1 Motivation

One of the key problems in non-parametric data analysis is to infer good models of probability distributions, assuming we are given as data a finite sample from that distribution. This problem is ill-posed for continuous distributions, even when they are low-dimensional: with finite data, there is no way to distinguish between (or exclude) distributions that are not regularized a priori, for example, by assuming that they are smooth in some sense. The question then becomes about how to formulate the problem of choosing the “best” distribution among the distributions that are “sufficiently smooth” mathematically; intuitively, more data should allow us to consider less smooth distributions.

One way of constructing such continuous distribution estimates is by kernel density estimation (KDE), which literally involves smoothing (or convolving) the data with a particular kernel function. Assuming a metric on the state space, the kernel specifies how far away from individual data points the corresponding contributions to estimated distribution extend. Another possibility is to describe the data with a parametric family of distributions that is sufficiently rich so that, in the large sample limit, it can describe an arbitrary smooth distribution. A well-known example from this class are Mixtures of Gaussians. Both of these approaches are reasonably simple to implement, both can be seen as instances of probabilistic (maximum likelihood or Bayesian) inference for their parameters, and both belong to a standard set of unsupervised learning tools for data exploration / modeling.

Yet another versatile framework for modeling both continuous and discrete distributions of potentially high dimensionality given a finite sample is the maximum entropy (ME) approach. Here, we are looking for the most random distribution (= maximum entropy) that exactly reproduces a chosen set of statistics which can reliably be estimated from data. The assumption of maximum entropy is a formal version of Occam’s razor: one chooses distributions of a particular form that contain a minimal amount of structure that is nevertheless sufficient to explain selected aspects of observations (constraints). Seen another way, the maximum entropy assumption is just another form of regularization, the intuition being that the most regular smooth distributions are uniform distributions which maximize the entropy by definition. What makes maximum entropy models special are certain uniqueness theorems and links to information-theoretic quantities.

Going beyond these standard tools, one can consider unsupervised deep learning models, restricted Boltzmann machines, field-theoretic methods for learning probability distributions and Gaussian Processes (in a supervised setting) that we will consider in the next lecture.
2 Goals

- Introduce kernel density estimation for 1D and 2D continuous data. Illustrate the optimal choice of kernel smoothing.
- Apply KDE to Sachs data.
- Introduce Mixtures of Gaussians and the expectation-maximization algorithm.
- Introduce maximum entropy models; examples of several well-known maximum entropy distributions.

3 Data

Single-cell measurements of network activity in a MAP signaling pathway of human T cells; for a primary reference on this dataset, consult [1]. Signaling nodes were immunostained in different color channels such that a FACS readout of intensity in each channel corresponds to the activity level of a given node. Each measurement therefore corresponds to an 11-dimensional readout for a single cell. Cells were imaged in 9 conditions, at 600 cells per condition; conditions are arranged in order in the datafile (i.e., columns 1 to 600 are condition 1, columns 601 to 1200 condition 2 etc.). A condition was a particular combination of chemicals that either stimulate or repress various parts of the signaling network. In the original publication, interaction diagram between the 11 nodes was reconstructed from this dataset. The nodes in the file are arranged as follows: RAF, MEK, PLCgamma, PIP2, PIP3, ERK, AKT, PKA, PKC, P38, JNK.

4 Kernel Density Estimation (KDE)

Consider a set of $N$-dimensional data vectors $\mathcal{D} = \{x^t\}, \ t = 1, \ldots, T$, that we assume have been drawn from a continuous probability distribution $P(x)$ which we don’t have direct access to, but would like to model. Formally, one could construct an empirical approximation to the distribution,

$$
\hat{P}(x) = T^{-1} \sum_{t=1}^{T} \delta(x - x^t),
$$

where Dirac-delta functions represent infinitely sharp peaks centered at the $T$ sample points. It is clear that this is a very bad model for the distribution if we have a prior expectation that the distribution should be smooth. As an extreme example, consider the case where $x$ are actually drawn from a uniform (true) distribution over a finite domain, whereas the empirical distribution is composed of infinitely sharp delta-peaks. However, we can generalize the form of Eq. (1) and write instead

$$
\hat{P}(x) = T^{-1} \sum_{t=1}^{T} K_\theta(x - x^t),
$$

where $K_\theta$ is a particular kernel function optionally parametrized by a set of parameters, $\theta$. A common choice is a Gaussian kernel, where

$$
K_C(x) = (2\pi)^{-N/2}(\det C)^{-1/2} \exp \left( -\frac{1}{2} x^T C^{-1} x \right);
$$

