Computer Vision (CS763)

Teaching cameras to "see"

Robust Methods in Computer Vision

Arjun Jain

Most slides courtesy Ajit Rajwade

https://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/ImageAlignment.pdf

Least Squares Estimates

- Consider a quantity y related to quantity x in the form $y = f(x; \boldsymbol{a})$
- Here, $\boldsymbol{a}$ is a vector of parameters for the function f. For example, if $y = mx + c$ then $\boldsymbol{a} = [\mathrm{m\ c}]^T$
- Now suppose we have N data points (x_i, y_i) where y_i is corrupted by noise, and we want to estimate $\boldsymbol{a}$
- We have done this in the past by finding $\boldsymbol{a} = \mathrm{argmin}_{\boldsymbol{a}} \sum_{i=1}^{N} (y_i - f(x_i; \boldsymbol{a}))^2$

Least Squares Estimates

- Why did we minimize the squared error loss? What would happened if we changed the power to 4? Or 1 or 3 (with absolute value)?

Least Squares Estimates: Gaussian Noise Model

- Let us assume that the noise ϵ_i affecting y_i is
  1. Additive
  2. Gaussian with 0 mean and some known standard deviation σ i.e. $\epsilon_i \sim N(0, \sigma)$

$$y_i = f(x_i; \boldsymbol{a}) + \epsilon_i$$

Observed Input Parameters Noise

Random Variable Recap:

- Say X is a random variable (R.V.) with mean μ and std. σ

- Then, $\mathrm{Y} = \alpha\mathrm{X} + \gamma$ is also a R.V. with mean γ and std. $\alpha\sigma$

- Given x_i, here ϵ_i's are our X's and and y_i's are our Y's with $\alpha = 1$ and $\gamma = f(x_i; \boldsymbol{a})$

Least Squares Estimates: Gaussian Noise Model

- Let us also assume that the noise ϵ_i affecting different y_i is are independent of each other
- Now given some value of $\boldsymbol{a}$ and some x_i, the probably density of y_i is:

$$p(y_i|x_i, \boldsymbol{a}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - f(x_i;\boldsymbol{a}))^2}{2\sigma^2}}$$

Likelihood of y_i

Least Squares Estimates: Gaussian Noise Model

- Now given some value of $\boldsymbol{a}$ and some x_i, the probably density of y_i is:

$$p(y_i|x_i,\boldsymbol{a}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - f(x_i;\boldsymbol{a}))^2}{2\sigma^2}}$$

- And (due to independence of noise)

$$p\left(\{y_i\}_{i=1}^N \middle| \{x_i\}_{i=1}^N, \boldsymbol{a}\right) = \prod_{i=1}^{N} p(y_i|x_i,\boldsymbol{a})$$

Likelihood of $\{y_i\}$, i=1 to N

Least Squares Estimates:

- We want to find a value of $\boldsymbol{a}$ which **maximizes** this probability density, this likelihood of $\{y_i\}$, i=1 to N. This is called **maximum likelihood estimate (MLE)** of $\boldsymbol{a}$
- .. equivalent to **maximizing** the log of this probability density (why?)
- ..equivalent to **minimizing** the negative log of this probability density, i.e.

$$\mathbf{J}(\boldsymbol{a}) = \sum_{i=1}^{N} \frac{(y_i - f(x_i;\boldsymbol{a}))}{2\sigma^2} + \log(\sigma\sqrt{2\pi})$$

Least Squares Estimates:

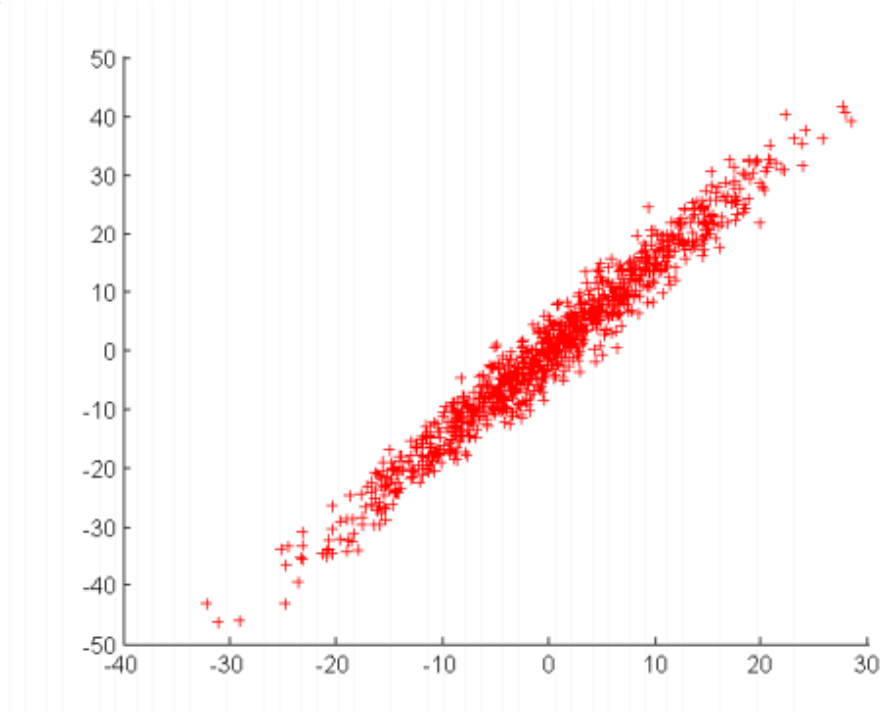
- This shows us that the least squares estimate is the same as the **maximum likelihood estimate** under the assumption that the noise affecting different samples was **independent** and **Gaussian** distributed with **fixed** variance and mean 0.

