Data Science and Scientific Computation Core Course

Understanding and visualizing data

March 20, 2023

Contents

1 Introduction

More and more frequently we are faced with large datasets, e.g., long temporal signals or high-dimensional samples of interesting processes. These data can be either continuous (e.g., as is typical of physics experimental data) or discrete (e.g., DNA / text sequences, qualitative descriptors). Here we will be concerned mostly with continuous signals, although many approaches we will describe have been developed also for the discrete cases. Non-trivial statistical structure in the data, which we can model or exploit for prediction, almost by definition must be due to some sort of correlation structure, and our goal here will be to quantify and visualize the simplest types of correlation.

We will introduce and review, as necessary, basic and generically applicable approaches to characterizing a new dataset. We will want to know, for example, if the collected samples are statistically independent, or whether a certain temporal signal is stationary or not. Further, we will examine the first- and second-order statistical structure in the data. This is useful for several reasons, out of which it is worth highlighting the following three.

First, knowing the low-order statistical structure in the data often allows us to optimize further data analysis, either by picking a more appropriate method (e.g., some methods work best if the data is nearly Gaussian), by preprocessing the signals (e.g., using filtering or applying nonlinear transformations), or by visualizing the data in an intuitively informative way (e.g., projecting the data along axes with largest variance or clustering the data).

Second, datasets are often of sufficient size to permit direct sampling of first- and second-order statistics, allowing us to approach the data in an as assumption-free way as possible. Sometimes, data collection or experimental setups leave their imprints on low-order statistics and it is useful to understand how such systematic effects may impact the results of various analyses.

Third, there exist processes that can be fully described by their first- and second-order statistics: if we are lucky, then our data is described by such simple processes, but even if not, these processes may serve as good baseline or null models against which to compare real data.

1.1 Topics

- Histograms obtained by binning the data, empirical PDFs and CDFs, mean / std / skewness / kurtosis, quantile statistics.
- Two-point statistics (covariance), different normalization conventions (covariance, Pearson correlation), covariance eigendecomposition and principal component analysis (PCA).
- Subsampling and error bar estimation using boostrap / jackknife methods.
- Stationary and non-stationary processes, timescales and auto-correlation functions.
- Looking for higher-order statistical structure: K-means clustering, multi-dimensional scaling (MDS), t-SNE, and independent component analysis (ICA).

1.2 Data

This is a hands-on course. You will apply the principles from the lectures to the actual data. The used data sets are described below: understanding of the data acquisition can aid the data analysis.

Microelectrode array data. Two voltage traces (`trace1, trace2`) from two single electrodes

in a microelectrode array that records extracellular activity of retinal ganglion cells. This is a (pre)amplified signal that contains a mixture of noise, background "rumbling" of cells that are a significant distance away from the electrode but nevertheless induce voltage fluctuations in it, and closeby cell(s) that, when they spike, cause a large, sharp echo of the action potential, or spike, on the electrode. Spikes are unitary neural events that neurons use for communication; they consist of stereotyped voltage fluctuations, usually $\sim 1-3$ ms long, across the neuron's membrane, and they can propagate along the neuron's axon over long distances. Usually, these spikes are signals of interest that need to be isolated from experimental data. The data vector consists of $2 \cdot 10^6$ voltage samples (in μV), sampled at $2 \cdot 10^4$ Hz, for a total of 100 seconds of recording. Data is by Olivier Marre, details in Ref [1]. We will use this voltage trace as a running example to demonstrate the topics discussed in the course.

Natural images data. An ensemble of 45 calibrated grayscale natural images from Ruderman et al, *Phys Rev Lett* **73** (1994), with resolution of 256×256 pixels, used for some homeworks.

Read the spike sorting paper by Olivier Marre et al., "Mapping a complete neural population in the retina," Journal of Neuroscience **32:** 14859–73 (2012) [1]. This is a technical paper; solving homeworks below does not require such detailed understanding. The main purposes of this reading assignment are to: **(i)** give you some background on the actual experiment where our data comes from and introduce you to the spike sorting problem; **(ii)** to demonstrate a difficult data analysis problem in a realistic setup where theoretical ideas are a useful starting point, but for an actual working algorithm one needs to pay attention to the details of the system and the apparatus; direct applications of out-of-the-box methods tend not to do well on this problem.

1.3 Why are we analyzing data and building models?

One way to answer this question is by thinking of what types of insight about the world we can extract from data. Broadly, we may be interested in **making predictions**, **testing hypotheses in a statistically rigorous fashion,** or looking for **principled understanding** of the natural phenomena. These goals are by no means exclusive, but focusing on one aspect rather than the other may lead us to pick a different data analysis or modeling approach. For example, predictive models can be very successful when evaluated on withheld / future data generated by the same process that created the training data, and if the training data is plentiful, such models can be very complex, with millions of parameters. This may lead to superb performance but very low interpretability; i.e., it may be difficult for humans to understand precisely *what* structure of the data the model has learned and made use of for prediction. In contrast, models chosen to support human interpretability are usually simpler, and their components (e.g., parameters or assumed processes) usually map in a relatively straightforward fashion to physical reality. Here, the ability to generalize to withheld data generated by the same process is desired but not the ultimate objective; most often, generalization of model predictions to qualitatively new kind of data is preferred.

Different disciplines also emphasize to various amounts different tradeoffs faced in model construction and inference. The principal balance studied by statistical learning theory is the balance between model complexity and amount of training data (to control for overfitting of the models to training data). The principal balance emphasized in natural sciences is, in contrast, the balance between the model complexity and the richness of model predictions, assuming

sufficient data to actually perform model inference.

Similarly, different fields also differ in their fundamental conception about what constitutes the model and its specification. Models in statistics or machine learning thus span the range from "null models" used for hypothesis testing or linear regression models, where the hypothesis and the assumptions are made explicit, the models are interpretable and usually data is not limiting, to models that can capture arbitrary functional relationships, such as neural networks, which are powerful, have large numbers of parameters that are hard to interpret, and are usually poorly constrained by data. In biology, models are often graphical schemes resembling engineering block diagrams that summarize biological processes and their interactions. It is hard to use these models for quantitative predictions (except for certain cases where these graphical models can be mapped to, e.g., systems of differential equations for chemical kinetics but which still feature lots of typically unknown parameters). On the other hand, these models make explicit and experimentally testable qualitative predictions, e.g., what happens when a process is disrupted or a component removed from a system. Lastly, a typical model in natural sciences is, for instance, a "natural law", say, Newton's law of gravitation, with $F_g = Gm_1m_2/r^2$. While this is a very simple equation, the key to its development has been to identify what are the relevant quantities (masses, forces, distance) that enter the equation, and identify how these quantities fit with the consistency requirements from other physical laws. To "infer" or learn the law of gravity, it was also crucial to abstract away from the raw data (e.g., from data on the proverbial measurements of falling objects from the leaning tower of Pisa) the systematic effects of air friction, which dominates our everyday experience. Once learned, however, this equation provides tremendous generalization performance, which extends across spatial scales from particles invisible to the eye to stellar objects. Again, these are not typical, but rather extremal examples from each discipline, chosen to illustrate the variability of modeling approaches; the situation in reality is more mixed.

Lastly, one should also consider the appropriate scale at which the modeling and analysis should take place. Here one can differentiate between **mechanistically detailed models** and **phenomenological models**; in the first class, the observed results are explained in terms of certain "elementary processes" given by prior / theoretical knowledge (e.g., molecular dynamics, chemical reactions, etc), usually parametrized by many parameters. In contrast, in the phenomenological approach microscopic details are often abstracted away to get an effective model with a small number of parameters. Sometimes, the transition from the microscopic into effective model is done formally (as in the application of statistical physics or renormalization group apparatus). Typical examples (going from physical sciences towards life sciences) include: the mechanics of colliding hard spheres in a box (detailed mechanical model) can be understood as ideal gas (a course grained, "thermodynamic" model); metal atoms in a crystal lattice can be understood as giving rise to Ohm's law or Ising magnetism; all-atom protein dynamics can be understood phenomenologically as a network of stochastic conformational transitions; and a detailed, ODE-based model of neural spike generation (Hodgkin-Huxley model of nonlinear differential equations with 20 parameters) can be simplified into a leaky integrate-and-fire model of neural spiking (one equation with a few parameters). Often, phenomenological models can be more precise or predictive for the *selected* question of interest, but their parameters may be harder to map to experimentally controllable quantities than in case of the microscopic models.

An interesting biological example of the two scales of description is provided by the problem of the so-called Planar Cell Polarity in epithelial tissues. An epithelium is a 2D tissue (like the skin), in which cells have a well-defined top (apical) and bottom (basal) surface. Often, on such tissues, a secondary macroscopic direction emerges: a studied case has been in the fruit fly *Drosophila*, where each epithelial cell grows a hair, and all the hairs share a common orientation in the plane perpendicular to the apical-basal normal. The question of how all the cells break the symmetry and decide on a common direction for the hairs has deserved substantial modeling effort. Burak and Shraiman (*PLOS Comput Biol* **5:** e1000628 (2009)) present a phenomenological, two-equation model with ~ 5 parameters for the process, to be contrasted with the microscopically-detailed model by Amonlirdviman et al (*Science* **307:** 423 (2005)), which is specified by a system of reaction-diffusion equations with roughly 40 parameters. The issue with the latter approach is not only the fact that most of these parameters are not known and exploring robustness within such a large space is hard, but that the actual list of processes that give rise to the equations is not known to be correct or complete. On the positive side, predicting the effects of mutations in detailed models is trivial, since most mutations simply correspond to running the model with certain chemical species or reactions missing. In contrast, a single phenomenological model can encompass many microscopic models and distills down the essential components needed to explain the phenomenon, but may be harder to link to accessible experimental perturbations.

Taken together, these – very broad and philosophical – considerations nevertheless should motivate you to think very carefully about the following questions: *Why you are building a model in the first place? What do you hope to extract and learn from it? What kind of the model is most appropriate for the question at hand, or does even specifying the model mathematically already raise interesting new questions?* These considerations will strongly influence the best approach to take.

2 Basic inspection of the data

A segment of the roughly 100 s worth of data in `trace1` is shown in Fig 1. A quick look suggests that the trace can be understood as a superposition of slow baseline fluctuations (happening on a ~ 0.5 s timescale), large excursions in voltage that on the plot appear almost instantaneous and likely correspond to neural spikes that we wish to find, and the residual, fast timescale fluctuation. In the course of the lectures we will make these intuitions more precise.

Always make sure to do a "sanity check" of the data: look at the beginning and end of traces in case something unusual happens there (in case of spiking data, perhaps the data acquisition was already / still going on while the retina was not being stimulated), check for gaps in the data (e.g., due to experimental failures etc). This already requires you to have some understanding of how the data was collected and how you should interpret it (does a segment of zeros correspond to something that could actually have happened or is it a sign of experimental failure?). Sometimes different parts of the data are not of the same quality (here, the retina could be slowly dying already towards the end of the experiment), which needs to be taken into account. All these considerations are not restricted to this example: *It makes sense to invest some time in the beginning and understand the data collection process well before plugging the data into any kind of analysis!*

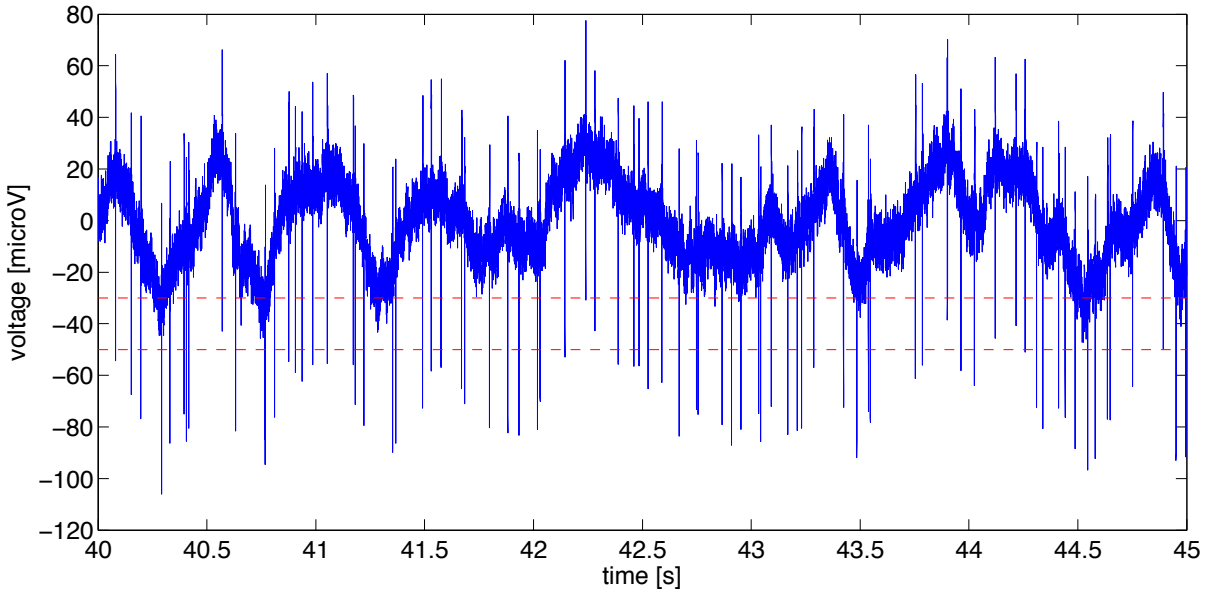

Figure 1: A 5 s example trace of the retinal recording on the multielectrode array (MEA).

2.1 Absolute vs arbitrary units

Check what are the units in which the data is gathered: are values reported in physical units or some "arbitrary units" [a.u.]? In particular, do the absolute zero and the dynamic range have any intrinsic meaning or are they something that can be chosen arbitarily by "data centering" (i.e., subtracting the mean from the time series to center the data at 0) and "normalization" (e.g., by dividing the data by maximum or standard deviation over the time series, to get a unitless measurement)? These considerations can have very direct and practical effects. For example, if the same experiment is repeated many times (or we are comparing the signal on multiple electrodes as in our example), then — if what is being measured are really physical quantities — the signals from different electrodes or repeats should be comparable directly and plotted on the same scale. It is precisely to guarantee the same absolute zero that we ground the electrical measurements to the same reference. If the units and zero are arbitrary, i.e., a consequence of experimental measurement that can vary from repeat to repeat rather than being intrinsic to the system being measured, then the data needs to be normalized before comparing across repeats. This is very prevalent in life science, where the measurements are often quantitative (i.e., give numerical values) but not physical (numerical values don't have absolute units, but are only a proxy for some underlying physical quantity). For example, we are often interested in the protein concentration (which would have units of, e.g., nanomole per liter) but measure instead the average light intensity of a fluorescently tagged protein. The measured quantity depends not only on concentration, which is intrinsic to the biological system, but also on the microscopy settings, staining or imaging protocols, etc.

An instructive example of how important data normalization can be to scientific conclusions is provided by the measurement of Bicoid morphogen concentration gradients in developing fruit fly (*Drosophila melanogaster*) embryos. Spatial gradients of morphogens are important, since they instruct cells of a multicellular organism to differentiate into different tissues; the precision of these concentration gradients limits how precisely the cells can differentiate. In Houchmandzadeh et al, *Nature* **415:** 798 (2002) the authors measure the concentration gradients, $c_i(x)$, of Bicoid morphogen in many embryos, $i = 1, \ldots, N$, as a function of the spatial coordinate, x, that extends from anterior to posterior. Since the measurements are not of a direct physical quantity but consist of immunostaining data supposedly linearly related to the concentration, the authors first normalize the gradients before plotting them on top of each other: they subtracted the offset from every gradient so that at its minimum the gradient was zero, and divided each gradient by an amplitude so that at the maximum it was one. Then they computed the standard deviation over these gradients to quantify their embryo-to-embryo reproducibility or precision (Fig 2, left), and concluded that this precision is too small to explain the precision by which cells later decide about their differentiation fate, implying the presence of unknown other biological signals in the system. Later, Gregor et al, *Cell* **130:** 153 (2007) remeasured the same gradients and reanalyzed also the original data from Houchmandzadeh et al, by normalizing the gradients not to minimum and maximum but so that they overlapped as well as possible in the χ^2 sense. Now, the aligned gradients were much less variable in the middle, where cells decide on their fate, implying that no extra signal is needed (Fig 2, right). In retrospect, it is clear that the initial normalization to min/max squeezed out, by construction, all the variability at low and high x, and squeezed it into the middle; the χ^2 normalization by Gregor et al, in contrast, made no such biased choice. This has two implications: (i) data normalization / alignment can be very important and needs to be thought through carefully; (ii) if the measurement was of the actual physical quantity (concentration) or the absolute calibration of the staining was performed, no normalizations / alignments would be needed and all profiles could simply be compared to each other directly. This approach, however, requires much greater experimental effort; cf. Dubuis et al, *Molecular Systems Biology* **9:** 639 (2013).

Remember that similar types of considerations apply to *any* data, not just the examples from life sciences discussed here – be it genomic sequences, physics measurements, images taken from the internet as training data for image classifiers, etc. The problem of offset and scale is very generic: when referees or professors grade the students, they do so on a fixed scale (say 1 to 10), but different people grade by using different dynamic ranges (some squeeze all the grades on a scale between 8 and 10, whereas some use the whole dynamic range): how does one normalize for that? Similarly, when digitalizing signals, analog values are packed into, e.g., a discrete set of 10-bit digital values $(0, \ldots, 1023)$, losing the unit, offset and the absolute value of the dynamic range in the process, which have to be recorded separately. This is relevant, for instance, for digital images, which do not record physical quantities (light fluxes). The first step in analyzing the data is to understand such issues of data representation and biases in data collection, so that you can later asses their impact on your conclusions.

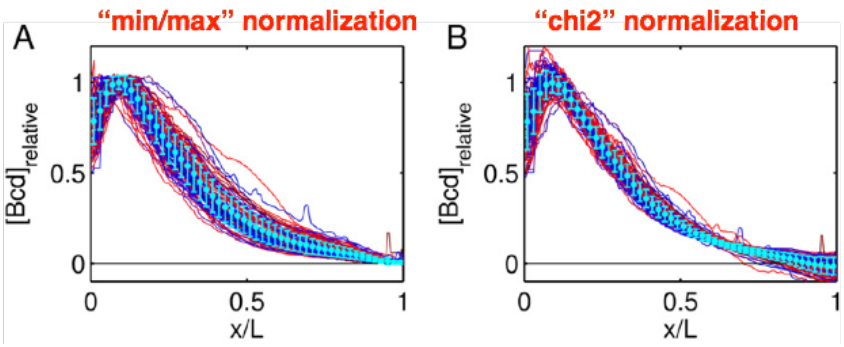

Figure 2: Comparison between normalizing each Bicoid gradient, [Bcd](x), by its minimum and maximum (to align all gradients between 0 and 1) in (A), vs aligning the gradients by adjusting an additive offset and multiplicative scale per gradient so that they best align to each other in a χ^2-sense in (B). In **(A)**, most of the variability between gradients (cyan errorbars) is squeezed from $x = 0$ and $x = 1$ towards close to the middle, at $x \sim 0.5$, whereas in **(B)** the gradients are very precise in the middle. See the corresponding text box for references and details; figure reproduced from Gregor et al, *Cell* **130:** 153 (2007).

3 Descriptive statistics

Suppose you are given N samples of some data that we will jointly denote as $\mathcal{D} = \{\mathbf{x}_t\}$, with $t = 1, \ldots, N$ indexing the samples. For now, let's think of $\mathbf{x}$ as real-valued data of dimension D, i.e., $\mathbf{x} \in \mathbb{R}^D$. In our case of the `trace1` voltage trace, the situation is simple if we view voltages as $D = 1$ dimensional scalars, and if we don't worry about the correlation between successive time points (more formally, here we will start the discussion assuming that the samples are *independent and identically distributed* (IID) – that means that the probability of observing a certain sample is independent of all the other observed samples, and that all samples are generated by the same distribution). Now, for our timeseries data, there obviously *are* temporal correlations and the IID assumption is thus patently false, but one can nevertheless look at the marginal statistics, as below (and be on a lookout about what could go wrong when making unsubstantiated IID assumptions).

We will introduce some basic notions here:

- **Empirical distribution.** $P_{\text{emp}}(\mathbf{x}) = \frac{1}{N}\sum_{t=1}^{N} \delta(\mathbf{x} - \mathbf{x}_t)$, where $\delta(\mathbf{x})$ is a Dirac-delta function, with the property that $\int_{-\infty}^{\infty} d\mathbf{x}\ \delta(\mathbf{x}) = 1$. Empirical distribution is a way to represent the given samples in $\mathcal{D}$ as a distribution composed of N infinitely sharp peaks, each corresponding to one observation. For IID data, this distribution contains equal information as the list of samples $\mathcal{D}$.

- **"True" data generating distribution** $P(\mathbf{x})$. If we repeated the same experiment or data collection, this is the distribution we would be drawing the new samples from (each time obtaining a different empirical distribution). We usually don't know P, but would like to learn about its properties from the data $\mathcal{D}$. Sometimes we make assumptions about the form of this distribution (e.g., by assuming it is a Gaussian) but treat its parameters (e.g., mean and variance) as unknowns to be determined from data; in this case we talk about *parametric statistics*. In other cases we don't want to assume the functional form for P, but nevertheless reason about it; in this case we talk about *non-parametric statistics*. For IID data, finding the true generating distribution would give us the complete statistical

model of the process of interest.

- **Sample statistic** is some function that can be applied to data $\mathcal{D}$. Descriptive statistics are quantities that summarize or describe the data; estimators are functions of the data that attempt to extract or approximate the parameters of the generating distributions from data. For example, in $D = 1$

$$\bar{x} = \frac{1}{N} \sum_{t=1}^{N} x_t \tag{1}$$

is a descriptive statistic that corresponds to the *mean* of the data. In case I believed that the generating distribution is Gaussian with the mean parameter m (i.e., $\int dx\, xP(x) = m$), the formula in Eq (1), solely viewed as a function of the data, would also become the *estimator* for the mean m of the Gaussian distribution.

- **Bias and variance of the estimators.** While statistics can be any functions of the data, when they are used for estimation of the parameters of the generating distributions, there are two desired properties that good estimators should share. First, they should be *unbiased*:

$$\langle \bar{x} - m \rangle = 0, \tag{2}$$

i.e., on our Gaussian example, the estimator of Eq (1) averaged over many draws from the (by assumption Gaussian) distribution $P(x)$, will be equal to the true mean, m; here, the averaging over draws is denoted by brackets, $\langle \cdot \rangle$. Second, the estimator should not only be unbiased, but also efficient; intuitively, it should give the smallest possible variance around the true parameter value given some number of samples, N. In statistics, the study of good estimators is a large subject that we won't go into, and efficient estiamtors of course depend on what parameter is being estimated. But a general rule of thumb is that if samples are IID (independent, identically distributed), the variance of efficient estimators should scale as

$$\mathrm{Var}[\bar{x}] \sim N^{-1}. \tag{3}$$

Thus, the "error bar" on statistical estimates from N IID samples should be expected to generically decrease as $1/\sqrt{N}$. The simplest example of this scaling is shown in Fig 3. Make sure that you understand well the difference between a statistic (say, for the mean), its "error bar" (standard error of the mean; SEM), and in the Gaussian case the standard deviation of the Gaussian distribution. Standard error is a property of a particular estimator which should shrink with more samples; standard deviation of the underlying Gaussian distribution is a parameter of that distribution that is independent of the number of samples.