2
here, $C$ is the covariance matrix. Intuitively, each data point is now smeared into a ball whose size is set by $C$, and each single data point induces a nonzero probability across the whole domain of $x$. Choosing the variances of the Gaussian involves a tradeoff: too small a variance brings us towards non-smooth distributions (delta functions in the limit), and too large of a variance will smooth over the interesting features in the distribution. How does one set the parameters $\theta$ of a kernel (and what kernel does one choose)? There is no universal answer, but there are (at least) two general approaches. In the first approach, one assumes that the data is generated by some distribution (e.g., Gaussian), and then computes the best smoothing kernel parameter $\theta$, which minimizes some measure of error (e.g., L2 norm between the true distribution and the one obtained from KDE; or Kullback-Leibler divergence between the two distributions). An example is the “Silverman’s rule of thumb” for selecting the Gaussian kernel parameter: $\sigma \approx 1.06\Omega T^{-1/5}$. This is derived assuming that the data is generated by an underlying 1D Gaussian distribution with std $\Omega$, and we want to minimize the least square error between the real distribution and the KDE model.

In an alternative approach one can use the cross-validation to set the smoothing. In cross-validation, which works well independently of the assumptions about the generating distributions, the data points that support the distribution in Eq. (2) are taken as the training data, while the parameters $\theta$ are set such as to maximize the likelihood on withheld (testing) data.

5 Mixture of Gaussians

For Gaussian mixtures, we look for a model of the form

$$P(x|\theta) = \sum_{i=1}^{M} w_i \mathcal{G}(x; \mu_i, C_i).$$

(4)

Here, $\mathcal{G}$ are $N$-dimensional Gaussian distributions with means $\mu_i$ and covariance matrices $C_i$, and the model is written as a weighted average (with weights $w_i$ such that $\sum_{i=1}^{M} w_i = 1$) of $M$ Gaussians. The parameters of the model $\theta = \{w_i, \mu_i, C_i\}$ (that is, $M-1$ weights plus $M \times N$ parameters for the mean and $M \times N(N+1)/2$ parameters for the covariance) can be estimated using maximum likelihood / Bayesian inference, or using a version of the expectation-maximization (EM) algorithm.

We will not use the mixture models, but it is instructive to discuss the similarities and differences to KDE, where the distribution is also a superposition of Gaussian functions. With mixtures, one decides on the number of gaussians that best describes the data (e.g., by cross-validation), but many times true distributions can be described by just a few Gaussians (e.g., a two-peaked distribution might be well described by a two-gaussian mixture); the point here is to estimate well the mixing weight and the (possibly high dimensional) means and covariance parameters. In contrast, in KDE the number of gaussians is given by the number of data points, their width is chosen by cross validation (with the symmetry of the gaussians assumed, i.e., by fixing the covariance to be a multiple of the identity matrix, in which case KDE is a one-parameter inference problem), and the mean is fixed by the data point itself.

5.1 Expectation Minimization (EM) algorithm

An interesting feature of the mixture models is that they can be viewed as an instance of hidden (latent) variable models: for every data point $x^t$, there is a hidden variable $z^t$ (the “cause”), taking on integer values that specify from which mixture component the data point was drawn.
This makes it possible to rewrite the probability distribution in Eq. (4) and construct the likelihood of the following extended form:

\[ P(D, z|\theta) = \prod_t \sum_i \delta(z^t - i) w_i \mathcal{G}(x^t; \mu_i, C_i). \]  

(5)

The log likelihood can also be written as:

\[ L(\theta; D, z) = \sum_t \sum_i \delta(z^t - i) \left[ \log w_i - \frac{1}{2} \log |C_i| - \frac{1}{2} (x^t - \mu_i)^T C_i^{-1} (x^t - \mu_i) \right]. \]  

(6)

You can check that the marginal likelihood of the extended model is equal to the likelihood of the mixture of gaussians, i.e.,

\[ P(D|\theta) = \sum_z P(D, z|\theta). \]  

(7)