- Why maximum likelihood estimate of $\boldsymbol{a}$? Intuitively, it is the value of $\boldsymbol{a}$ that best agrees with or supports the observations $\{y_i\}_{i=1}^{N}$ given $\{x_i\}_{i=1}^{N}$

Least Squares Fit of a Line

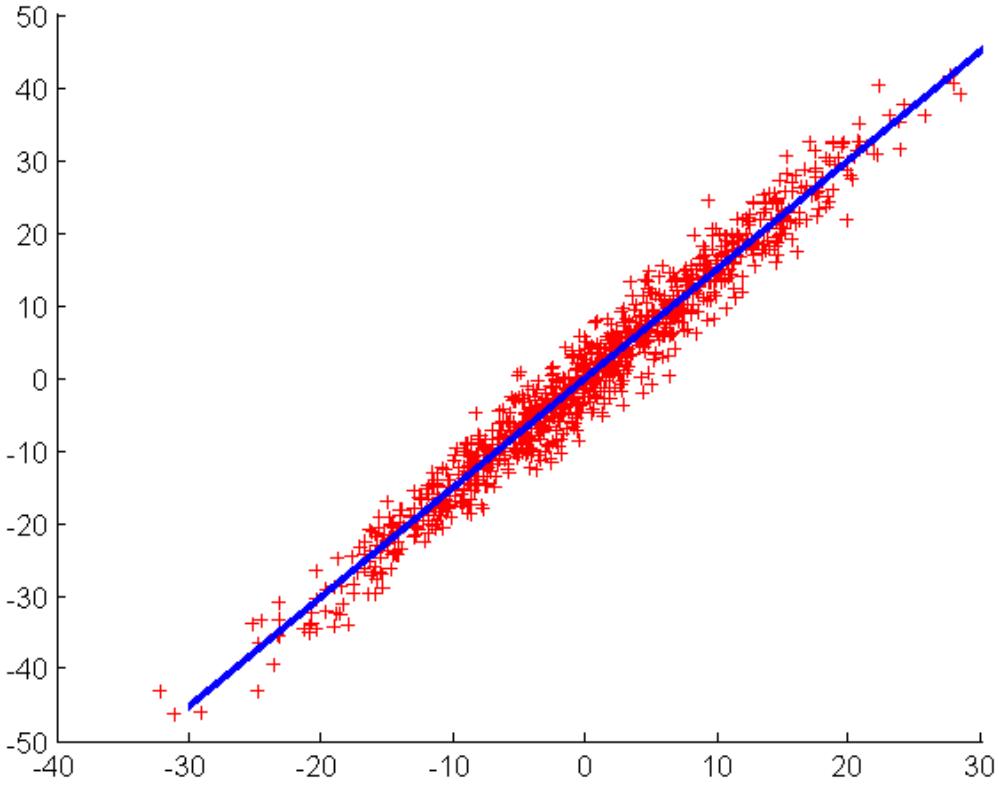
$$(m^*, c^*) = \arg\min J(m,c) = \sum_{i=1}^{N} (y_i - mx_i - c)^2$$

$$\therefore \begin{pmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & N \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{pmatrix}$$




```
N = 1000;x = 10*randn(N,1);
y = 1.5*x + randn(N,1)*3;
 scatter(x,y,'r+')
```

