Appendix A

Statistics

A.1 Decision Theory and Loss Functions

In any decision problem [238, 63, 431], one is led to define a loss function (or equivalently a reward function) to measure the effect of one's action on a given state of the environment. The fundamental theorem of decision theory is that under a small set of sensible axioms used to describe rational behavior, the optimal strategy is the one that minimizes the expected loss, where expectation is defined with respect to a Bayesian probabilistic analysis of the uncertain environment, given the available knowledge. Note that several of the tasks undertaken in purely scientific data analysis endeavors—such as data compression, reconstruction, or clustering—are decision-theoretic in nature and therefore require the definition of a loss function. Even prediction falls into this category, and this is why in regression, $\mathbf{E}(y|x)$ is the best predictor of y given x, when the loss is quadratic (see below).

When one of the goals is to pick the "best" model, as is often the case throughout this book, the expected loss function is equal to the negative log-likelihood (or log-prior). But in general the two functions are distinct. In principle, for instance, one could even have Gaussian data with quadratic negative log-likelihood, but use a quartic loss function.

Two loss functions f_1 and f_2 can be equivalent in terms of minimization properties. This is the case if there is an order-preserving transformation g (if $u \leq v$, then $g(u) \leq g(v)$) such that $f_2 = gf_1$. Then f_1 and f_2 have the same minima. This of course does not imply that minimization (i.e., learning) algorithms applied to f_1 or f_2 behave in the same way, nor that f_1 and f_2 have the same curvature around their minima. As briefly mentioned in chapter 5, a good example is provided by the quadratic function $f_1(y) = \sum_1^K (p_i - y_i)^2/2$ and the cross-entropic function $f_2(y) = -\sum_1^K p_i \log y_i$, when $\sum p_i = 1$. Both functions are convex in y, and have a unique global minimum at $y_i = p_i$, provided f_2 is restricted to $\sum y_i = 1$. In fact, by Taylor-expanding f_2 around p_i, we have

$$f_2(y) = -\sum_1^K p_i \log(p_i + \epsilon_i) \approx \mathcal{H}(p) + \sum_1^K \frac{\epsilon_i^2}{2p_i} \tag{A.1}$$

with $y_i = p_i + \epsilon_i$ and $\sum \epsilon_i = 0$. Therefore, when $p_i = 1/K$ is uniform, one has the even stronger result that $f_2 \approx \mathcal{H}(p) + Kf_1$. Therefore, apart from constant terms, the quadratic and cross-entropy loss f_1 and f_2 coincide around the same optimum and have the same curvature. In the rest of this appendix, we concentrate on the most common quadratic loss functions (or Gaussian likelihoods), but many of the results can be extended to other loss functions, using the remarks above.

A.2 Quadratic Loss Functions

A.2.1 Fundamental Decomposition

To begin, consider a sequence of numbers $y_1, \ldots, y_K$ and the quadratic form $f(y) = \sum_1^K (y - y_i)^2/K$, that is the average square loss. Then f has a unique minimum at the average $y^* = \mathbf{E}(y) = \sum_1^K y_i/K$. This is easily seen by using Jensen's inequality (appendix B), or more directly by writing

$$\begin{aligned} f(y) &= \frac{1}{K}\sum_1^K (y - y^* + y^* - y_i)^2 \\ &= (y - y^*)^2 + \frac{1}{K}\sum_1^K (y^* - y_i)^2 + \frac{2}{K}\sum_1^K (y - y^*)(y^* - y_i) \\ &= (y - y^*)^2 + \frac{1}{K}\sum_1^K (y^* - y_i)^2 \geq f(y^*). \end{aligned} \tag{A.2}$$

Thus f can be decomposed into the sum of the bias $(y - y^*)^2$ and the variance $\sum_1^K (y^* - y_i)^2$. The bias measures the distance from y to the optimum average, and the variance measures the dispersion of the y_is around the average. This decomposition of quadratic loss functions into the sum of two quadratic terms (Pythagoras' theorem) with the cancellation of any cross-product terms is essential, and will be used repeatedly below in slightly different forms. The above result remains true if the y_i occur with different frequencies or strengths $p_i \geq 0$, with $\sum p_i = 1$. The expected quadratic loss is again minimized by the the weighted average $y^* = \mathbf{E}(y) = \sum p_i y_i$ with the decomposition

$$\mathbf{E}\left[(y-y_i)^2\right] = \sum_1^K p_i(y-y_i)^2 = (y-y^*)^2 + \sum_1^K p_i(y^*-y_i)^2. \tag{A.3}$$

We now show how this simple decomposition can be applied to regression problems, and in several directions, by using slightly different expectation operators, including averaging over different training sets or different estimators.

A.2.2 Application to Regression

Consider a regression problem in which we are trying to estimate a target function $f(x)$ and in which the x, y data are characterized by a distribution $P(x, y)$. For simplicity, as in chapter 5, we shall assume that as a result of "noise," different possible values of y can be observed for any single x. For any x, the expected error or loss $\mathbf{E}[(y - f(x))^2|x]$ is minimized by the conditional expectations $y^* = \mathbf{E}(y|x)$, where now all expectations are taken with respect to the distribution P, or approximated from corresponding samples. Again this is easily seen by writing

$$\mathbf{E}\left[(y-f(x))^2|x\right] = \mathbf{E}\left[(y-\mathbf{E}(y|x)+\mathbf{E}(y|x)-f(x))^2|x\right] \tag{A.4}$$

and expanding the square. The cross-product term disappears, leaving the bias/variance decomposition

$$\mathbf{E}\left[(y-f(x))^2|x\right] = [\mathbf{E}(y|x)-f(x)]^2 + \mathbf{E}\left[(y-\mathbf{E}(y|x))^2|x\right]. \tag{A.5}$$

A.3 The Bias/Variance Trade-off

Consider the same regression framework as above, but where different training sets D are available. For each training set D, the learning algorithm produces a different estimate $f(x, D)$. The performance of such an estimator can be measured by the expected loss $\mathbf{E}[(y - f(x, D))^2|x, D]$, the expectation again being with respect to the distribution P. The usual calculation shows that

$$\mathbf{E}\left[(y-f(x,D))^2|x,D\right] = \\ [f(x,D)-\mathbf{E}(y|x)]^2 + \mathbf{E}\left[(y-\mathbf{E}(y|x))^2|x,D\right]. \tag{A.6}$$

The variance term does not depend on the training sample D. Thus, for any x, the effectiveness of the estimator $f(x, D)$ is measured by the bias $[f(x, D) - \mathbf{E}(y|x)]^2$, that is, by how it deviates from the optimal predictor $\mathbf{E}(y|x)$. We can now look at the average of such error over all training sets D of a given size. Again writing

$$\mathbf{E}_D\left[(f(x,D)-\mathbf{E}(y|x))^2\right] =$$

$$\mathbf{E}_D\left[(f(x,D)-\mathbf{E}_D(f(x,D))+\mathbf{E}_D(f(x,D))-\mathbf{E}(y|x))^2\right], \quad \text{(A.7)}$$

cancellation of the cross-product term leaves the bias-variance decomposition

$$\mathbf{E}_D\left[(f(x,D)-\mathbf{E}(y|x))^2\right] =$$

$$[\mathbf{E}_D(f(x,D))-\mathbf{E}(y|x)]^2+\mathbf{E}_D\left[(f(x,D)-\mathbf{E}_D(f(x,D)))^2\right]. \quad \text{(A.8)}$$

The bias/variance decomposition corresponds to a sort of uncertainty principle in machine learning: it is always difficult to try to decrease one of the terms without increasing the other. This is also the basic trade-off between underfitting and overfitting the data. A flexible machine with a large number of parameters that can cover a large functional space typically achieves a small bias. The machine, however, must be sensitive to the data and therefore the variance associated with overfitting the data tends to be large. A simple machine has typically a smaller variance, but the price to pay is a larger underfitting bias.

A.4 Combining Estimators

As mentioned in chapter 4, it can be useful at times to combine different estimators $f(x,w)$, using a discrete (or even continuous) distribution $p_w \geq 0$, ($\sum_w p_w = 1$) over parameters w associated with each estimator. As in (A.8), the different estimators could, for example, correspond to different training sets. By taking expectations with respect to w, (A.8) can be generalized immediately to

$$\mathbf{E}_w\left[(f(x,w)-\mathbf{E}(y|x))^2\right] =$$

$$[\mathbf{E}_w(f(x,w)-\mathbf{E}(y|x))]^2+\mathbf{E}_w\left[(f(x,w)-\mathbf{E}_w(f(x,w)))^2\right]. \quad \text{(A.9)}$$

Thus the loss for the weighted average predictor $f^*(x)=\mathbf{E}_w(f(x,w))$, sometimes also called ensemble average, is always less than the average loss:

$$\mathbf{E}_w\left[(f(x,w)-\mathbf{E}(y|x))^2\right] \geq [f^*(x)-\mathbf{E}(y|x)]^2. \quad \text{(A.10)}$$

In fact, we can average (A.9) over all possible xs, using the distribution P to obtain "generalization" errors:

$$\mathbf{E}_X \left[f^*(x) - \mathbf{E}(y|x)\right]^2 =$$

$$\mathbf{E}_X \mathbf{E}_w \left[(f(x,w) - \mathbf{E}(y|x))^2\right] - \mathbf{E}_X \mathbf{E}_w \left[(f(x,w) - f^*(x))^2\right]. \quad \text{(A.11)}$$

This is the relation used in [340, 339]. The left-hand term is the expected loss of the ensemble. The first term on the right-hand side is the expected loss across estimators, and the second term is called the ambiguity. Clearly, combining identical estimators is useless. Thus a necessary condition for the ensemble approach to be useful is that the individual estimators have a substantial level of disagreement. All else equal, the ambiguity should be large. One way to achieve this is to use different training sets for each estimator (see [340], where algorithms for obtaining optimal weighting schemes p_w—for instance, by quadratic programming—are also discussed). One important point is that all the correlations between estimators are contained in the ambiguity term. The ambiguity term does not depend on any target values, and therefore can be estimated from unlabeled data.

A.5 Error Bars

For illustration, consider a modeling situation with one parameter w, and a uniform prior. Let $f(w) = -\log \mathbf{P}(D|w)$ be the negative log-likelihood of the data. Under mild differentiability conditions, a maximum likelihood estimator w^* satisfies $f'(w^*) = 0$. Therefore, in the neighborhood of w^*, we can expand $f(w^*)$ in a Taylor series:

$$f(w) \approx f(w^*) + \frac{1}{2} f''(w^*)(w - w^*)^2 \quad \text{(A.12)}$$

or

$$\mathbf{P}(D|w) = e^{-f(w)} \approx Ce^{-\frac{1}{2}f''(w^*)(w-w^*)^2}, \quad \text{(A.13)}$$

where $C = e^{-f(w^*)}$. Thus the likelihood and the posterior $\mathbf{P}(w|M)$ locally behave like a Gaussian, with a standard deviation $1/\sqrt{f''(w^*)}$, associated with the curvature of f. In the multidimensional case, the matrix of second-order partial derivatives is called the Hessian. Thus the Hessian of the log-likelihood has a geometric interpretation and plays an important role in a number of different questions. It is also called the Fisher information matrix (see also [5, 16, 373]).

A.6 Sufficient Statistics

Many statistical problems can be simplified through the use of *sufficient statistics*. A sufficient statistic for a parameter w is a function of the data that summarize all the available information about w. More formally, consider a random variable X with a distribution parameterized by w. A function S of X is a sufficient statistic for w if the conditional distribution $P(X = x|S(X) = s)$ is independent of w with probability 1. Thus $P(X = x|S(X) = s)$ does not vary with w, or

$$\mathbf{P}(X = x|S = s, w) = \mathbf{P}(X = x|S = s). \tag{A.14}$$

This equality remains true if we replace X by any statistics $H = h(X)$. Equivalently, this equality yields $\mathbf{P}(w|X, S) = \mathbf{P}(w|S)$. All information about w is conveyed by S, and any other statistic is redundant. In particular, sufficient statistics preserve the mutual information I (see appendix B): $I(w, X) = I(w, S(X))$.