Whenever we encounter inference of continuous parameters \( \theta \) given continuous or discrete observable data \( x \) in a model with discrete latent variables, \( z \), where there is one latent variable per data point, we can use expectation maximization (EM) scheme. Generally, maximizing the marginal (over latent variables) likelihood is difficult because it contains an intractable sum over the states of the latent variables. EM scheme, however, can find local maxima of the marginal likelihood in tractable manner. This scheme iterates between the E and M steps so as to perform the following operations in iteration \( q \):

- **E step:** Compute the expected value of the log likelihood of the parameters over the hidden (latent) variables, \( z \), given current estimate of parameters, \( \theta^{(q)} \), i.e., \( Q(\theta|\theta^{(q)}) = \langle L(\theta; D, z) \rangle_{P(z|D,\theta^{(q)})} \). Q lower bounds the marginal likelihood everywhere, so improvements to it imply improvements to the objective function that our goal is to maximize.

- **M step:** Maximize the expected value of the log likelihood of the parameters with respect to the parameters to get \( \theta^{(q+1)} \), i.e., \( \theta^{(q+1)} = \arg\max_{\theta} Q(\theta|\theta^{(q)}) \).

The algorithm terminates when the improvement in the expected value of log likelihood is below some threshold \( \epsilon \). For the case of Gaussian mixtures, the above steps have a simple interpretation:

- **E-step:** First, use Bayes theorem to compute, given the current parameters \( \theta^{(q)} \), the posterior probability that data point \( t \) is generated by gaussian mixture component \( k \), \( P(z^t = k|x^t, \theta^{(q)}) \). This is simply:

\[ P(z^t = k|x^t, \theta^{(q)}) = \frac{w_k^{(q)} \mathcal{G}(x^t; \mu_k^{(q)}, C_k^{(q)})}{\sum_{k'} w_{k'}^{(q)} \mathcal{G}(x^t; \mu_{k'}^{(q)}, C_{k'}^{(q)})}. \]  

(8)

- **E-step:** Take the expectation of log likelihood for Eq. (6) with respect to the above posterior probability. Only the delta function depends on \( z \), and averages into the posterior probability itself (which only depends on previous values of parameters). Maximizing the expectation of the log likelihood in parameters \( w, \mu, C \) is then standard, and its functional form is the same as for maximum likelihood inference of the covariance matrix and the mean from data, with quickly computable analytical results.
An interesting remark about Gaussian mixtures expressed in terms of latent variable models is that they can be viewed as performing clustering: when the model has been inferred, one can ask about the probability of any point to belong to any of the $M$ mixture components, by evaluating the posterior in Eq. (8); hard clustering is obtained by picking the mixture (or cluster) assignment with the highest posterior. K-means clustering can be seen as an example of such hard EM iteration (with spherical distances, and the iterative estimation of the cluster means).

6 Maximum Entropy models

The groundwork for maximum entropy and its relation to statistical physics was laid down in the work of ET Jaynes [2]. We start by choosing a set of $M$ functions (“operators”) on the state of the system, $\hat{f}_\mu(x)$, $\mu = 1, \ldots, M$, and estimate the empirical averages of these operators over the data, i.e.,

$$\langle \hat{f}_\mu \rangle_{\text{data}} = T^{-1} \sum_{t=1}^{T} \hat{f}_\mu(x^t).$$

These expectations are also known as “constraints.” Our model $\hat{P}(f_1, \ldots, f_M)(x)$ (or $\hat{P}$ for short) will be constructed according to the following two conditions:

- Expectation values of the $M$ functions over the model distribution will be exactly equal to the constraints, which are evaluated over empirical data (Eq. (9)).

- The distribution will be maximally unstructured, specifically, it will have maximum entropy, $S[\hat{P}(x)] = -\sum_x \hat{P}(x) \log_2 \hat{P}(x)$.

Note that maximum entropy framework doesn’t specify which constraints to use, but merely tells us what the form of the distribution should be given the chosen constraints. To derive the maximum entropy distribution, one forms a functional optimization problem:

$$\mathcal{L} = -\sum_x \hat{P}(x) \log \hat{P}(x) - \Lambda \left( \sum_x \hat{P}(x) - 1 \right) - \sum_\mu g_\mu \left( \sum_x \hat{P}(x) f_\mu(x) - \langle f_\mu \rangle_{\text{data}} \right),$$

where the first term will maximize the entropy, the second (with conjugate Lagrange multiplier $\Lambda$) will enforce the normalization of the distribution, and the last sum will make sure that all the expectation values over the model distribution equal the empirical constraints. We solve $\delta \mathcal{L}/\delta \hat{P}(x)$ to find:

$$\hat{P}(x) = \frac{1}{Z(\{g_\mu\})} \exp \left\{ \sum_\mu g_\mu f_\mu(x) \right\},$$

where $Z$ is the normalization constant and the multipliers (“couplings”) need to be set such that

$$\langle f_\mu(x) \rangle_{\hat{P}} \equiv \frac{\partial \log Z}{\partial g_\mu} = \langle f_\mu(x) \rangle_{\text{data}}.$$