Least Squares Fit of a Line



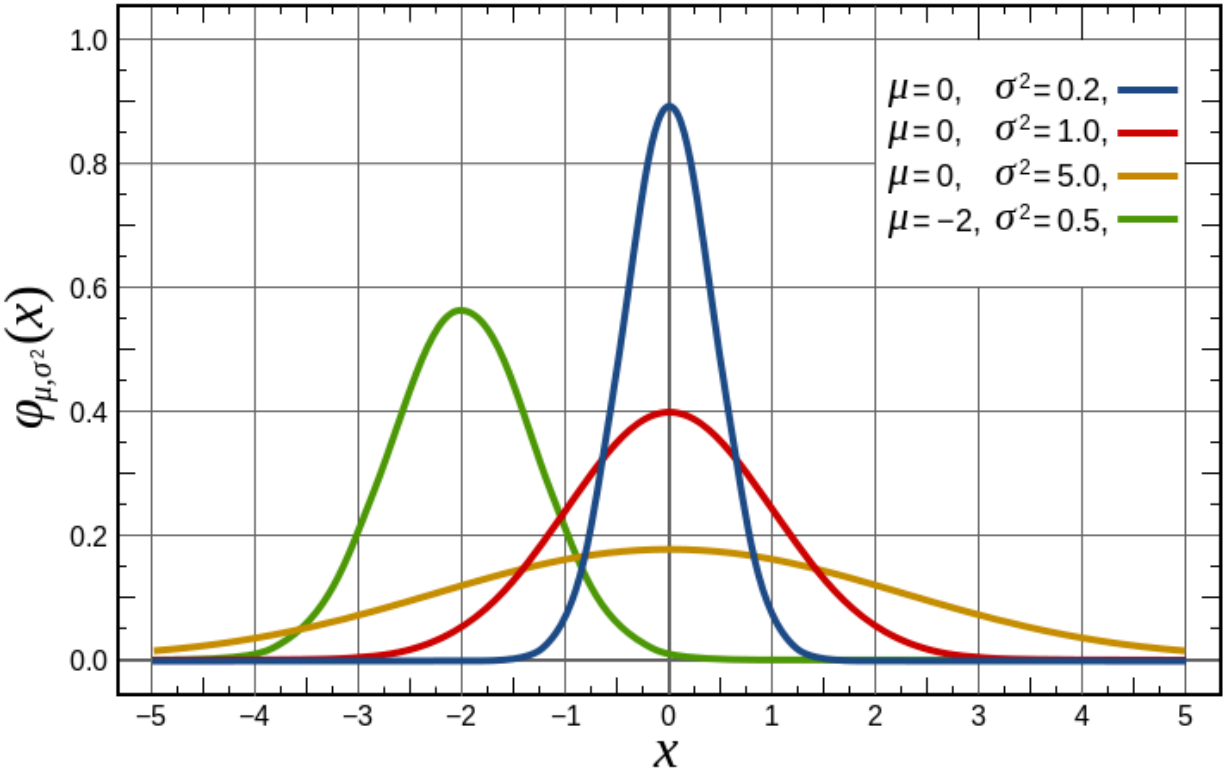
Result of a least-squares estimate under Gaussian noise:

Estimated slope: 1.5015 (versus 1.5)
Estimated intercept: 0.088 (versus 0)

Other Least-squares Solutions in Computer Vision

- Camera calibration – SVD
- Parametric motion estimation – SVD or pseudo-inverse (homography, affine, etc.)
- Fundamental/essential matrix estimation - SVD (we will study this later in stereo-vision)
- Optical Flow (Horn-Shunck as well as Lucas-Kanade) (we will study this later)