- **Moments of the distribution.** Other useful descriptive statistics that we give as examples here are the variance, $\sigma^2 = \frac{1}{N-1} \sum_{t=1}^{N} (x_t - \bar{x})^2$, and higher order moments. If you wonder about the denominator, $(N-1)$, in formula for variance, I side with the classic textbook *Numerical Recipes in C* by Flannery et al: any standard statistical textbook will explain why unbiased variance estimate when the mean is *a priori* unknown uses $(N-1)$ instead of N, but if this distinction is important for your data, you are anyway most likely making dangerous inferences in a data-limited regime. Higher-order moments of the distribution are defined as:

$$M_k = \frac{1}{N} \sum_{t=1}^{N} \frac{(x_t - \bar{x})^k}{\sigma^k}, \tag{4}$$

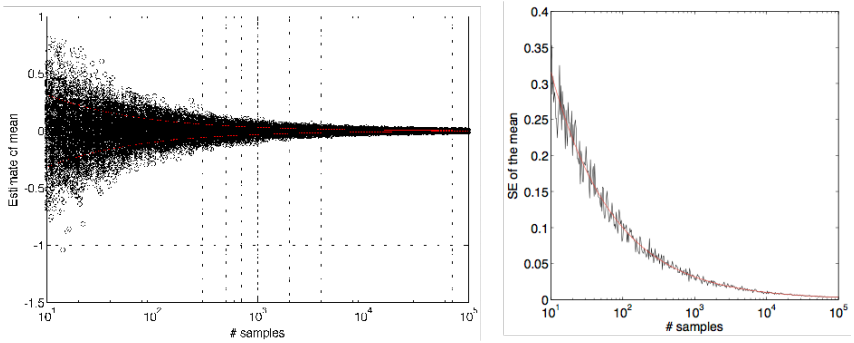

Figure 3: Individual data points are drawn IID from the standard Gaussian distribution with mean $m = 0$ and variance $\sigma^2 = 1$. **Left.** Different number of samples N are used to compute estimates of the mean according to Eq (1); each estimate is plotted as black circle. Since the estimator is unbiased, it converges to zero with increasing N, as desired. The standard deviation of the estimator is shown as the red envelope. **Right.** Standard deviation of the mean estimate, also known as the standard error of the mean (SEM), is plotted in black; superposed in red is the theoretical curve, $\mathrm{Std}[\bar{x}] = \sigma/\sqrt{N}$, a well-known formula for SEM for Gaussian distributions.

where the deviation from the mean is divided by powers of σ so that the resulting moment, M_k, is dimensionless. The well-known cases for $k = 3$ (skewness) and $k = 4$ (kurtosis) we will encounter later in the course.

4 Histograms and probability distributions

Note that for IID data all statistics (mean, moments, other functions) can be seen—and mathematically written—as functions of the empirical distribution, $P_{\mathrm{emp}}(\mathbf{x})$. A lot of information should thus be contained in a suitable representation and visualization of that distribution. To this end, we usually construct and plot raw histograms, by choosing some particular binning of the x variable (e.g., defined by the bin boundaries $x_0 \leq x_1 \leq \cdots \leq x_L$) and then plotting the number of samples, X_j, that fall into bin j. Mathematically, this corresponds to plotting

$$X_j = \sum_t (x^t \geq x_j) \wedge (x^t < x_{j+1}), \tag{5}$$

where the formula on the right-hand side simply gives a 1 if the sample falls between two bin boundaries, x_j and x_{j+1}, and 0 otherwise. This is done for `trace1` data for two different choices of data binning in Fig 4. Note that if data is intrinsically discrete, then no explicit binning is required. For continuous data, however, raw histograms with a certain choice of binning scheme give rise to several considerations:

1. Density estimation (estimating true $P(x)$ from finite number of samples for continuous x) is a hard problem that can only be solved given prior assumptions on P (e.g., on its smoothness). The simplest methods tend to convolve each data point, x^t, with a narrow Gaussian to smoothen the data and averaging across all data points, in a process known as kernel density estimation (KDE). This is beyond the scope of this course, but is a standard approach that has also been a topic in Methods of Data Analysis course.

2. Constructing raw histograms gives an obvious sense of statistical power in individual bins (since we see directly the number of samples per bin); on the other hand, the count values

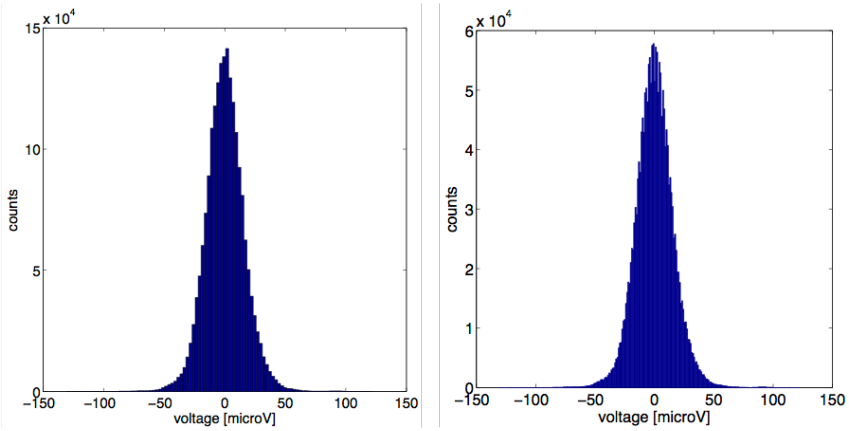

Figure 4: Raw histograms of data in `trace1`, with 100 bins **(left)** and 1000 bins **(right)**, uniformly spaced between the maximal and minimal observed values in the full trace. Interestingly, the histogram at right reveals curious "throughs" between the bins: on closer inspection it turns out that our data is not really continuous but discrete (you can check that there are only 2022 different levels in the two-million sample dataset), due to the digitalization of extracellular voltages by the recording equipment. This is a typical, rather than exceptional case, whereever physical data is captured by devices with limited resolution. For many analyses this does not matter, but for some it can: so check if your continuous-looking data is truly discrete.

are not directly comparable across the same data histogrammed using different binning widths (cf. Fig 4 left and right).

3. If bins are unequally sized, raw histograms are very hard to interpret.

4.1 Estimating PDFs

To show a distribution in a way that is independent of binning, construct an estimate of the (normalized) probability density function (PDF), $\hat{P}$, by normalizing the histogram so that:

$$\int dx\ \hat{P}(x) = 1 \tag{6}$$

If the histogram is constructed using uniform binning, i.e., $\Delta = x_1 - x_0 = x_2 - x_1 = \cdots = x_L - x_{L-1}$ with bin counts X_j in different bins, one can approximate the $\hat{P}$ as

$$\hat{P}(x_j) = \frac{1}{\Delta}\frac{X_j}{\sum_k X_k}. \tag{7}$$

Note that both histograms and estimated PDFs, Eqs (5-7), are just particular classes of descriptive statistics of the data, so that general considerations about estimation apply to them as well as to the more common statistics such as the moments.

There are a few points worth making about the choice of binning:

- You want to choose the number of bins that is high enough to capture the details of the distribution (i.e., the value of the distribution should not change much between two neighboring bins), but still small enough that each bin contains the number of counts $X_j \gg 1$, so that the empirical probability can be well estimated. There is no universal rule for doing this, and obviously there is a tradeoff between the resolution and sampling power depending on the choice of bin width, Δ.

- In the limit of $\Delta \to 0$, the estimate of the PDF, $\hat{P}$, will converge to the empirical distribution, P_{emp}, defined above. No information about data $\mathcal{D}$ is lost by such fine binning, but it provides a poor estimate to the true generating distribution, $P(x)$, since it is completely overfit to the samples.

- There are a number of small details about putting Eq (7) into practice: given that we used a discrete binning scheme to assign data points to bins, and have normalized those bins, to what value of x should the estimated value of the PDF be assigned (i.e., to the left boundary of the bin, the right one, the center, etc)? With fine enough binning this usually does not matter and the only advice is to keep in mind that the exact values depend on this consideration, but it may matter in the case of non-uniform binning (see below).

- You may choose bins that are not uniformly spaced. One popular choice is adaptive binning where bin boundaries are selected such that each bin roughly contains equal number of counts. This ensures the same statistical power in each bin, but could mean that bins differ widely in size: the bins are very densely distributed close to the peak of the distribution, and are very rare and large in the tails. In this case it does matter how bin centers are defined in the tails; for example, see Fig 5 middle right. Note that in the case of nonuniform binning of continuous variables, it is essential to plot normalized PDFs, since raw histograms no longer can be graphically interpreted. Such adaptive binning schemes are useful for higher-dimensional histograms which otherwise suffer from the curse of dimensionality. For D dimensional data uniform binning often results in large numbers of bins in the tails of the high-dimensional distribution—indeed, maybe the majority of all bins—being completely empty, which can complicate certain analyses. Instead, one can adaptively make the tail bins very large: this trades resolution in the tails for statistical power.

An example PDF estimate for our data is shown in Fig 5 at left, created from a normalized 100-bin histogram. A necessary step in inspecting the empirical distributions is to always plot them on a logarithmic scale. This quickly reveals interesting structure in the tails, as shown in Fig 5 at middle left: we can guess that the excess of low voltage excursions are exactly due to the spiking events in our electrode trace, and a naive way to set a threshold for discriminating spikes from background would be to set the threshold at the minimum separating the bulk of the distribution from the low voltage peak. This threshold would roughly correspond one of the red dashed lines in Fig 1, but you should quickly be able to convince yourself that that simple criterion will miss some of the spikes and thus generate false negatives.

4.2 Cumulative distributions

An alternative way of showing the histogram that is less dependent on the binning is by means of the cumulative density function (CDF), formally defined as:

$$C(x) = \int_{-\infty}^{x} dx' \; P(x'). \tag{8}$$

Given N samples, the estimation can be done in a nearly binning-free way, by directly using the empirical distribution, $P_{\text{emp}}(x)$, in Eq (8). For example, the following Matlab script will effectively plot a CDF estimate of `data` (make sure you understand how this works; the plot is shown in Fig 5 at right):

```
plot(sort(data,'ascend'),(1:numel(data))./numel(data),'k.');
```

CDFs are useful in particular if the underlying true distribution is mixed, i.e., contains a continuous component as well as point probability masses; in this case, no binning scheme for PDF estimation is convenient, but the CDF is well-behaved, making large discrete steps at the locations of point masses (when CDF are estimated from finite data as above, the CDF is anyway composed of unitary steps on the y axis that have a magnitude of the inverse number of samples). Note that CDFs are also unit-free and range from 0 (at minimum of x or formally at $x = -\infty$) to 1 (at maximum of x or formally at $x = \infty$). If used to compare multiple distributions on a CDF plot, the clearest comparison is around the median, but it is hardest to compare visually the tails; for that, instead, it is often convenient to plot $1 - C(x)$ on the logarithmic scale. Kolmogorov-Smirnov test that we will introduce later to compare distributions also operates on the CDFs and similarly has highest sensitivity around the median.

An advantage of the CDF for visualization is the ease with which we can directly read off the quantile statistics. In comparison with central statistics / moments, defined by mean and variance formulas and by Eq (4), quantile statistics ask about values on the x axis of the CDF at which the cumulative data crosses some threshold probability; mathematically, quantile Q_θ for the threshold θ is the value where:

$$\theta = \int_{-\infty}^{Q_\theta} dx' \; P(x') = C(Q_\theta). \tag{9}$$

For instance, median (a special name for the $\theta = 0.5$ quantile) is the value at which the CDF crosses 0.5; similarly, one can look at the points where the CDF crosses 0.25 and 0.75, and define the *interquartile range* (IQR) as the range of x-values that contain 50% of the total probability weight (that is, the range between $Q_{0.25}$ and $Q_{0.75}$). This is an example of robust statistic, since its value is independent of the presence of small number of extreme outliers—in contrast with, say, variance. A simple way to compare two distributions graphically, including in the tails, is to make a quantile-quantile plot: each point represents the value of a particular quantile in the first distribution (to be shown on x-axis) vs the value of the same quantile the second distribution (on y-axis). Another robust statistic for quantifying the spread of the distribution (although not a quantile statistic) is the *mean absolute deviation*,

$$AD = \frac{1}{N} \sum_{t=1}^{N} |x^t - \bar{x}|. \tag{10}$$

4.3 Gaussian distributions and z-scoring

Returning now to normalized PDFs, one important feature of plotting a normalized PDF on the log plot is that you can easily compare it with standard distributions, e.g., the normal distribution with mean and variance selected to match the empirical estimate from the data (you should know the Gaussian distribution by heart, including the normalization factor):

$$\mathcal{N}(x; \bar{x}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\bar{x})^2}{\sigma^2}}. \tag{11}$$

Such a comparison is shown in Fig 5 in the middle left; together with the ability to estimate the error bars on our PDF estimates that we will discuss next, this brings us closer to being able to assess the significance of deviations between data and model PDFs, such as the normal distribution. (As a side note: despite the extremely simple nature of these suggested steps, I

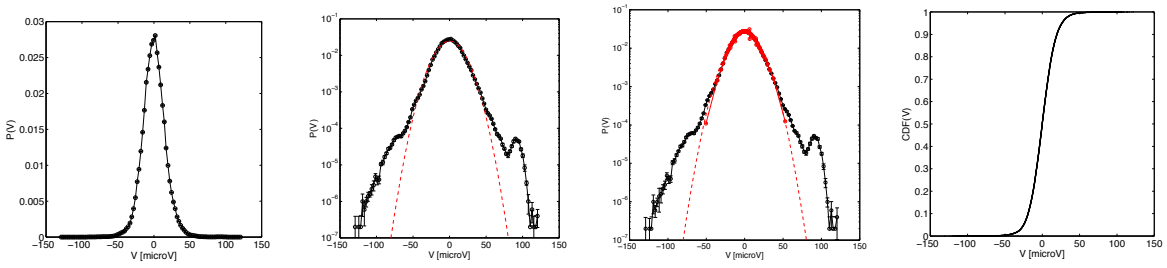

Figure 5: **Left.** Histogram of `trace1` data constructed over 100 equidistant bins and normalized into an estimate of the PDF, as in Eq (7). Note that PDFs have units, in this case $1/\mu V$, since they have to integrate to 1. **Middle left.** Logarithmic plot of the same PDF and comparison to the Gaussian distribution of matching mean and variance (dashed red line) plotted directly from Eq (11), shows the excess weight in the tails of the data. A useful reference to remember when interpreting PDF estimates is the value of the PDF that corresponds to seeing the data point once in the whole dataset (e.g., the left-most extremal values at the logarithmic plot). **Middle right.** Comparison of the distribution constructed from equally spaced bins (black) with the distribution constructed using 100 adaptive bins selected such that the number of samples per bin is equal (and the resulting error bars, see below, are also equal). This provides a good agreement near the mode of the distribution, but can lose details in the tail, where the bins span large segments of the x-axis. **Right.** Cumulative distribution of the same data can be constructed without binning and is useful for reading off quantile statistics.

regularly see students plot raw histograms also for continuous data and have trouble plotting model distributions on top of their data correctly—make sure that you can do all these steps routinely and automatically.)

At this point it is useful to also write down the generalization of the Gaussian distribution to the multivariate case,

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{C}) = (2\pi)^{-D/2} |\mathbf{C}|^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T \mathbf{C}^{-1} (\mathbf{x}-\bar{\mathbf{x}})}, \tag{12}$$

where $\mathbf{C}$ is the $D \times D$ covariance matrix and $\bar{\mathbf{x}}$ is the D-dimensional mean column vector. A few basic facts to remember about Gaussian distributions:

- They are symmetric around the mean and have a single peak, so that the mode, median, and mean coincide. They have a strong central tendency, i.e., a faster-than-exponential drop in probability with the distance from the mean.
- They are fully specified by the first (mean) and second (covariance) moments. All higher moments, M_k, can be expressed in terms of the low-order moments. All odd-order moments ($k \geq 3$, k is odd) are zero due to symmetry. For one-dimensional case:

$$M_k = \begin{cases} 0 & \text{if } k \text{ is odd} \\ \sigma^k (k-1)!! & \text{if } k \text{ is even} \end{cases} \tag{13}$$

- Integrals of the Gaussian distributions are analytically tractable; in the multi-variate case, integrating over any subset of components of $\mathbf{x}$ also results in a Gaussian (marginal) distribution.
- In 1D, interval $[\bar{x} - \sigma, \bar{x} + \sigma]$ contains $\sim 68\%$ of the total PDF weight, and $[\bar{x} - 2\sigma, \bar{x} + 2\sigma]$ contains $\sim 95\%$ of the weight.

- They are the most random distributions for continuous real-valued variables with a given mean and covariance; formally, they are distributions that maximize the entropy, $S[P] = -\int dx\ P(x) \log P(x)$, with given first- and second-order moments.
- Gaussian distributions are limiting distributions for sums of IID random variables that are, each, drawn from an underlying distribution with a finite mean and variance. In that case, the mean (variance) of the sum is the sum of the means (variances) of the underlying variables. This is known as the Central Limit Theorem (CLT), which also underlies many arguments about the $N^{-1/2}$ scaling of the standard error of various efficient estimators that we mentioned previously.

Figure 5 should convince you that how we visualize data is important: plotting on a linear scale reveals no surprising features, while plotting the same data on the logarithmic scale does. As illustrated by the caveat example below, the "visualization" should not be taken too far, to manipulate scientific conclusions.

An example of data analysis where data visualization probably played a critical role for advancing a dubious scientific statement is in Bar-Even et al, *Nature Genet* **38:** 636 (2006). This paper, appearing in a reputable journal, was published there mainly due to its title claim, that "Noise in protein expression scales with natural protein abundance". More precisely, the authors implied that the variance in protein expression normalized by the squared mean expression (i.e., the coefficient of variation, CV, squared) is related to the mean protein expression linearly on a log-log plot. To show that linearity, the authors scatter-plotted the two quantities of interest on the log-log plot across many genes and many conditions in Figure 2 of their paper. These points involve a lot of scatter: so the authors first excluded the region of high and low abundance at arbitrary thresholds ("...chosen by eyeballing" from the Methods section of the paper), and then iteratively linearly fit the data, excluding outliers *and plotting those outliers in light gray in Figure 2.* This way of visualizing the points included (and arbitrarily excluded) from the fit plays well with the human visual system so as to clearly imply a linear relationship where there hardly is one.

As in the example of morphogen gradients in the fruit fly development where the gradients from different embyos had to be "normalized" and compared, we often need to compare distributions of a certain quantity, either across repeats of the same experiment or accumulated across some related phenomena. If such distributions differ in mean or variance, comparing their shape is difficult; a useful approach is to "z-score" the values, i.e., transform data x^t into:

$$z^t = \frac{x^t - \bar{x}}{\sigma_x}. \tag{14}$$

This subtracts from each data point the mean over all data and divides by the standard deviation. If the actual data $\mathcal{D}$ were drawn from a Gaussian distribution, the resulting distribution of z should be the standard zero-mean unit-variance Gaussian. If not, we can show the actual distribution of z-scored values and compare it across conditions to see if the conditions *only* differed in mean/variance, or also in the full distribution shape (i.e., higher-order moments).

Homework 1. How would you propose to generalize "z-scoring" (e.g., subtraction of the mean, normalization by the standard deviation) from the 1D case to the multivariate case, where $\mathbf{x} \in \mathbb{R}^D$? Generate a synthetic dataset with 10^4 data points drawn from bivariate Gaussian distribution with different means and standard deviations for both variables (e.g., $\bar{x}_1 = 10$, $\bar{x}_2 = -1$, and $\sigma_1 = 2$, $\sigma_2 = 1$), and for three different correlation coefficients (e.g., $\rho = 0, 0.5, 0.95$). Does your proposed transformation alter the covariance matrix?

Sometimes, z-scoring can lead to a case where distributions of values generated by similar, but not identical, processes (that differ in mean and variance) collapse onto a universal distribution. This is referred to as a "data collapse," which can indicate several important aspects about the data. One possibility is that the data acquisition method (experimental protocol) induces variation in mean/variance from repeat to repeat of the experiment and z-scoring is a way to eliminate that variation. A more interesting possibility is that the underlying process that generates the data is universal (hence has the same z-scored distribution) but is modulated by some low-dimensional factors that subsequently affect the mean and variance. An example of this is the distribution of light intensities across pixels of natural images: the distribution has a universal shape given by the laws of optics, statistics of the objects in our visual environment, and their reflectivity. This universal distribution is modulated strongly in its mean simply by the fact that we observe nature at different overall amounts of light illumination (due to variation in time-of-day etc). A less intuitive but no less intriguing example is the recent work of Brenner et al, *arxiv.org:1503.01046*, where the authors showed that the distribution of protein amount expressed from different promoters in *Escherichia coli* bacterium, accumulated from single-cell measurements across time, collapses onto a universal distribution when z-scored. The authors claim that this is not due to experimental variability, implying that the gene expression could be described by a (unknown) universal stochastic process that can be modulated by a single extra variable which jointly influences its mean and variance.

Homework 2. Plot the voltage signal $x(t)$ `trace1` from the microelectrode array and visually examine it. Spikes are very fast downward voltage excursions (sometimes reaching to $\sim -100\ \mu V$), followed by a small overshoot (zoom in to a few spikes to see how they typically look like).

To get some sense of the signal, plot a probability distribution function (properly normalized, so that $\int dx P(x) = 1$), of $x(t)$. Estimate the error bars on the PDF by splitting the data multiple times into halves and compute the SD over PDF estimates constructed from halves of the data. Is there any obvious feature for negative voltages in the histogram where you could draw a threshold to recognize the spikes easily? To identify the spikes, you can set a threshold. Scan a range of thresholds, from $-70\ \mu V$ and $-30\ \mu V$; whenever the signal crosses the threshold in a downward direction (please pay attention to this definition!), identify a putative spike, and plot the number of spikes as a function of the threshold. By examining the trace in detail, can you claim that any specific threshold is a good choice for spike detection?

The problem seems to be in slow baseline fluctuations in the recorded voltage, $x(t)$, an intuition that could be made precise using spectral (Fourier) methods. For now, estimate the slow baseline variation by smoothing the original signal over the timescale of $T = 100$ consecutive time points. The simplest way to do this is to define a new time series, $\tilde{x}(t)$, such that each value of $\tilde{x}$ corresponds to a moving average of the original time series, e.g.,

$$\tilde{x}(t) = \frac{1}{T} \sum_{t'=t-T/2+1}^{t+T/2} x(t') \tag{15}$$

Subtract this slowly varying component from the original signal. Do you see spikes more clearly now? Plot the number of detected spikes as a function of the threshold, for thresholds between -75 μV and -30 μV. How dependent is the number of spikes on the threshold now?

Now we will construct a curve similar to an often-used performance measure for binary classification: the "receiver-operating characteristic" or an ROC curve[a]. In classification scenarios, one often uses a threshold to determine whether some event belongs to a particular class (is "positive" or P, when, e.g., above the threshold) or not (is "negative" or N, when, e.g., below the threshold). If we posses the "ground truth", that is, we know with certainty how each event should be assigned to P and N categories, we can compare the threshold-based classifier with this ground truth, by computing two quantities: the "true positive rate" (TP) and the "false positive rate" (FP). These quantities obviously depend on the value of the threshold, and when plotted one against the other as a function of that threshold, we obtain an ROC curve.

In our case, to see how important it is to subtract the slowly varying baseline, let's start by picking a particular threshold of -50 μV. Then, on the baseline-subtracted trace, identify all the spikes and declare them to be correct identifications ("ground truth"). Now, go back to to the non-baseline subtracted case, and identify the spikes using different threshold values. For every spike identified on the non-baseline-subtracted trace, you can ask whether that was a true detection or not compared to your ground truth. Plot the TP and FP rates as a function of the threshold – this is a plot closely analogous to the ROC curve.

Why is this plot is closely analogous, but not exactly equal to standard ROC curve, and what is the reason for the difference? What kind of shape on TP vs FP plot corresponds to good classification performance? Related to that, check out what AOC means and how it relates to the ROC curve – this is one of the standard measures of classifier performance.

[a] Check out the Wikipedia page for ROC curve.

5 Estimating error bars

In this section we will describe how to obtain error bars on various statistics, including on the PDF estimates. Until now, we only mentioned the standard error of the mean (SEM), which is connected to the variance of the underlying Gaussian distribution (generic, since it emerges under the central limit theorem) by a known formula; and we have discussed the *scaling* of typical errors for statistics (as inverse square root of the number of *independent* samples). For

quantitative estimates, however, we need **(i)** to determine not only the scaling, but the actual numerical value for the error bar given some number of samples; **(ii)** to treat non-IID samples.