Maxent models given by Eq. (11) are therefore exponential family models whose sufficient statistics are given by the functions $f_\mu$.

To infer a maxent model, we need to find $\{g_\mu\}$. Equation (12) is a set of $M$ nonlinear equations for $\{g_\mu\}$. It can be shown that for maximum entropy problems, these equations have
a solution, which (for all reasonable cases of non-degenerate constraints and finite couplings) is unique. Nevertheless, solving the problem by means of solving a set of nonlinear equations is often hard (especially because it requires the evaluation of the partition sum, or normalization constant, \( Z \)) except for the case of a small number of constraints and low dimensional distributions.

Observe that the forms of Eqs (11, 12) are exactly identical to the Boltzmann distribution in statistical physics and the role of the partition function (or normalizing constant) \( Z \) as the generating function for the observables. Indeed, maximum entropy can be viewed as inverse statistical mechanics. Specifically, Boltzmann distribution of statistical physics is a maximum entropy distribution over microstates of the system where the sole constraint is that on the average energy (and thus the distribution has the form \( P(x) = Z^{-1} \exp(-\beta E(x)) \), where \( \beta^{-1} = k_B T \) is now seen as the Lagrange multiplier enforcing the constraint of average energy). For physicists, this is precisely the definition of the canonical ensemble.

Another approach to find the couplings is to view the inference of couplings as maximum likelihood inference. Writing out the dependence on parameters explicitly, the maxent model is \( \hat{P}(x|\{g_\mu\}) \) and we find the set of \( \{g_\mu\} \) that maximize the log likelihood of the data, \( \mathcal{L}(g) = \sum_i \log \hat{P}(x^i|\{g_\mu\}) \). The gradient of the log likelihood function can be evaluated easily:

\[
\frac{\partial \mathcal{L}}{\partial g_\mu} = \frac{\partial}{\partial g_\mu} \sum_i \left[ -\log Z(g) + \sum_\nu g_\nu f_\nu(x^i) \right] = -T \left( \langle f_\mu(x) \rangle_{\hat{P}} + g_\mu \langle f_\mu \rangle_{\text{data}} \right)
\]

Consequently, a gradient ascent on the log likelihood leads to the learning iterative scheme:

\[
g_\mu^{(q+1)} = g_\mu^{(q)} - \alpha \left( \langle f_\mu(x) \rangle_{\hat{P}} - \langle f_\mu \rangle_{\text{data}} \right), \tag{13}
\]

where the learning rate \( \alpha \) must slowly be decreased as the scheme iterates (\( q \) denotes the iteration index), and \( \hat{P} \) at the right hand side is evaluated at the current estimates, \( g^{(q)} \). For that, only needs expectation values over the model distribution \( \hat{P} \), which can be obtained by using Monte Carlo sampling for high-dimensional distributions, without the need to explicitly compute \( Z \). Maxent problems can thus be viewed as a specific subset of maximum likelihood problems, in which the constraints can be reproduced exactly (not just in a best-fit manner), and where the form of the distribution is “derived” from the chosen constraints. The scheme proposed in Eq. (13) has the correct fixed point and will converge to the true solution (because maxent inference is a convex problem), but it is not an optimal scheme; in principle, more efficient schemes are possible where learning is implemented as a version of (quasi-)Newton iterations, with the second-derivative (Hessian) matrix being simply related to the susceptibility matrix of the Boltzmann-like distribution in Eq. (11).