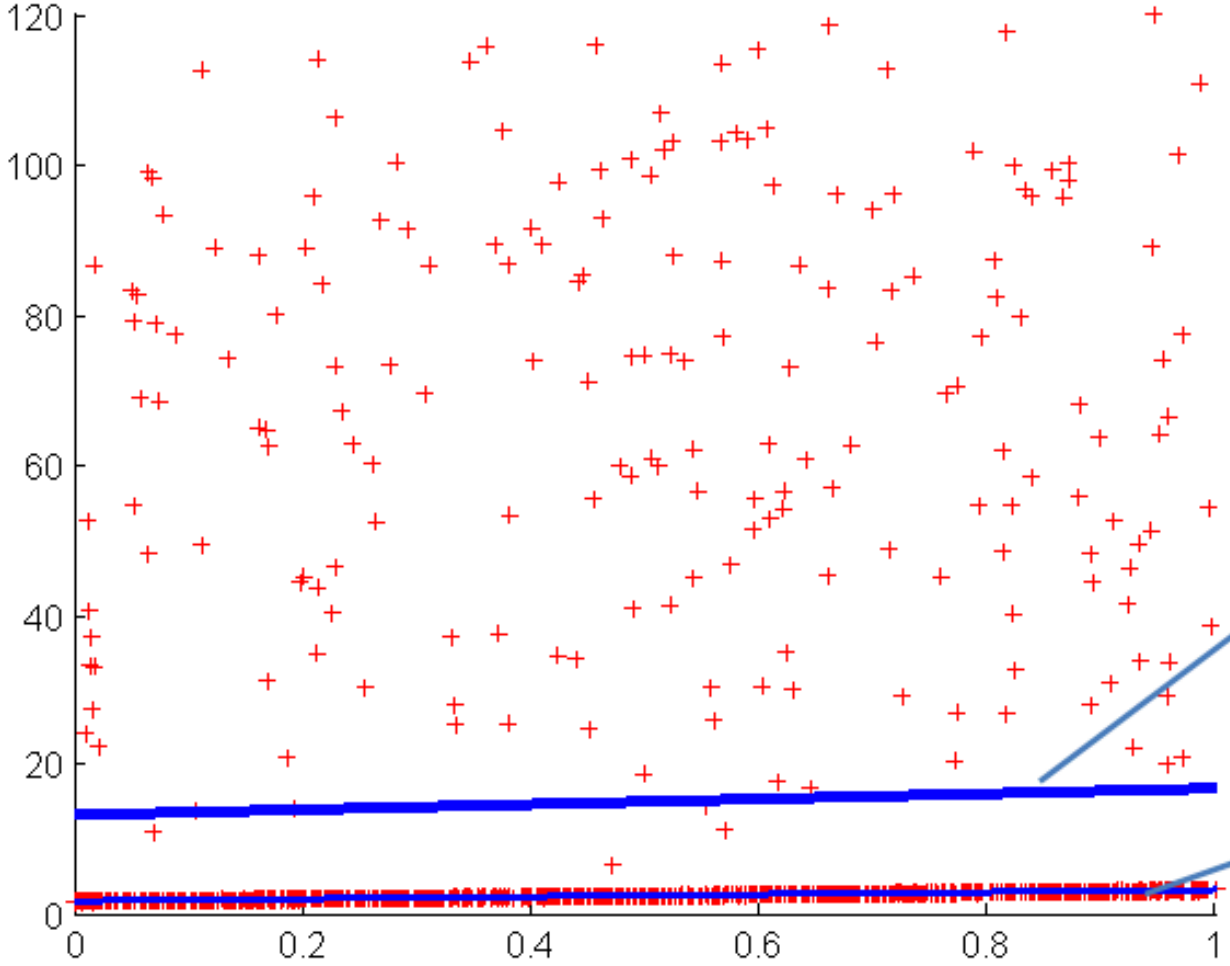
Outliers and Least-squares Estimates



Observation: Let *x* be a random variable with a Gaussian distribution. Then the probability that *x* takes on any value in a small range far away from the mean (typically at a distance of more than +/- 3σ) is **very low.** See diagram above.

Outliers and Least-squares Estimates

- The learning from the previous observation is this: the least squares estimate assumes that most points will lie **close** to the true (unknown) model else their probability would be very low.
- Now, suppose the given dataset contains wild outliers, i.e. stray points that simply do not obey the gaussian noise model.
- These outliers will skew the least squares estimate – as it tries to force a solution which maximizes the likelihood of the **outliers** as well.
- Since outliers were extremely unlikely under the Gaussian probability density, the model (during maximum likelihood estimation) has to change itself to make the outliers more likely.



20% of the points are outliers. They have skewed the estimate of the slope from 1.5 to 3.6 and the intercept from 2 to 13.5.

But outliers do occur in Real Life!

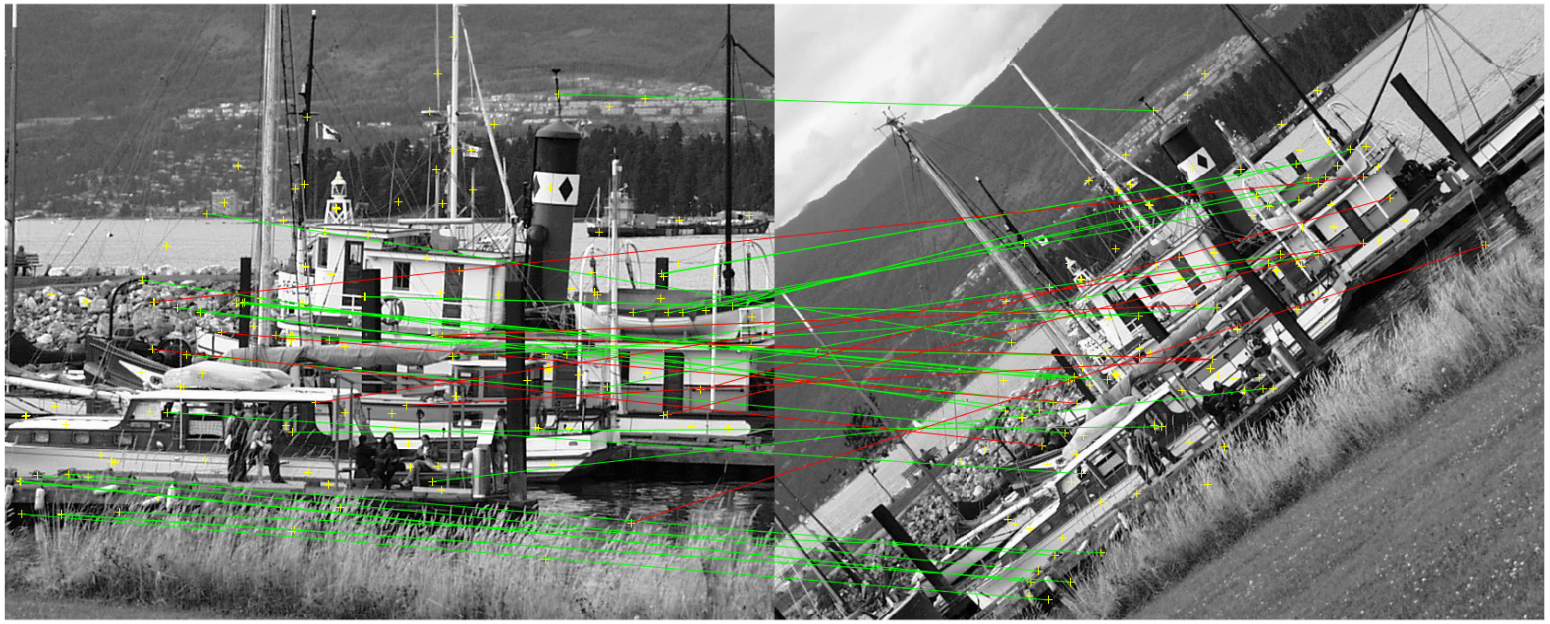
Outlier Example 1

- Salt and pepper noise in images

Salt and pepper noise (a special case of impulse noise)