A non-parametric way to estimate error bars is by means of *bootstrap resampling*. To illustrate the idea on a concrete example, let's consider the simplest estimation problem. Imagine we have collected N total samples and each sample is either white or black. In the dataset, there are M black samples, and we would like to estimate the frequency of black samples. You can see this as the simplest problem of histogram / probability distribution estimation: we are estimating probabilities of white vs black, i.e., a histogram or distribution that has support for two different values. Our considerations will generalize to examples with multiple bins.

Clearly, $\hat{p} = M/N$ is the empirical estimate for the probability of obtaining a black sample. But what is the error on that estimate? Well, if we considered a repeat experiment where we were again collecting N samples, and with probability $\hat{p}$ each would be black, we have binomial expectation for the number of black samples Z. On average, we expect $\hat{p}N$ black balls, and the variance in repeat observations in the number of black Z would be $\mathrm{Var}[Z] = N\hat{p}(1-\hat{p})$. Since on repeat experiments I would estimate probability of black as $q = Z/N = \hat{p}$, it follows that $\mathrm{Var}[q] = \mathrm{Var}[Z]/N^2$, and thus $\mathrm{Std}[q] = \frac{1}{\sqrt{N}}\sqrt{\hat{p}(1-\hat{p})}$. In case black is rare, $\hat{p} \ll 1$ (this is the typical regime relevant for constructing empirical PDFs, where we select binning such that the number of samples in any one bin is a small fraction of the total number of samples), and then the standard deviation of my estimate of probability q (which I take to be the error on the empirical estimate $\hat{p}$):

$$\mathrm{Std}[\hat{p}] = \mathrm{Std}[q] \approx \sqrt{\frac{\hat{p}}{N}}. \tag{16}$$

How would boostrapping work in this case, where the data is identically distributed and independent?

One version of bootstrap would proceed as follows:

1. Subsample the data $\mathcal{D}$, by selecting randomly only half of the data points, to get a subset $\mathcal{D}_\mu$ of size $N/2$.

2. Evaluate the statistic, f, on the subsampled data, to get f_μ.

3. Repeat the procedure 1-2 multiple times (say, 10-100 times) and approximate the error bar on f as $\mathrm{Std}[f_\mu]$ over subsamples μ.

Let's check what we would expect in our problem with black and white samples. When we draw a subset of half the samples, above, let's consider how we draw black samples assuming, again, that they are rarer than white ($\hat{p} \ll 1$). Then I am sampling each of M black samples with a probability 1/2 into my subsample; the rest of the samples will be white. On average, I expect $\frac{1}{2}M$ black balls in my subsample, with the variance of $\frac{1}{4}M$ from binomial theorem across the subsamples. From this, you can estimate the variance in the estimate of probability of black across subsamples (with factors of 2 cancelling out), to get again $\mathrm{Std}[\hat{p}] \approx \sqrt{\frac{\hat{p}}{N}}$, as above.

The key to bootstrapping is to look at the variability in the statistic over artificially constructed subsets of data and use that as a proxy for the variability over the complete dataset. Clearly, this is only a rough estimate, but we can see how well it does for estimating the errors on PDFs, as illustrated in Fig 6. In that case, as above, we can reason what the errors should be independently of the resampling process: namely, if we observe X_j counts in bin j in the raw histogram and can think of those counts approximately as multinomial or Poisson samples (the distinction is not crucial so long as no single bin contains the large majority of the sampled events — e.g., under the same assumption as our simple example above), then the error on the

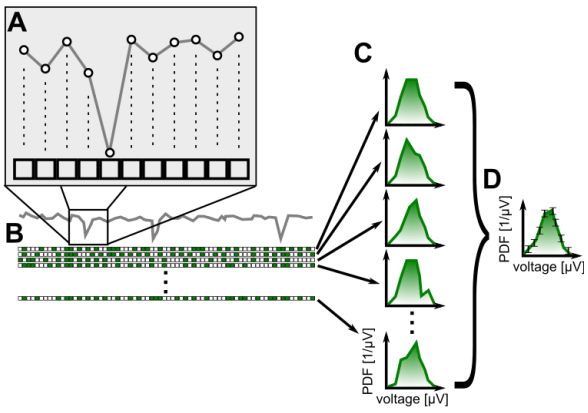

Figure 6: **Schematic of bootstrapping by selecting halves of data to determine error bars on empirical PDF estimate.** This prescription makes sense for uncorrelated data (independent samples). For time series data, such as our voltage example, the independence assumption is likely violated. **(A)** Example time series data (gray line). Zoom in: dots represent measurements, squares below illustrate uniform discrete sampling in time. **(B)** Construction of random resamplings without replacement. Each resampling (a single row) consists of a choice of half of the time points (denoted by green boxes), chosen randomly. **(C)** Given the selected time points, each resampling yields a PDF, estimated over an identical binning grid. **(D)** Standard deviation across individual resampled PDFs yields error bars on the PDF bins.

raw histogram in bin j should be $\approx \sqrt{X_j}$, from which you can derive the error in the estimated PDF. We can then compare whether the simple bootstrap estimates of the error, following the above prescription, match the Poisson expectation in Fig 7. While the match in Fig 7 looks encouraging and the deviation of our PDF estimate from Gaussian clearly looks significant, we should bear in mind the following caveats:

- The bootstrap prescription described above chooses random halves of the data, but the fact that we are choosing halves (not thirds, etc.) appears not to enter our reasoning. In fact, that is not entirely true: you can check analytically that if exactly halves of the data are chosen without replacement for the bootstrap outlined above, the error estimates from the bootstrap will agree with the poisson expectation.

- The poisson estimate is not exact if the total number of samples observed, N, is considered fixed. With a fixed total number of samples, a better model is multinomial (i.e., where the counts drawn in each bin occur with an unknown true probability p_j, and then the expected variance in that bin is $Np_j(1-p_j)$). This would, e.g., properly account for the fact if a single bin contains all the counts, its empirical error is zero given a fixed number of samples observed, N.

- The fact that the total number of observed samples is given, the error estimates on various bins are not independent, as you can convince yourself by thinking of an extreme example where the histogram only has two bins. Estimating a histogram can thus be seen as an estimation problem where multiple statistics at the same time are being estimated, and the error structure of this estimation is fully captured by the covariance matrix of errors. In practice, for estimating PDFs this is very rarely relevant, but it may be for other types of estimates.

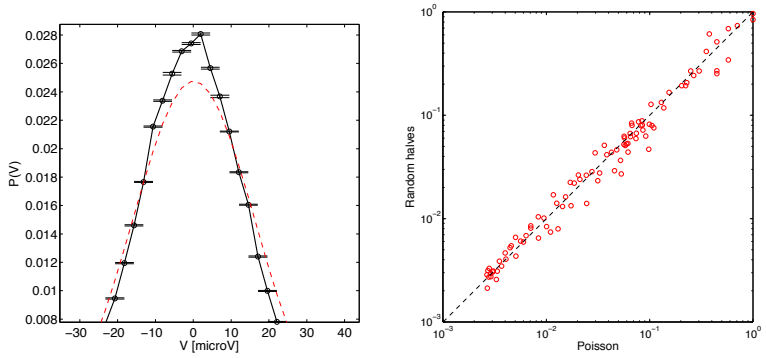

Figure 7: **Left.** Zoom-in of the peak of the histogram of `trace1` values from Fig 5 with bootstrap estimates of the error bars (black), compared to the Gaussian expectation for the peak (dashed red line). **Right.** Error estimated as standard deviation over halves of the data (y-axis) compared to the Poisson (square-root of the number of counts) based estimate on x-axis. Each point is an error bar for one of the 100 bins used to construct the PDF estimate at right.

- The major error in our application of reasoning above to the `trace1` data is the assumption of IID samples, which—as we will see—is patently false. Subsequent samples acquired at 10 kHz are by no means independent: while we nominally have 2M samples, the *effective number of independent samples* is much smaller, and our error estimates should take this into account.

Before addressing these concerns on real data, let's look at the estimation of central moments and quantiles on synthetic data, where we can guarantee that the IID sampling is correct. The goal here is to demonstrate in practice that estimates of the central moments can be very sensitive to outliers and thus many samples might be required to obtain central moments with small error bars; in contrast, quantile statistics are much more robust. In Fig 8 at left we show two very similar looking distributions: a normal (Gaussian) in red, and the log-normal distribution (where the logarithms of the random variable are Gaussian distributed) in black. Skewmess, the third central moment, M_3, of Eq. (4) is zero for the Gaussian distribution and ≈ 0.3 for the lognormal distribution shown here. Kurtosis, the fourth central moment, M_4, is exactly 3 for the Gaussian distribution (which is why people often report normalized kurtosis, $K = M_3 - 3$) and 3.17 for the lognormal distribution. By these two measures, therefore, the lognormal differs only marginally from the Gaussian distribution.

Figure 8 middle shows the estimates of the skewness and normalized kurtosis as a function of the number of IID samples drawn from the lognormal distribution. Error bars are estimated as the std of these statistics over repeated independent draws. The convergence of the kurtosis and skewness to their true values is slow, with $\geq 10^3$ samples needed for the estimates to stabilize and their error bars to become significantly different from 0. This can be contrasted with the quantile estimation shown in Fig 8 at right, where the convergence of quantiles is fast and error bars comparatively small already at an order of magnitude fewer samples.

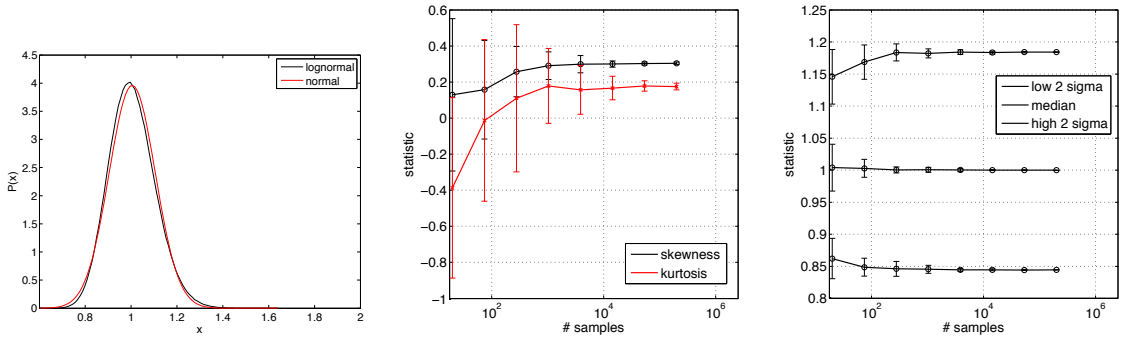

Figure 8: **Left.** Normal (red) and lognormal (black) distributions from which we will draw IID samples. $M_3 = 0$ (0.3) for the normal (lognormal, respectively) distribution; $M_4 = 3$ (3.17) for normal (lognormal, respectively) distribution. Middle: M_3 (black) and $K = M_4 - 3$ (red) estimates from the lognormal distribution. **Right.** Quantile estimates (middle line: median, top line: quantile corresponding to $+2\sigma$ Gaussian deviation, bottom line: quantile corresponding to -2σ Gaussian deviation.

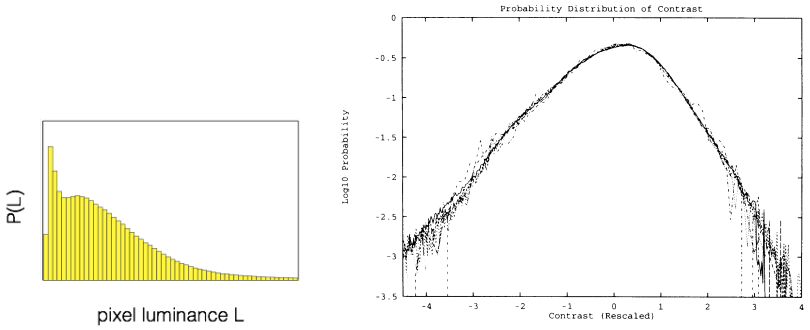

Figure 9: **Left.** The distribution of pixel luminance across a calibrated natural image dataset from Tkacik et al, *PLOS One* **6:** e20409 (2011) shows a large, right-ward skew. This is due to a combination of factors, including the time-of-day and weather variability as well as the overrepresentation of the dark patches in natural scenes due to shade and occlusion with very bright pixels. **Right.** Distribution of $\log L/L_0$, where L_0 is the mean contrast over each calibrated image; the scaling by the mean collapses the distributions sampled from multiple images onto a single distribution. Log-transform reduces the right-ward skew: the resulting distribution is much better described as central and nearly symmetric, with exponential tails. Figure at right reproduced from Ruderman et al, *Phys Rev Lett* **73** (1994).

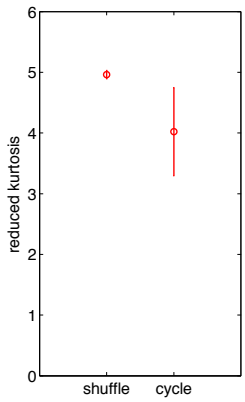

Figure 10: Mean and error bars on kurtosis, M_4, estimated by a simple bootstrap resampling procedure where subsamples of 1/20th of the data are selected randomly from the trace multiple times ("shuffle", at left) or contiguous blocks of 1/20th of the time series are taken multiple times, and the error bars are taken to be the standard deviation of the kurtosis statistic across these subsamples.

It is potentially useful to transform the raw data nonlinearly so that its distribution takes a form more amenable to analysis. Here the goal is not to reduce or reject "outliers" (i.e., large possibly wrongly measured data points) but to deal with distributions that in their raw form have special shapes that make parameter estimation difficult, e.g., long, power-law decaying tails. An intuitive example is the luminance of natural images. One can take a calibrated camera and collect natural imagery; because it is calibrated, it is possible to transform the output of (today, digital) cameras into physical units, i.e., energy fluxes at a certain wavelength. The luminance distribution over pixels of multiple images is a non-negative quantity with a very large, rightward skew, as shown in Fig 9 at left. Using a log transform and normalization per-image, as in Fig 9 at right, achieves a data collapse that we discussed in the context of gene expression before, and ensures that the resulting distribution is much better described by central moments. Sometimes, nonlinear transformations of the data can also be motivated by theoretical considerations: e.g., in chemical kinetics, the relevant quantity is often log concentration (chemical potential) as we know from the form of the relevant equations, so even if what is measured is absolute concentration, it may make sense to log-transform that before analysis. Make sure you understand analytically how distributions transform under nonlinear functions.

After this detour, let's return to estimating the error bars on statistics on real, possibly correlated data. As with your lognormal synthetic example, we can ask about the skewness and kurtosis of the `trace1` time series, to find $M_3 = 0.11$ and $M_4 = 4.97$; i.e., we see slightly higher than Gaussian skew and considerably higher kurtosis, as expected due to heavy tails. Let's now carry out a simple bootstrap resampling procedure to look at the error bars on the moments, and consider two just slightly different resampling scenarios: (i) in the "shuffle" case, select fractions (e.g. 1/20ths) by picking data points randomly; (ii) in the "cycle permute" case, take contiguous blocks of data of the same (e.g., 1/20th) size. As shown in Fig 10, this leads to error bars that are different almost by a factor of 10, a huge difference!

As you might have guessed, the difference comes because the data samples in the time trace

are not IID samples: there is a strong correlation between the successive values in the time series. In the first, "shuffle," estimation, we do bootstrap resampling as if the data were IID, leading us to think that there are more independent samples than there really are. In the second, "cycle," method, we don't break any temporal correlations within the contiguous blocks that we take from `trace1`, properly maintaining temporal correlations. By comparing the two estimated error bars, we may even estimate how many independent samples there really are in the data: the ratio of both standard deviations is about 10, thus the ratio of the two variances is ~ 100. Since the variance of an estimator typically drops with N^{-1} as we discussed previously, that implies that the real number of effectively independent samples is $\sim 100\times$ less than the nominal number of samples ($2 \cdot 10^6$), so about $2 \cdot 10^4$.

In sum, bootstrap procedure depends strongly on whether the data is correlated. For **uncorrelated, IID data**:

- Resample (with replacement, or with synthetically added noise if the noise distribution is known) from the empirical data many times N samples.
- Evaluate the error bar of the statistic of interest as a standard deviation of the statistic evaluated over resampling draws.

Alternatively, you may resample (without replacement) exactly halves of the data, as we did above; in this case, factors of 2 happen to cancel out to give you proper estimates.

For **correlated data** where the dominant source of correlation is serial (i.e., neighboring values are most strongly correlated) and of finite range, much shorter than the full dataset size:

- Split the data into contiguous blocks, for blocks of different sizes.
- Evaluate the statistic on each block and compute its Std across the blocks of a given size.
- Extrapolate the Std of the statistic, as a function of the block size, towards the whole data set size.

The last, extrapolation step is necessary to address one of the bootstrapping concerns mentioned above: the dependence on our choice of the size of the subsample. Clearly, if I compute the statistic on half of the data, I should be overestimating the error bar by roughly $\sqrt{2}$. Bootstrapping prescription above takes this scaling into account by empirical extrapolation, illustrated in Fig 11, for estimating the variance of the "residual" fast-noise part of our `trace1` signal.

Mathematical aside: Error bars for an Ornstein-Uhlenbeck process. Let's verify that the bootstrapping procedure for correlated data makes sense on data that we can synthetically generate and where we have a full control over its statistics. The simplest stochastic process that results in a Gaussian marginal distribution of values as well as exponentially decaying autocorrelation function is the so called Ornstein-Uhlenbeck process, defined by the solution of the following differential equation:

$$\dot{x} = -\gamma x + \xi(t) \tag{17}$$

where $\xi(t)$ is uncorrelated zero-mean Gaussian noise with standard deviation A, i.e., $\langle \xi(t) \rangle = 0$ and $\langle \xi^2 \rangle = A^2$. This process, defined in continuous time, can be simulated in discrete time (with proper care in how the random term is implemented); the discrete time version of this process is known in statistics as an AR(1) (auto-regressive process of order 1). The process is stationary and has a Gaussian distribution of x, with zero mean and variance $\sigma_x^2 = \frac{A^2}{2\gamma}$, as well as exponentially decaying autocorrelation function $C(t) = C_0 \exp(-\gamma t)$.
We can simulate this process numerically for three values of $\gamma = 0.1, 1, 10$, which drastically change correlation length in the timeseries, while keeping the number of samples fixed to $N_{\text{tot}} = 10^7$. We adjust A to keep the std of the distribution of x constant ($\sigma_x = 0.2$). On these three synthetic timeseries, we can now implement the bootstraping scheme to estimate the error bar on the standard deviation in x, as shown in Fig 12. We observe log-log bootstrap extrapolation slope for the error bar with inverse data set size close to -0.5, as expected for efficient estimators. Assuming independence of the timeseries, we estimate the estimator error at $E_0 \sim 4 \cdot 10^{-5}$, obviously independently of γ (and thus correlation time); this we can do by choosing random halves of the data, as discussed in the notes. Our bootstrap analysis with correlated data chunks, in contrast, suggests $E(\gamma = 1) \approx 0.00067$, $E(\gamma = 0.1) \approx 0.0023$, $E(\gamma = 10) \approx 0.00022$.
Do these values make sense? Let's first compare the $\gamma = 1$ analysis with assumed IID samples. The ratio error variances is $(E(\gamma = 1)/E_0)^2 = (6.7 \cdot 10^{-4}/4.1 \cdot 10^{-5})^2 \sim 260$. The discretization used is $\Delta t = 0.005$, which yields 200 timebins per correlation time for $\gamma = 1$. Thus, indeed, the number of independent samples is by a factor of ~ 200 smaller than the total number of samples, as observed.
How about the comparison between different error estimates for three values of γ? Observe that $(E(\gamma = 0.1)/E(\gamma = 1))^2 \sim 12$, and $(E(\gamma = 1)/E(\gamma = 10))^2 \sim 9$. Both fractions are close to 10, which is the ratio of the correlation timescales given by γ. The bootstrap error estimate therefore correctly captures the scaling of the error with the effective number of independent samples $N_{\text{eff}} \approx N_{\text{tot}} \gamma \Delta t$.

It is worth mentioning a few other uses of resampling methods:

- **Jackknife estimate of variance.** A simple way to estimate an error bar on the statistic of interest, f, is to evaluate f on the full dataset, $\mathcal{D}$, as well as on all subsamples of $\mathcal{D}$ where exactly one sample is left out, assuming IID data. The resulting error bar on the statistic is then given by

$$\sigma_f^2 \approx \frac{N-1}{N} \sum_{t=1}^{N} \left(f_{\mathcal{D}\setminus t} - f_{\mathcal{D}} \right)^2, \tag{18}$$

  where $f_{\mathcal{D}}$ indicates evaluation on the full data, and $f_{\mathcal{D}\setminus t}$ indicates leave-one-out evaluation. For small datasets, this is a fast approximation to the full bootstrap.

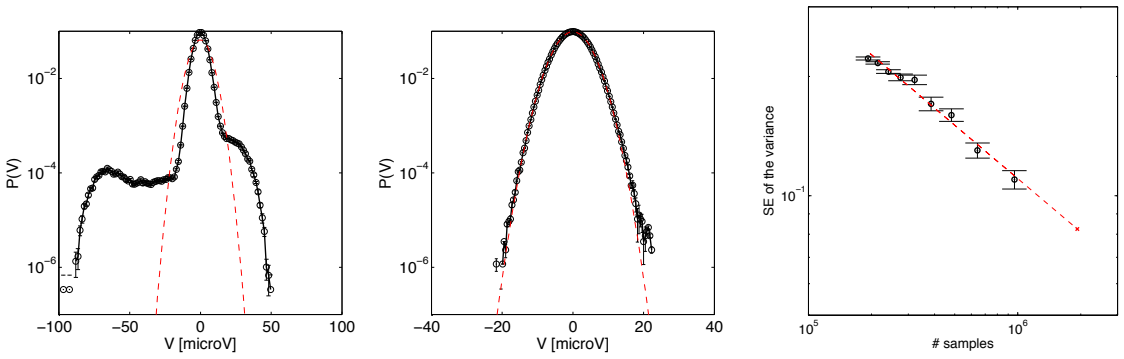

Figure 11: **Left.** Histogram of `trace1` voltage values with slow baseline fluctuations subtracted (as in the homework assignment). Note the much clearer peak at negative voltages corresponding to spiking events. **Middle.** Histogram of baseline-subtracted `trace1` signal where have further been removed, by clipping out small windows (of $\sim$ 60 timebins in length) around the spike, matches the Gaussian distribution much more closely, with variance of $\sigma^2 \approx 16.6\ \mu\text{V}^2$. **Right.** Bootstrap resampling procedure for estimating the error bar on the variance. Different data points are standard deviations of the variance estimate computed over contiguous blocks of data of the length indicated on the x-axis. Red dashed line is the linear extrapolation on the log-log plot (remember, the expectation is that the error bar on the variance decreases $\sim N^{-1/2}$, which is roughly the case here as well). We are interested in the extrapolated value (red cross) at the x-axis which corresponds to the block size equal to the total data set size, i.e., $2 \cdot 10^6$. This yields an error bar estimate of $\sigma^2 = 16.6 \pm 0.09\ \mu\text{V}^2$, substantially different from the error bar estimate assuming IID, which would be $\sim 0.01\ \mu\text{V}^2$.

- **Resampling for bias correction.** Some estimators are strongly biased, and the bias dominates the estimator variance (error bar) for realistic data set sizes. One such example is the naive estimator for the entropy—see the text box below.

- **Resampling for hypothesis testing.** We can often use resampling to generate null distributions against which to test certain chosen statistics. In this case, resampling procedure is crafted by hand to implement the null hypothesis. An example in Fig 13 tests whether two sets of samples have different skewness. We construct them such that they do in the case at hand, by drawing 1000 samples from normal distribution (distribution A, with true skewness zero, but due to finite sampling, any estimator will give some small nonzero value), and 1000 samples from a weakly lognormal distribution (distribution B). Let our statistic of interest be the difference of skewness ("delta skewness") between 1000 draws from each of the distributions. To create the null distribution for this "delta skewness" statistic, one can generate many resampled pairs of datasets, where a random half of the samples are taken from A and the other half from B and vice versa; by construction, such a pair of samples is statistically equivalent. We can evaluate the "delta skewness" statistic over many such pairs of samples and create the null distribution, against which one can evaluate the p-value for the statistic evaluated over the initial draw (where we compare skewness of all samples drawn from A vs all samples drawn from B).