As an example, consider a sample $X = (X_1, \ldots, X_N)$ drawn from a random variable $\mathcal{N}(\mu, \sigma^2)$, so that $w = (\mu, \sigma)$. Then (m, s) is a sufficient statistic for w, with $m = \sum_i X_i/N$ and $s^2 = \sum_i (X_i - m)^2/(N - 1)$. In other words, all the information about μ contained in the sample is contained in the sample mean m, and similarly for the variance.

A.7 Exponential Family

The *exponential family* [94] is the most important family of probability distributions. It has a wide range of applications and unique computational properties: many fast algorithms for data analysis have some version of the exponential family at their core. Many general theorems in statistics can be proved for this particular family of parameterized distributions. The density in the one-parameter exponential family has the form

$$\mathbf{P}(x|w) = c(w)h(x)e^{q(w)S(x)}. \tag{A.15}$$

Most common distributions belong to the exponential family, including the normal (with either mean or variance fixed), chi square, binomial and multinomial, geometric and negative binomial, exponential and gamma, beta, Poisson, and Dirichlet distributions. All the distributions used in this book are in the exponential family. Among the important general properties of the exponential family is the fact that a random sample from a distribution in the one-parameter exponential family always has a sufficient statistic S. Furthermore, the sufficient statistic itself has a distribution that belongs to the exponential family.

A.8 Additional Useful Distributions

Here we briefly review three additional continuous distributions used in chapter 12.

A.8.1 The Scaled Inverse Gamma Distribution

The scaled inverse gamma distribution $\mathcal{I}(x; \nu, s^2)$ with $\nu > 0$ degrees of freedom and scale $s > 0$ is given by:

$$\frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)} s^{\nu} x^{-(\nu/2+1)} e^{-\nu s^2/(2x)} \tag{A.16}$$

for $x > 0$. The expectation is $(\nu/\nu - 2)s^2$ when $\nu > 2$, otherwise it is infinite. The mode is always $(\nu/\nu + 2)s^2$.

A.8.2 The Student Distribution

The Student-t distribution $t(x; \nu, m, \sigma^2)$ with $\nu > 0$ degrees of freedom, location m and scale $\sigma > 0$ is given by:

$$\frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)\sqrt{\nu\pi}\sigma} (1 + \frac{1}{\nu}(\frac{x-m}{\sigma})^2)^{-(\nu+1)/2}. \tag{A.17}$$

The mean and the mode are equal to m.

A.8.3 The Inverse Wishart Distribution

The inverse Wishart distribution $\mathcal{I}(W; \nu, S^{-1})$, where ν represents the degrees of freedom and S is a $k \times k$ symmetric, positive definite scale matrix, is given by

$$\begin{aligned} &(2^{\nu k/2} \pi^{k(k-1)/4} \prod_{i=1}^{k} \Gamma(\frac{\nu+1-i}{2}))^{-1} |S|^{\nu/2} |W|^{-(\nu+k+1)/2} \\ &\exp(-\frac{1}{2} tr(SW^{-1})) \end{aligned} \tag{A.18}$$

where W is also positive definite. The expectation of W is $E(W) = (\nu - k - 1)^{-1} S$.

A.9 Variational Methods

To understand this section one must be familiar with the notion of relative entropy (appendix B) . In the Bayesian framework, we are often faced with high–dimensional probability distributions $P(x) = P(x_1, ..., x_n)$ that are intractable, in the sense that they are too complex to be estimated exactly. The basic idea in variational methods is to approximate $P(x)$ by constructing a tractable family $Q(x, \theta)$ of distributions parameterized by the vector θ and choosing the element in the family closest to P. This requires a way of measuring distances between probability distributions. In variational methods this is generally done using the relative entropy or KL distance $\mathcal{H}(Q, P)$. Thus we try to minimize

$$\mathcal{H}(Q,P) = \sum Q \log \frac{Q}{P} = -\mathcal{H}(Q) + \mathbf{E}_Q(-\log P). \tag{A.19}$$

When P is represented as a Boltzmann–Gibbs distribution $P = e^{-\lambda \mathcal{E}}/Z(\lambda)$, then

$$\mathcal{H}(Q,P) = -\mathcal{H}(Q) + \lambda \mathbf{E}_Q(\mathcal{E}) + \log Z(\lambda) = \lambda \mathcal{F} + \log Z(\lambda) \tag{A.20}$$

where $\mathcal{F}$ is the free energy defined in chapter 3. Since the partition function Z does not depend on θ, minimizing $\mathcal{H}$ is equivalent to minimizing $\mathcal{F}$. From Jensen's inequality in appendix B, we know that, for any approximating Q, $\mathcal{H} \geq 0$ or, equivalently, $\mathcal{F} \geq -\log Z(\lambda)/\lambda$. Equality at the optimum can be achieved only if $Q^* = P$.

In modeling situations we often have a family of models parameterized by w and P is the posterior $\mathbf{P}(w|D)$. Using Bayes' theorem and the equation above, we then have

$$\mathcal{H}(Q,P) = -\mathcal{H}(Q) + \mathbf{E}_Q[-\log \mathbf{P}(D|w) - \log \mathbf{P}(w)] + \log \mathbf{P}(D) \tag{A.21}$$

with $\lambda = 1$ and $E = -\log \mathbf{P}(D|w) - \log \mathbf{P}(w)$. Again, the approximating distributions must satisfy $\mathcal{H} \geq 0$ or $\mathcal{F} \geq -\log \mathbf{P}(D)$.

In a sense, variational methods are close to higher levels of Bayesian inference since they attempt to approximate the entire distribution $\mathbf{P}(w|D)$ rather than focusing on its mode, as in MAP estimation. At an even higher level, we could look at a distribution over the space Q rather than its optimum Q^*. We leave as an exercise for the reader to study further the position of variational methods within the Bayesian framework and to ask, for instance, whether variational methods themselves can be seen as a form of MAP estimation.

But the fundamental problem in the variational approach is of course the choice of the approximating family $Q(x, \theta)$ or $Q(w, \theta)$. The family must satisfy two conflicting requirements: it must be simple enough to be computationally tractable, but not too simple or else the distance $\mathcal{H}(Q, P)$ remains too large. By computationally tractable we mean that one ought to be able to estimate, for instance, $\mathcal{F}$ and $\partial\mathcal{F}/\partial\theta$. A simple case is when the family Q is factorial. Q is a factorial distribution if and only if it has the functional form $Q(x_1, \dots, x_n) = Q(x_1) \dots Q(x_n)$. Mean field theory in statistical mechanics is a special case of variational method with factorial approximation (see also [582]). More generally, the construction of a suitable approximating family Q is problem–dependent and remains an art more than a science. In constructing Q, however, it is often useful to use:

- Mixture distributions
- Exponential distributions
- Independence assumptions and the corresponding factorizations (appendix C).

For instance, Q can be written as a mixture of factorial distributions, where each factor belongs to the exponential family. The parameters to be optimized can then be the mixture coefficients and/or the parameters (mean, variance) of each exponential member.

This page intentionally left blank

Appendix B

Information Theory, Entropy, and Relative Entropy

Here we briefly review the most basic concepts of information theory used in this book and in many other machine learning applications. For more in-depth treatments, the reader should consult [483], [71], [137], and [577]. The three most basic concepts and measures of information are the entropy, the mutual information, and the relative entropy. These concepts are essential for the study of how information is transformed through a variety of operations such as information coding, transmission, and compression. The relative entropy is the most general concept, from which the other two can be derived. As in most presentations of information theory, we begin here with the slightly simpler concept of entropy.

B.1 Entropy

The entropy $\mathcal{H}(P)$ of a probability distribution $P = (p_1, \ldots, p_n)$ is defined by

$$\mathcal{H}(P) = \mathbf{E}(-\log P) = -\sum_{i=1}^{n} p_i \log p_i. \tag{B.1}$$

The units used to measure entropy depend on the base used for the logarithms. When the base is 2, the entropy is measured in bits. The entropy measures the prior uncertainty in the outcome of a random experiment described by P, or the information gained when the outcome is observed. It is also the minimum average number of bits (when the logarithms are taken base 2) needed to transmit the outcome in the absence of noise.

The concept of entropy can be derived axiomatically. Indeed, consider a random variable X that can assume the values $x_1, \ldots, x_n$ with probabilities $p_1, \ldots, p_n$. The goal is to define a quantity $\mathcal{H}(P) = \mathcal{H}(X) = \mathcal{H}(p_1, \ldots, p_n)$ that measures, in a unique way, the amount of uncertainty represented in this distribution. It is a remarkable fact that three commonsense axioms, really amounting to only one composition law, are sufficient to determine $\mathcal{H}$ uniquely, up to a constant factor corresponding to a choice of scale. The three axioms are as follows:

1. $\mathcal{H}$ is a continuous function of the p_i.
2. If all p_is are equal, then $\mathcal{H}(P) = \mathcal{H}(n) = \mathcal{H}(1/n, \ldots, 1/n)$ is a monotonic increasing function of n.
3. Composition law: Group all the events x_i into k disjoint classes. Let A_i represent the indices of the events associated with the ith class, so that $q_i = \sum_{j \in A_i} p_j$ represents the corresponding probability. Then

$$\mathcal{H}(P) = \mathcal{H}(Q) + \sum_{i=1}^{k} q_i \mathcal{H}\left(\frac{\bar{P}_i}{q_i}\right), \tag{B.2}$$

where $\bar{P}_i$ denotes the set of probabilities p_j for $j \in A_i$. Thus, for example, the composition law states that by grouping the first two events into one,

$$\mathcal{H}(1/3, 1/6, 1/2) = \mathcal{H}(1/2, 1/2) + \frac{1}{2}\mathcal{H}(2/3, 1/3). \tag{B.3}$$

From the first condition, it is sufficient to determine $\mathcal{H}$ for all rational cases where $p_i = n_i/n$, $i = 1, \ldots, n$. But from the second and third conditions,

$$\mathcal{H}(\sum_{i=1}^{n} n_i) = \mathcal{H}(p_1, \ldots, p_n) + \sum_{i=1}^{n} p_i \mathcal{H}(n_i). \tag{B.4}$$

For example,

$$\mathcal{H}(9) = \mathcal{H}(3/9, 4/9, 2/9) + \frac{3}{9}\mathcal{H}(3) + \frac{4}{9}\mathcal{H}(4) + \frac{2}{9}\mathcal{H}(2). \tag{B.5}$$

In particular, by setting all n_i equal to m, from (B.4) we get

$$\mathcal{H}(m) + \mathcal{H}(n) = \mathcal{H}(mn). \tag{B.6}$$

This yields the unique solution

$$\mathcal{H}(n) = C \ln n, \tag{B.7}$$

with $C > 0$. By substituting in (B.4), we finally have

$$\mathcal{H}(P) = -C \sum_{i=1}^{n} p_i \log p_i. \tag{B.8}$$

The constant C determines the base of the logarithm. Base-2 logarithms lead to a measure of entropy and information in bits. For most mathematical calculations, however, we use natural logarithms so that $C = 1$.

It is not very difficult to verify that the entropy has the following properties:

- $\mathcal{H}(P) \geq 0$.
- $\mathcal{H}P|Q) \leq \mathcal{H}(P)$ with equality if and only if P and Q are independent.
- $\mathcal{H}(P_1, \dots, P_n) \leq \sum_{i=1}^{n} \mathcal{H}(P_i)$ with equality if and only if P and Q are independent.
- $\mathcal{H}(P)$ is convex ($\cap$) in P.
- $\mathcal{H}(P_1, \dots, P_n) = \sum_{i=1}^{n} \mathcal{H}(P_i|P_{i-1}, \dots, P_1)$.
- $\mathcal{H}(P) \leq \mathcal{H}(n)$ with equality if and only if P is uniform.