A particularly insightful construction in the maximum entropy framework is obtained when one builds a sequence of approximating distributions that are constrained by marginals of a higher and higher order [3]. Consider for instance a distribution over binary variables, \( P(\{\sigma_i\}) \), where \( \sigma_i = \{0, 1\} \). Then, the first-order marginals are just \( P(\sigma_i) \), which are fully specified by the mean value of each variable, \( \langle \sigma_i \rangle \). One can construct the maxent distribution consistent with these means, which is the factor (independent) distribution: \( \hat{P}(\sigma_i)(\{\sigma_i\}) = \prod_{i=1}^N P(\sigma_i) \). The next approximating distribution is a maxent distribution consistent with all second-order marginals, that is, distributions \( P(\sigma_i, \sigma_j) \) for every pair \( (i, j) \). This is equivalent to specifying all \( \langle \sigma_j \rangle \) and, for all pairs, \( C_{ij} = \langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle \langle \sigma_j \rangle \). The maximum entropy distribution then must follow the ansatz: \( P(\sigma_i, \sigma_j) = Z^{-1} \exp \left( \sum_i h_i \sigma_i + \sum_{i<j} J_{ij} \sigma_i \sigma_j \right) \), i.e., the distribution has an Ising-model
like form. Constraining higher-order correlations results in exponential family distributions that contain terms with successively higher-order products in the energy function. But specifying higher and higher correlation orders stops when we constrain the $N$-th order marginal table (and thus $N$-th order correlation), which is equal to the distribution itself, so that model is exact. A sequence of maxent models constrained by progressively higher-order correlations thus generates a hierarchy that starts with the independent distribution and captures systematically interactions between more and more variables, until all the structure has been captured and the model is exact. Note that constructing even a pairwise model for binary variables from data is very powerful (called Boltzmann machine learning in the machine learning community) but very hard numerically. On the other hand, often even capturing the lower-order interactions can explain very complex collective effects—recently, a number of papers built pairwise maxent models for real data sets that predicted extremely well the measured higher-order statistics of the data (note that this is a nontrivial statement for maxent models, which by construction are the most random models consistent with, say, pairwise observables).

As discussed in the previous lecture, a sequence of maximum entropy distributions, $\hat{P}(k)$ consistent with marginals up to order $k$ defines a sequence of entropies of the corresponding models,

$$S^{(1)} \geq S^{(2)} \geq \ldots S^{(N)},$$  

(14)

where $S^{(1)}$, sometimes denoted as $S_0$, is the “independent entropy”, or the entropy of the factorized model, and $S^{(N)}$ is the true entropy of the distribution, consistent with all marginals of all orders. The inequalities hold because adding higher-order marginal constraints can only make the distribution less random and thus of lower entropy. Defining now the connected correlation of order $k$ as $I^{(k)} = S^{(k-1)} - S^{(k)}$, we see that we can write:

$$I[P(x)] = S_0[P(x)] - S[P(x)] = \sum_{k=2}^{N} I^{(k)},$$  

(15)

which is the unique decomposition of multi-information, $I$, introduced in the previous lecture as measure of the total amount of statistical dependence, into a unique series of non-negative terms that capture the statistical structure order-by-order (uniqueness follows from the uniqueness of maximum entropy constructions).

Maximum entropy models also have an interesting interpretation in that they separate correlations (observable statistics; constraints) from the true underlying interactions (terms in the exponential model). Concretely, in case of the binary pairwise maximum entropy model (Ising-like model), the correlations $C_{ij}$ are simple to estimate from data, but map in a potentially highly complicated way to a matrix of underlying interactions, $J_{ij}$. They can be viewed as an instance of an inverse problem: in (statistical) physics one often assumes the probabilistic model and the values of interactions in the energy function, and then computes the values of the observables. Maxent approach starts with measured observables and infers what interactions should be put into the energy function to explain the measurements. Note that this connection between correlations and real underlying interactions can be very complicated: in binary (spin-like) systems, even a modest nearest-neighbour coupling $J$ can introduce long-range correlations and complicated collective behaviors such as phase transitions.

7 Restricted Boltzmann Machines

One can combine the latent variable ideas and the maximum entropy models (with pairwise-interacting binary units) to define a class of probabilistic models called Restricted Boltzmann
Machines. Here, we are again modeling a distribution over e.g., \( N \) visible binary units, \( x \), but the system is augmented with \( M \) hidden units, \( y \), also assumed binary. A pairwise interacting model without any restrictions would have an Ising-like (or Boltzmann machine) model form, which is difficult to infer. But we can make a simplifying assumption that visible units have no pairwise interactions between themselves, and the same for the hidden units. In this case, the joint probability distribution can be written as:

\[
P(x, y) = \frac{1}{Z} \exp \left( \sum_{i \leq N, j \leq M} J_{ij} x_i y_j + \sum_i h_i x_i + \sum_j k_j y_j \right)
\]

where \( J_{ij} \) again represent the interactions, and \( h \) and \( k \) represent the bias terms. The absence of visible-visible and hidden-hidden interactions is key: it implies that conditional distributions are tractably computable, i.e., \( P(x|y) = \prod_i P(x_i|y) \), which, due to symmetry, also holds for \( P(y|x) \). Moreover, the marginalization over the hidden units (which we need to do inference) can be carried out most of the way analytically:

\[
P_x(x) = \frac{1}{Z} \prod_i e^{h_i x_i} \prod_j \left( 1 + e^{k_j + \sum_{i'} J_{ji'} x_i'} \right).
\]

We can next define the log likelihood of parameters given the data, \( \mathcal{L}(J, h, k) = \sum_t \log P_x(x^t) \), but to maximize it, we need to do something about the intractable partition function, \( Z \). Interestingly, when the number of hidden units \( M \) is small enough (typically \( M < 20 \)), we can evaluate \( Z \) by explicit summation: instead of marginalizing over \( y \) as we did in Eq. (17), we marginalize over \( x \) to get \( P_x(y) \), which can then be summed over the state space for \( y \) explicitly to get \( Z \). If both \( M \) and \( N \) are intractably large, this trick doesn’t work; in this case we have to use Monte Carlo sampling techniques which have been developed specially for RBMs (but we don’t go into details here). Suffice it to say that Gibbs sampling for RBMs can be very effective, because the configuration of all hidden/visible nodes can be sampled in one draw given the configuration of all visible/hidden nodes.

RBMIs are very powerful unsupervised models, because they (unlike maximum entropy models) can learn any distribution with sufficiently large number of hidden units (and enough data not to overfit). This is because at most with \( M = 2^N \) hidden units one can “encode” any probability distribution in the model weights. The number of hidden units is usually chosen by cross-validation. Unlike maxent models, however, RBM performance may be hard to interpret, since the model class has no well-defined set of sufficient statistics: marginalization over hidden units can model interactions of (in principle) any order between the visible units. In this case, maxent models can provide the hierarchy of null-models against which RBMs can be compared (by asking questions such as how much information beyond pairwise maxent model can RBMs capture).

8 Study literature

- Notes on nonparametric statistics including kernel density estimation

- Chapter 22 of MacKay’s book (and other chapters therein), for mixture models

- A list of online resources for maximum entropy models
  http://homepages.inf.ed.ac.uk/lzhang10/maxent.html
9 Homework

1. Let’s check if the scaling for kernel width predicted by the Silberman rule also applies in the likelihood cross-validation scenario. Draw, e.g., \( n = 100, 200, 500, 1000, 2000 \) “training” points from a Gaussian with unit variance (\( \Omega = 1 \)), as well as 100 “testing” points, 500 times at each \( n \). Set a range of \( \sigma \) to try out, e.g., \( \sigma = 0.05, 0.1, 0.2, 0.3, \ldots, 1.5 \). For each random draw of points, construct a KDE model using the training points, and compute the log likelihood of that model on the testing points, for all values of \( \sigma \). Average the log likelihood across 500 random draws for each \( \sigma \), and plot it for various \( n \) as a function of \( \sigma \) to see if the average log likelihood is maximized at a particular \( \sigma^*(n) \) for different data sizes, \( n \). To smoothly find the value of \( \sigma \) at which the cross-validated log likelihood is maximized, one approximate rule-of-thumb way is to first fit a parabola or similar low-order polynomial to the log likelihood as a function of \( \sigma \) (at every \( n \)), and then to compute the optimal \( \sigma^* \) exactly from the fitted coefficients. You can alternatively use other methods of smoothing to find the optimal \( \sigma^* \), but please describe briefly what you did. Show \( \sigma^* \) as a function of \( n \) on a log log plot, and compare with the Silverman’s rule of thumb on the same plot.