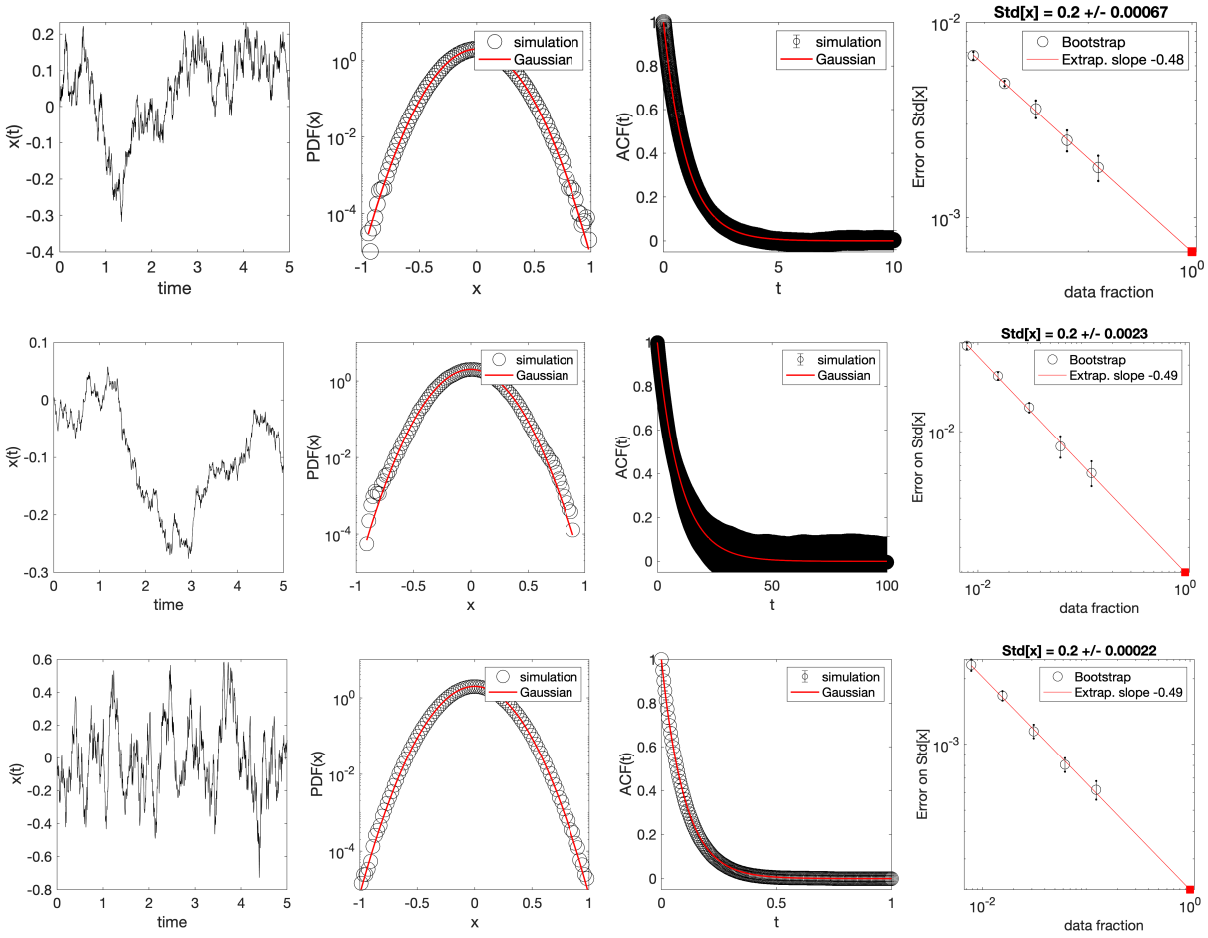

Figure 12: **Estimating the variance of Ornstein-Uhlenbeck (OU) process using bootstrap.** Each row corresponds to an instance of OU process with different correlation time (top: $\gamma = 1$; middle: $\gamma = 0.1$; bottom: $\gamma = 10$). For each process, the first column shows a sample realization of $x(t)$ for $t \in [0, 5]$. The second row shows the marginal distribution of x from the simulation (black), overlaid by the theoretical Gaussian expectation with analytically computed variance (red; here $\sigma_x = 0.2$). The third column shows the autocorrelation function estimated from the simulation (black), overlaid by the theoretical exponentially decaying function with analytically computed timescale (red). The fourth column shows the bootstrap estimate of the error bar on σ_x (the empirical mean and bootstrap-estimated error bar shown in the title of the plot). Contiguous chunks of timeseries for different fractions of the data (data fraction on horizontal axes) are chosen from the timeseries, and the error on std is computed across these chunks at each data fraction, and extrapolated linearly on log-log plot to full data set (fraction = 1, red square). Slope of the extrapolating line is in the legend.

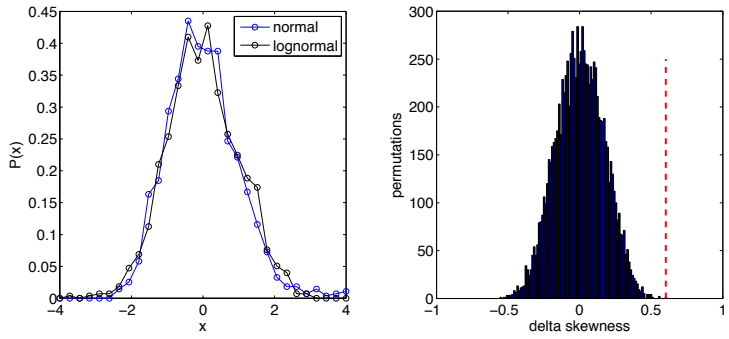

Figure 13: **Left.** A normal (blue) and weakly lognormal (black) PDF estimate, constructed from $N = 1000$ samples IID drawn from each (see main text). Both distributions have zero mean, unit variance, but different skewness. **Right.** The null distribution for the test statistic (blue), constructed as explained in the text, by computing 10^4 times the difference in skewness on resampled data. This is compared with the true difference in skewness (dashed red line), which lies far outside the null distribution, with a significance $p < 10^{-4}$.

A statistic that is known to be strongly biased is the so-called "naive" or maximum likelihood estimator for the entropy of the (discrete) distribution p, i.e., $S[p] = -\sum_i p_i \log_2 p_i$. Here, the index i runs through all discrete bins of the distribution, and p_i is an empirical estimate of the probability, i.e., $p_i = N_i/N$, where N_i is the raw counts of occurrence of state i, and $N = \sum_i N_i$ is the total number of counts. Figure 14 shows how bad the problem really is for undersampled distributions: here, the true distribution is a uniform distribution over $m = 500$ bins, which should have the entropy of $S = \log_2 500 \approx 8.97$ bits. With only a $N = 100$ samples, the naive estimator gives an entropy of ≈ 6.3 bits with a small error bar—so the dominant error of the estimator is the bias (the estimation is wrong on average), not the variance. Only in the regime where $N \gg m$ is the bias of the naive estimator small. One way to "debias" the estimate is to empirically assess how the estimate systematically changes with the sample size, N, and then extrapolate away this dependence in the infinite sample limit, as shown in Fig 14 at right. This method is most powerful when we have a theoretical expectation for the scaling of the bias with N, as we do for the entropy, and the extrapolation is used only to obtain the scaling parameters.

Homework 3. Try three different bootstrapping methods for estimating the error bar on a statistic for a synthetic data that is IID by construction. For your data set, draw a hundred samples from a normal distribution with zero mean and unit variance. On this dataset, estimate the standard deviation (ground truth value of which is unity), and its error bar. In the first method for error bar estimation, do bootstrapping by selecting random halves of the data (e.g., 1000 random splits). In the second method, do bootstrapping by resampling (e.g., 1000 times) with replacement from the full dataset. In the third method, apply the Jackknife estimate. Compare the obtained error bar estimates.

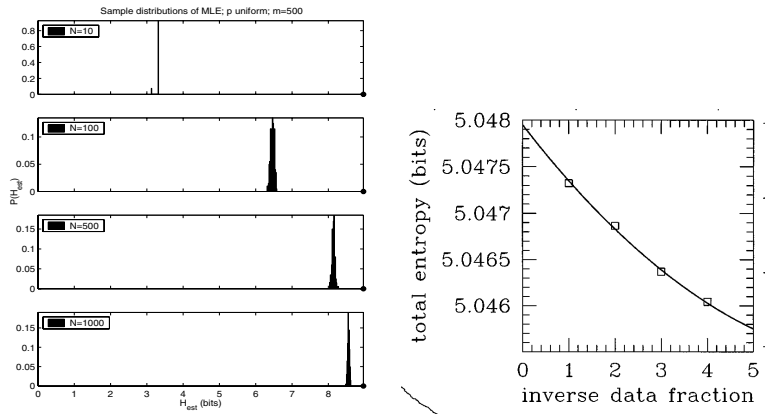

Figure 14: **Left.** Distributions of the "naive" estimator for entropy (see text), as a function of the sample size (N, in the legends), for an uniform distribution over $m = 500$ bins; the correct value of the entropy is almost 9 bits (black dot in the rightmost corner of the plot). The bias of the estimator, here the difference between the mean of the distribution and the black dot, is much larger than the error bar of the estimator (width of the distribution). Reproduced from Paninski et al, *N*eural Comput **15** (2003). **Right.** Since the bias of the naive estimator for entropy is known to scale as $1/N$ to leading order, i.e., $S(N) = S_{\rm true} + S_1/N + S_2/N^2 + \ldots$, we can undo it by subsampling: entropy is evaluated over many subsamples of the data of different size (here, "inverse data fraction" on x-axis, with, e.g., 5 representing 20% of total samples, etc.). These estimates at different fractions of the data are used to fit the extrapolation curve for $S(N)$, here quadratic, and find the true estimate, $S_{\rm true}$ (the value of the smooth curve at "inverse data fraction" 0, which corresponds to the $N \to \infty$ limit. Reproduced from Strong et al, *Phys Rev Lett* **80** (1998), where they estimated the entropy of neural spike trains from actual data.

Homework 4. Use your detection of spikes on background-subtracted `trace1` data from previous homework to define a new vector z, of the same length as `trace1`, which only contains zeros or ones: ones should indicate downward threshold voltage crossings in `trace1`, with threshold you selected for spike detection, and other time bins should be zero; in other words, z should have a one whenever a spike occurred in the original voltage trace; $\sum_t z_t$ should be the total number of detected spikes in `trace1`. Note: z should only indicate threshold *crossings*, i.e., cases where voltage was above the threshold in the previous timebin and is below the threshold in the next timebin. Let's now do the following: (i) take many contiguous chunks of z of length $500, 750, 1000, 2500, 5000, \ldots, 100000$ time bins, and on each chunk estimate the "spike rate", as the number of spikes in that chunk divided by the length of that chunk; (ii) estimate the error in the spike rate by computing the SD over chunks of the same length, as in the classic bootstrap; (iii) plot this error as a function of the chunk length, on the semilogx plot. Does the error bar decrease with the chunk length as you would expect? (iv) To get a better intuition, compare this plot of spike rate error bar vs chunk length for the vector z whose elements you first randomly permuted to break all temporal correlations. Check that in this permuted control, the spike rate error bar decrease as (chunk length)$^{-1/2}$. If you see a deviation from that behavior of the spike rate error bar for the original z, what could be the source of such deviations?

6 Comparing distributions

6.1 Kolmogorov-Smirnov test

Now that we know how to create histograms and estimate PDFs with error bars from data, we can look for the ways of comparing whether two distributions are the same, or, alternatively, of testing whether a data-derived distribution matches a theoretical model. For continuous data a standard test for this is the Kolmogorov-Smirnov (KS) test, which has three clear benefits: **(i)** it is non-parametric, i.e., it makes no modeling assumptions about the data; **(ii)** it doesn't require binning the data first to create histograms; and **(iii)** it is reparametrization-invariant (which means that test results are independent if carried out on the original data, x, or on some nonlinear monotonic transformation of the data, e.g., $f(x)$). Two potential drawbacks are that the test is only applicable to 1D distributions, and that its sensitivity is greatest for distribution discrepancies near the median, rather than in the tails (although generalizations of the test exist that correct for that).

Given two sets of samples of size N, the test first creates the two corresponding cumulative distribution functions, $C_1(x)$ and $C_2(x)$, and then defines as a test statistic the distance D, where $D = \max_x |C_1(x) - C_2(x)|$. The key property of this choice is that the probability of D being equal to or larger than some observed value given that the distributions don't differ (the null hypothesis) is a universal function only of D and the $\sqrt{N}$, the number of samples, which is known analytically (see *Numerical Recipes in C* if you want to know the expression) and is built in into most numerical packages.

An interesting application of the KS test is to ask whether, for instance, the background Gaussian-like noise distribution, shown in Fig 11 middle, could be *stationary*, i.e., statistically indistinguishable between the first and second part of the experiment. Indeed, up to now we

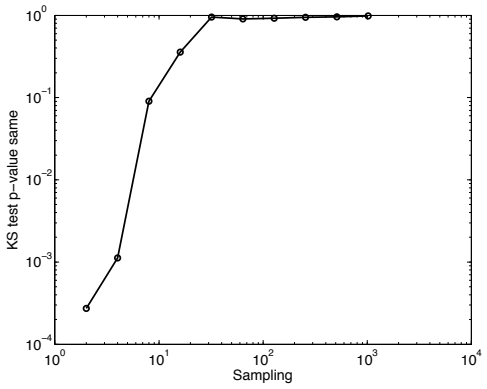

Figure 15: Raw `trace1` data has been background subtracted and spikes have been removed as in Fig 11, middle. The time series is then split into two halves, each consisting of roughly 1M samples. From this, we subsample every $k-th$ datapoint (denoted on the x-axis), and use the KS test to ask whether we can reject the null hypothesis that the (subsampled) data in the first and second halves of the trace comes from the same distribution; the resulting p-value is shown on the y-axis.

have been assuming the stationarity of our time series data, that is, the stability of all the statistics at different times, without explicitly checking for it. Figure 15 suggests that the first and the second half of the trace are nominally different (the null hypothesis that they are the same could be rejected with low p-value), but only if we don't subsample the data. Yet we know that there must be strong correlations in the data which is thus not IID, and have roughly estimated before that the effective number of independent samples could be ~ 100 times lower than the nominal value. Indeed, if we subsample correspondingly (only take every 32th point or even more sparsely), the gaussian distributions in the first and second half of the trace are not significantly different. Note that we could use the KS test in a straightforward fashion not only to compare the first and second parts of the trace, but also to compare each to the Gaussian model rigorously, but same caveats about non-IID samples apply.

6.2 χ^2 test

Another standard way of comparing the distributions is useful when we are working with raw histograms, that is, with binned data. Let counts X_j constitute integer counts in a raw histogram that is to be compared to an expectation, Y_j, derived from a model (Ys need not be integers). Since we expect (from the model) the error in each bin to be $\sqrt{Y_j}$, we can form a χ^2 statistic for the deviation between the observed counts and the model:

$$\chi^2 = \sum_j \frac{(X_j - Y_j)^2}{Y_j}. \tag{19}$$

The significance of this deviation can then be looked up from the standard χ^2 distribution to assess whether the null hypothesis that the data X and the model distribution Y are the same can be rejected. Nominally, χ^2-distribution gives the probability that the sum of squares of independent gaussian variables is greater than some threshold. In Eq. (19) individual terms are clearly not Gaussian (since X are discrete), but if the counts are sufficiently large and we are summing over sufficient number of bins, the use of χ^2 is appropriate. The number of degrees-of-freedom (DoF) for the comparison is typically equal to the number of the histogram bins minus

1, if the model distribution for Y is forced by normalization to have the same total number of events, $N = \sum_j X_j$, observed in the data.

The same test can be also used to compare two data-derived histograms, as we did before when we looked at the first and second half of the time series. The only change is that now both X and Y are integers, and the denominator of Eq (19) changes into $X_j + Y_j$, since the uncertainty is in both histograms.

Lastly, χ^2 is often used as an objective function to fit theoretical distribution models to data. As an example, we fit a mixture of three Gaussians to the distribution of voltage values for `trace1` in Fig 16. This amounts to fitting 8 parameters in total (3 means and variances, and 2 prefactors) that determine the model, Y, variable in the χ^2 given by Eq (19), which is minimized using a generic nonlinear fitting routine, e.g., Levenberg-Marquardt; there is no guarantee of finding the global χ^2 minimum. There is no theoretical reason for why our data distribution should be composed of three gaussians (and, indeed, the deviations of the model from the data are quite significant), but a mixture of Gaussians is often a flexible and versatile way to phenomenologically model complex, multi-peaked distributions. It is also interesting to contrast fitting such models to data using χ^2 minimization on binned histograms vs. doing a maximum likelihood fit of the mixture of Gaussians (which is a probabilistic model) directly to unbinned data $\mathcal{D}$.

One should understand well the nature of the statistical test being done when computing, e.g., χ values in Eq. (19). If χ values across bins are squared and summed together, then *one* test is performed to assess whether a sample as a whole comes from a given distribution. A very different situation arises when deviations in individual bins are tested for significance: here, we have multiple hypothesis testing (across all bins), and our significance thresholds should be adjusted according to how many hypotheses we are testing (e.g., using Bonferroni correction). A typical case is in high energy experiments (HEP) where the Standard Model theory predicts how many events of a particular type should be detected in the particle colliders as a function of some kinematic variable, and this is compared bin-by-bin to actual data. A good example was the excitement in 2016 (which turned out to be a statistical fluctuation) of a localized deviation from Standard Model, which could be a putative particle, in *arxiv.org:1506.00962*. A localized deviation was highly significant, at 3σ; but this was correctly decreased for the "look-elsewhere effect" (physicist's slang for multiple hypothesis testing), and fell considerably short of the 5σ standard for proclaiming new discovery (to be compared to the standard life science thresholds of $p = 0.05$ which lead to many significance reports highlighted recently). This problem is very acute when the number of effective hypotheses is immense. In fMRI scans, for instance, the brain is divided into local volume elements or "voxels" where the BOLD—blood oxygenation signal, a proxy for local energy utilization by neural processing—is measured. The number of voxels in new scanners is $> 10^5$, and with standard significance thresholds not properly corrected for this number one is likely to observe "significant" activity even when there can be none, as demonstrated by the famous fMRI scan of a dead atlantic salmon (Bennett et al, `http://prefrontal.org/files/posters/Bennett-Salmon-2009.pdf`).

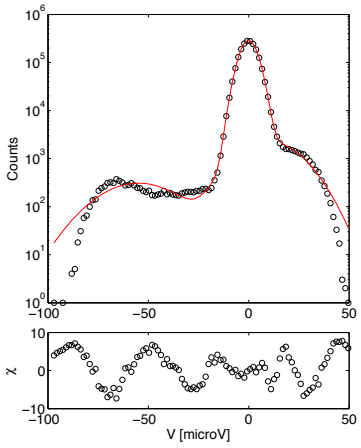

Figure 16: The raw histogram of background-subtracted voltage in `trace1` and the best-fit mixture of three Gaussians (red line), using χ^2 minimization. The plot below shows the per-bin contributions to the total χ^2. While χ^2 can be used as a good metric for fitting models to data, its precise value cannot be used directly for statistical significance testing since the data is not IID.

Homework 5. Estimate the distribution of log luminance levels in the Ruderman natural image dataset. For each image, transform it by taking the log of the pixel luminance values, and subtracting the mean value of log luminance across each image. Show the estimate of the error bars on this distribution of mean-subtracted log-luminances, y, by thinking about what "independent samples" are in this dataset (you do not need to do a full bootstrap analysis). Show that the left- and right- tails of this distribution are approximately exponential, i.e., $P_+(y) \propto \exp(-\mu_+|y|)$ and $P_-(y) \propto \exp(-\mu_-|y|)$, where $P_\pm$ stand for the positive (or, respectively, the negative) tails of the log-luminance distribution. Do this by finding the two best-fitting constants, μ_+ and μ_- and plotting the tails on top of the normalized distribution for y. A good choice to define the positive and negative tails is to use the threshold of $|y| > 1$. How you do the fitting (linear regression in log-luminance space, χ^2 fitting of raw histograms) is up to you, but explain clearly what you did.

7 Measuring correlations

Here we learn to quantify different types of (pairwise) correlations. By pairwise correlations we mean statistical relationships, in particular, deviation from statistical independence, that are evident at the level of pairs of components of data vectors, $\mathbf{x}$, or pairs of values in a time series. Pairwise independence would imply that, for a given pair of components x_μ and x_ν:

$$P(x_\mu, x_\nu) = P_\mu(x_\mu)P_\nu(x_\nu), \tag{20}$$

i.e., that the joint distribution factorizes into marginal distributions. In discretely-sampled time series, you can think of a true generating distribution over all possible sequences that could have been observed, $P(x_1, x_2, \ldots, x_N)$; samples from this distribution can be thought of as repeats of an experiment that generates the time series many times. Pairwise independence in this context

implies

$$P(x_t, x_{t'}) = P_t(x_t)P_{t'}(x_{t'}) \tag{21}$$

for any pair of time indices, t and t'. If, further, the time series is stationary (and thus its statistics, including the marginal distribution of values, P_0, are independent of the absolute time) then the joint probability of observing two values is the product of probabilities of observing each evaluated in the same distribution, and independence can be tested from the single realization of the time series. So, under stationarity, it is further true that:

$$P_t(x_t)P_{t'}(x_{t'}) = P_0(x_t)P_0(x_{t'}). \tag{22}$$

Generically, "correlation" could be understood as any type of deviation from statistical independence. This is to be contrasted, as we will explain below, with *linear correlation*, which is the (commonly used) subtype of statistical dependence: there exist pairwise correlations that are not linear.

The problem with testing for statistical independence (or measuring statistical dependence) is that true generating distribution, $P(\mathbf{x})$, is not available directly; we are only given a finite number of samples, $\mathcal{D}$, drawn from it. Depending on whether components of x are discrete categorical, discrete ordinal, or continuous, we can devise various ways of testing for statistical dependence, as we explain below.

Multi-point correlations generalize the notions of statistical dependence beyond Eq (20) to dependence between three, four, etc. variables. Importantly, it is easy to demonstrate cases where multiple variables are statistically independent at the level of all pairs, but statistically dependent in an intrinsically multi-point sense, implying that checking for pairwise statistical independence does not demonstrate full statistical dependence in the data. Studying multi-point (higher-order) correlations is a very interesting topic, but is beyond the scope of this lecture; even defining multi-point correlations systematically—and ignoring the estimation issues—is a non-trivial task; yet as we will see at the end of this course, there are examples where the relevant information is present exactly in such multi-point (also called "higher order") correlations.

We have seen multiple indications that there are serial (temporal) correlations in the `trace1` data, but haven't properly measured their strength and evaluated their significance.

7.1 Discrete data, contingency tables

Let's first consider the case where the variables could be discrete, without any natural ordering (e.g., categorical discrete variables). The components of $\mathbf{x}$ take on K possible values (let's enumerate them with $1, \ldots, K$), and for each component μ, the marginal distribution $P_i(x_\mu)$, is a discrete distribution over K values with normalization constraint, $\sum_{x_\mu} P_\mu(x_\mu) = 1$. Similarly, any pairwise distribution $P_{\mu\nu}(x_\mu, x_\nu)$, is a joint (2D) distribution of size $K \times K$, with one normalization constraint. In case of statistical independence, marginal distributions P_μ and P_ν (given by $2K-2$ independent values) fully determine the joint distribution $P_{\mu\nu}$ (given by K^2-1 values).