B.2 Relative Entropy

The relative entropy between two distributions $P = (p_1, \dots, p_n)$ and $Q = (q_1, \dots, q_n)$, or the associated random variables X and Y, is defined by

$$\mathcal{H}(P, Q) = \mathcal{H}(X, Y) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}. \tag{B.9}$$

The relative entropy is also called cross-entropy, or Kullback–Liebler distance, or discrimination (see [486], and references therein, for an axiomatic presentation of the relative entropy). It is viewed as a measure of the distance between P and Q. The more dissimilar P and Q are, the larger the relative entropy. The relative entropy is also the amount of information that a measurement gives about the truth of a hypothesis compared with an alternative hypothesis. It is also the expected value of the log-likelihood ratio. Strictly speaking, the relative entropy is not symmetric and therefore is not a distance. It can be made symmetric by using the divergence $\mathcal{H}(P, Q) + \mathcal{H}(Q, P)$. But in most cases, the symmetric version is not needed. If $U = (1/n, \dots, 1/n)$ denotes the uniform density, then $\mathcal{H}(P, U) = \log n - \mathcal{H}(P)$. In this sense, the entropy is a special case of cross-entropy.

By using the Jensen inequality (see section B.4), it is easy to verify the following two important properties of relative entropies:

- $\mathcal{H}(P,Q) \geq 0$ with equality if and only if $P = Q$.
- $\mathcal{H}(P,Q)$ is convex ($\cap$) in P and Q.

These properties are used throughout the sections on free energy in statistical mechanics and the EM algorithm in chapters 3 and 4.

B.3 Mutual Information

The third concept for measuring information is the mutual information. Consider two distributions P and Q associated with a joint distribution R over the product space. The mutual information $\mathcal{I}(P,Q)$ is the relative entropy between the joint distribution R and the product of the marginals P and Q:

$$\mathcal{I}(P,Q) = \mathcal{H}(R, PQ). \tag{B.10}$$

As such, it is always positive. When R is factorial, i.e. equal to the product of the marginals, the mutual information is 0. The mutual information is a special case of relative entropy. Likewise, the entropy (or self-entropy) is a special case of mutual information because $\mathcal{H}(P) = \mathcal{I}(P,P)$. Furthermore, the mutual information satisfies the following properties:

- $\mathcal{I}(P,Q) = 0$ if and only if P and Q are independent.
- $\mathcal{I}(P_1,\ldots,P_n,Q) = \sum_{i=1}^n \mathcal{I}(P_i, Q|P_1,\ldots,P_{i-1})$.

It is easy to understand mutual information in Bayesian terms: it represents the reduction in uncertainty of one variable when the other is observed, that is between the *prior* and *posterior distributions*. If we denote two random variables by X and Y, the uncertainty in X is measured by the entropy of its prior $\mathcal{H}(X) = \sum_x \mathbf{P}(X = x) \log \mathbf{P}(X = x)$. Once we observe $Y = y$, the uncertainty in X is the entropy of the posterior distribution, $\mathcal{H}(X|Y = y) = \sum_x \mathbf{P}(X = x|Y = y) \log \mathbf{P}(X = x|Y = y)$. This is a random variable that depends on the observation y. Its average over the possible ys is called the *conditional entropy*:

$$\mathcal{H}(X|Y) = \sum_y P(y)\mathcal{H}(X|Y = y). \tag{B.11}$$

Therefore the difference between the entropy and the conditional entropy measures the average information that an observation of Y brings about X. It is straightforward to check that

$$\mathcal{I}(X,Y) = \mathcal{H}(X) - \mathcal{H}(X|Y) =$$

$$\mathcal{H}(Y) - \mathcal{H}(Y|X) = \mathcal{H}(X) + \mathcal{H}(Y) - \mathcal{H}(Z) = \mathcal{I}(Y, X) \tag{B.12}$$

where $\mathcal{H}(Z)$ is the entropy of the joint variable $Z = (X, Y)$. or, using the corresponding distributions,

$$\mathcal{I}(P, Q) = \mathcal{H}(P) - \mathcal{H}(P|Q) =$$
$$\mathcal{H}(Q) - \mathcal{H}(Q|P) = \mathcal{H}(P) + \mathcal{H}(Q) - \mathcal{H}(R) = \mathcal{I}(Q, P). \tag{B.13}$$

We leave for the reader to draw the classical Venn diagram associated with these relations.

B.4 Jensen's Inequality

The Jensen inequality is used many times throughout this book. If a function f is convex (∩) and X is a random variable, then

$$\mathbf{E}f(X) \le f\mathbf{E}(X). \tag{B.14}$$

Furthermore, if f is strictly convex, equality implies that X is constant. This inequality becomes graphically obvious if one thinks in terms of center of gravity. The center of gravity of $f(x_1), \ldots, f(x_n)$ is below $f(x^*)$, where x^* is the center of gravity of $x_1, \ldots, x_n$. As a special important case, $\mathbf{E} \log X \le \log \mathbf{E}(X)$. This immediately yields the properties of the relative entropy.

B.5 Maximum Entropy

The maximum entropy principle was discussed in chapters 2 and 3 for the case of discrete distributions. The precise statement of the maximum entropy principle in the continuous case requires some care [282]. But in any case, if we define the *differential entropy* of a random variable X with density P to be

$$\mathcal{H}(X) = -\int_{-\infty}^{+\infty} P(x) \log P(x) dx, \tag{B.15}$$

then of all the densities with variance σ^2, the Gaussian $\mathcal{N}(\mu, \sigma)$ is the one with the largest differential entropy. The differential entropy of a Gaussian distribution with any mean and variance σ^2 is given by $[\log 2\pi e \sigma^2]/2$. In n dimensions, consider a random vector X with vector mean μ, covariance matrix C, and density P. Then the differential entropy of P satisfies

$$\mathcal{H}(P) \le \frac{1}{2} \log(2\pi e)^n |C| = \mathcal{H}(\mathcal{N}(\mu, C)) \tag{B.16}$$

with equality if and only if X is distributed according to $\mathcal{N}(\mu, C)$ almost everywhere. Here $|C|$ denotes the determinant of C.

These results have a very simple proof using the derivation of the Boltzmann–Gibbs distribution in statistical mechanics. For instance, in the one-dimensional case, a Gaussian distribution can be seen as a Boltzmann–Gibbs distribution with energy $\mathcal{E}(x) = (x-\mu)^2/2\sigma^2$ and partition function $\sqrt{2\pi}\sigma$, at temperature 1. Thus the Gaussian distribution must have maximum entropy, given that the only constraint is the observation of the expectation of the energy. The mean of the energy is given by $\int (x-\mu)^2/2\sigma^2 P(x)dx$, which is constant, equivalent to the statement that the standard deviation is constant and equal to σ.

This can be generalized to the members of the exponential family of distributions. In the case of the Dirichlet distributions, consider the space of all n-dimensional distributions $P = (p_1, \ldots, p_n)$. Suppose that we are given a fixed distribution $R = (r_1, \ldots, r_n)$, and define the energy of a distributions by its distance, measured in relative entropy, from R:

$$\mathcal{E}(P) = \mathcal{H}(R, P) = \sum_i r_i \log r_i - \sum_i r_i \log p_i. \tag{B.17}$$

If all we observe is the average D of $\mathcal{E}$, then the corresponding maximum entropy distribution for P is the Boltzmann–Gibbs distribution

$$\mathbf{P}(P) = \frac{e^{-\lambda \mathcal{E}}}{Z} = \frac{e^{-\lambda \mathcal{H}(R,P)}}{Z} = \frac{e^{\lambda \mathcal{H}(R)} \prod_i p_i^{\lambda r_i}}{Z(\lambda, R)}, \tag{B.18}$$

where λ is the temperature, which depends on the value D of the average energy. Now, if we let $\alpha = \lambda + n$ and $q_i = (\lambda r_i + 1)/(\lambda + n)$, this distribution is in fact the Dirichlet distribution $\mathcal{D}_{\alpha Q}(P)$ with parameters α and Q (note that $\alpha \geq 0$, $q_i \geq 0$, and $\sum_i q_i = 1$). If r_i is uniform, then q_i is also uniform. Thus any Dirichlet distribution can be seen as the result of a MaxEnt calculation.

B.6 Minimum Relative Entropy

The minimum relative entropy principle [486] states that if a prior distribution Q is given, then one should choose a distribution P that satisfies all the constraints of the problem and minimizes the relative entropy $\mathcal{H}(P, Q)$. The MaxEnt principle is obviously a special case of the minimum relative entropy principle, when Q is uniform. As stated, the minimum relative entropy principle is a principle for finding posterior distributions, or for selecting a praticular class of priors. But the proper theory for finding posterior distributions is the Bayesian theory, and therefore the minimum relative entropy principle (or

MaxEnt) cannot have any universal value. In fact, there are known examples where MaxEnt seems to give the “wrong” answer [229]. Thus, in our view, it is unlikely that a general principle exists for the determination of priors. Or if such a principle is really desirable, it should be that the most basic prior of any model should be uniform. In other words, in any modeling effort there is an underlying hierarchy of priors, and priors at the zero level of the hierarchy should always be uniform in a canonical way. It is instructive to look in detail at the cases where the minimum relative entropy principle yields the same result as a Bayesian MAP estimation (see chapter 3).

This page intentionally left blank

Appendix C

Probabilistic Graphical Models

C.1 Notation and Preliminaries

In this appendix, we review the basic theory of probabilistic graphical models [557, 348] and the corresponding factorization of high-dimensional probability distributions. First, a point of notation. If X and Y are two independent random variables, we write $X \perp Y$. Conditional independence on Z is denoted by $X \perp Y | Z$. This means that $\mathbf{P}(X, Y|Z) = \mathbf{P}(X|Z)\mathbf{P}(Y|Z)$. It is important to note that conditional independence implies neither marginal independence nor the converse. By $G = (V, E)$ denote a graph with a set V of vertices and a set E of edges. The vertices are numbered $V = \{1, 2, ..., n\}$. If the edges are directed, we write $G = (V, \vec{E})$. In all the graphs to be considered, there is at most one edge between any two vertices, and there are no edges from a vertex to itself. In an undirected graph, $N(i)$ represents the sets of all the neighbors of vertex i and $C(i)$ represents the set of all the vertices that are connected to i by a path. So,

$$N(i) = \{j \in V : (i, j) \in E\}. \tag{C.1}$$

If there is an edge between any pair of vertices, a graph is said to be *complete*. The *cliques* of G are the subgraphs of G that are both complete and maximal. The *clique graph* G^C of a graph G is the graph consisting of a vertex for each clique in G, and an edge between two vertices, if and only if the corresponding cliques have a nonempty intersection.

In a directed graph, the direction of the edges will often represent causality or time irreversibility. We use the obvious notation $N^-(i)$ and $N^+(i)$ to denote all the parents of i and all the children of i, respectively. Likewise, $C^-(i)$ and $C^+(i)$ denote the ancestors, or the "past," and the descendants of i, or the "future," of i. All these notations are extended in the obvious way to any set of vertices I. So for any $I \in V$,

$$N(I) = \{j \in V : i \in I \quad \text{and} \quad (i,j) \in E\} - I. \tag{C.2}$$

This is also called the boundary of I. In an undirected graph, a set of vertices I is *separated* from a set J by a set K if and only if I and J are disjoint and any path from any vertex in I to any vertex in J contains a vertex in K.

We are interested in high-dimensional probability distributions of the form $\mathbf{P}(X_1, ..., X_n)$, where the X variables represent both hidden and observed variables. In particular, we are interested in the factorization of such distributions into products of simpler distributions, such as conditionals and marginals. Obviously, it is possible to describe the joint distribution using the marginals

$$\mathbf{P}(X_1, ..., X_n) = \prod_{i=0}^{n-1} \mathbf{P}(X_{i+1}|X_1, ..., X_i). \tag{C.3}$$

The set of complete conditional distributions $\mathbf{P}(X_i|X_j : j \neq i)$ also defines the joint distribution in a unique way, *provided they are consistent* (or else no joint distribution can be defined) [68, 20]. The complete set of marginals $\mathbf{P}(X_i)$ is in general highly insufficent to define the joint distribution, except in special cases (see factorial distributions below). The problem of determining a multivariate joint distribution uniquely from an arbitrary set of marginal and conditional distributions is examined in [198]. As we shall see, graphical models correspond to joint distributions that can be expressed economically in terms of local conditionals, or joint distributions over small clusters of variables. Probabilistic inference in such models allows one to approximate useful probabilities, such as posteriors. A number of techniques are typically used to carry inference approximations, including probability propagation, Monte Carlo methods, statistical mechanics, variational methods, and inverse models.