Outlier Example 2

- Estimation of the spatial transformation between images (could be translation, rotation, affine or homography) requires N+ pairs of corresponding points. Some of these correspondences can be faulty.



Outlier Example 3

Outlier Example 4: Matching

- Some of these correspondences can be faulty for various reasons
  - occlusions/ difference in field of view / shadows
  - identical objects in the scene
  - change in the position of some objects in the scene (even though the global motion is homography, these objects will not conform to that same motion model).

http://petapixel.com/2012/10/22/an-eerie-time-lapse-of-seattle-minus-all-the-humans/

Dealing with Outliers

1. Heavier Tailed Distributions

2. RANSAC

3. LMeds

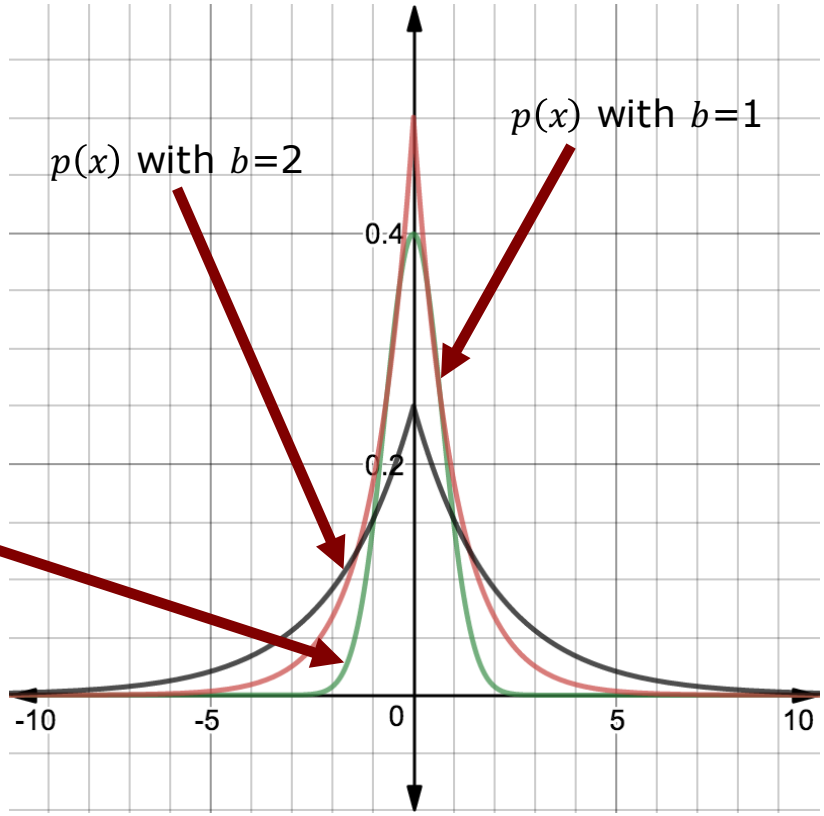
Dealing with Outliers

1. **Heavier Tailed Distributions**
2. RANSAC
3. LMeds

Dealing with Outliers
1. Heavier Tailed Distributions

- Plot the function below with different b and $\mu = 0$

$$p(x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$



$p(x) =$ Laplacian probability density function

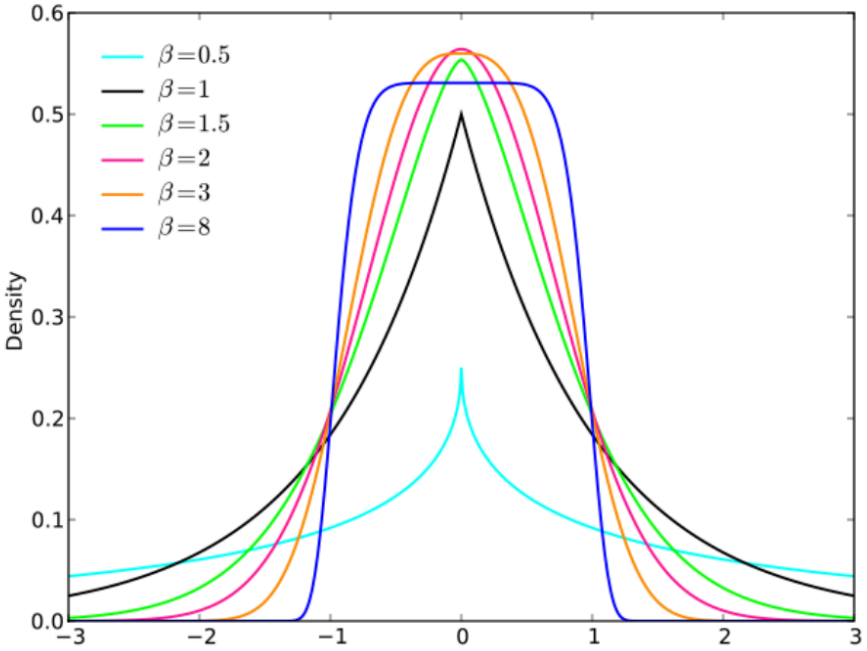
https://www.desmos.com/calculator/xyswczsnmv