A simple way to test for pairwise independence is to construct a statistic that measures the deviation between the joint histogram and the independence model and compare it against its null distribution. Let N_{ij} be the number of counts in the data $\mathcal{D}$ when $x_\mu = i$ and $x_\nu = j$, $N_i^1 = \sum_j N_{ij}$ and $N_j^2 = \sum_i N_{ij}$. We can then define the χ^2 statistic as

$$\chi^2 = \sum_{ij} \frac{(N_{ij} - n_{ij})^2}{n_{ij}}, \tag{23}$$

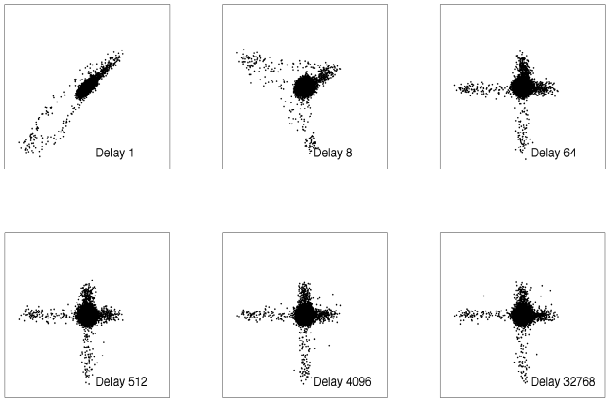

Figure 17: Statistical dependencies between pairs of values, $x(t)$ and $x(t')$, in `trace1`, for different delays $|t - t'|$ (different panels), sampled assuming the stationarity of the time series. Obvious dependencies at short delays (e.g., 1 or 8) dissipate by a delay of ~ 64.

where n_{ij} is the expectation under independence, i.e., $n_{ij} = N\frac{N_i^1}{N}\frac{N_j^2}{N}$. Following the same arguments that led us to use the χ^2 distribution as the null distribution when we compared 1D histograms, here too we can evaluate whether a given value of χ^2 is consistent with the null hypothesis of pairwise independence, by looking up the χ^2 distribution with $K \times K - 1 - 2(K-1)$ degrees of freedom (i.e., the number of degrees of freedom in the pairwise histogram minus the number of degrees of freedom in two marginal histograms).

To illustrate this test on our data, we can imagine the two sets of values as being the voltage trace values at a given time delay, sampled from `trace1`. Example pairwise scatterplots are shown in Fig 17, showing clear short delay dependencies which vanish (at least visually) at large delays. To apply our statistical testing framework and ask whether the dependencies at long delays are significant, we can first turn the (pseudo)continuous voltage trace values into discrete values by binning them into, e.g., $K = 100$ equi-populated bins (i.e., we set the bin boundaries at equally spaced quantiles), and constructing the resulting pairwise histogram, shown in Fig 18 at left. Paralleling our discussion of PDF estimation in Section 4, you can recognize this histogram construction as exactly the one that is not very useful for visualization, since it employs non-uniform binning and doesn't result in a uniform and interpretable PDF. In contrast, this construction is suitable for statistical testing, since equi-populated binning guarantees that individual bins have sufficient number ($\gg 1$) of counts, and consequently the distribution of z-scores (χ values), or individual normalized deviations from the independence expectation, is nearly Gaussian, as shown in Fig 18 at right.

The resulting value of χ^2 can now be tested against the null expectation (the χ^2 distribution with 9801 degrees of freedom) to yield $P(\chi^2) = 0.58$, or a high likelihood that the data is consistent with the null model of pairwise independence, as expected.

Another statistic for measuring pairwise statistical dependence is the *mutual information*,

$$I[P(x_\mu, x_\nu)] = \sum_{x_\mu, x_\nu} P(x_\mu, x_\nu) \log_2 \frac{P(x_\mu, x_\nu)}{P_\mu(x_\mu) P_\nu(x_\nu)}. \tag{24}$$

Here, we wrote the mutual information (a non-negative scalar dependency measure usually expressed in bits) as a functional of the joint distribution, $P(x_\mu, x_\nu)$. The power of this statistic lies in its theoretical underpinnings which go to the foundations of Shannon's information theory; importantly, I is zero if and only if x_μ and x_ν are statistically independent. This is a universal

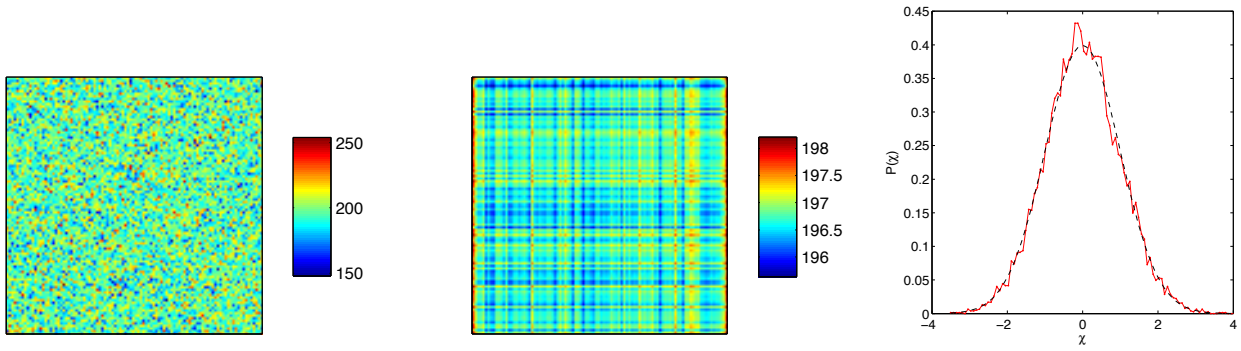

Figure 18: **Left.** Raw histogram (counts, N_{ij}, colorbar at right) for `trace1` voltage values, discretized into 100 equi-populated bins, at a large delay of 32768 time bins. **Middle.** Count expectation in case of independence (n_{ij}, see text). **Right.** The distribution of χ values from Eq (23) for each of the 100×100 entries in the raw histogram (red line), compared to the standard normal distribution (dashed black line). Since the z-scores are Gaussian distributed, the application of χ^2 test is warranted.

result and is true irrespective of the distribution of the x values, whether they are continuous or discrete, etc. Again, we need to devise estimators for I that will act on data $\mathcal{D}$ directly and not on the unknown generating distributions, and that is a formidable task that we cannot go into here. One way is the "naive" estimate, analogous to the "naive" estimate for the entropy discussed in Section 5: if the data is naturally discrete or is binned as in the χ^2 example above, we can approximate the distributions involved simply with normalized counts, e.g., $P(x_\mu = i, x_\nu = j) \approx N_{ij} / \sum_{ij} N_{ij}$, and similarly for the marginal distributions. Such estimator suffers from large biases, just like the entropy estimator to which it is related, but for the purpose of hypothesis testing (for pairwise independence; here we simply want a test whether $I = 0$ which indicates independence) this is much less relevant than for the purposes of information estimation (where we want a precise value of I).

In short, we can compute the information, Eq (24), by the "naive" or any other estimation method, to find a nonzero value—and now we need to decide whether this value is significant or not. For this, we need a null distribution, yet unfortunately there is no analytic formula or simple numerical expression for one. We can, however, reuse our ideas for resampling to construct the null distribution empirically. Pairs of values of voltage used to construct the joint histogram can be randomly reshuffled to break any correlations (i.e., values of voltage at time t_1 are paired with random values of voltage at time t_2, such that the marginal distributions remain unaffected). For each such shuffle, we estimate the information statistic of Eq (24), and by repeating shuffle many times, create empirical null distribution against which the statistic can be tested as shown in Fig 19.

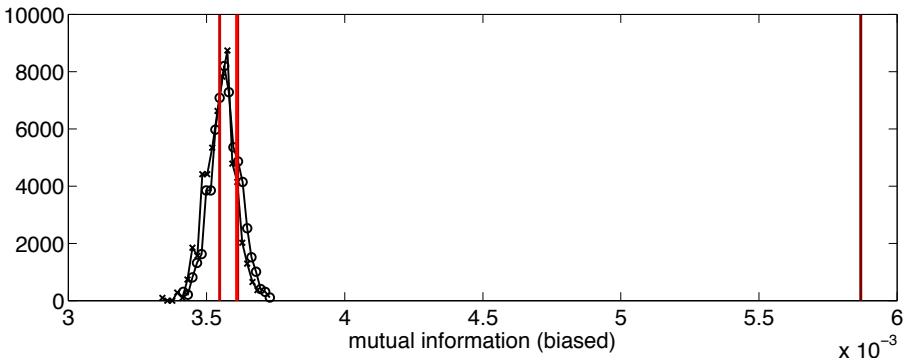

Figure 19: The value of naive (biased) mutual information statistic of Eq (24) evaluated on `trace1` data binarized into 100 equi-populated bins for different delays (dark red to bright red corresponds to delays of $64, 512, 4096, 32768$ time bins). Shown in black are the null distributions of the same statistic after random reshufflings of the data. While hard to observe in Fig 18, this test demonstrates that the correlation between voltage values separated by 64 bins is small (uncorrected $I \sim 6 \cdot 10^{-3}$ bits) but statistically significant, whereas at larger separations the correlation is not significant, consistent with χ^2 analysis results for 32768 time bin delay. For reference, the same statistic computed at time delay of 1 bin (where strong correlation is obviously present) yields 0.6 bits of mutual information.

Occasionally, alternative measures of pairwise correlation emerge that are supposed to retain the universal power of mutual information to detect statistical dependencies, while improving on its other properties, mainly in terms of interpretability and ease of estimation. Given the firm mathematical basis of mutual information in information theory such suggestions need to be treated with care and a healthy dose of skepticism. A good lesson in this vein has been the paper by Reshef et al, *Science* **334:** 1518 (2011). This paper introduced a new statistic, called Maximal Information Coefficient (or MIC), similar to mutual information with some crucial differences. In particular, the authors asserted that MIC has a new property of *equitability*, i.e., the ability of the statistic to assign the same value to all (stochastic) dependencies between two variables (say x and y) that have the same amount of noise, irrespective of the form of the underlying dependence (illustrated in Fig 20); mutual information was claimed not to have this property. This claim was supported only by simulation evidence. Later, Kinney & Atwal, *PNAS* **111:** 3354 (2014), showed that the claims of the original paper were faulty in several key aspects: **(i)** they mathematically proved that no non-trivial measure of dependence can exist that has the equitability property, so MIC cannot have it either; **(ii)** MIC violates several intuitive notions of correlation that any reasonable measure of statistical dependence should have; **(iii)** the original numerical evidence for the equitability of MIC and against the equitability of mutual information was obtained with far too few samples, and when the sampling is increased, MIC is seen to be less equitable than mutual information. While the original Science paper appeared at a key moment when datasets have been rapidly growing and good measures of dependency are crucial, the story of these two papers illustrates the dangers of trading mathematical rigor for the attractiveness of a momentary "hot topic." Despite this, the Science paper in 2023 already has more than 2600 citations.

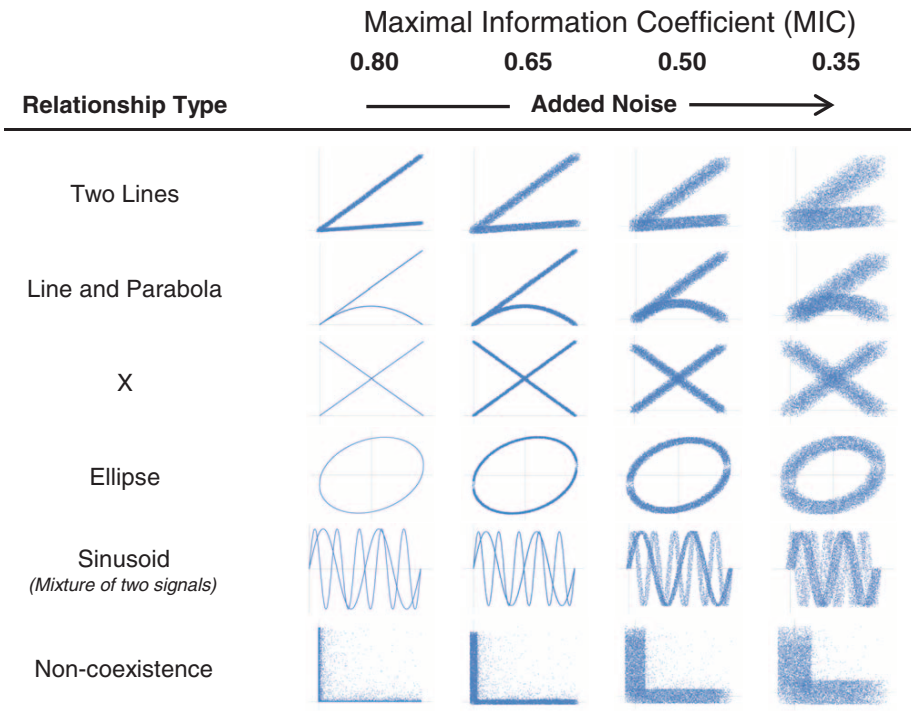

Figure 20: Illustration of *equitability*, the claimed property of the MIC statistic, introduced in and reproduced from Reshef et al, *Science* **334:** 1518 (2011). Different example of functional dependency between two variables, x and y (table rows) are corrupted by different amounts of noise (table columns). MIC is claimed to return the normalized score (between 0 and 1) that measures statistical dependence in a way which is only a function of the noise magnitude (column index), but not of the underlying functional dependency or the way this was sampled (row index).

7.2 Correlation between ordinal variables

When the variables are ordinal (i.e., there exists a natural ordering between the possible values of the variables, defined by the less than / greater than relationships), simpler measures of pairwise correlation exist. The most common one is the linear or Pearson correlation coefficient r, or also the normalized covariance, defined by

$$r = \frac{\mathrm{Cov}(x,y)}{\sigma_x \sigma_y}, \tag{25}$$

where the sample covariance is

$$\mathrm{Cov}(x,y) = \frac{1}{N}\sum_{t=1}^{N}(x_t - \bar{x})(y_t - \bar{y}), \tag{26}$$

with $\bar{x}, \bar{y}$ being the empirical means over the N samples of x and y, respectively, and σ_x^2, σ_y^2 are the standard variances, e.g.,

$$\sigma_x^2 = \frac{1}{N}\sum_{t=1}^{N}(x_t - \bar{x})^2. \tag{27}$$

Pearson correlation r is normalized to lie between -1 (perfect linear anticorrelation) and 1 (perfect linear correlation), with 0 indicating the absence of linear statistical dependence. It is intimately correlated to the problem of least-squares linear regression, where the relationship between x and y is modeled as $y = A^*x + B^*$, with the coefficients A^* and B^* determined by minimizing χ^2:

$$\{A^*, B^*\} = \arg\min_{A,B} \chi^2(A, B) = \arg\min_{A,B} \sum_t (y_t - (Ax_t + B))^2 . \tag{28}$$

Here, the smallest achievable values of χ^2 at $\{A^*, B^*\}$ is related to r^2 by the relation

$$\chi^2 = (1 - r^2) \sum_t (y_t - \bar{y})^2. \tag{29}$$

This is why the quality of the linear fit is often given in terms of the r^2 value, which is tightly related to the Pearson correlation, r; r^2 is also the "fraction of variance explained" (and $1 - r^2$ thus "fraction of variance unexplained") in a regression model. For linear regression, you can clearly see this from Eq (29), where the right-hand side is the total variance in y, while the left-hand side χ^2 is the square residual (the unexplained variance).

In addition to interpretability, the other advantage of Pearson correlation is the ease with which it is computed. The major drawbacks are as follows: **(i)** it is not guaranteed to detect non-linear dependencies; **(ii)** the universal null distribution against which to assess statistical significance does not exist. Here, by "universal" we mean a distribution that is independent of the marginal distributions, $P_x(x)$ and $P_y(y)$ that generated the data. Proper statistical significance tests can then only be made against analytically known null distributions if the marginals take a special form, which needs to be independently verified; or against empirically constructed null distributions obtained by Monte Carlo sampling / shuffling of the data. Generally, for central marginals (i.e., unimodal distributions with tails that decay exponentially or faster) the Pearson correlation estimator of Eq (25) on independent data will result in an unbiased estimate (i.e., $\langle r \rangle = 0$) whose error bar decays as $\sigma_r \sim N^{-1/2}$, as expected.

Some of the drawbacks of the Pearson correlation are remedied by the Spearman or rank correlation. Here, values for x and y are first sorted in ascending order and assigned a rank in the ordered list, thereby transforming the data arrays for x and y into two lists of integer ranks, R_x and R_y. For example, the smallest value in x is assigned rank 1 and thus the entry corresponding to x in R_x is 1; similarly, the second-smallest value is assigned an entry 2 in R_x, until the maximal value in x which is assigned the rank N. There is a small potential complication in case several values of x have the same values, in which case fractional ranks can be introduced; usually this is already handled by the numerical software.

Spearman rank correlation, r_s, on the original data pairs $(x, y)_t$ is then defined simply as a Pearson correlation estimator on the *rank* variables, i.e., over pairs $(R_x, R_y)_t$. It thus retains the ease of computability of linear correlation. The main benefits are the ability to detect some nonlinear dependencies, especially monotonic ones; furthermore, there is an exact statistical test for significance. This is because the marginal distributions of ranks (as opposed to values) are known and are simply uniform. Under the null hypothesis of statistical independence, one defines the statistic t as follows:

$$t = r_s \sqrt{\frac{N - 2}{1 - r_s^2}}, \tag{30}$$

which for large N has a Student-t distribution with $N - 2$ degrees of freedom, where N is the number of data points.

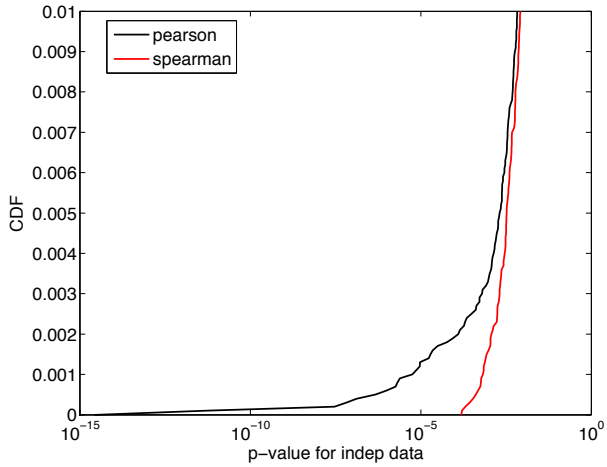

Figure 21: The low-end tail of the cumulative distribution function for the p-value of correlation (linear in black, rank in red) for synthetic data that is independent by construction. 1000 (x, y) pairs were drawn from lognormal distributions IID and tested for dependence 10^4 times to generate the plotted CDF. p-values for Pearson correlation (which Matlab, in this case, computes by assuming normal marginal distributions) extend to very small values, wrongly rejecting the null hypothesis, while rank correlation is much more robust.

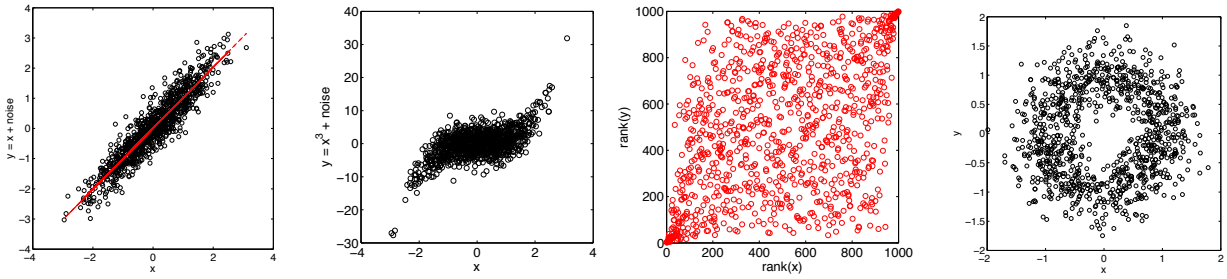

Figure 22: Synthetic IID drawn pairs of data points, (x, y), shown as black circles, displaying different patterns of correlation, and measures of correlation. **Left.** Linearly correlated data, where x and y are jointly Gaussian, or equivalently, $y = x + \eta$ and the noise, η, and $P_x(x)$ are IID Gaussian. $r \approx r_s \approx 0.92$ and $I \approx 1.45$ bits. For jointly Gaussian variables, mutual information I and Pearson correlation are related analytically, $I(x; y) = -\frac{1}{2}\log_2(1 - r^2)$. As a rule of thumb, 1 bit of mutual information among jointly Gaussian variables roughly amounts to a linear correlation of 0.9, and 3 bits to 0.99. **Left middle.** Nonlinear relationship with $y = x^3 + \eta$. Here, $r \approx 0.64$ and $I \approx 1.05$ bits. **Right middle.** Explicit construction for rank correlation for the nonlinear example; scatter-plotted are R_x and R_y values from the previous panel, showing linear correlation in ranks that yields $r_s \approx 0.6$. **Right.** Circular joint dependence is not detected by either linear or rank correlation, which give values close to 0 that are not significant. In contrast, $I \approx 0.17$ bits, significantly above zero.

Figure 22 illustrates the application of various measures of statistical dependence on synthetic data. While linear and rank correlation are easy to use, they can miss some forms of statistical dependence (Fig 22, right) which the more versatile mutual information will detect. Unfortunately, estimating mutual information on continuous data is cumbersome, since it requires either binning the data with variable number of bins and extrapolating the bins away (e.g., *arxiv.org:CS.IT/0502017*), or using binless estimators which, however, make further assumptions on the joint distribution, $P(x, y)$ (e.g., the k-nearest-neighbour, or kNN, estimator by Kraskov et al).

7.3 Time series correlation

In case we are given a time series, $x(t)$, where real-valued data are indexed by an ordinal t, or equivalently, data that is indexed by spatial coordinates (e.g., light intensity in an image, $I(x, y)$, where x and y are discrete pixel locations), we can define a *correlation function* as a set of (usually linear) correlation values indexed by the time or location coordinates.

In general, if the original data is indexed by 1D index (as with time series), then the full correlation function is 2D:

$$C(t, t') = \langle (x(t) - \langle x(t) \rangle)(x(t') - \langle x(t') \rangle), \tag{31}$$

where brackets $\langle \cdot \rangle$ denote averaging across multiple repeats of the experiment in which the time series $x(t)$ has been measured under identical conditions. Mathematically, $C(t, t')$ is a real valued function of two real valued coordinates; when implemented numerically, time series are usually discretely sampled (i.e., t and t' now denote time bins), such that $C(t, t')$ is a matrix of dimension $N \times N$, where N is the number of time bins in the series. This matrix is a *covariance matrix*, and is thus symmetric with non-negative eigenvalues.

If the time series is *stationary*, statistics are invariant to absolute time shifts, thus $C(t, t') = C(t - t_0, t' - t_0)$, where t_0 is some arbitrary time shift, and the correlation can only depend on the difference in time, $\tau = t - t'$, between the two points in the time series where we are computing the correlation. By denoting $\tau = t - t'$ write

$$C(\tau) = C(t, t') = \langle (x(t_0 + \tau) - \langle x \rangle)(x(t_0) - \langle x \rangle) \rangle_{t_0}, \tag{32}$$

where, for the stationary time series, the correlation is now *only* a function of a single parameter, τ, and can be computed by averaging across the *single* time series by picking different values of t_0 (this is meant by the notation $\langle \cdot \rangle_{t_0}$ in Eq (32), instead over averaging multiple repeats of the experiment (ergodicity). Due to stationarity, the mean values of time series also cannot depend on absolute time, $\langle x(t) \rangle_{\text{repeats}} = \langle x(t) \rangle_t$, simplifying the expression.

In practice, this means that the correlation function on stationary data can be estimated by: **(i)** segmenting the original time series of contiguous blocks of length L; **(ii)** estimating $C(\tau)$ on each block; **(iii)** averaging the estimates of correlation function across blocks and computing standard error of this mean across the blocks to get the error bars. This procedure is valid, provided that the correlations in the original time series decay sufficiently quickly, i.e., that $C(L) \approx 0$, and thus different blocks of length L provide IID samples for estimating the correlation function.

Correlation functions are normally often normalized by dividing by the variance of the time series, σ_x^2, in which case $C(\tau = 0) = 1$, as in the case shown in Fig 23 for our `trace1` data.

When we talk about the *scale* (e.g., τ_0) of correlation function, we usually mean that $C(\tau) = C_0 \exp(-\tau/\tau_0)$, so that (at least within some interval of τ) correlation function decays sufficiently quickly for the time series values to become essentially independent for $\tau \gg \tau_0$.