For technical reasons [557], we assume that $\mathbf{P}(X_1, ..., X_n)$ is positive everywhere, which is not restrictive for practical applications because rare events can be assigned very small but nonzero probabilities. We consider graphs of the form $G = (V, E)$, or $G = (V, \vec{E})$, where each variable X_i is associated with the corresponding vertex i. We let X_I denote the set of variables $X_i : i \in I$, associated with a set I of indices. For a fixed graph G, we will denote by $\mathcal{P}(G)$ a family of probability distributions satisfying a set of independence assumptions embodied in the connectivity of G. Roughly speaking, the absence of an edge signifies the existence of an independence relationship. These independence relationships are defined precisely in the next two sections, in the two main cases of undirected and directed graphs. In modeling situations, the real probability distribution may not belong to the set $\mathcal{P}(G)$, for any G. The goal then is to find a G and a member of $\mathcal{P}(G)$ as close as possible to the real distribution—for instance, in terms of relative entropy.

C.2 The Undirected Case: Markov Random Fields

In the undirected case, the family $\mathcal{P}(G)$ corresponds to the notion of Markov random field, or Markov network, or probabilistic independence network, or, in a slightly different context, Boltzmann machine [272, 2]. Symmetric interaction models are typically used in statistical mechanics—for example, Ising models and image processing [199, 392], where associations are considered to be more correlational than causal.

C.2.1 Markov Properties

A Markov random field on a graph G is characterized by any one of the following three equivalent Markov independence properties. The equivalence of these properties is remarkable, and its proof is left as an exercise.

1. **Pairwise Markov Property.** Nonneighboring pairs X_i and X_j are independent, conditional on all the other variables. That is, for any $(i, j) \notin E$,
$$X_i \perp X_j | X_{V-\{i,j\}}. \tag{C.4}$$
2. **Local Markov Property.** Conditional on its neighbors, any variable X_i is independent of all the other variables. That is, for any i in V,
$$X_i \perp X_{V-N(i)\cup\{i\}} | X_{N(i)}. \tag{C.5}$$
3. **Global Markov Property.** If I and J are two disjoint sets of vertices, separated by K, the corresponding set of variables is independent conditional on the variables in the third set:
$$X_I \perp X_J | X_K. \tag{C.6}$$

These independence properties are equivalent to the statement
$$\mathbf{P}(X_i | X_{V-\{i\}}) = \mathbf{P}(X_i | X_{N(i)}). \tag{C.7}$$

C.2.2 Factorization Properties

The functions $\mathbf{P}(X_i | X_j : j \in N(i))$ are called the *local characteristics* of the Markov random field. It can be shown that they uniquely determine the global distribution $\mathbf{P}(X_1, ..., X_n)$, although in a complex way. In particular, and unlike what happens in the directed case, the global distribution is not the product of all the local characteristics. There is, however, an important theorem that relates Markov random fields to Boltzmann–Gibbs distributions. It can be shown that, as a result of the local independence property, the global distribution of a Markov random field has the functional form

$$\mathbf{P}(X_1, ..., X_n) = \frac{e^{-f(X_1,...,X_n)}}{Z} = \frac{e^{-\sum_C f_C(X_C)}}{Z}, \tag{C.8}$$

where Z is the usual normalizing factor. C runs over all the cliques of G, and f_C is called the *potential* or clique function of clique C. It depends only on the variables X_C occurring in the corresponding clique. f is also called the energy. In fact, $\mathbf{P}$ and G determine a Markov random field if and only if (C.8) holds [500].

It is easy to derive the local characteristics and marginals from the potential clique functions by applying the definition in combination with the Boltzmann–Gibbs representation. The potential functions, on the other hand, are not unique. The determination of a set of potential functions in the general case is more elaborate. But there are formulas to derive the potential functions from the local characteristics. There is an important special case that is particularly simple. This is when the graph G is *triangulated.* A graph G is triangulated if any cycle of length greater than or equal to 4 contains at least one chord. A singly connected graph (i.e. a tree) is an important special case of a triangulated graph. A graph is triangulated if and only if its clique graph has a special property called the *running intersection* property, which states that if a vertex of G belongs to two cliques C_1 and C_2 of G, it must also belong to all the other cliques on a path from C_1 to C_2 in the clique graph G^C. The intersection of two neighboring cliques C_1 and C_2 of G—that is, two adjacent nodes of G^C—is called a *separator.* In a triangulated graph, a separator of C_1 and C_2 separates them in the probabilistic independence sense defined above.

Another important characterization of triangulated graphs is in terms of *perfect numbering.* A numbering of the nodes in V is perfect if for all i, $N(i) \cap \{1, 2, ..., i-1\}$ is complete. A graph is triangulated if and only if it admits a perfect numbering (see [512], [350], and references therein).] The key point here is that for Markov random fields associated with a triangulated graph, the global distribution has the form

$$\mathbf{P}(X_1, ..., X_n) = \frac{\prod_C \mathbf{P}(X_C)}{\prod_S \mathbf{P}(X_S)}, \tag{C.9}$$

where C runs over the cliques and S runs over the separators, occurring in a *junction tree*, that is, a maximal spanning tree of G^C. $\prod_C \mathbf{P}(X_C)$ is the marginal joint distribution of X_C. The clique potential functions are then obvious.

A very special case of the Markov random field is when the graph G has no edges. This is the case when all the variables X_i are independent and $\mathbf{P}(X_1,...,X_n) = \prod_{i=1}^n \mathbf{P}(X_i)$. Such joint distributions or Markov random fields are called *factorial*. Given a multivariate joint distribution P, it is easy to see that among all factorial distributions, the one that is closest to P in relative entropy is the product of the marginals of P.

C.3 The Directed Case: Bayesian Networks

In the directed case, the family $\mathcal{P}(G)$ corresponds to the notions of Bayesian networks, belief networks, directed independence probabilistic networks, directed Markov fields, causal networks, influence diagrams, and even Markov meshes [416, 557, 121, 106, 286, 246] (see [322] for a simple molecular biology illustration). As already mentioned, the direction on the edges usually represents causality or time irreversibility. Such models are common, for instance, in the design of expert systems.

In the directed case, we have a directed graph $G = (V, \vec{E})$. The graph is also assumed to be *acyclic*, that is, with no directed cycles. This is because it is not possible to consistently define the joint probability of the variables in a cycle from the product of the local conditioning probabilities. That is, in general the product $\mathbf{P}(X_2|X_1)\mathbf{P}(X_3|X_2)\mathbf{P}(X_1|X_3)$ does not consistently define a distribution on X_1, X_2, X_3. An acyclic directed graph represents a partial ordering. In particular, it is possible to number its vertices so that if there is an edge from i to j, then $i < j$. In other words, the partial ordering associated with the edges is consistent with the numbering. This ordering is also called a topological sort. We will assume that such an ordering has been chosen whenever necessary, so that, the past of i $C^-(i)$ is included in $\{1,2,...,i-1\}$, and the future $C^+(i)$ is included in $\{i+1,...,n\}$. The *moral* of $G = (V, \vec{E})$ is the undirected graph $G^M = (V, E + M)$ obtained by removing the direction on the edges of G and by adding an edge between any two nodes that are parents of the same child in G (if they are not already connected, of course). The term "moral" was introduced in [350] and refers to the fact that all parents are "married." We can now describe the Markov independence properties of graphical models with an underlying acyclic directed graph.

C.3.1 Markov Properties

A Bayesian network on a directed acyclic graph G is characterized by any one of a number of equivalent independence properties. In all cases, the basic

Markov idea in the directed case is that, conditioned on the present, the future is independent of the past or, equivalently, that in order to predict the future, all the relevant information is assumed to be in the present.

Pairwise Markov Property

Nonneighboring pairs X_i and X_j with $i < j$ are independent, conditional on all the other variables in the past of j. That is, for any $(i, j) \notin \vec{E}$ and $i < j$,

$$X_i \perp X_j | X_{C^-(j)-\{i\}}. \tag{C.10}$$

In fact, one can replace $C^-(j)$ with the larger set $\{1, ..., j-1\}$. Another equivalent statement is that, conditional on a set of nodes I, X_i is independent of X_j if and only if i and j are *d-separated*, that is, if there is no d-connecting path from i to j [121]. A d-connecting path from i to j is defined as follows. Consider a node k on a path from i to j. The node k is called linear, divergent, or convergent, depending on whether the two edges adjacent to it on the path are incoming and outgoing, both outgoing, or both incoming. The path from i to j is d-connecting with respect to I if and only if every interior node k on the path is either (1) linear or diverging and not a member of I, or (2) converging, and $[k \cup C^+(k)] \cap I \neq \emptyset$. Intuitively, i and j are d-connected if and only if either (1) there is a causal path between them or (2) there is evidence in I that renders the two nodes correlated with each other.

Local Markov Property

Conditional on its parents, a variable X_i is independent of all other nodes, except for its descendants. Thus

$$X_i \perp X_j | X_{N^-(i)}, \tag{C.11}$$

as long as $j \notin C^+(i)$ and $j \neq i$.

Global Markov Property

If I and J are two disjoint sets of vertices, we say that K separates I and J in the directed graph G if and only if K separates I and J in the moral undirected graph of the smallest ancestral set containing I, J, and K [349]. With this notion of separation, the global Markov property is the same—that is, if K separates I and J,

$$X_I \perp X_J | X_K. \tag{C.12}$$

It can be also be shown [557] that the directed graph G satisfies all the Markov independence relationships of the associated moral graph G^M. The converse is not true in general, unless G^M is obtained from G by removing edge orientation only, that is, without any marriages. Finally, any one of the three Markov independence properties is equivalent to the statement

$$\mathbf{P}(X_i|X_{C^-(i)}) = \mathbf{P}(X_i|X_{N^-(i)}). \tag{C.13}$$

In fact, $C^-(i)$ can be replaced by the larger set $\{1, ..., i-1\}$.

C.3.2 Factorization Properties

It is not difficult to see, as a result, that the unilateral local characteristics $\mathbf{P}(X_i|X_{N^-(i)})$ are consistent with one another, and in fact uniquely determine a Bayesian network on a given graph. Indeed, we have

$$\mathbf{P}(X_1, ..., X_n) = \prod_i \mathbf{P}(X_i|X_{N^-(i)}). \tag{C.14}$$

This property is fundamental. The local conditional probabilities can be specified in terms of lookup tables, although this is often impractical due to the size of the tables. A number of more compact but also less general representations are often used, such as noisy OR- [416] or NN-style representations, such as sigmoidal belief networks [395] for binary variables, where the characteristics are defined by local connection weights and sigmoidal functions, or the obvious generalization to multivalued variables using normalized exponentials. Having a local NN at each vertex to compute the local characteristics is another example of hybrid model parameterization.

C.3.3 Learning and Propagation

There are several levels of learning in graphical models in general and Bayesian networks in particular, from learning the graph structure itself to learning the local conditional distributions from the data. With the exception of section C.3.6, these will not be discussed here; reviews and pointers to the literature can be found in [106, 246]. Another fundamental operation with Bayesian networks is the propagation of evidence, that is, the updating of the probabilities of each X_i conditioned on the observed node variables. Evidence propagation is NP-complete in the general case [135]. But for singly connected graphs (no more than one path between any two nodes in the underlying undirected graph), propagation can be executed in time linear with n, the number of nodes, using a simple message-passing approach [416, 4]. In the general case, all known exact algorithms for multiply connected networks rely on the construction of an equivalent singly connected network, the junction tree, by clustering the original variables, according to the cliques of the corresponding triangulated moral graph ([416, 350, 467], with refinements in [287]).

