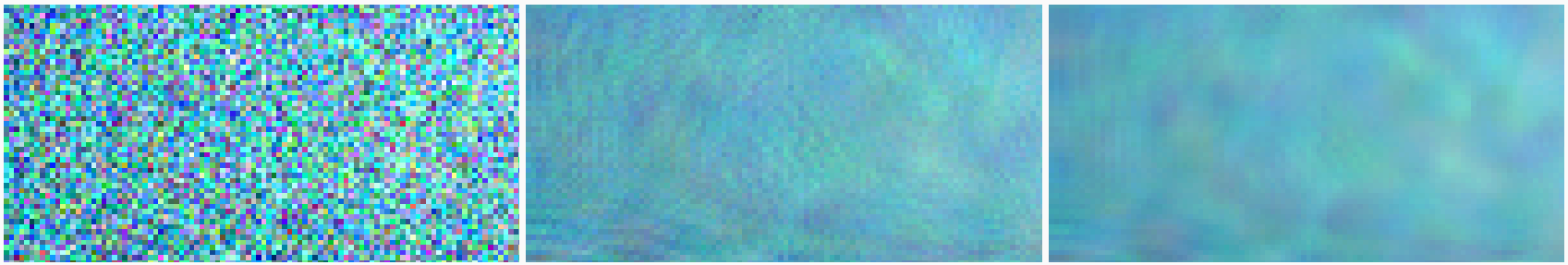
$$\hat{\boldsymbol{x}}_j = \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{L}}_i(\hat{\boldsymbol{L}}_i + \boldsymbol{\Sigma})^{-1}(\boldsymbol{y}_j - \hat{\boldsymbol{\mu}}_i)$$

- Note: As $\hat{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{L}}_i$ depends on $\boldsymbol{y}_j$, in fact **non-linear** filter.
- Reproject each patch at their original location,
- Average overlapping patches together.



Non-Local Bayes [Lebrun, Buades, Morel, 2013]

- By estimating the statistics of similar patches,
- The estimators may amplify random patterns in the noise.



(a) Noisy image (b) 1st step (c) 2nd step

Workaround: 2 steps filtering

- Repeat the filter a second time,
- Estimate $\boldsymbol{\mu}_i$ and $\boldsymbol{L}_i$ from patches of the first estimate,
- Assume them to be clean: $\hat{\boldsymbol{L}}_i = \hat{\boldsymbol{C}}_i$.

Non-Local Bayes

(a) Noisy image

(b) 1st step

(c) 2nd step

Non-Local Bayes

(d) Block Non-Local means

Non-Local Bayes

(a) Noisy image

Non-Local Bayes

(b) 1st step

Non-Local Bayes

(c) 2nd step

Non-Local Bayes

(d) Block Non-Local means

Non-Local Bayes and LMMSE

Pros and cons (compared to Non-Local means)

By taking into account the covariance (2nd order moment):

- More degrees of freedom

  $\Rightarrow$ capture complex patterns/textures even with low SNR,
- Too much flexibility, over-fit the low-frequency components of the noise,

  $\Rightarrow$ Use a multi-scale approach + Trick for homogeneous regions

  [Lebrun, Colom, Morel, 2015]

Conclusions about LMMSE

- Except if it is made spatially adaptive for patches (hence non-linear),

  $\Rightarrow$ The LMMSE is linear, thus inappropriate for image processing.
- As: Assuming Gaussian noise + Gaussian prior $\Rightarrow$ LMMSE,

  $\Rightarrow$ The limitation of the LMMSE means that:

  Natural clean images are far from being Gaussian distributed.

Questions?

Next class: Shrinkage and wavelets

Sources, images courtesy and acknowledgment

L. Condat

A. Horodniceanu

S. Kay

R. Willet

Wikipedia