This is in contrast to the "scale-free" case where correlation only decays as a power law in τ and thus correlations are long-ranged. Analysis of such data is beyond the scope of this course. The longest timescale in the problem is important because it sets the number of effectively independent samples to $\sim N/\tau_0$. The detailed structure of the correlation functions is, moreover, important when confronting various theoretical models with data, since the signatures of the dynamics, even in steady state when the mean is largely uninformative, shape correlation functions.

Homework 6. Compute the pairwise correlation function of log-intensity in the Ruderman natural images dataset, and plot it for the range from 1 to 100 pixels. Start by first estimating a 1D correlation function that is analogous to the one in Fig 23: the horizontal axis should be separation (in pixels) along x-image direction at a fixed y-image direction, and you can average over different random choices of y. Estimate error bars on this correlation function (and argue briefly why your method of assessing errors makes sense).
Since the images are two-dimensional, think about generalizing your 1D correlation function construction from above to two dimensions. A full correlation function should depend on 2 variables, i.e., on separation in x and separation in y coordinates of the image. Plot such 2D correlation function (you do not need to estimate its error bars), and by inspecting it, argue whether natural images can be isotropic (that is, have statistics that are independent of the direction in the image plane)? Would you expect isotropy in this image set, or in general, in natural images? Assuming isotropy, the correlation should be a function of one variable only (instead of 2 dimensions), that is, of the Euclidean distance between the two points in the image independent of direction; this would be a large simplification in the statistical structure of natural images.
Is there a characteristic spatial scale in the correlation functions you estimated from natural images? Do you expect there should be one (why or why not)?

7.4 Covariance matrices and Principal Component Analysis

We briefly mentioned the covariance matrix in the previous chapter and when discussing the multivariate Gaussian distribution, but let us now introduce it more formally in a way that is not specifically tied to the time series. Suppose we are given data vectors $\mathbf{x}$ of dimension D that are assumed to be drawn IID from some unknown $P(\mathbf{x})$, and let's further assume that these vectors have been mean-subtracted so that $\langle \mathbf{x} \rangle = 0$. Then the covariance matrix is defined as

$$\mathbf{C} = \langle \mathbf{x}\mathbf{x}^T \rangle = \frac{1}{N} \sum_{t=1}^{N} x_i^t x_j^t, \tag{33}$$

where the upper index t denotes the samples, while the lower indices denote the components of the data vectors. Since the covariance matrix and the means are the sufficient statistics of the multivariate Gaussian distribution of Eq (12) which can be seen as the simplest approximation to true $P(\mathbf{x})$ that matches the data in the first- and second-order statistics, we can interpret the covariance matrix graphically as estimating equi-probability contours of our data (the estimates would be correct if the data distribution really were exactly Gaussian). Setting the exponent of the multivariate Gaussian, $\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x}$ to some fixed value K defines a set of values for $\mathbf{x}$ that solve this quadratic equation for different values of K, and the locus of such $\mathbf{x}$ are ellipsoids centered at zero, whose axes (and their directions) are defined by the properties of $\mathbf{C}$.

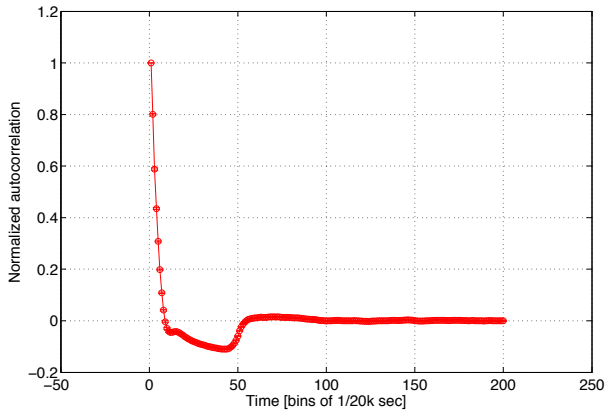

Figure 23: Correlation function, normalized to 1 at $\tau = 0$, of the `trace1` data with slow baseline fluctuations removed. At least two timescales are apparent: the fast timescale that decays over ~ 4 time bins, possibly due to correlated fast voltage fluctuations, and the intermediate timescale (with anticorrelation, $C < 0$) that extends until ~ 50 timebins, in line with the typical duration of spike waveforms. Beyond 100 time bins the correlation is zero within error bars, consistent with our previous tests of statistical dependence. Error bars are standard errors of the mean over correlation functions estimated across contiguous 200-time-bin blocks.

Since the axes of the ellipsoid are orthogonal, there exists a rotation in the space of $\mathbf{x}$ that will align the components of the coordinate system with the principal axes of the ellipsoid. In this rotated space there is no correlation between individual components of the data vectors, so the total variance in the data is equal to the sum of the variances along the individual axes (see Fig 24). How can we mathematically find the axes of the equi-probability ellipsoid?

Key to that are two properties of the covariance matrix. Equation (33) shows that the covariance matrix is symmetric, $C_{ij} = C_{ji}$. It is also positive semidefinite, that is, for any nonzero vector $\mathbf{y}$ it holds true that $\mathbf{y}^T\mathbf{C}\mathbf{y} \geq 0$. This is easy to see from the definition of the covariance, as

$$\mathbf{y}^T\mathbf{C}\mathbf{y} = \mathbf{y}^T\langle\mathbf{x}\mathbf{x}^T\rangle\mathbf{y} = \langle(\mathbf{x}^T\mathbf{y})^2\rangle \geq 0 \tag{34}$$

and the last inequality holds because the left-hand side is the expectation of a square term that must be nonnegative.

Symmetric real positive semidefinite matrices can be decomposed as

$$C = \sum_{i=1}^{D} \lambda_i \mathbf{v}_i \mathbf{v}_i^T = VDV^T, \tag{35}$$

where $\mathbf{v}_i$ are the D *eigenvectors* and λ_i are the corresponding eigenvalues, which are non-negative due to the positive semidefiniteness of the covariance matrices. This decomposition can also be written in matrix notation, where V is a matrix whose columns are eigenvectors $\mathbf{v}_i$, and D is a diagonal matrix whose diagonal entries are the eigenvalues, λ_i.

Consistent with our geometrical intuition (of equi-probability ellipsoid), the eigenvectors constitute an orthonormal system such that $\mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij}$ where $\delta_{ij} = 1$ for $i = j$ and 0 otherwise. D-dimensional vectors $\mathbf{v}_i$ point precisely in the directions of the principal axes of the ellipsoid; λ_i are the associated variances of the data in those directions.

Because eigenvectors of the covariance matrix form an orthonormal system, it is possible to

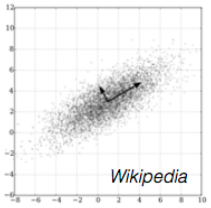

Figure 24: Scatter plot of points drawn from a jointly Gaussian distribution, denoted on it are the two principal axes of the equi-probability ellipsoid; the longer axis corresponds to the principal component with the dominant variance. In the (x, y) coordinate frame there is a correlation between x and y. Rotating the points into the eigensystem of the covariance matrix aligns the new coordinate axes with the principal axes; in the new coordinate system, the linear correlation between the new coordinates is zero.

expand every data vector in the eigenspace of the covariance matrix:

$$\mathbf{x}^t = \sum_{i=1}^{D} a_i^t \mathbf{v}_i, \tag{36}$$

where a_i^t are the components of the data vector in the new system. This representation has several nice properties. First, by construction, the components a are decorrelated, e.g., $\langle a_i^t a_j^t \rangle = 0$ for $i \neq j$.

Mathematical aside: Variance decomposition. The "total variance" in multivariate data is often defined as the second moment of the data vectors (assuming we centered the data vectors so that $\langle \mathbf{x}^t \rangle = 0$, where $\langle \cdot \rangle$ again denotes an empirical averege over samples indexed by t):

$$\mathrm{Var}[\mathbf{x}] = \frac{1}{N} \sum_{t=1}^{N} \mathbf{x}^t \cdot \mathbf{x}^t = \frac{1}{N} \sum_{t=1}^{N} \left(\sum_{i=1}^{D} a_i^t \mathbf{v}_i \right)^2 . \tag{37}$$

Using a bit of linear algebra you can show that:

$$\mathrm{Var}[\mathbf{x}] = \frac{1}{N} \sum_t \sum_i (a_i^t)^2 = \sum_{i=1}^{D} \mathrm{Var}[a_i]. \tag{38}$$

So the total variance in the data is the sum of variances of components a_i along eigendirections of the covariance matrix. This has to be the case, since the new components are decorrelated (their covariance has no off-diagonal elements), and so the total variance is the sum of variances along individual eigenvectors.
Can this total variance be related to eigenvalues? We manipulate the decomposition of covariance matrix as follows:

$$C = VDV^T = \frac{1}{N} \sum_{t=1}^{N} \mathbf{x}^t (\mathbf{x}^t)^T. \tag{39}$$

Multiply the above expression with eigenvector matrix to isolate eigenvalues in D:

$$D = \frac{1}{N} \sum_{t=1}^{N} (V^T \mathbf{x}^t) \cdot (V^T \mathbf{x}^t)^T. \tag{40}$$

The trace of the matrix D is the sum of eigenvalues, which, when you take the trace of the equation above, you can show to be

$$\mathrm{Tr}[D] = \sum_{i=1}^{D} \lambda_i = \frac{1}{N} \sum_t \mathrm{Tr}[(V^T \mathbf{x}^t) \cdot (V^T \mathbf{x}^t)^T] = \sum_{i=1}^{D} \mathrm{Var}[a_i]. \tag{41}$$

Together with Eq (38), this tells us that the total variance in the data is equal to the sum of the eigenvalues of the covariance matrix.

Second, if we sort the eigenvalues and eigenvectors in decreasing order of the eigenvalues, we can define *fraction of variance explained* by $K \leq D$ principal components as:

$$FV = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{D} \lambda_i}; \tag{42}$$

here, the denominator is the total variance in the data, while the numerator is the variance in the subspace subtended by K first components. This eigendecomposition is called the Principal Component Analysis (PCA) and is one of the basic data analysis tools. It serves the following purposes:

- If the large fraction of variance (e.g., $FV > 0.8$ or $FV > 0.9$) is captured by a small number of components, then the data is effectively much lower dimensional than D. As an

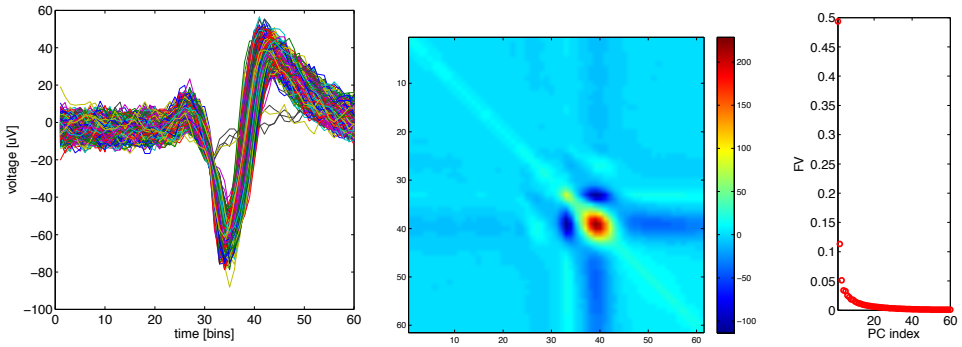

Figure 25: **Left.** 1200 isolated spikes from slow-baseline-subtracted `trace1` data. **Middle.** Covariance matrix estimated over spike samples at left. **Right.** Eigenvalues of the covariance matrix, sorted in decreasing order, and normalized to full variance (i.e., the sum of all eigenvalues).

extreme example consider three-dimensional data that lies on or close to a plane; in this case, the two principal components that span the plane will have large variance, while the third component (the plane normal) will have a much smaller eigenvalue. PCA is thus a simple way to estimate or bound the intrinsic dimensionality of the data.

- Since data can always be rotated into the eigensystem defined by the covariance matrix, this is a usual and useful preprocessing for many other tasks, performed after data centering (subtracting the mean). In this space, correlations become trivial. One can make a further processing step, which—after rotating the data—divides every rotated component with a square root of its corresponding eigenvalue, $\sqrt{\lambda_i}$. Such data is said to have been *whitened*: whitening transformations generate data that have trivial first- and second-order statistical structure, i.e., they are zero-mean and have unit (identity) covariance matrix.

- PCA can also be used for dimensionality reduction. In this case, data vectors $\mathbf{x}$ of dimension D are linearly projected into a subspace of the top K principal components, as defined by $\mathbf{y} = [\mathbf{v}_1 \cdot \mathbf{x}, \mathbf{v}_2 \cdot \mathbf{x}, \ldots, \mathbf{v}_K \cdot \mathbf{x}]^T$; the $\mathbf{y}$ are now only K dimensional but retain most of the variance in the original data.

The first steps of the PCA are illustrated in Fig 25. From slow-background-subtracted `trace1` data we collected ~ 1200 instances of spike waveforms by finding downward threshold crossings, and collected them together as $D = 60$ dimensional vectors, overlaid in Fig 25, left. Covariance matrix of this data is shown in Fig 25, middle. An element at position (t, t') in this matrix indicates the linear correlation across spike samples in voltage at t and t': thus, negative values at $(35, 40)$ mean, for instance, that spike waveforms that fluctuate downwards from the mean at $t = 35$ tend to fluctuate upwards at $t = 40$. The covariance matrix is clearly symmetric, and for early times, $t, t' < 20$ shows a time-translation invariant (stationary), band-diagonal structure. The width of the band is simply the timescale of the fast decay of the autocorrelation function, ~ 4 timebins, as shown in Fig 23 (make sure you understand why this is the case).

By decomposing the spike covariance matrix, we can plot the spectrum of (sorted) eigenvalues in Fig 25, right, which is very inhomogenous: the first 10 components capture more than 80% of the variance, suggesting that the dimensionality of the nominally 60-dimensional spike vectors could be reduced.

Before looking at such a reduction, let us remark on a few issues that can arise when empirically estimating covariance matrices from a finite number of samples:

- In the regime when the number of samples is less than the dimension of the data vectors, $N < D$, the covariance matrix will be rank-deficient, i.e., it will have zero eigenvalues. This is possible for new, very high-dimensional datasets, and the statistical tools used in that case need to be very different than the well-sampled, $N \gg D$ regime we are considering here. The spectral decomposition of Eq (35) can be replaced by Singular Value Decomposition (SVD), but this is beyond the scope of our course.

- Often, estimating the covariance will be difficult due to non-stationarities in the data. For instance, in financial data estimating the covariance between 500 stocks in the Standard & Poors index requires more than 500 trading days worth of samples (with one sample of stock prices at day close), which is ~ 2 years. On that scale financial markets usually undergo systematic and cyclical changes, making all related time series strongly non-stationary. Some of this can be corrected for by "de-trending" the data, but that comes with its own set of strong assumptions about the latent statistical structure responsible for systematic, non-stationary variation.

- Often, all dimensions of the data vectors of interest cannot be measured simultaneously. For instance, if gene regulatory network is given by the simultaneous activity of $D = 10$ nodes, but only subsets of two of these proteins can be fluorescently tagged and observed in a particular experiment, each experiment will only provide an estimate of the 2×2 submatrix of the whole $D \times D$ covariance. In the infinite data limit and with no experiment-to-experiment systematic variability, one could measure all these submatrices and simply assemble them, element-by-element into the full covariance. In practice, this does not work, and the resulting assembled matrix is often not a proper covariance (i.e., has negative eigenvalues). The problem of estimating the full covariance then needs to be seen as an inference problem of an unknown matrix given partial observations, and is solved by, e.g., maximum likelihood methods.

- Empirical estimation of covariance matrices from finite data will in general lead to a structure of eigenvalues that is a combination of the "true" structure that persists also in the infinite-data limit and structure due to finite sampling. For example, in Fig 25, right, the very small eigenvalues are non-zero, but that may be consistent with a spectrum of a matrix estimated from N appropriately random data vectors. In our case, we could take, for instance, $N = 1200$ clips of voltage time traces that *do not* contain spikes, and calculate their eigenvalue spectrum as a null distribution against which to test the statistical significance of the spike covariance matrix (i.e., this would test for the *excess* covariance structure due to the spikes, beyond the fast-correlation band-diagonal structure that is present in the whole `trace1` data).

- Results obtained by PCA can be difficult to interpret. For example, if we have a dataset of gene expression for different genes, the eigenvectors of the covariance matrix are a linear combination of gene expression profiles. How do you interpret such mixtures? This issue becomes worse when different *physical* quantities are analyzed (e.g., measurement of speed and power), where the quantities even have different units. PCA might allow for dimensionality reduction, fitting with fewer features, data visualization, etc. but it might not help with *understanding* of the data.

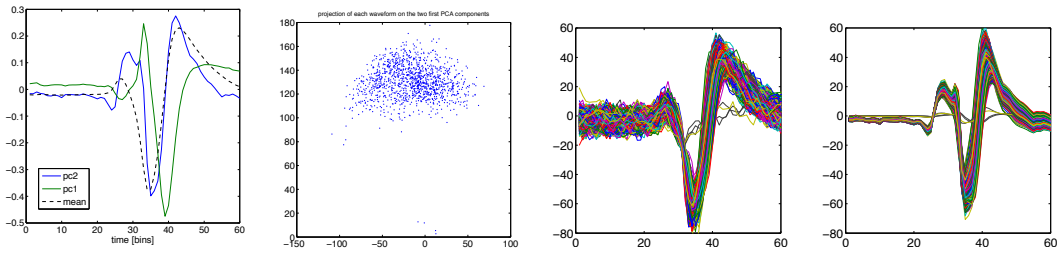

Figure 26: **Left.** The mean spike waveform (average over 1200 examples shown in Fig 25, left) in dashed black line. Green and blue curves show the first and second principal component, respectively. The first principal component corresponds closely to the time derivative of the mean spike waveform, which is expected if most of the variability between the data vectors (individual spike waveforms) consists of small temporal shifts of the prototypical or average waveform. Note that the principal components (blue, green) are normalized to unit norm and are orthogonal. **Middle left.** Projection of the original 60−D spikes into the two-dimensional space formed by the first two principal components; each spike is now represented in the plane as a 2D point. Coordinates of each point are obtained as a dot product between the individual spike waveforms and the principal components. **Middle right, right.** Original spike waveforms and their reconstructions using only the top two principal components that capture most of the variance.

- In some cases data must not only be centred ($\langle \mathbf{x} \rangle = 0$), but also standardized, i.e., z-scored. PCA finds the vectors in the multidimensional space along which the variance is maximal. Variance of a variable will increase in absolute terms if we, for example, change its unit (e.g., switching from km to m, the variance will go up by a factor of 10^6, and the corresponding covariances will be affected by a factor of 10^3). This will skew the PCA result towards the variable with the highest variance. By z-scoring the variables (by dividing individual data components with their standard deviations), PCA gets independent of measurement units, which, depending on the context, can be desirable. In this case, we effectively perform PCA decomposition on a correlation (and not covariance) matrix. Always be mindful of the effects of data preprocessing on the resuts of statistical analyses.

- PCA identifies *linear* statistical dependencies between observed variables. If the two variables are dependent, but not in a nonlinear manner (e.g., the first two eigenworm components in Fig 27), nonlinear transformations might exist that can make data more suitable for subsequent PCA analysis.

Figure 26 shows the first two principal components of the spike waveforms, and the projections of individual spikes onto those two components, yielding a large dimensionality reduction to two dimensions, which is, however, highly effective, as can be seen in Fig 26, right, from the reasonably good approximation of individual spikes based solely on two principal components.

A nice application of PCA for dimensionality reduction is in Stephens et al, *PLOS Comput Biol* **4** (2008), who analyzed the motion of the *C. elegans* worm by precise optical tracking. As shown in Fig 27, right, worm images were acquired and processed into body outlines from which midlines could be extracted. The worm shape could thus be described as a midline at every timepoint, but midlines are high-dimensional (they were represented as ~ 100 dimensional vectors of local curvature at every point). Stephens et al. found that all these worm shapes live in a much smaller dimensional space: 4 top eigenmodes accounted for more than 90% of the variance, as shown in Fig 27, middle left. Moreover, these "eigenworms" had a clear interpretation. The first two modes corresponded to the decomposition of the sinusoidal undulation of worm's body, characteristic of its typical motion pattern, and the magnitude of those two modes (i.e., the projection coefficients or dot products of the worm midline shape with the first two eigenvectors) were jointly distributed on the circle, as in Fig 27, middle right, so that they could be further summarized by one variable, the phase angle. Here is a good example of the space which is even simpler (only one dimensional), not two dimensional as PCA would suggest; PCA however cannot find this extra simplicity, since the remaining correlation between the first two mode coefficients is nonlinear (coefficients lie on the circle). The third "eigenworm" (principal component), shown in Fig 27, right, corresponds to extreme turns, known in biological literature as "omega turns", stereotyped manoeuvres that the worms use for large-magnitude changes in direction. This mode has a strongly kurtotic, non-Gaussian distribution, where most of the time its values are concentrated at zero (corresponding to the majority of worm shape snapshots when the worm is not doing omega-turns), with a few specific timepoints that correspond to large angle turns.

Homework 7. Analyze `trace2` signals (note that these are signals different from the `trace1` electrode we have been analysing up to now). First remove the slow baseline fluctuations by subtracting a smoothed version of the signal, as in the previous homework. Find the spikes by thresholding the signal: extract the waveforms extending 29 timebins before and 30 timebins after the spike and represent them as 60-dimensional vectors. By looking at these samples, do you think signals on the `trace2` electrode come from an individual neuron (compare with the analysis in Fig 26)? Compute the covariance matrix of samples and do PCA. Project the samples onto the first two principal components and show all spike samples as points on the plane (let's denote the first projection by y_1 and second by y_2). Can you clearly define clusters of spikes in this plane? How many spikes would you assign to each cluster? Plot the mean spike shape in each cluster. How would you find a direction vector in the (y_1, y_2) plane that best separates spikes from first vs second cluster (in other words, can you find a linear combination of y_1 and y_2, i.e., $z = Ay_1 + By_2$ for some coefficients A and B, such that z values for waveforms from the first cluster cleanly separate from z values for waveforms from the second cluster, e.g., by means of a threshold on z)? Do you know how to visualize this direction vector (or the linear combination) as a waveform in the original, 60-dimensional space?

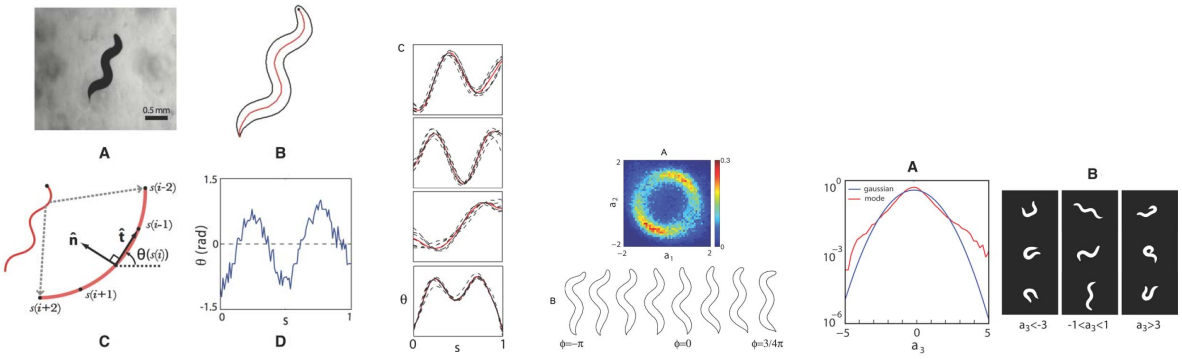

Figure 27: Analysis of *C. elegans* worm shape using the Principal Component Analysis, all figures reproduced from Stephens et al, *PLOS Comput Biol* **4** (2008). **Left.** Snapshot of the worm in A and its body outline (black) plus extracted midline (red) in B. The midline can be parametrized as a sequence of tangent angles θ in C, so that at each instant in time the shape of the worm is a vector θ_i with $i = 1, \ldots, D$ components. **Middle left.** The shape vectors lie in effectively only 4-dimensional space which captures $> 90\%$ of the variance; shown here are the top 4 eigenshapes, or "eigenworms". **Middle right.** The first two eigenworm coefficients trace out a joint distribution on a circle and are thus captured by a single parameter, or phase ϕ, as in A. Worm shapes corresponding to different values of the phase, in B, correspond to sinusoidal undulation of the worm's body during movement. Since PCA is a linear analysis, it is incapable of finding this single-dimensional representation; instead, two shapes (similar to the sine and cosine shapes) that form a quadrature pair $\pi/2$ out of phase are required to account for the sinusoidal motion. **Right.** The third eigenworm corresponds to sharp "omega" turns where the worm crosses its own path.