A similar algorithm for the estimation of the most probable configuration of the variables X_i is given in [145]. Schachter et al. [468] show that all the known exact inference algorithms are equivalent in some sense to the algorithms in [287] and [145]. An important conjecture, supported both by emprirical evidence and results in coding theory, is that the simple message-passing algorithm of [416] yields reasonably good approximations in the multiply connected case (see [385] for details).

C.3.4 Generality

It is worth noting that the majority of models used in this book can be viewed as instances of Bayesian networks. Artificial feed-forward NNs are Bayesian networks in which the local conditional probability functions are delta functions. Likewise, HMMs and Markov systems in general have a very simple Bayesian network representation. In fact, HMMs are a special case of both Markov random fields and Bayesian networks. We leave as a useful exercise for the reader to derive these representations, as well as the Bayesian network representation of many other concepts such as mixtures, hierarchical priors, Kalman filters and other state space models, and so on. The generality of the Bayesian network representation is at the root of many new classes of models currently under investigation. This is the case for several generalizations of HMMs, such as input-output HMMs (see chapter 9), tree-structured HMMs [293], and factorial HMMs [205].

When the general Bayesian network propagation algorithms are applied in special cases, one “rediscovers” well-known algorithms. For instance, in the case of HMMs, one obtains the usual forward–backward and Viterbi algorithms directly from Pearl’s algorithm [493]. The same is true of several algorithms in coding theory (turbo codes, Gallager–Tanner–Wiberg decoding) and in the theory of Kalman filters (the Rauch–Tung–Streibel smoother), and even of certain combinatorial algorithms (fast Fourier transform) [4, 204]. We suspect that the inside–outside algorithm for context-free grammar is also a special case, although we have not checked carefully. While belief propagation in general remains NP-complete, approximate algorithms can often be derived using Monte Carlo methods such as Gibbs sampling [210, 578], and variational methods such as mean field theory (appendix A and [465, 276, 204]), sometimes leveraging the particular structure of a network. Gibbs sampling is particularly attractive for Bayesian networks because of its simplicity and generality.

C.3.5 Gibbs Sampling

Assuming that we observe the values of the variables associated with some of the visible nodes, we want to sample the value of any other node i according to its conditional probability, given all the other variables. From the factorization (C.14), we have

$$\mathbf{P}(X_i|X_{V-\{i\}}) = \frac{\mathbf{P}(X_V)}{\mathbf{P}(X_{V-\{i\}})} = \frac{\prod_j \mathbf{P}(X_j|X_{N^-(j)})}{\sum_{x_i} \mathbf{P}(X_1, \ldots, X_i = x_i, \ldots, X_n)}, \tag{C.15}$$

which yields, after simplifications of common numerator and denominator terms,

$$\mathbf{P}(X_i|X_{V-\{i\}}) = \frac{\mathbf{P}(X_i|X_{N^-(i)}) \prod_{j \in N^+(i)} \mathbf{P}(X_j|X_{N^-(j)})}{\sum_{x_i} \mathbf{P}(X_i = x_i|N^-(i)) \prod_{j \in N^+(i)} \mathbf{P}(X_j|X_{N^-(j)})}. \tag{C.16}$$

As expected, the conditional distributions needed for Gibbs sampling are local and depend only on i, its parents, and its children. Posterior estimates can then be obtained by averaging simple counts at each node, which requires very little memory. Additional precision may be obtained by averaging the *probabilities* at each node (see [396] for a partial discussion). As in any Gibbs sampling situation, important issues are the duration of the procedure (or repeated procedure, if the sampler is used for multiple runs) and the discarding of the initial samples ("burn-in"), which can be nonrepresentative of the equilibrium distribution.

C.3.6 Sleep–Wake Algorithm and Helmholtz Machines

A theoretically interesting, but not necessarily practical, learning algorithm for the conditional distributions of a particular class of Bayesian networks is described in [255, 146]. These Bayesian networks consist of two inverse models: the recognition network and the generative network. Starting from the input layer, the recognition network has a feed-forward layered architecture. The nodes in all the hidden layers correspond to stochastic binary variables, but more general versions—for instance, with multivalued units—are possible. The local conditional distributions are implemented in NN style, using combinational weights and sigmoidal logistic functions. The probability that unit i is on is given by

$$\mathbf{P}(X_i = 1) = \frac{1}{1 + e^{-\sum_{k \in N^-(i)} w_{ik} x_k + b_i^l}}, \tag{C.17}$$

where x_k denotes the states of the nodes in the previous layer. The generative network mirrors the recognition network. It is a feed-forward layered network that begins with the top hidden layer of the recognition network and ends up with the input layer. It uses the same units but with a reverse set of connections. These reverse connections introduce local loops so the combined architecture is not acyclic. This is not significant, however, because the networks are used in alternation rather than simultaneously.

The sleep–wake algorithm, named after its putative biological interpretation, is an unsupervised learning algorithm for the forward and backward connection weights. The algorithm alternates between two phases. During each phase, the unit activities in one of the networks are used as local targets to train the weights in the opposite network, using the delta rule. During the wake phase, the recognition network is activated and each generative weight w_{jk} is updated by

$$\Delta w_{jk} = \eta x_k (x_j - p_j), \tag{C.18}$$

where x_j represents the state of unit j in the recognition network and p_j the corresponding probability calculated as in (C.17), using the generative connections. A symmetric update rule is used during the sleep phase, where the fantasies (dreams) produced by the generative network are used to modify the recognition weights [255, 574].

Appendix D

HMM Technicalities, Scaling, Periodic Architectures, State Functions, and Dirichlet Mixtures

D.1 Scaling

As already pointed out, the probabilities $\mathbf{P}(\pi|O, w)$ are typically very small, beyond machine precision, and so are the forward variables $\alpha_i(t)$, as t increases. A similar observation can be made for the backward variables $\beta_i(t)$, as t decreases. One solution for this problem is to scale the forward and backward variables at time t by a suitable coefficient that depends only on t. The scalings on the αs and βs are defined in a complementary way so that the learning equations remain essentially invariant under scaling. We now give the exact equations for scaling the forward and backward variables, along the lines described in [439].[1] For simplicity, throughout this section, we consider an HMM with emitting states only. We leave as an exercise for the reader to adapt the equations to the general case where delete states are also present.

[1]The scaling equations in [439] contain a few errors. A correction sheet is available from the author.

D.1.1 Scaling of Forward Variables

More precisely, we define the scaled variables thus:

$$\hat{\alpha}_i(t) = \frac{\alpha_i(t)}{\sum_j \alpha_j(t)}. \tag{D.1}$$

At time 0, for any state i, we have $\alpha_i(0) = \hat{\alpha}_i(0)$. The scaled variables $\hat{\alpha}_i(t)$ can be computed recursively by alternating a propagation step with a scaling step. Let $\hat{\hat{\alpha}}_i(t)$ represent the propagated $\hat{\alpha}_i(t)$ before scaling. Assuming that all variables have been computed up to time $t-1$, we first propagate $\hat{\alpha}_i$ by (7.5):

$$\hat{\hat{\alpha}}_i(t) = \sum_{j \in N^-(i)} \hat{\alpha}_j(t-1) t_{ij} e_{iX^t}, \tag{D.2}$$

with $\hat{\hat{\alpha}}_i(0) = \alpha_i(0)$. The same remarks as for the propagation of the $\alpha_i(t)$ apply here. Therefore, using (D.1),

$$\hat{\hat{\alpha}}_i(t) = \frac{\alpha_i(t)}{\sum_j \alpha_j(t-1)}. \tag{D.3}$$

We then scale the $\hat{\hat{\alpha}}(t)$s, which by (D.3) is equivalent to scaling the αs:

$$\frac{\hat{\hat{\alpha}}_i(t)}{\sum_j \hat{\hat{\alpha}}_j(t)} = \frac{\alpha_i(t)}{\sum_j \alpha_j(t)} = \hat{\alpha}_i(t). \tag{D.4}$$

This requires computing at each time step the scaling coefficient $c(t) = \sum_i \hat{\hat{\alpha}}_i(t)$. From (D.3), the relation between $c(t)$ and the scaling coefficent $C(t) = \sum_i \alpha_i(t)$ of the αs is given by:

$$C(t) = \prod_{\tau=1}^{t} c(\tau). \tag{D.5}$$

D.1.2 Scaling of Backward Variables

The scaling of the backward variables is slightly different, in that the scaling factors are computed from the forward propagation rather than from the βs. In particular, this implies that the forward propagation must be completed in order for the backward propagation to begin. Specifically, we define the scaled

$$\hat{\beta}_i(t) = \frac{\beta_i(t)}{D(t)}. \tag{D.6}$$

The scaling coefficient is defined to be

$$D(t) = \prod_{\tau=t}^{T} c(\tau). \tag{D.7}$$

The reason for this choice will become apparent below. Assuming all variables have been computed backward to time $t+1$, the $\hat{\beta}$s are first propagated backward using (7.10) to yield the variables

$$\hat{\hat{\beta}}_i(t) = \sum_{j \in N^+(i)} \hat{\beta}_j(t+1) t_{ji} e_{jX^{t+1}}. \tag{D.8}$$

The $\hat{\hat{\beta}}_i(t)$ are then scaled by $c(t)$, to yield

$$\hat{\beta}_i(t) = \frac{\hat{\hat{\beta}}_i(t)}{c(t)} = \frac{\beta_i(t)}{D(t)} \tag{D.9}$$

as required by (D.6).

D.1.3 Learning

Consider now any learning equation, such as the EM equation for the transition parameters (7.31):

$$t_{ji}^{+} = \frac{\sum_{t=0}^{T} \gamma_{ji}(t)}{\sum_{t=0}^{T} \gamma_i(t)} = \frac{\sum_{t=0}^{T} \alpha_i(t) t_{ji} e_{jX^{t+1}} \beta_j(t+1)}{\sum_{t=0}^{T} \sum_{j \in S} \alpha_i(t) t_{ji} e_{jX^{t+1}} \beta_j(t+1)}. \tag{D.10}$$

Any product of the form $\alpha_i(t)\beta_j(t+1)$ is equal to $C\hat{\alpha}_i(t)\hat{\beta}_j(t+1)$, with $C = C(t)D(t+1) = \prod_1^T c(t)$ independent of t. The constant C cancels out from the numerator and the denominator. Therefore the same learning equation can be used by simply replacing the αs and βs with the corresponding scaled $\hat{\alpha}$s and $\hat{\beta}$s. Similar remarks apply to the other learning equations.

D.2 Periodic Architectures

D.2.1 Wheel Architecture

In the wheel architecture of chapter 8, we can consider that there is a *start* state connected to all the states in the wheel. Likewise, we can consider that all the states along the wheel are connected to an *end* state. The wheel architecture contains no delete states, and therefore all the algorithms (forward, backward, Viterbi, and scaling) are simplified, in the sense that there is no need to distinguish between emitting and delete states.

D.2.2 Loop Architecture

The loop architecture is more general than the wheel architecture because it contains delete states, and even the possibility of looping through delete states. We introduce the following notation:

- h is the anchor state of the loop. The anchor state is a delete (silent) state, although it is not associated with any main state.

- L denotes the set of states in the loop.

- κ denotes the probability of going once around the loop silently. It is the product of all the t_{ji} associated with consecutive delete states in the loop.

- t^d_{ji} is the probability of the shortest direct silent path from i to j in the architecture.

- t^D_{ji} is the probability of moving silently from i to j. For any two states connected by at least one path containing the anchor, we have $t^D_{ji} = t^d_{ji}(1 + \kappa + (\kappa^2)\ldots) = t^d_{ji}/(1-\kappa)$.