2. Implement a 2D KDE for Sachs data. First, visualize several pairwise cross-sections through the data (look in particular at node 4 vs node 5, and node 1 vs node 3). Look at the data if you scatterplot the log activity levels of pairs of proteins against each other. Which representation (raw vs log) do you find more useful? Using cross-validation, determine the best \( \sigma \) to use for making a 2D KDE model for the log activities for both pairs of variables (1 vs 3, 4 vs 5); you will have to determine best \( \sigma \)'s separately for different pairs of variables. Using the KDE model, visualize the joint probability distributions of the pairwise distributions by making a contour plot using the optimal value of the \( \sigma \), and using five-fold too large and five-fold too small values. Concretely, you can make your kernels be 2D Gaussians with diagonal covariance that has \( \sigma_x, \sigma_y \) widths in the two orthogonal directions. Finally, to examine the effects of smoothing, discretize the \( x \) and \( y \) axis into bins of size \( \sigma_x \times \sigma_y \), and estimate an empirical distribution (by counting) over this discretized domain. Over the same domain, define another discrete probability distribution, which is equal to your continuous KDE model evaluated in the bin centers (and normalized). Compute and compare the entropy of both discrete distributions.

3. Derive the: (i) a continuous maximum entropy distribution with a given mean \( \langle x \rangle \) for \( x \in [0, \infty) \) (i.e., what is the form for the distribution, and what is the value of the Lagrange multiplier conjugate to the constraint function in terms of \( \langle x \rangle \)); (ii) a continuous maximum entropy distribution with a given mean \( \langle x \rangle \) and a given variance, \( \sigma^2 \), and, in general, for the multivariate case, with mean \( \langle x \rangle \) and covariance matrix \( C \), where \( x \in (-\infty, \infty) \). For the multivariate case, a well-argued educated guess is sufficient, if you do not want to carry out multidimensional integrals (note: carrying out high-dimensional Gaussian integration is very applicable to many fields, so you are encouraged to learn about it; by using several tricks it is possible to compute essentially any moment without actually doing the integration).

4. Let’s try to infer interactions from Sachs et al data. First, try the naive way, by computing the correlation coefficient between every pair of signaling nodes across all of 5400 samples. Plot the \( 11 \times 11 \) matrix of correlation coefficients. If you were to threshold that matrix using different thresholds to find the underlying interactions, could you reproduce, even
approximately, the network of known interactions as reported in Ref [1]? Second, try to construct a maximum entropy model that is consistent with means and covariances (e.g., a pairwise model you derived in the previous exercise) of the activation levels. If you take the resulting matrix of interactions (specifically, their absolute values) and threshold it, does it recover the known interactions better than the naive approach? What happens if you repeat the same procedure only on the first 2 conditions (1200 samples) that correspond to the signaling network driven by its “natural” stimulation? A third possible approach, using Bayes network inference, has been attempted in the paper by Sachs et al – you are welcome to read about that in the original publication.

5. Maximum entropy for binary variables. Let’s try to do maximum entropy in case the signaling levels are binary. To this end, take again the first two conditions, and binarize each variable such that the discretized value is 0 for samples below the median, and 1 for samples above the median. This should result in a binary data matrix of dimension $11 \times 1200$. To build a pairwise maximum entropy model for this data, you need to infer a vector of local fields, $h_i$ (11 variables) and a matrix of pairwise couplings $J_{ij}$ of dimension $11 \times 11$ (with $11 \times 10/2$ free parameters due to symmetry), in a model distribution of the form $P(\sigma_i, \sigma_j) = Z^{-1} \exp \left( \sum_i h_i \sigma_i + \sum_{i<j} J_{ij} \sigma_i \sigma_j \right)$, such that the mean values $\langle \sigma_i \rangle$ and all pairwise terms $\langle \sigma_i \sigma_j \rangle$ of the model exactly match the data. For 11 binary variables it is possible to enumerate all $2^{11}$ states, compute the normalization constant $Z$ and the probability of each state, and thus evaluate the expectation values of the means and pairwise terms. Use the gradient ascent learning (see introduction; you can use any other method if you know how to implement it) to learn the parameter values for $h$ and $J$, by decreasing the learning rate $\alpha$ slowly (hint: two possibilities to try are e.g. $\alpha(q) = q^{-0.5}$ or $\alpha(q) = q^{-0.25}$). Iterate the learning until the relative error on the means and covariances is below 1%. How to the interactions $J_{ij}$ in this binarized case compare to the previous exercise where you looked at the continuous activation levels? If your code for 11 nodes together runs too slowly, try with a smaller set of nodes (this is always useful to debug your algorithm anyway)—you should definitely have no problems in terms of computational time for a system of size $N = 5$ or 6 nodes...

References