8 Exploring higher order structure in the data

8.1 K-means clustering

The PCA example in the homework demonstrated that dimensionality reduction can be followed up by a thresholding operation to separate examples of data points into classes, or *clusters*. The clustering problem is, however, more general: it can happen that clusters clearly separate in some low dimensional projection space, but this need not be the case. It is, therefore, useful to think of how to formulate the clustering problem more abstractly, in a way that does not depend on e.g., PCA.

Typically, clustering is viewed as a mapping from observed (possibly high-dimensional) samples, $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ into a set of clusters, S_i, where $i \leq N$. The mapping can either be deterministic, where each sample is assigned to precisely one cluster, or probabilistic, where the mapping is defined by a probability distribution, $P(\mathbf{x}_t|S_i)$, that the sample $\mathbf{x}_t$ belongs to cluster S_i. Probabilistic mappings can be turned into deterministic (hard clustering solutions) by simply picking the cluster assignment that maximizes the likelihood, i.e.,

$$S(\mathbf{x}_t) = \arg\max_i P(\mathbf{x}_t|S_i). \tag{43}$$

The central idea is to group into the same cluster those samples that are more similar among themselves than they are to samples from other clusters. The definition of "similarity" is either built into the clustering algorithm, or is a pluggable similarity measure (i.e., measure specified by the person doing the clustering, so far as it satisfies constraints posed by the clustering algorithm). There is a large number of clustering approaches, of which we will only describe possibly the best known and universally used one, the *k-means clustering*.

Results of the clustering procedure can be interpreted in multiple ways:

- **As an implicit or explicit model for the generating distribution,** $P(\mathbf{x})$. In this case, clustering can be viewed as an unsupervised learning procedure. As we will see, k-means clustering provides an implicit model of the data. Alternatively, probabilistic models such as the mixture of Gaussians, where $P(\mathbf{x}) = \sum_{i=1}^{k} \omega_i \mathcal{N}(\mathbf{x}; \mu_i, \mathbf{C}_i)$ a distribution over data is modeled as a linear superposition (with coefficients ω_i) of Gaussian distributions (with means μ_i and covariance matrices $\mathbf{C}_i$) can also be viewed as clustering methods (since every sample can be assigned into the most likely Gaussian component).

- **As a dimensionality reduction technique.** D-dimensional vectors $\mathbf{x} \in \mathbb{R}^D$ are being assigned to one of the clusters determined by the cluster index, i. Usually, the number of desired clusters k, $1 \leq i \leq k$, is the parameter of the clustering procedure, and setting it determines the "compression" of the data: from the limit where all samples are assigned to a single cluster and all information is lost, to the limit where each sample resides in its own cluster (when $k = N$), and no information about the samples is lost. Since the input vectors typically live in continuous space and the cluster-index output is discrete, this method can also be viewed as *vector quantization*, or a scheme to discretize / quantize real-valued vectors. Note that the question of whether the data possesses an *intrinsic* number of clusters, k^*, and how to infer that number, is usually treated separately from the clustering problem itself.

- **As feature- or dictionary-learning technique.** Here, one thinks of the clustering as identifying in the data instances of "prototypical" patterns that occur repetitively with some degree of variability from occurrence to occurrence. Once clusters are identified, the

cluster prototypes are usually taken to be either representative samples from each cluster, or some function thereof (e.g., the mean or centroid vector of the cluster). In this case, the objects returned still live in $\mathbb{R}^D$ space. These centroids can the form the basis of a classification algorithm: when a new sample is encountered, it is classified into the cluster whose centroid is closest (in the distance measure used for initial clustering) to the given sample.

It is clear that a well-designed clustering scheme should be sensitive to statistics in the data vectors $\mathbf{x}$ beyond the second order. The easiest way to see this is to consider the role of clustering as an unsupervised procedure that learns $P(\mathbf{x})$. If clustering were only learning statistics up to the second order, then it would treat as identical all the generating distributions that are equal to a multivariate Gaussian with a matched mean and covariance matrix. In particular, multi-peaked distributions (e.g., given by mixture-of-Gaussians) would be treated equally to a single Gaussian with average mean / covariance (which must have a single peak), rather than being identified with several clusters corresponding to mixture components.

This is perhaps best illustrated in the *k-means* clustering case. Here, the task is to partition N samples into k disjoint sets S_i, $i = 1, \ldots, k$, to minimize within-cluster distance D:

$$D = \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2, \tag{44}$$

where μ_i are cluster centers, also known as "centroids" (basically, these are simple center-of-mass points of each cluster):

$$\mu_i = \frac{1}{|S_i|} \sum_{\mathbf{x}_i \in S_i} \mathbf{x}. \tag{45}$$

Implementations of k-means clustering differ in how the optimization over D (by assigning samples to clusters, which is a discrete combinatorial problem) is performed. Before we look at such an algorithm, let's comment first on the structure of the distance measure D. Clearly, the assumed distance measure is Euclidean, which implies a local spherical symmetry on the space of $\mathbb{R}^D$ in which sample points live. Indeed, k-means clustering can be seen as making an implicit model of the generating distribution of the data:

$$P(\mathbf{x}) = \sum_{i=1}^{k} \omega_i P_0 \left(||\mathbf{x} - \mu_i|| \right), \tag{46}$$

where the weights ω_i are given simply by the number of points in each cluster, and the distribution P_0 only depends on the Euclidean distance between the sample and the cluster center. Cluster assignments fully specify the centroids and the weights. This simplicity makes k-means easy to scale to very large data with lots of clusters, but also causes certain inflexibilities in the data, especially compared to more powerful models, such as the mixture of Gaussians, as shown in Fig 28. Specifically, k-means as a tool to recover the underlying cluster structure in the data works best for equal cluster sizes, and for locally spherical cluster structure.

Because k-means operates on absolute distance measures, simple effects of different scaling of each coordinate axis can bias the clustering results. Sometimes, it makes sense to preprocess the data (e.g., by centering it and rescaling each axis by its standard deviation). It is also possible to fully whiten the data (as in the project) so that the transformed data, $\tilde{\mathbf{x}}$, is zero mean and unit covariance, i.e., $\mathrm{Cov}(\tilde{x}_i, \tilde{x}_j) = \delta_{ij}$; this, under certain sparsity assumptions of the data, recovers the Independent Components (see Section 8.3).

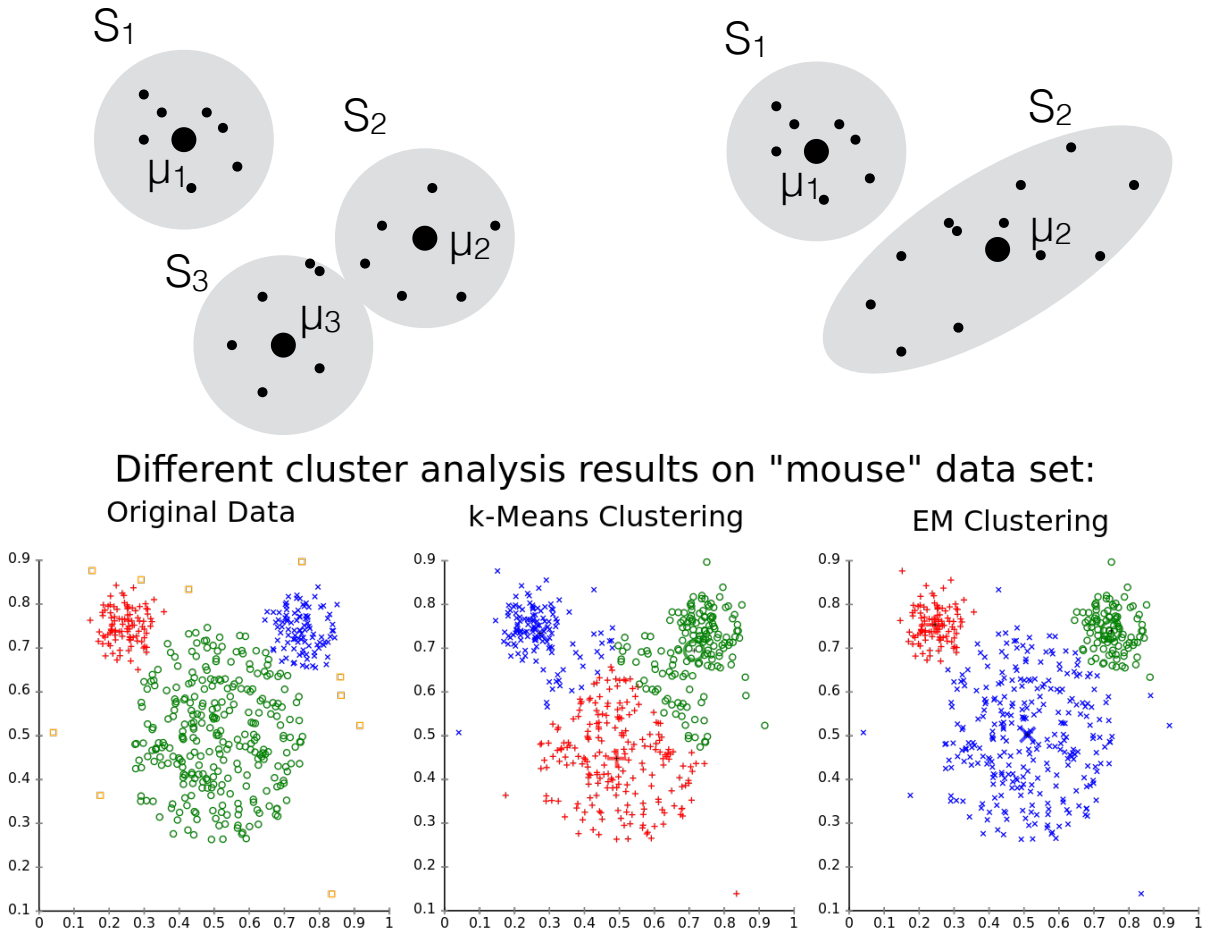

Figure 28: Schematic of statistical structure in data recovered using different clustering schemes. **Top Left.** The 3-cluster structure recovered using a k-means-like algorithm. Centroids are large black dots, sample points are small black dots. The group of points in the lower right is resolved into two spherical clusters. k-means finds cluster assignments (and thus centroids as well as cluster weights) by solving a discrete optimization problem, thereby implying a probabilistic model of data given by Eq (46). **Top Right.** In contrast, the more flexible mixture-of-Gaussians model may provide a better description of data in terms of only two components, where the second component is a larger, correlated set of points. Mixture-of-Gaussians explicitly assumes $P(\mathbf{x}) = \sum_{i=1}^{k} \omega_i P(\mathbf{x}|S_i) = \sum_{i=1}^{k} \omega_i \mathcal{N}(\mathbf{x}; \mu_i, \mathbf{C}_i)$ and infers parameters $\{\omega_i, \mu_i, \mathbf{C}_i\}$ by maximum likelihood from data (usually using an expectation-maximization algorithm), by solving a continuous optimization problem. Cluster assignments of sample points to mixture components ("clusters") proceed trivially, by exact enumeration as in Eq (43). **Bottom.** Similar example from *Wikipedia.org* where clusters are of very different sizes. Data (at left), generated by a true 3-Gaussian mixture model. Cluster assignment found by k-means with a particular initialization (middle). Cluster assignment by EM inference of a mixture recovers the correct partitioning (at right).

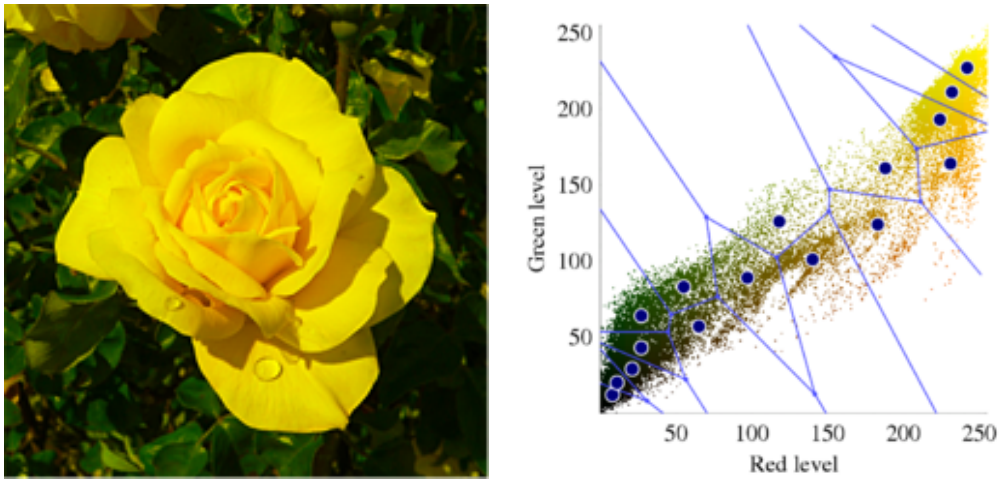

Figure 29: Example of k-means being used for vector quantization, reproduced from *Wikipedia.org*. **Left.** The original color image. **Right.** Scatterplot, where each point represents an image pixel plotted in a system with two color coordinates (green and red channels). K-means clustering partitions the points into 16 clusters that define the Voronoi tesselation of the plane (blue separating lines); centroids are denoted by thick blue dots. Note the density matching property (denser centroids where there are more data points). The image could be compressed by replacing the color coordinates of each pixel by the centroid coordinates of the corresponding cluster.

Finding the global minimum for k-means is intractable (NP-hard in general), but a simple iterative algorithm, known as *Lloyd's algorithm*, which provably converges to a local minimum exists:

1. Do initial assignments of each of the N sample points to one of the k clusters, compute centroids.
2. Assign each point to the cluster whose centroid is closest in Euclidean sense.
3. Given new assignments, recompute centroids.
4. Iterate 2 and 3 until the assignments keep changing.

Since steps 2 and 3 both decrease D, D is bounded from below, and there only exists a finite number of partitions of sample points into clusters, the algorithm will converge. This is not ensured for arbitrary choices of distance functions, but is true for Euclidean distance.

Typically, the algorithm is initialized either by randomly selecting the cluster index for each data point, or by randomly selecting k data points to be the initial centroids of the clusters.

While the assumptions of the k-means account for most of its failure modes when recovering true structure of the data, they also make k-means very useful for vector quantization. This is because k-means can be shown to perform density matching (i.e., making partitioning into finer clusters in regions where there are more data points, as a consequence of the implicit probabilistic model that favors equally-sized clusters), with clusters forming a Voronoi tesselation on the original space $\mathbb{R}^D$, as illustrated in Fig 29. We can apply k-means clustering also to the isolated spike waveforms from `trace2` to see how clusters can be extracted from data automatically, as shown in Fig 30. In this case, we don't have problems connected to different meaning or

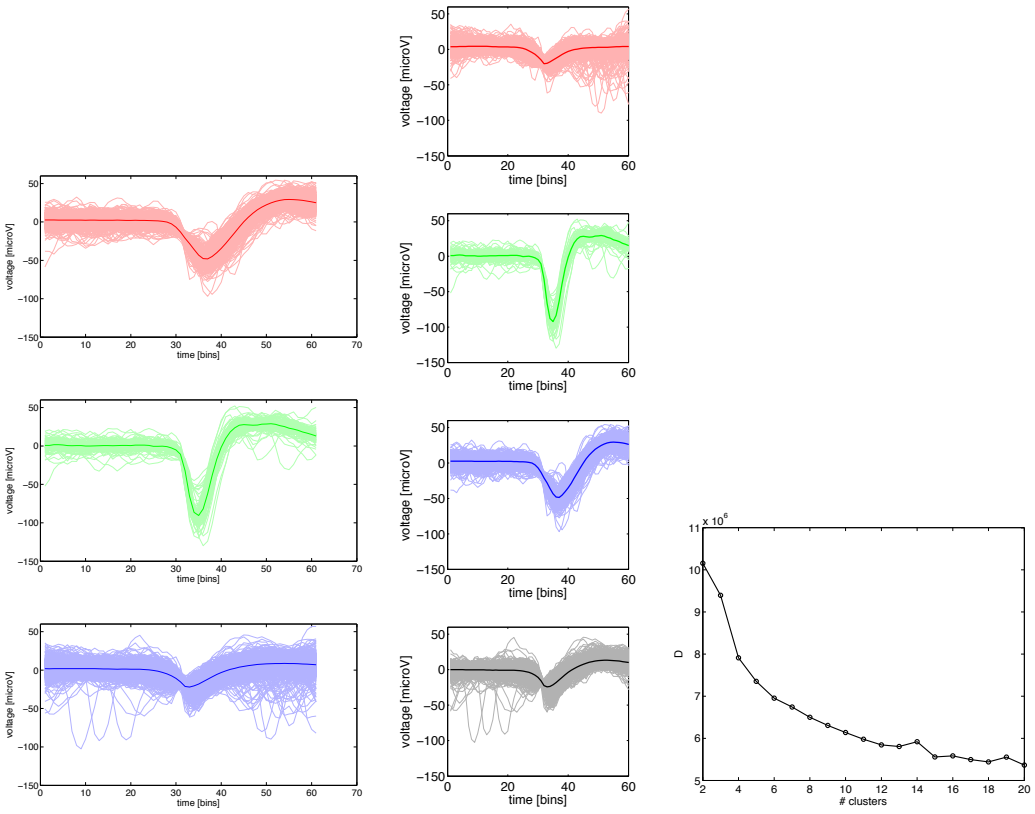

Figure 30: K-means clustering applied to spikes isolated from baseline subtracted `trace2` data. **Left.** Individual samples and the mean (centroid) assigned to 3 different clusters (red, green, blue). **Middle.** Same analysis for 4 different clusters. The green cluster stays the same as for the 3-cluster case; other three clusters reveal a finer discretization of the intermediate-magnitude spike waveforms. **Right.** The dependence of within-cluster distance D from Eq (44) on the number of clusters.

scale of different components of the data vectors, since all components are temporal samples from the same time series. It is, however, difficult to tell what is a "good" number of clusters: as expected, D decreases with k, and usual heuristics look for a point where this decrease slows down (on training or, if doing cross-validation, on test data). One can further look at how clusters identified at $(k+1)$ correspond to clusters identified at (k), i.e., by evaluating a similarity measure on the cluster centroids, see which clusters remain stable or split up when introducing one more cluster. Even if data is intrinsically multi-model (with $k*$ true clusters), having $k > k^*$ will allow k-means to further quantize existing clusters into finer partitions.

8.2 Multidimensional scaling and t-SNE

8.3 Independent component analysis

Independent component analysis (ICA), also known as "sparse coding", is a way of explicitly searching for certain kinds of statistical structure beyond second-order. It is motivated by the observation that many interesting signals have sparse structure, which can be most simply

captured by the following generative (probabilistic) model:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A} \cdot \mathbf{z}_t = \sum_{i=1}^{k} \mathbf{A}^{(i)} z_t^{(i)}, \\ z_t^{(i)} &\sim Sparse{-}dist, \end{aligned} \tag{47}$$

where $\mathbf{x}$ are D-dimensional, real valued vectors that represent our data samples, indexed here by t. $\mathbf{x}$ are a linear mixture of "template vectors" or "features" $\mathbf{A}^{(i)} \in \mathbb{R}^D$ (usually of unit norm), of which there are k in total, each of which can occur with a coefficient $z_t^{(i)}$; these coefficients vary from sample to sample. The dimensionality of feature space, k, need not be equal to the dimension of the feature vectors or $\mathbf{x}$, both of which are D-dimensional (it can be either smaller or larger). Importantly, considered as a generative model, coefficients $z^{(i)}$ are drawn IID from some "sparse distribution": a distribution whose kurtosis is substantially greater than for the Gaussian (which has a kurtosis of 3). Recall that sparse distributions have a large probability of observing values close to zero, and an above-Gaussian probability of observing values far in the tails. One particular example is the Laplace distribution,

$$P(z) = \frac{\lambda}{2} e^{-\lambda |z|}. \tag{48}$$

Equations (47), together with a choice of a sparse distributon as in Eq (48), specify a probabilistic model over vectors $\mathbf{x}$, which might be a good model for various data.

Let's consider for a while what we would expect for $P(\mathbf{x})$ if z were drawn IID from a Gaussian distribution. In that case, x would consist of a linear mixture of Gaussian random variables, which is always a Gaussian random variable. If z were zero mean, so would be the linear mixture, i.e., $\langle \mathbf{x} \rangle = 0$, and the covariance would be given by

$$C_{ij} = \langle x_i x_j \rangle = \langle \sum_k A_{ik} z_k \sum_l A_{jl} z_l \rangle = \sum_{k,l} A_{ik} A_{jl} \langle z_k z_l \rangle = \sum_{k,l} A_{ik} A_{jl} \delta_{kl} = \sum_k A_{ik} A_{jk}, \tag{49}$$

since z are IID unit normal. In this trivial case, one could estimate the covariance matrix, $\mathbf{C}$, from the samples, and look to recover the mixing coefficients, $\mathbf{A}$, under certain assumptions by various types of matrix decompositions (PCA, SVD, Cholesky, etc). In any case, however, the resulting $\mathbf{x}$ would be Gaussian and thus these simplest analyses on the covariance matrix would give the full account of the statistical structure.

The situation is different if the z are generated by a sparse distribution which is explicitly non-Gaussian. But how do we learn about the mixing coefficients and even the dimensionality of z, which are latent (unobserved) variables; in other words, we are given only $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the data samples, and know nothing of the latent structure.

Independent Component Analysis (ICA) is a tractable procedure to invert the generative model of Eqs (47). The basic idea is to consider applying a linear transformation, $\mathbf{w}$, to the data vectors,

$$z_t = \mathbf{w} \cdot \mathbf{x}_t, \tag{50}$$

and think about what the *empirical* distribution of z would look like were the $\mathbf{x}_t$ generated by the sparse model described above. As we argued before, if $\mathbf{x}$ were Gaussian, then any projection (i.e., z computed with any $\mathbf{w}$) would also be Gaussian. But this is true even more generally: for most choices of $\mathbf{w}$ one could expect the distribution of $\mathbf{z}$ to be close to Gaussian by the central limit theorem. Only if the $\mathbf{w}$ corresponded to one of the original features, $\mathbf{A}^{(i)}$, that generated the data, would the empirical distribution of z become very sparse (kurtotic). This can be turned

into a computational algorithm for ICA that we will introduce below: *optimize* over vectors $\mathbf{w}$ such that the empirical distribution of projections is as sparse as possible. Equivalently, this amounts to searching for projections / features of the data that are maximally non-Gaussian.

More precisely, the structure of the ICA algorithm that looks for a single sparse component is as follows:

1. **Center and whiten the data.** First, subtract the mean from all data vectors: $\mathbf{x}_t \leftarrow \mathbf{x}_t - \langle \mathbf{x} \rangle$. Second, compute the covariance matrix, $\mathbf{C} = \langle \mathbf{x}\mathbf{x}^T \rangle$ and decompose it into its eigensystem: $\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^T$, where $\mathbf{D}$ is the diagonal matrix of eigenvalues, and $\mathbf{V}$ is the matrix of eigenvectors that form an orthonormal basis, i.e., $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ (where $\mathbf{I}$ is the identity). Apply the whitening transform: $\mathbf{x}_t \leftarrow \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{x}_t$. After whitening, the data vectors are centered and have unit covariance matrix, so that there is no residual statistical structure of the first- and second-order.