Dealing with Outliers
1. Heavier Tailed Distributions

- Laplacian probability density function is a distribution which has heavier tails than the Gaussian. This means that if a random variable is Laplacian distributed, the probability that it can take values far away from the mean, is higher than what it would be if the variable were Gaussian distributed.

The Laplacian (or Laplace) pdf is not to be confused with the Laplacian Operator for a function f(x,y). Do you remember?

Dealing with Outliers
1. Heavier Tailed Distributions

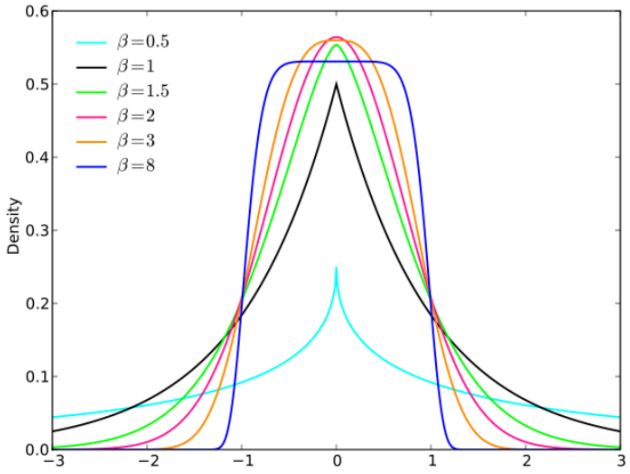


$$p(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-\left(\frac{|x-\mu|}{\alpha}\right)^{\beta}}$$

The Laplacian probability density function is a special case of the family of **Generalized Gaussian probability density functions** with **shape parameter** β and **scale parameter** α. As β reduces below 1, the density function becomes heavier tailed.

Question: Why do we care for heavier tails? Because they ensure that the wild outliers are more likely to occur (than the Gaussian pdf). Consequently, a maximum likelihood estimation assuming heavy-tailed noise models will be less affected by outliers.

Dealing with Outliers
1. Heavier Tailed Distributions

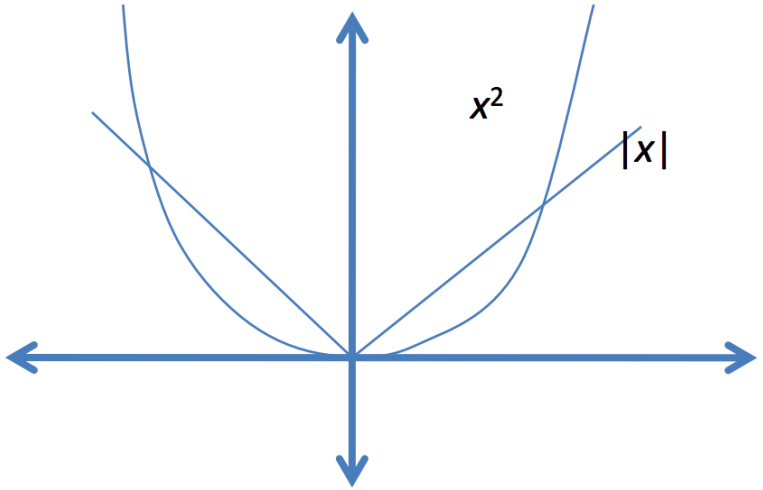


$$p(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-\left(\frac{|x-\mu|}{\alpha}\right)^{\beta}}$$

Note that x^p grows faster than x^q beyond |x| = 1, for p > q

Why do these Generalized Gaussians with shape parameter β < 2 have heavier tails than the usual Gaussian (i.e. β = 2)?

Consider a zero-mean Gaussian and a zero-mean Laplacian (without loss of generality), i.e. β = 1.
The term inside the exponential in a Gaussian is x^2, where it is $|x|$ for a Laplacian.



Dealing with Outliers
1. Heavier Tailed Distributions

- Assume the noise has a Laplacian distribution.

- Then what will be the likelihood of $\{y_i\}$, i=1 to N?