Forward Propagation Equations

Forward propagation equations are true both for instantaneous propagation and at equilibrium. For any emitting state $i \in E$,

$$\alpha_i(t+1) = \sum_{j \in N^-(i)} \alpha_j(t) t_{ij} e_{i\mathrm{X}^{t+1}}. \tag{D.11}$$

For any silent state i, including the anchor state,

$$\alpha_i(t+1) = \sum_{j \in N^-(i)} \alpha_j(t+1) t_{ij}. \tag{D.12}$$

For the anchor state, one may separate the contribution from the loop and from the flanks as

$$\alpha_h(t+1) = \sum_{j \in N^-(h)-L} \alpha_j(t+1) t_{hj} + \sum_{j \in N^-(h)\cap L} \alpha_j(t+1) t_{hj}. \tag{D.13}$$

Implementations

There are three possible ways of implementing the propagation. First, iterate instantaneous propagation equations until equilibrium is reached. Second, iterate the equilibrium equations only once through the loop, for the anchor state. That is, write $x = \alpha_h(t+1)$, forward-propagate the above equations once through the loop as a function of x, and solve for x at the end. Once the loop is completed, this yields an equation of the form $x = ax + b$ and so $x = b/(1-a)$. Then replace x by its newly found value in the expression of $\alpha_i(t+1)$ for all $i \in L$.

Third, solve analytically for x. That is, directly find the equilibrium value of $x = \alpha_h(t+1)$ (i.e., a and b above). For this, note that the paths leading to the expression of $\alpha_h(t+1)$ can be partitioned into two classes depending on whether X^{t+1} is emitted inside or outside the loop:

$$\alpha_h(t+1) = \sum_{j\in N^-(h)-L} \alpha_j(t+1)t_{hj}(1+\kappa+\kappa^2+\ldots) + \sum_{j\in E\cap L} \alpha_j(t+1)t^D_{hj}. \quad \text{(D.14)}$$

Thus the second term in the right-hand side accounts for the case where the emission of X^{t+1} inside the loop is followed by any number of silent revolutions terminating with the anchor state. This term contains unknown quantities such as $\alpha_j(t+1)$. These are easy to calculate, however, using the values of $\alpha_j(t)$ that are known from the previous epoch of the propagation algorithm. So finally,

$$\alpha_h(t+1) = \frac{1}{1-\kappa} \sum_{j\in N^-(h)-L} \alpha_j(t+1)t_{hj} + \sum_{j\in E\cap L}\sum_{k\in N^-(j)} \alpha_k(t)a_{jk}e_{j\mathsf{X}^{t+1}}t^D_{hj}. \quad \text{(D.15)}$$

For the specific calculation of the last sum above, we consider the following implementation, where we forward-propagate two quantities, $\alpha_i(t)$ and $\alpha_i^L(t)$. $\alpha_i^L(t)$ is to be interpreted as the probability of being in state i at time t while having emitted symbol t *in the loop and not having traversed the anchor state yet again.* For any emitting state i in the loop, the propagation equations are

$$\alpha_i(t+1) = \alpha_i^L(t+1) = \sum_{j\in N^-(i)} \alpha_j(t)t_{ij}e_{i\mathsf{X}^{t+1}}. \quad \text{(D.16)}$$

For any mute state (delete states and anchor) i in the loop, the propagation equations are

$$\alpha_i^L(t+1) = \sum_{j\in N^-(i)\cap L} \alpha_j^L(t+1)t_{ij}. \quad \text{(D.17)}$$

These equations should be initialized with $\alpha_h^L(t+1) = 0$ and propagated all the way once through the loop to yield, at the end, a new value for $\alpha_h^L(t+1)$.

We then have

$$\alpha_h(t+1) = \frac{1}{1-\kappa}[\sum_{j \in N^-(h)-L} \alpha_j(t+1)t_{hj} + \alpha_h^L(t+1)]. \tag{D.18}$$

At time 0, initialization is as follows:

- $\alpha_i(0) = 0$ for any emitting state
- $\alpha_i^L(0) = 0$ for any state, including the anchor
- $\alpha_h(0) = \sum_{j \in N^-(h)-L} \alpha_j(0)t_{hj}/(1-\kappa)$
- $\alpha_i(0) = \sum_{j \in N^-(i)} \alpha_j(0)t_{ij}$ for any mute state in the loop exept the anchor

All variables can be computed with one pass through the loop by using propagating $\alpha(t)$ and $\alpha^L(t)$ simultaneously through the loop, in the following order. At step t, assume that $\alpha_i(t)$ is known for the anchor state and all emitting states. Then:

- Set $\alpha_h^L(t+1)$ to 0.
- Forward-propagate simultaneously through the loop the quantities $\alpha_i(t)$ for mute states (D.12), $\alpha_i(t+1) = \alpha_i^L(t+1)$ for emitting states (D.16), and $\alpha_i^L(t+1)$ for all mute states (D.17).
- Calculate $\alpha_h(t+1)$ by (D.18).

Backward propagation and scaling equations for the loop architecture can be derived along the same lines.

D.3 State Functions: Bendability

As discussed in chapters 7 and 8, any function that depends on the local amino acid or nucleotide composition of a family, such as entropy, hydrophobicity, or bendability, can be studied with HMMs. In particular, the expectation of such a function computed from the HMM backbone probabilities enhances patterns that are not always clearly present in individual members of the family. This expectation is straightforward to compute when the corresponding function or scale is defined over single alphabet letters (entropy, hydrophobicity). A little more care is needed when the function depends on adjacent pair or triplet of letters, usually DNA dinucleotides or trinucleotides (bendability, nucleosome positioning, stacking energies, propeller twist). Convolving several functions with the HMM backbone can help determine structural and functional properties of the corresponding family. Over 50 different functions are available in our current HMM simulator. Here we show how to compute such expectations in the case of bendability, which is a little harder because of its dependence on triplets rather than single letters.

D.3.1 Motivation

Average bending profiles can be computed directly from a multiple alignment of the available sequences to avoid the risk of introducing exogenous artifacts. It is useful, however, to be able to define and compute bending profiles directly from an HMM, for several reasons.

- The computation is faster because it can be executed as soon as the HMM is trained, without having to align all the sequences to the model.
- In many of the cases we have tried, the profiles derived from the HMM and the multiple alignment have very similar characteristics. Consistency of the two bending profiles can be taken as further evidence that the HMM is a good model of the data. Discrepant cases may yield additional insights.
- In certain cases—for example, when few data are available—a well-regularized HMM may yield better bending profiles.

D.3.2 Definition of HMM Bending Profiles

We assume a standard linear HMM architecture, but similar calculations can be done with the loop or wheel architectures. In the definition of an HMM bending profile, it is natural to consider only HMM main states $m_0, \ldots, m_{N+1}$, where m_0 is the *start* state and m_{N+1} is the *end* state (unless there are particularly strong transitions to insert states or delete states, in which case such states should be included in the calculation). The bendability $B(i, O)$ of a sequence $O = (\mathsf{X}_O^1, \ldots, \mathsf{X}_O^N)$ at a position i, away from the boundary, can be defined by averaging triplet bendabilities over a window of length $W = 2l + 1$:

$$B(i, O) = \frac{1}{W} \sum_{j=i-l}^{i+l-2} b(\mathsf{X}_O^j, \ldots, \mathsf{X}_O^{j+2}), \tag{D.19}$$

where $b(\mathsf{X}, \mathsf{Y}, \mathsf{Z})$ denotes the bendability of the XYZ triplet according to some scale ([96] and references therein). The bendability $B(i)$ of the family at position i is then naturally defined by taking the average over all possible backbone sequences:

$$B(i) = \sum_O B(i, O)\mathbf{P}(O). \tag{D.20}$$

This approach, however, is not efficient because the number of possible sequences is exponential in N. Fortunately, there exists a better way of organizing this calculation.

D.3.3 Efficient Computation of Bendability Profiles

From (D.20), we find

$$B(i) = \sum_O B(i, O) \prod_{k=1}^{N} e_{k\mathsf{X}_O^k} \prod_{k=0}^{N+1} t_{m_k m_{k+1}}. \tag{D.21}$$

The last product is the product of all HMM backbone transitions and is equal to some constant C. Substituting (D.19) in (D.21), we have

$$B(i) = \frac{C}{W} \sum_O \sum_{j=i-l}^{i+l-2} b(\mathsf{X}_O^j, \ldots, \mathsf{X}_O^{j+2}) \prod_{k=1}^{N} e_{k\mathsf{X}_O^k}. \tag{D.22}$$

Interchanging the sums yields

$$B(i) = \frac{C}{W} \sum_{j=i-l}^{i+l-2} \sum_O b(\mathsf{X}_O^j, \ldots, \mathsf{X}_O^{j+2}) \prod_{k=1}^{N} e_{k\mathsf{X}_O^k}. \tag{D.23}$$

To sum over all sequences, we can partition the sequences into different groups according to the letters X, Y, and Z appearing at positions $j, j+1$, and $j+2$. After simplifications, this finally yields

$$B(i) = \frac{C}{W} \sum_{j=i-l}^{i+l-2} \sum_{\mathsf{X,Y,Z}} b(\mathsf{X}, \mathsf{Y}, \mathsf{Z}) e_{j\mathsf{X}} e_{j+1\mathsf{Y}} e_{j+2\mathsf{Z}}. \tag{D.24}$$

Thus the definition in (D.20) is equivalent to the definition in (D.24), where summations within a window occur over all possible alphabet triplets weighted by the product of the corresponding emission probabilities at the corresponding locations. Definition (D.24) is of course the easiest to implement and we have used it to compute bending profiles from trained HMMs, usually omitting the constant scaling factor C/W. In general, boundary effects for the first and last l states are not relevant.

D.4 Dirichlet Mixtures

First recall from chapters 2 and 3 that the mean of a Dirichlet distribution $\mathcal{D}_{\alpha Q}(P)$ is Q, and the maximum is reached for $p_{\mathsf{X}} = (\alpha q_{\mathsf{X}} - 1)/(\alpha - |A|)$

provided $p_X \geq 0$ for all X. A mixture of Dirichlet distributions is defined by $\mathbf{P}(P) = \sum_1^n \lambda_i \mathcal{D}_{\alpha_i Q_i}(P)$, where the mixture coefficients must satisfy $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. The expectation of the mixture is $\sum_i \lambda_i Q_i$, by linearity of the expectation. For a Dirichlet mixture, the maximum in general cannot be determined analytically.

D.4.1 Dirichlet Mixture Prior

Now consider the problem of choosing a prior for the emission distribution $P = (p_X)$ associated with an HMM emitting state or, equivalently, the dice model associated with a column of an alignment. Thus here p_X are the parameters of the model. The data D consists of the letters observed in the column with the corresponding counts $D = (n_X)$, with $\sum_X n_X = N$. The likelihood function for the data is given by

$$\mathbf{P}(D|M) = \mathbf{P}(n_X|p_X) = \prod_X p_X^{n_X}. \tag{D.25}$$

We have seen that a natural prior is to use a single Dirichlet distribution. The flexibility of such a prior may sometimes be too limited, especially if the same Dirichlet is used for all columns or all emitting states. A more flexible prior is a Dirichlet mixture

$$\mathbf{P}(P) = \sum_{i=1}^{n} \lambda_i \mathcal{D}_{\alpha_i Q_i}(P) \tag{D.26}$$

as in [489], where again the same mixture is used for all possible columns, to reflect the general distribution of amino acid in proteins. The mixture components $\mathcal{D}_{\alpha_i Q_i}$, their number, and the mixture coefficients can be found by clustering methods. An alternative for protein models is to use the vectors Q_i associated with the columns of a PAM matrix (see chapter 10 and [497]). Note that the present mixture model is different from having a different set of mixing coefficients for each column prior. It is also different from parameterizing each P as a mixture in order to reduce the number of HMM emission parameters, provided $n < |A|$ ($n = 9$ is considered optimal in [489]), in a way similar to the hybrid HMM/NN models of chapter 9. We leave it as an exercise for the reader to explore such alternatives.