2. **Select the sparsity function**, such that the expectation value of that function over the (transformed) samples is monotonically related to kurtosis of the data. For instance, using an empirical average of $f(z) = z^4$ would serve as a direct estimator for the fourth moment of zero-mean data, but other functions are also permissible and can indeed be more numerically stable for ICA.

3. **Solve a continuous optimization problem:**

$$\max_{||\mathbf{w}||_2=1} \langle f\left(\mathbf{w} \cdot \mathbf{x}_t\right) \rangle_t, \tag{51}$$

   i.e., maximize the empirical average of the sparsity function of the projection of the transformed data, over the projection vector $\mathbf{w}$, constrained to have unit norm (otherwise the sparsity function can typically always be maximized by simply taking any $\mathbf{w}$ and increasing its norm).

Before we continue with the examples of ICA in practice, we should discuss the following questions:

- **How are "sparsity" and "non-Gaussianity" related to "independence" in Independent Component Analysis?** As we argued above, for Gaussian data a *whitening transform* will remove all pairwise correlations. There are many possible whitening transforms (linear transformations of the data), of which the whitening in the PCA basis is only one particular choice. After whitening with any whitening transform, Gaussian data is spherically symmetric. In contrast, sparse data is not spherically symmetric after an arbitrary whitening transformation, but will be maximally statistically independent after the ICA transformation.

  Consider a set of linear transformations, $z_t^{(i)} = \mathbf{w}^{(i)}\mathbf{x}_t$ for all i, or in matrix notation, $\mathbf{z}_t = \mathbf{W}\mathbf{x}_t$, where $\mathbf{W}$ contains orthogonal vectors $\mathbf{w}^{(i)}$. Let $\mathbf{x}$ be whitened, zero-mean data vectors. Let's look for a transformation $\mathbf{W}$ that will make the components of $\mathbf{z}$ as statistically independent as possible. Formally, maximizing statistical independence means minimizing multi-information (a generalization of mutual information to multiple variables) between the components:

$$I(\mathbf{z}) = I(z^{(1)}, \ldots, z^{(k)}) = S[P_{\text{ind}}(\mathbf{z})] - S[P(\mathbf{z})], \tag{52}$$

  where $S[P_{\text{ind}}(\mathbf{z})]$ is the entropy of the factorized distribution for $\mathbf{z}$, i.e., $S[P_{\text{ind}}(\mathbf{z})] = \prod_i P_i(z^{(i)})$. Because entropy is unchanged by orthogonal transformations, $S[P(\mathbf{z})] =$

$S[P(\mathbf{x})]$, the entropy of the data; so this term does not depend on $\mathbf{W}$. Thus, maximizing statistical independence between components of $\mathbf{z}$ means minimizing the first term of Eq (52):

$$\max_{\mathbf{W}} \left(-\sum_i S[P_i(z^{(i)})] \right), \tag{53}$$

over the set of orthonormal matrices, $\mathbf{W}$. Because the projections have fixed mean (to zero) and variance (to 1), and the Gaussian distribution is the distribution with maximum entropy with a fixed mean and variance, minimizing the entropy of the projections will find projections of the data that have maximally "non-Gaussian" distributions. If, furthermore, the data are generated by a latent model as in Eq (47), maximally non-Gaussian will mean maximally sparse / kurtotic. This suggests that the information-theoretic formulation of ICA is more general (since it is looking for maximally independent projections without the assumption of sparsity).

- **Why pick kurtosis for the sparsity function?** $f(z) = z^4$ is by no means the only permissible function; indeed, it can be shown that any convex function of $u = z^2$ will suffice, and the choice is often made for computational convenience. Examples, which are more numerically stable for optimization, include $f(u) = -\log\cosh\sqrt{u}$ and $f(u) = -\sqrt{u}$. As pointed out before, one can also maximize (minus the) entropy of the projections. The degeneracy of these choices stems from the fact that there are many ways to measure non-Gaussianity; many of these yield identical results on the data, but not all (if, e.g., we were maximizing skewness, not kurtosis).

- **How to generalize the single-component ICA algorithm above multiple independent components,** as in the generating model of Eq (47)? Theoretically, this is easy. Suppose we have already identified k independent components, $\mathbf{w}_\mu$, with $\mu = 1, \ldots, k$. We can look for the subsequent component $\mathbf{w}_{k+1}$ by carrying out the optimization of Eq (51) under constraints such that $\mathbf{w}_{k+1} \cdot \mathbf{w}_\mu = \delta_{k+1,\mu}$ for all μ. Technically, this can be carried out in different ways, e.g., by Gram-Schmidt orthogonalization during optimization steps. Note that Eq (51) is quite similar to a formulation of PCA where we look for projections of data vectors that maximize the projection variance (this, by definition, is the first principal component): if, in the ICA algorithm, the data were *not* whitened (only centered), and $f(\cdot)$ were selected as the variance estimator, $f(z) = z^2$, the ICA procedure would become the PCA procedure, and finding multiple principal components would also proceed by sequential optimization, enforcing orthogonality of principal components through Gram-Schmidt orthogonalization. While PCA generates a clear ordering among principal components (in decreasing order of variance), ICA, however, generates no similar ordering into "more" or "less" important independent components.

  Alternative formulations of ICA allow all components to be extracted in parallel (as in the so-called infomax ICA); for more information see an excellent reference on natural scene statistics and ICA [5].

9 Excursion: ICA and the heart rate

The analysis and explanation in this section is courtesy of Bor Kavčič.

Different dimensionality-reduction techniques are a powerful tool for data analysis. Below we will discuss an example that demonstrates the power of Independent Component Analysis (ICA). This techniques decomposes the measurements $\mathbf{x}(t)$ into the source signals $\mathbf{s}(t)$. The measurements and sources signals are linearly mixed by a mixing matrix $\mathbf{A}$ as $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$. This mixing matrix is unknown. We would like to obtain an unmixing matrix $\mathbf{W}$, given as $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}$. This matrix would convert measurements into original signals: $\mathbf{s}(t) = \mathbf{W}\mathbf{x}(t)$. We will attempt to find rows of matrix $\mathbf{W}$ such that the scalar product $\mathbf{w}_i^{\mathrm{T}}\mathbf{x}(t)$ will be as non-Gaussian as possible. This principle underlies ICA; algorithm of estimating the $\mathbf{w}_i$ will attempt to maximize the *negentropy* $I(y) = H(y_{\mathrm{Gauss}}) - H(y)$ where $y_{\mathrm{Gauss}} \sim \mathcal{N}(0,1)$ and $H(p) = -\int p(y)\log P(y)\mathrm{d}y$ is the entropy of probability distribution $p(y)$.

To expand on the topics covered in the lectures, we will implement FastICA algorithm [6, 7] to analyze an actual dataset. In this case, the will approximate the negentropy [since we do not know the $p(y)$] with an appropriate non-quadratic function $G(y)$: $I(y) \approx [\langle G(y)\rangle - \langle G(y_{\mathrm{Gauss}})\rangle]^2$. There are different functions that can tick the requirements, but we will use $G(y) = \log\cosh(y)$ with $G'(y) = \tanh(y)$, and $G''(y) = 1 - \tanh^2(y)$. We will comment on the algorithm as we work out the example.

9.1 ECG data

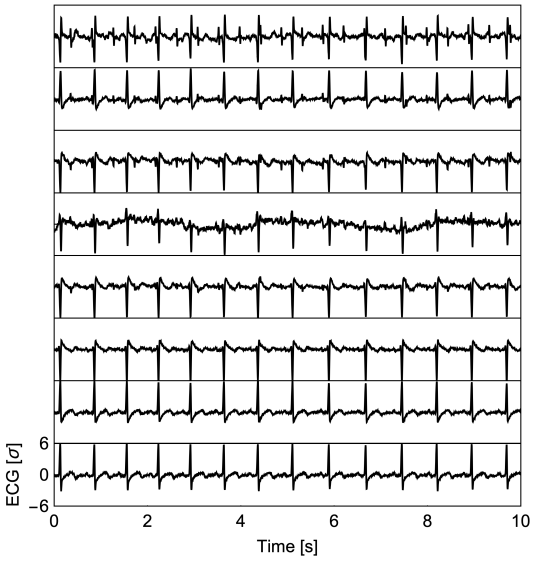

Figure 31: **Standardized ECG readings.** Readings from eight electrodes. Data were centered and rescaled by corresponding standard deviations.

We will use the data from an electrocardiogram of a patient. The dataset contains 10 s of data, sampled at 250 Hz from 8 electrodes placed on different points on the body (Fig. 31). This example is taken from Ref. [7] and data from website [8]. We evaluate the covariance (correlation, due to standardization) matrix $\mathbf{\Sigma}$. We calculate the corresponding eigenvalues λ_i and eigenvectors $\mathbf{u}_i$, and we construct a diagonal matrix $\mathbf{\Lambda}^{-1/2} = \mathrm{diag}\{1/\sqrt{\lambda_i}\}$ and orthogonal matrix $\mathbf{U}$ of the eigenvectors. We then transform the standardized data as $\tilde{\mathbf{x}}(t) = \mathbf{\Lambda}^{-1/2}\mathbf{U}^{\mathrm{T}}\mathbf{x}(t)$. This transformation yields the set $\{\tilde{\mathbf{x}}(t)\}$, whose covariance matrix is $\mathbf{Id}$. The data has been

deprived of all linear dependencies.

With this data at hand, we can start the FastICA algorithm with the goal of finding the $\mathbf{W}$. We begin by first deciding how many signals we seek. We will try to find r independent signals. We initialize the algorithm by randomly picking r vectors $\mathbf{w}_k$ with (in our case) eight components. These vectors form a matrix $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_8)^{\mathrm{T}} \in \mathbb{R}^{r\times 8}$ We normalize those vectors such that they have a unit length. Then, we:

1. Orthogonalize the matrix as $\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^{\mathrm{T}})^{-1/2}\mathbf{W}$. Inverse square root is of the matrix calculated from, say, Cholesky decomposition and its singular decomposition. Alternatively, one can iterate $\mathbf{W} \leftarrow (3/2)\mathbf{W} - (1/2)\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{W}$ until converged.

2. For $k = 1, \ldots, r$ we compute new vectors as

$$\mathbf{w}_k \leftarrow \frac{1}{n}\left[\sum_i \tilde{\mathbf{x}}_i G'\left(\mathbf{w}_k^{\mathrm{T}}\tilde{\mathbf{x}}_i\right) - \mathbf{w}_k \sum_i G''\left(\mathbf{w}_k^{\mathrm{T}}\tilde{\mathbf{x}}_i\right)\right], \tag{54}$$

   which we then normalize.

3. Repeat steps 1 and 2 until sufficient convergence is achieved. This can be checked by, *e.g.,* computing the scalar product between $\mathbf{w}_k$ from current and previous iteration. Sufficient convergence is achieved when this scalar product does not differ from 1 by some small real number.

Upon finding the estimate for $\mathbf{W}$, we calculate the r signals as $\mathbf{S} = \mathbf{W}\tilde{\mathbf{X}}^{\mathrm{T}}(t)$ – (Fig. 32).

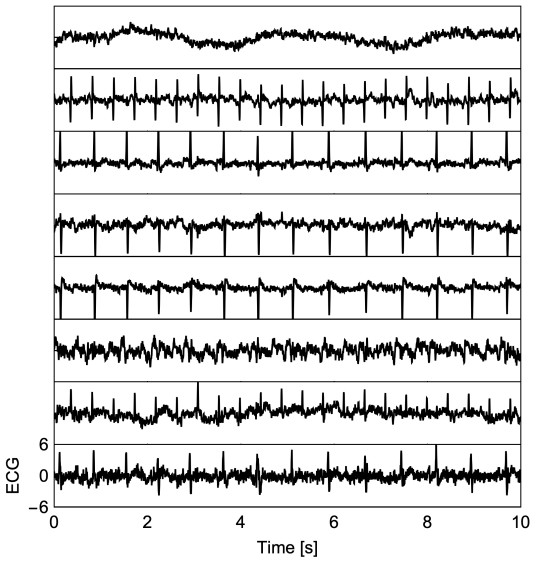

Figure 32: **Estimated independent signals.** Topmost signal is a slowly varying background (breathing?). Signals 2 and 7 have a distinctly faster heartbeat compared to the rest: they belong to the heartbeat of the unborn child, while the remaining ones are mother's heartbeat.

The methods extracted the background as one of the components, and it isolated two distinct ECG traces: one with a slow and one with more rapid heart beat – the ECG was performed

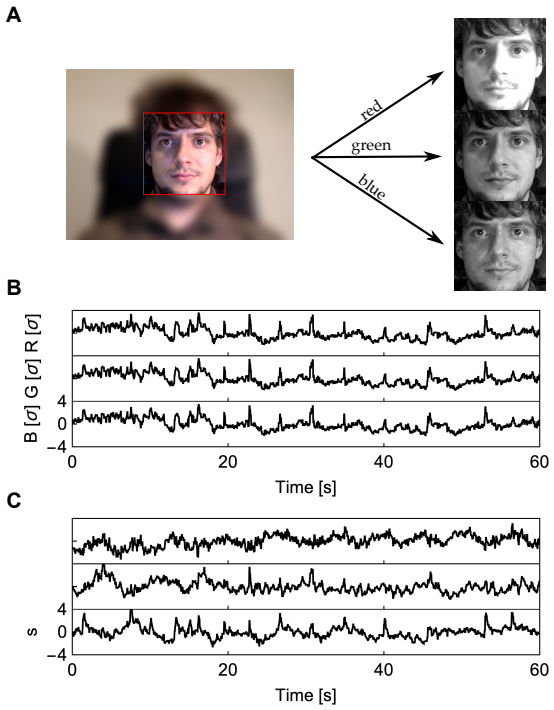

Figure 33: **Webcam image decomposition and reconstruction of signals. (A)** An example frame as recorded by the webcam with isolated face patch (left); this segment is decomposed into red, green, and blue component (right). **(B)** Standardized RGB signals. Note, that the signals are nearly identical to the naked eye. **(C)** Reconstructed signals. Oscillations are most prominent in signal 2 (middle).

on a pregnant mother. The child's heart beat is almost invisible on the original data – Fig. 31. Note, that this method loses the scale of the data and the order of signals. Contrary to PCA, the "strength" of the signal is not conveyed by a scalar (as is with the eigenvalues in PCA).

This segment demonstrated how ICA can uncover signals that are well "hidden" in the data. The next section will illustrate a more "MacGyver" technique that uses FastICA at its core to extract a quantity of interest.

9.2 Measuring heart rate at home

Above we used professionally-measured heart rate data obtained using precise and purpose-built device: ECG machine. We will now demonstrate how one can determine the heart rate using ICA-based method. We will use the principle of photoplethysmography (PPG). Without going too much into detail, the technique is based on the variable light path length due to pulsing lumen of vessels in upper layers of the skin. If we record the series of images of a skin patch, these changes are observed in the amount of light reflected. This version of PPG is called reflective photoplethysmography. We will use the signal from a normal webcam on a laptop to do this measurement, loosely following Ref. [9].

Webcam sensor is made of three different light sensor arrays with different spectral sensitivities, traditionally covering red, green, and blue channel (RGB). This detector will measure the mixture of PPG signal, changes in the ambient lighting, movement (blinking, breathing, eye movement). We will try to uncover the PPG signal or at least its frequency signature.

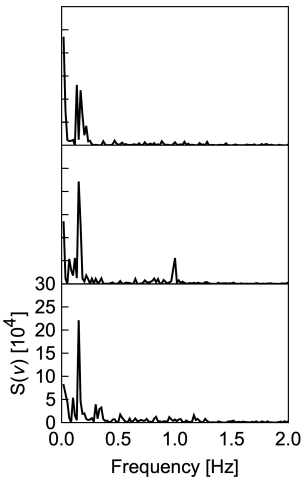

Figure 34: **Power-spectrum of the signals.** There is a distinct peak at ~ 1 Hz in the power-spectrum of signal 2, indicative of the PPG signal. Highest peak is around 0.2 Hz, which is roughly the frequency of breathing.

We record a 60 s video with a certain frame rate (25 Hz) in resolution 640×480 pixels, resulting in 1,500 frames. We isolate individual frames from the video and algorithmically find a face in each image. We split each face-patch into RGB channels; we collapse these separated channels into a single number by averaging each channel data. This yields three measured time traces, each with 1,500 data points.

We then follow the procedure described in the previous section. We standardize the data, and we further whiten it to obtain uncorrelated traces. We then run a FastICA algorithm on it, parsing three independent "signals." We see that one of them contains oscillations in it. To quantify the frequency of these oscillations, we perform a discrete Fourier transform given as

$$f_k = \sum_{n=0}^{N-1} x_n \mathrm{e}^{-(2\pi \mathrm{i} k n)/N}. \tag{55}$$

By looking at its power spectrum given as $f_k f_k^*$, we then seek an apparent peak around 0.5-3 Hz (equivalent to 30-180 beats per minute).

10 Mini-project

(5 points) Collect many 7×7-pixel patches of natural images from the Ruderman dataset and represent the patches as 49-component vectors of log-normalized luminance (i.e., prior to collecting the patches, take the log of the absolute luminance and subtract the mean log luminance separately for every image). Do principal component analysis on the vectors, show principal components (eigenvectors) as 7×7 image patches sorted in decreasing order of eigenvalues, and plot the corresponding sorted eigenvalues. Do you see any regularity in the eigenvectors and eigenvalues? Can you qualitatively explain what you see in relation to the spatial symmetries in our physical world from which the images have been sampled?

(3 points) Let's find the higher order statistics in small image patches that are characteristic of natural scenes: that is, we are looking for statistical structure that is present beyond the mean and the covariance structure among, e.g., patches of 7×7 pixels. To do that, we will use a poor man's version of ICA (Independent Component Analysis). As in the previous problem, collect T (say $T = 10000$) 7×7 pixel patches from all images of the Ruderman dataset (you don't need to do the log transform), and represent them as a 10000×49 matrix $\mathbf{X}$; every row, $\mathbf{x_t}$, of this matrix represents a 49-dimensional patch, with patches enumerated by $t = 1, \ldots, T$. First, let's remove the mean over the samples (so that the data is centered, or zero mean), and then whiten the data. With patches, the most convenient way to whiten (completely remove pairwise correlations over the collection of patches) is to compute the eigenvectors, $\mathbf{V}$, and eigenvalues, $\mathbf{D}$, of the 49×49 covariance matrix of $\mathbf{X}$, and then transform $\mathbf{X}$ into $\mathbf{XVD^{-1/2}V^T}$. If the inversion of the diagonal matrix is nearly singular, use a numerically stable version (e.g., a pseudo-inverse), implemented in most numerical packages. Convince yourself that the covariance of the transformed data is equal to identity. On the first- and second-moment level, the data is now as boring as it can get: it looks like an uncorrelated (= no linear dependencies left) unit-variance Gaussian. Our next task will be to check if the data contains some additional higher-order statistical structure.

To look for higher-order statistical structure inspired by ICA, we can proceed as follows. We will look for a 7×7 linear filter (represented as a 49-component vector) $\mathbf{w}$ of unit norm ($||\mathbf{w}|| = 1$), with the property that the projections of the data on this filter, $\mathbf{Xw}$, will be *sparse*: mostly very close zero, but sometimes taking on very large (positive or negative) values, relative to what would be expected from a Gaussian. One statistic with that property, for example, would the the kurtosis of $\mathbf{Xw}$, and we could try to find the filter by maximizing the kurtosis of the projections. Note if the data, $\mathbf{X}$, truly were Gaussian, there would be no such special filter: every linear projection of a multivariate Gaussian is Gaussian distributed, so this procedure really is looking for features in the data that are higher-order, beyond the Gaussian model. Maximizing kurtosis, however, is numerically unstable, since kurtosis strongly depends on the outliers (although you can try and see if you find anything interesting by using kurtosis as the objective function). Another way to look for filters with sparse projections is to maximize an average over patches of some nonlinear transformation of the data. Experience has shown that maximizing the average of $(-\log\cosh(\mathbf{Xw}))$ over patches is quite stable numerically.

(5 points) We can phrase our problem as follows: **(i)** center and whiten the collection of image patches as described above; **(ii)** given some filter $\mathbf{w}$, normalize it to have unit norm (this is important), compute the projection, $\mathbf{w} \cdot \mathbf{x_t}$, of every whitened patch, $\mathbf{x_t}$, onto the filter $\mathbf{w}$, compute $-\log\cosh(\mathbf{w} \cdot \mathbf{x_t})$ and average this over T projections, i.e., compute $f = -T^{-1} \sum_{t=1}^{T} \log\cosh(\mathbf{w} \cdot \mathbf{x_t})$; **(iii)** adjust the filter parameters $\mathbf{w}$ to maximize f, while keeping $\mathbf{w}$ fixed to unit norm. There are multiple ways to do this, and you are free to implement any of them provided you understand how they work. We make two suggestions, and you can choose to do either. **In the first suggestion,** you could acquaint yourself with the plain vanilla nonlinear optimization that your software supports (e.g., `fminunc` of Matlab), program the function above to be extremized, choose an initial random guess for $\mathbf{w}$, and run the optimizer until it converges to the local extremum. This will give you one 49-component filter that generates sparse projections over natural images—also known as a "feature"—which you can visualize as a 7×7 image patch. Run the optimization several times with different initial starting values and plot some examples of the features you find. **In the second suggestion,** you can implement the canonical FastICA algorithm described in Section 9.1. Here, too, find and plot multiple orthogonal sparse features. **In both cases,** how would you qualitatively describe the features you find?

(5 bonus points) Are pairwise correlations between pixel luminances in natural images sufficient for perception? That is, if we modify natural images so that their pairwise correlations are maintained but all other correlations are destroyed, do we still perceptually recognize the original image? Conversely, what if we modify natural images so that their pairwise correlations are removed but higher-order correlations are maintained? Do this problem only if you are familiar with the Discrete Fourier Transform (DFT) and its use.

To test this, take the first image from the Ruderman dataset and construct two new images from it: a "whitened" image, and a "phase-scambled" image, corresponding to the two hypotheses above. For the whitened image, do 2D Fourier transform of the original image, normalize the absolute values of each of the Fourier components to the same unit value, and transform the image back into the real domain. What is the correlation function of the transformed image? Show the original and the transformed image – is the transformed image still recognizable?

To create a phase-scrambled image, again Fourier transform the original image. Keep the amplitudes of Fourier components untouched, but assign each component a random phase (recall: each component is a complex number $z = x + iy = Ae^{i\phi}$, where $\phi \in [0, 2\pi)$ is the phase, and $A = \sqrt{x^2 + y^2}$ is the amplitude). When you scramble phases, you need to be careful, since not all phases in the signal are independent of each other – because the original image is real, the phases of certain Fourier components are related by conjugation... After phase-scrambling, transform the image back (as a check, you must obtain a real-valued image). Is the image still recognizable? What of the original image do you still recognize, now that pairwise correlations (encoded in the amplitude) have been maintained but higher-order correlations (encoded in the phase) have been removed?

What does this imply for perceptual salience: what kind of correlations must our visual systems (and machine vision applications) really extract? See also Ref [3].

Interestingly, while the retina approximately removes pairwise correlations as you did by whitening, the subsequent stage of processing in the brain, the visual cortex, looks for these sparse higher-order statistical features. A classic work on this topic that you may want to skim through is Ref [4].

References

[1] Marre O, Amodei D, Deshmukh N, Sadeghi K, Soo F, Holy TE, Berry MJ II (2012) Mapping a complete neural population in the retina. *J Neurosci* **32:** 14859.

[2] Ruderman DL, Bialek W (1994) Statistics of natural images: Scaling in the woods. *Phys Rev Lett* **73:** 814.

[3] Oppenheim AV, Lim JS (1981) The importance of phase in signals. *Proceedings IEEE* **69:** 529.

[4] Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381:** 607.

[5] Hyvärinen A, Hurri J, and Hoyer PO, *Natural Image Statistics* (Springer, 2009).

[6] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis", IEEE Trans. Neural Networks **10**, 626 (1999)

[7] Simon Širca and Martin Horvat, "Computational Methods for Physicists", Springer (2013)

[8] https://homes.esat.kuleuven.be/∼smc/daisy/daisydata.html

[9] Žiga Zaplotnik, master thesis, University of Ljubljana (2014)