- We need to maximize the likelihood in order to estimate **a**

Likelihood in case of Laplacian Noise

- Now given some value of $\boldsymbol{a}$ and some x_i, the probably density of y_i is:

$$p(y_i|x_i,\boldsymbol{a}) = \frac{1}{2b} e^{-\frac{|y_i - f(x_i;\boldsymbol{a})|}{b}}$$

- And (due to independence of noise)

$$p\left(\{y_i\}_{i=1}^N \middle| \{x_i\}_{i=1}^N, \boldsymbol{a}\right) = \prod_{i=1}^{N} p(y_i|x_i,\boldsymbol{a})$$

Likelihood of $\{y_i\}$, i=1 to N

Laplacian Noise MLE

- We want to find a value of $\boldsymbol{a}$ which **maximizes** this probability density, this likelihood of $\{y_i\}$, i=1 to N. This is called **maximum likelihood estimate (MLE)** of $\boldsymbol{a}$
- .. equivalent to **maximizing** the log of this probability density (why?)
- ..equivalent to **minimizing** the negative log of this probability density, i.e.

$$\mathbf{J}(\boldsymbol{a}) = \sum_{i=1}^{N} \frac{|y_i - f(x_i;\boldsymbol{a})|}{b} - \log(2b)$$

Laplacian Noise MLE

- **minimize** the negative log of this probability density, i.e.

$$\mathbf{J}(\boldsymbol{a}) = \sum_{i=1}^{N} \frac{|y_i - f(x_i;\boldsymbol{a})|}{b} - \log(2b)$$

- Unfortunately, there is no closed form solution (based on inverse/pseudo-inverse) in this case unlike the squared error

- One will need iterative methods like gradient descent

Adaptive Gradient Descent

Repeat till there is no change in $\mathbf{a}$

{

$\alpha = \alpha_{\max}$

while $(\alpha > \alpha_{\min})$

{

$$\mathbf{a}' = \mathbf{a} - \alpha \frac{dJ(\mathbf{a})}{d\mathbf{a}}, 0 < \alpha << 1$$

if $J(\mathbf{a}') > J(\mathbf{a}), \alpha = \alpha_{\max} / 2$

else $\{\mathbf{a} = \mathbf{a}'; \text{break}\}$

}

}

α = **step size** of gradient descent

Gradient descent converges to a local minimum of the energy function (or objective function), i.e. J(**a**) in this slide, if the step size α is "small enough" to never.

Unfortunately, too small a step size is too expensive. A large step size may lead to increase in the energy function across iterations.

So we pick the largest possible step-size (within a given range) that reduces the energy – this is called **gradient descent with adaptive step-size** or **adaptive gradient descent**.

Dealing with Outliers

1. Heavier Tailed Distributions

2. **RANSAC**

3. LMeds

RANSAC: Random Sample Consensus

1. Arbitrarily choose k out of N points where *k* is the smallest number of points required to determine **a**. Call this set C.

2. Determine **a** using an inverse (say) from C.

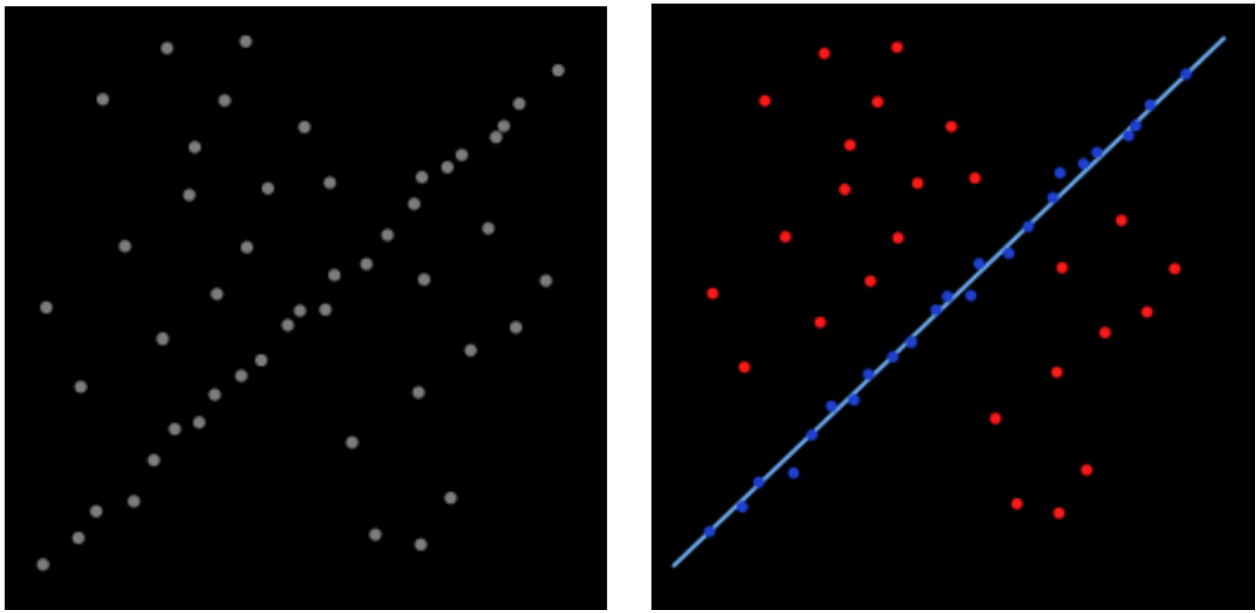
3. Determine and remember the squared residual errors for all the other N-k points, i.e. compute