Now, from the single Dirichlet mixture prior and the likelihood, the posterior is easily computed using Bayes' theorem as usual

$$\mathbf{P}(P|D) = \frac{1}{\mathbf{P}(D)} \sum_{i=1}^{n} \lambda_i \frac{B(\beta_i, R_i)}{B(\alpha_i, Q_i)} \mathcal{D}_{\beta_i R_i}(P). \tag{D.27}$$

The new mixture components are given by

$$\beta_i = N + \alpha_i \quad \text{and} \quad r_{i\mathrm{X}} = \frac{n_{\mathrm{X}} + \alpha_i q_{i\mathrm{X}}}{N + \alpha_i}. \tag{D.28}$$

The beta function B is defined as

$$B(\alpha, Q) = \frac{\prod_{\mathrm{X}} \Gamma(\alpha q_{\mathrm{X}})}{\Gamma(\alpha)}, \tag{D.29}$$

as usual with $\alpha \geq 0$, $q_{\mathrm{X}} \geq 0$, and $\sum_{\mathrm{X}} q_{\mathrm{X}} = 1$. The posterior of a mixture of conjugate distributions is also a mixture of conjugate distributions. In this case, the posterior is also a Dirchlet mixture, but with different mixture components and mixture coefficients. Since the integral of the posterior over P must be equal to one, we immediately have the evidence

$$\mathbf{P}(D) = \sum_{i=1}^{n} \lambda_i \frac{B(\beta_i, R_i)}{B(\alpha_i, Q_i)}. \tag{D.30}$$

As pointed out above, the MAP estimate cannot be determined analytically, although it could be approximated by some iterative procedure. The MP estimate, however, is trivial since it corresponds to the average of the posterior

$$p^*_{\mathrm{X}} = \frac{1}{\mathbf{P}(D)} \sum_{i=1}^{n} \lambda_i \frac{B(\beta_i, R_i)}{B(\alpha_i, Q_i)} r_{i\mathrm{X}}. \tag{D.31}$$

This provides a formula for the estimation of optimal model parameters in this framework. Numerical implementation issues are discussed in [489].

D.4.2 Hierarchical Dirichlet Model

In hierarchical modeling, we introduce a higher level of priors, for instance with a Dirichlet prior on the mixture coefficients of the previous model. This two-level model is also a mixture model in the sense that $\mathbf{P}(P|\lambda) = \sum \lambda_i \mathcal{D}_{\alpha_i Q_i}(P)$ but with

$$\mathbf{P}(\lambda) = \mathcal{D}_{\beta Q}(\lambda) = \frac{\Gamma(\beta)}{\prod_i \Gamma(\beta q_i)} \prod_{i=1}^{n} \lambda_i^{\beta q_i - 1}. \tag{D.32}$$

We then have

$$\mathbf{P}(P) = \int_{\lambda} \mathbf{P}(P|\lambda)\mathbf{P}(\lambda) d\lambda. \tag{D.33}$$

Interchanging sums and integrals yields

$$\mathbf{P}(P) = \sum_{i=1}^{n} \mathcal{D}_{\alpha_i Q_i}(P)[\int_{\lambda} \lambda_i \mathcal{D}_{\beta Q}(\lambda) d\lambda] = \sum_{i=1}^{n} q_i \mathcal{D}_{\alpha_i Q_i}(P), \tag{D.34}$$

the second equality resulting from the Dirichlet expectation formula. Thus this two-level hierarchical model is in fact equivalent to a one-level Dirichlet mixture model, where the mixture coefficients q_i are the expectation of the second-level Dirichlet prior in the hierarchical model.

This page intentionally left blank

Appendix E

Gaussian Processes, Kernel Methods, and Support Vector Machines

In this appendix we briefly review several important classes of machine learning methods: Gaussian processes, kernel methods, and support vector machines [533, 141].

E.1 Gaussian Process Models

Consider a regression problem consisting of K input-output training pairs (x_1, y_1),..., (x_K, y_K) drawn from some unknown distribution. The inputs x are n-dimensional vectors. For simplicity, we assume that y is one-dimensional, but the extension to the multidimensional case is straightforward. The goal in regression is to learn the functional relationship between x and y from the given examples. The Gaussian process modeling approach [559, 206, 399], also known as "kriging," provides a flexible probabilistic framework for regression and classification problems. A number of nonparametric regression models, including neural networks with a single infinite hidden layer and Gaussian weight priors, are equivalent to Gaussian processes [398]. Gaussian processes can be used to define probability distributions over spaces of functions directly, without any need for an underlying neural architecture.

A Gaussian process is a collection of variables $Y = (y(x_1), y(x_2), \ldots)$, with a joint Gaussian distribution of the form

$$\mathbf{P}(Y|C,\{x_i\}) = \frac{1}{Z}\exp(-\frac{1}{2}(Y-\mu)^T C^{-1}(Y-\mu)) \tag{E.1}$$

for any sequence $\{x_i\}$, where μ is the mean vector and $C_{ij} = C(x_i, x_j)$ is the covariance of x_i and x_j. For simplicity, we shall assume in what follows that $\mu = 0$. Priors on the noise and the modeling function are combined into the covariance matrix C. Different sensible parameterizations for C are described below. From (E.1), the predictive distribution for the variable y associated with a test case x is obtained by conditioning on the observed training examples. In other words, a simple calculation shows that y has a Gaussian distribution

$$\mathbf{P}(y|\{y_1,...,y_K\}, C(x_i,x_j), \{x_1,...,x_K,x\} = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y-y^*)^2}{2\sigma^2}) \tag{E.2}$$

with

$$y^* = k(x)^T C_K^{-1}(y_1,...,y_K) \quad \text{and} \quad \sigma = C(x,x) - k(x)^T C_K^{-1} k(x) \tag{E.3}$$

where $k(x) = (C(x_1,x),...,C(x_K,x))$ and C_K denotes the covariance matrix based on the K training samples.

E.1.1 Covariance Parameterization

A Gaussian process model is defined by its covariance function. The only constraint on the covariance function $C(x_i, x_j)$ is that it should yield positive semidefinite matrices for any input sample. In the stationary case, the Bochner theorem in harmonic analysis ([177] and given below for completeness) provides a complete characterization of such functions in terms of Fourier transforms. It is well known that the sum of two positive matrices (resp. positive definite) is positive (resp. positive definite). Therefore the covariance can be conveniently parameterized as a sum of different positive components. Useful components have the following forms:

- Noise variance: $\delta_{ij}\theta_1^2$ or, more generally, $\delta_{ij} f(x_i)$ for an input-dependent noise model
- Smooth covariance: $C(x_i,x_j) = \theta_2^2 \exp(-\sum_{u=1}^n \rho_u^2 (x_{iu} - x_{ju})^2)$
- And more generally: $C(x_i,x_j) = \theta_2^2 \exp(-\sum_{u=1}^n \rho_u^2 |x_{iu} - x_{ju}|^r$
- Periodic covariance: $C(x_i,x_j) = \theta_3^2 \exp(-\sum_{u=1}^n \rho_u^2 \sin^2[\pi(x_{iu} - x_{ju})/\gamma_u]$

Notice that a small value of ρ_u characterizes components u that are largely irrelevant for the output in a way closely related to the automatic relevance determination framework [398]. For simplicity, we write θ to denote the vector of hyperparameters of the model. Short of conducting lengthy Monte Carlo integrations over the space of hyperparameters, a single value θ can be estimated by minimizing the negative log-likelihood

$$\mathcal{E}(\theta) = \frac{1}{2}\log\det C_K + \frac{1}{2}Y_K^T C_K^{-1} Y_K + \frac{K}{2}\log 2\pi. \tag{E.4}$$

Without any specific shortcuts, this requires inverting the covariance matrix and is likely to require $O(N^3)$ computations. Prediction or classification can then be carried based on (E.3). A binary classification model, for instance is readily obtained by defining a Gaussian process on a latent variable Z as above and letting

$$\mathbf{P}(y_i = 1) = \frac{1}{1 + e^{-z_i}}. \tag{E.5}$$

More generally, when there are more than two classes, one can use normalized exponentials instead of sigmoidal functions.

E.2 Kernel Methods and Support Vector Machines

Kernel methods and support vector machines (SVMs) are related to Gaussian processes and can be applied to both classification and regression problems. For simplicity, we consider here a binary classification problem characterized by a set of labeled training example pairs of the form (x_i, y_i) where x_i is an input vector and $y_i = \pm 1$ is the corresponding classification in one of two classes H^+ and H^-. A a (0,1) formalism is equivalent but leads to more cumbersome notation. As an example, consider the problem of deciding whether a given protein (resp. a given gene) belongs to a certain family, given the amino acid sequences (resp. expression levels) of members within (positive examples) and outside (negative examples) the family [275, 95]. In particular, the length of x_i can vary with i. The label y for a new example x is determined by a discriminant function $\mathcal{D}(x; \{x_i, y_i\})$, which depends on the training examples, in the form $y = \text{sign}(\mathcal{D}(x; \{x_i, y_i\}))$. In a proper probabilistic setting,

$$y = \text{sign}(\mathcal{D}(x; \{x_i, y_i\})) = \text{sign}(\log \frac{\mathbf{P}(H^+|x)}{\mathbf{P}(H^-|x)}) \tag{E.6}$$

In kernel methods, the discriminant function is expanded in the form

$$\mathcal{D}(x) = \sum_i y_i \lambda_i K(x_i, x) = \sum_{H^+} \lambda_i K(x_i, x) - \sum_{H^-} \lambda_i K(x_i, x) \tag{E.7}$$

so that, up to trivial constants, $\log \mathbf{P}(H^+|x) = \sum_{H^+} \lambda_i K(x_i, x)$ and similarly for the negative examples. K is called the kernel function. The intuitive idea is to base our classification of the new example on all the previous examples weighted by two factors: a coefficient $\lambda_i \geq 0$ measuring the importance of example i, and the kernel $K(x_i, x)$ measuring how similar x is to example x_i. Therefore the expression for the discrimination depends *directly* on the training examples. This is different from the case of neural networks, for instance, where the decision depends indirectly on the training examples via the trained neural network parameters. Thus in an application of kernel methods two fundamental choices must be made regarding (a) the kernel K; and (b) the weights λ_i. Variations on these choices lead to a spectrum of different methods, including generalized linear models and SVMs.

E.2.1 Kernel Selection

To a first approximation, from the mathematical theory of kernels, a kernel must be positive definite. By Mercer's theorem of functional analysis (given later in the section E.3.2 for completeness), K can be represented as an inner product of the form

$$K_{ij} = K(x_i, x_j) = \phi(x_i)\phi(x_j). \tag{E.8}$$

Thus another way of looking at kernel methods is to consider that the original x vectors are mapped to a "feature" space via the function $\phi(x)$. Note that the feature space can have very high (even infinite) dimension and that the vectors $\phi(x)$ have the same length even when the input vectors x do not. The similiarity of two vectors is assessed by taking their inner product in feature space. In fact we can compute the euclidean distance $||\phi(x_i) - \phi(x_j)||^2 = K_{ii} - 2K_{ij} + K_{jj}$ which also defines a pseudodistance on the original vectors.

The fundamental idea in kernel methods is to define a linear or nonlinear decision surface in feature space rather than the original space. The feature space does not need to be constructed explicitly since all decisions can be made through the kernel and the training examples. In addition, as we are about to see, the decision surface depends *directly* on a *subset* of the training examples, the support vectors.