$$\{e_i = (y_i - f(x_i; \boldsymbol{a}))^2\}_{i \neq C}$$

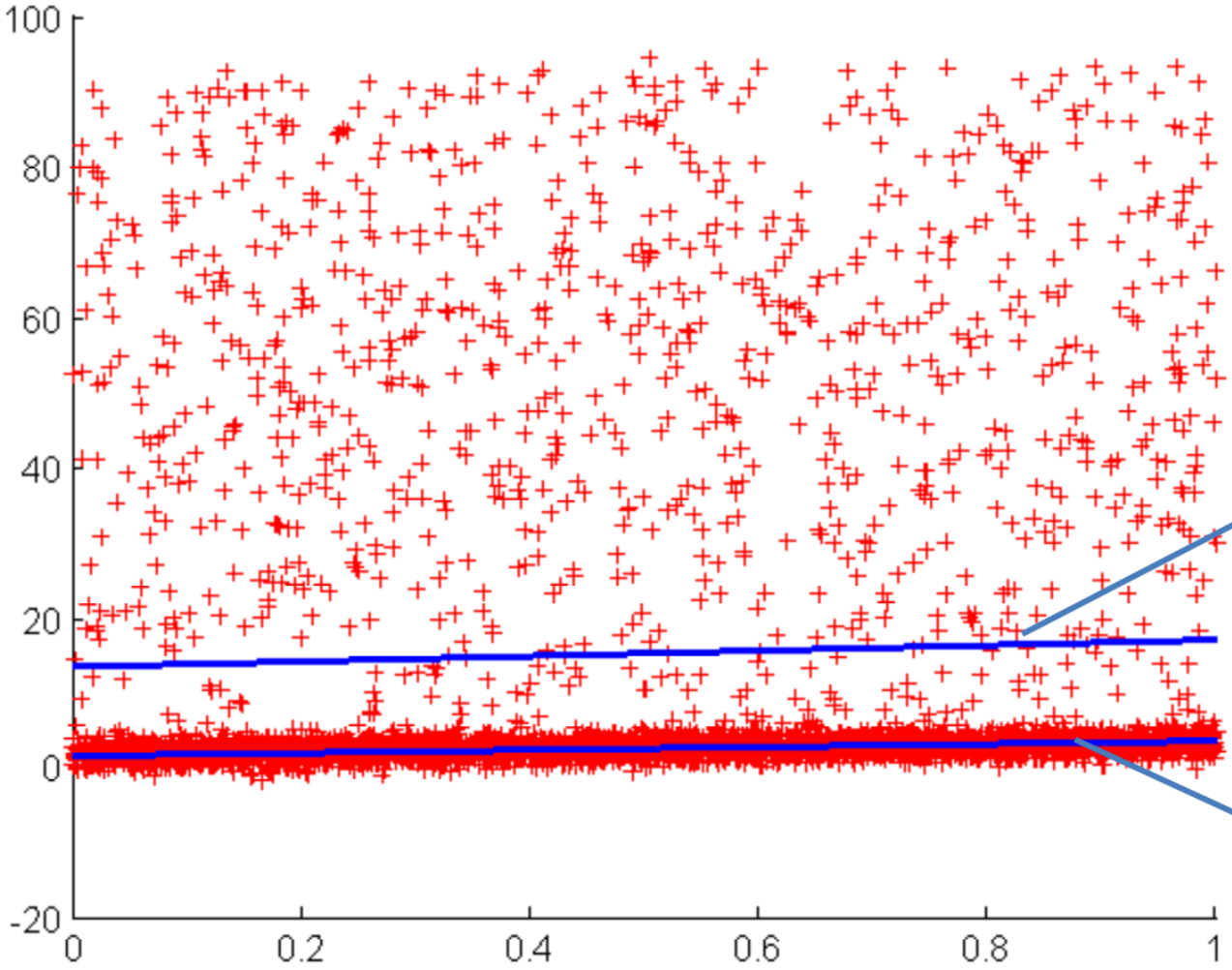
RANSAC: Random Sample Consensus

4. Count the number of points for which $e_i < \lambda$ where λ is some threshold. These points form the "consensus set" for the chosen model.

Sample result with RanSaC for line fitting.



RANSAC: Random Sample



Least squares estimate

RanSaC estimate

Slide Information

- The slide template has been created by Cyrill Stachniss (cyrill.stachniss@igg.uni-bonn.de) as part of the photogrammetry and robotics courses.
- A lot of material from Ajit Rajwade's CS763 course (https://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/ImageAlignment.pdf)
- Appendix A.7 of Trucco and Verri
- Article on Robust statistics by Chuck Stewart
- Original article on RanSaC by Fischler and Bolles
- Article on RanSaC variants by Torr and Zisserman
- **I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.**

Arjun Jain, ajain@cse.iitb.ac.in