Notice that a dot product kernel provides a way of comparing vectors in feature space. When used directly in the discrimination function, it corresponds to looking for linear separating hyperplanes in feature space. However more complex decision boundaries in feature spaces (quadratic or higher order) can easily be implemented using more complex kernels K' derived from the inner product kernel K, such as:

- Polynomial kernels: $K'(x_i, x_j) = (1 + K(x_i, x_j))^m$

- Radial basis kernels: $K'(x_i, x_j) = \exp -\frac{1}{2\sigma^2}(\phi(x_i) - \phi(x_j))^t(\phi(x_i) - \phi(x_j))$
- Neural network kernels: $K'(x_i, x_j) = \tanh(\mu x_i^t x_j + \kappa)$

E.2.2 Fisher Kernels

In [275] a general technique is presented for combining kernel methods with probabilistic generative models. The basic idea is that a generative model, such as an HMM, is typically trained from positive examples only and therefore may not be always optimal for discrimination tasks. A discriminative model, however, can be built from a generative model using both positive and negative examples and a kernel of the form $K(x_i, x_j) = U^t(x_i)F^{-1}U(x_j)$, where the vector U is the gradient of the log-likelihood of the generative model with respect to the model parameters $U(x) = \partial \log \mathbf{P}(x|w)/\partial w$. This gradient describes how a given value of w contributes to the generation of example x. For the exponential family of distributions, the gradient forms essentially a sufficient statistics. Notice again that $U(x)$ has fixed length even when x has variable length. For instance, in the case of an HMM trained on a protein family, $U(x)$ is the vector of derivatives that was computed in chapter 7. F is the Fisher information matrix $F = E(U(x)U^t(x))$ with respect to $\mathbf{P}(x|w)$, and this type of kernel is called a *Fisher kernel*. The Fisher matrix consists of the second-order derivatives of the log-likelihood and is therefore associated with the local curvature of the corresponding manifold (see, for instance, [15]). F defines the Riemannian metric of the underlying manifold. In particular, the local distance between two nearby models parameterized by w and $w + \epsilon$ is $\epsilon^t F\epsilon/2$. This distance also approximates the relative entropy between the two models. In many cases, at least asymptotically with many examples, the Fisher kernel can be approximated by the simpler dot product $K(x_i, x_j) = U_{x_i}^t U_{x_j}$. The Fisher kernel can also be modified using the transformations described above, for example in the form $K(x_i, x_j) = \exp -\frac{1}{2\sigma^2}(U(x_i) - U(x_j))^t(U(x_i) - U(x_j))$.

It can be shown that, at least asymptotically, the Fisher kernel classifier is never inferior to the MAP decision rule associated with the generative probabilistic model. An application of Fisher kernel methods to the detection of remote protein homologies is described in [275].

E.2.3 Weight Selection

The weights λ are typically obtained through an iterative optimization procedure on an objective function (classification loss). In general, this corresponds to a quadratic optimization problem. Often the weights can be viewed as Lagrange multipliers, or dual weights with respect to the original parameters of the problem (see section E.2.4 below). With large training sets, at the optimum many of the weights are equal to 0. The only training vectors that matter in a given decision are those with nonzero weights and these are called the support vectors.

To see this, consider an example x_i with target classification y_i. Since our decision is based on the sign of $\mathcal{D}(x_i)$, ideally we would like $y_i\mathcal{D}(x_i)$, the margin for example i, to be as large as possible. Because the margin can be rescaled by rescaling the λs, it is natural to introduce additional constraints such as $0 \leq \lambda_i \leq 1$ for every λ_i. In the case where an exact separating manifold exists in feature space, a reasonable criterion is to maximize the margin in the worst case. This is also called risk minimization and corresponds to $\max_\lambda \min_i y_i\mathcal{D}(x_i)$. SVMs can be defined as a class of kernel methods based on structural risk minimization (see section E.2.4 below). Substituting the expression for $\mathcal{D}$ in terms of the kernel yields $\max_\lambda \min_i \sum_j \lambda_j y_i y_j K_{ij}$. This can be rewritten as $\max_\lambda \min_i \sum_j A_{ij}\lambda_j$, with $A_{ij} = y_i y_j K_{ij}$ and $0 \leq \lambda_i \leq 1$. It is clear that in each minimization procedure all weights λ_j associated with a nonzero coefficient A_{ij} will either be 0 or 1. With a large training set, many of them will be zero for each i and this will remain true at the optimum. When the margins are violated, as in most real-life examples, we can use a similar strategy (an alternative also is to use slack variables as in the example given in section E.2.5 below). For instance, we can try to maximize the average margin, the average being taken with respect to the weights λ_i themselves, which are intended to reflect the relevance of each example. Thus in general we want to maximize a quadratic expression of the form $\sum_i \lambda_i y_i \mathcal{D}(x_i)$ under a set of linear constraints on the λ_i. Standard techniques exist to carry out such optimizations. For example, a typical function used for minimization in the literature is:

$$\mathcal{E}(\lambda_i) = -\sum_i [y_i\lambda_i\mathcal{D}(x_i) + 2\lambda_i]. \tag{E.9}$$

The solution to this constrained optimization problem is unique provided that for any finite set of examples the corresponding kernel matrix K_{ij} is positive definite. The solution can be found with standard iterative methods, although the convergence can sometimes be slow. To accommodate training errors or biases in the training set, the kernel matrix K can be replaced by $K + \mu D$, where D is a diagonal matrix whose entries are either d^+ or d^- in locations corresponding to positive and negative examples [533, 108, 141]. An example of application of SVMs to gene expression data can be found in [95].

In summary, kernel methods and SVMs have several attractive features. As presented, these are supervised learning methods that can leverage labeled data. These methods can build flexible decision surfaces in high-dimensional feature spaces. The flexibility is related to the flexibility in the choice of the kernel function. Overfitting can be controlled through some form of margin maximization. These methods can handle inputs of variable lengths, such as biological sequences, as well as large feature spaces. Feature spaces need not be constructed explicitly since the decision surface is entirely defined in terms of the kernel function and typically a sparse subset of relevant training examples, the support vectors. Learning is typically achieved through iterative solution of a linearly constrained quadratic optimization problem.

E.2.4 Structural Risk Minimization and VC Dimension

There are general bounds in statistical learning theory [533] that can provide guidance in the design of learning systems in general and SVMs in particular. Consider a family of classification functions $f(x;w)$ indexed by a parameter vector w. If the data points (x,y) are drawn from some joint distribution $\mathbf{P}(x,y)$, then we would like to find the function with the smallest error or risk

$$\mathcal{R}(w) = \int \frac{1}{2}|y - f(x;w)|d\mathbf{P}(x,y). \tag{E.10}$$

This risk, however, is in general not known. What is known is the empirical risk measured on the training examples:

$$\mathcal{R}_K(w) = \frac{1}{2K}\sum_1^K |y_i - f(x_i;w)|. \tag{E.11}$$

A fundamental bound of statistical learning theory is that for any $0 \leq \eta \leq 1$, with probability $1-\eta$, we have

$$\mathcal{R}(w) \leq \mathcal{R}_K(w) + \sqrt{\frac{h(\log 2K/h) + 1) - \log(\eta/4)}{K}} \tag{E.12}$$

where h is a non-negative integer called the Vapnik-Chervonenkis (VC) dimension [533].

The VC dimension is a property of a set of functions $f(x;w)$. If a given set of M points can be labeled in all possible 2^M ways using functions in the set, we say that the set of points is shattered. For instance, if $f(x,w)$ is the set of all lines in the planes, then every set of two points can easily be shattered, and most set of three points (except those that are collinear) can also be shattered. No set of four points, however, can be shattered. The VC dimension of the set of functions $f(x;w)$ is the maximum number of points for which at least one instance can be shattered. Thus, for instance, the VC dimension of all the lines in the plane is three and more generally, it can be shown that the VC dimension of hyperplanes in the usual n-dimensional Euclidean space is $n+1$.

The fundamental inequality of (E.12) embodies in some way the bias/variance or fitting/underfitting trade-off. It shows that we can control risk through two buttons: the empirical error (how well we fit the data) and the VC dimension or capacity of the set of functions used in learning. The structural risk minimization aims at optimizing both simultaneously by minimizing the right-hand side of (E.12).

E.2.5 Simple Examples: Linear and Generalized Linear Model

Consider first the family of linear models of the form $\mathcal{D}(x; w) = w_1^t x + w_2$ with $w = (w_1, w_2)$, where w_1 is a vector and w_2 is a scalar, scaled in such a way that $\min_i |\mathcal{D}(x_i; w)| = 1$. If R is the radius of the smallest ball containing the training examples and if $||w_1|| < A$, then it can be shown that the VC dimension h of this family of hyperplanes is bounded: $h < R^2A^2$. This bound can be much tighter than the $n+1$ bound above. Thus we can use A to control the capacity of the hyperplanes.

If a separating hyperplane exists, then the scaling above implies that $y_i\mathcal{D}(x; w) \geq 1$ for every example i. In the more general case where the constraints can be violated, we can introduce slack variables $\xi_i \geq 0$ and require $y_i\mathcal{D}(x; w) \geq 1 - \xi_i$. The support vector approach to minimize the risk bound in (E.12) is to minimize

$$\mathcal{E}(w) = w^t w + \mu \sum_i \xi_i \quad \text{subject to} \quad \xi_i \geq 0 \quad \text{and} \quad y_i\mathcal{D}(x; w) \geq 1 - \xi_i. \tag{E.13}$$

The first term in (E.13) favors small VC dimension and the second term small global error (empirical risk). Introducing Lagrange multipliers λ_i and using the Kuhn-Tucker theorem of optimization theory, one can show that the solution has the form $w = \sum_i y_i\lambda_i x_i$. Intuitively, this is also clear from geometric considerations since the vector w is orthogonal to the hyperplane. This results in the decision function $\mathcal{D}(x; w) = \sum_i y_i\lambda_i x_i^t x + w_2$ associated with a plain dot product kernel. The coefficients λ_i are nonzero only for the support vectors corresponding to the cases where the slack constraints are saturated: $y_i\mathcal{D}(x_i; w) = 1 - \xi_i$. The coefficient λ_i can be found by minimizing the quadratic objective function

$$\mathcal{E}(\lambda) = -\sum_i \lambda_i + \frac{1}{2}\sum_{ij} y_i y_j \lambda_i \lambda_j x_i^t x_j \quad \text{subject to} \quad 0 \leq \lambda_i \leq \mu \quad \text{and} \quad \sum_i \lambda_i y_i = 0. \tag{E.14}$$

In a logistic linear model, $\mathbf{P}(y) = \mathcal{D}(x) = \sigma(yw^t x)$ where w is a vector of parameters and σ is the logistic sigmoidal function $\sigma(u) = 1/(1 + e^{-u})$. A standard prior for w is a Gaussian prior with mean 0 and covariance C. Up to additive constants, the negative log-posterior of the training set is

$$\mathcal{E}(w) = -\sum_i \log \sigma(y_i w^t x_i) + \frac{1}{2} w^t C^{-1} w. \tag{E.15}$$

It is easy to check that at the optimum the solution must satisfy

$$w^* = -\sum_i y_i \lambda_i C x_i \tag{E.16}$$

with $\lambda_i = \partial \log \sigma(z)/\partial z$ taken at $z = y_i {w^*}^t x_i$. Thus we obtain a solution with the general form of (E.7) with the kernel $K(x_i, x_j) = x_i^t C x_j$.

E.3 Theorems for Gaussian Processes and SVMs

For completeness, here we state two useful theorems underlying the theory of kernel methods, SVMs, and Gaussian processes: Bochner's theorem in probability and harmonic analysis and Mercer's theorem in functional analysis.

E.3.1 Bochner's Theorem

Bochner's theorem provides a complete characterization of characteristic functions in terms of Fourier transforms, and as a byproduct establishes the equivalence between characteristic functions and covariance functions of continuous stationary processes.

Consider a complex process, that is, a family of complex random variables $\{X_t = U_t + iV_t\}$, with $-\infty < t < +\infty$. For simplicity, assume that $E(X_t) = 0$ and define the covariance by $\mathbf{Cov}(X_u, X_v) = E(X_u, \bar{X_v})$. We will assume that the process X_t is stationary and continuous, which means that the covariance function is continuous and satisfies

$$\mathbf{Cov}(X_s, X_{s+t}) = f(t). \tag{E.17}$$

Thus it depends only on the distance between variables. Under these assumptions, Bochner's theorem asserts that f satisfies

$$f(t) = \int_{-\infty}^{+\infty} e^{i\lambda t} \mu(d\lambda) \tag{E.18}$$

where μ is a measure on the real line with total mass $f(0)$. That is, f is positive definite and is the Fourier transform of a finite measure. If the variables X_t are real, then the measure μ is symmetric and

$$f(t) = \int_{-\infty}^{+\infty} \cos \lambda t \mu(d\lambda). \tag{E.19}$$

The measure μ is called the spectral measure of the process. Conversely, given any finite measure μ on the real line, it can be shown that there exists a stationary process X_t with spectral measure μ. The measure $\mu/f(0)$ is a probability measure and therefore the function f in (E.18) is a characteristic function. In other words, an equivalent theorem is that a continuous function $g(t)$ is the characteristic function of a probability distribution if and only if it is positive definite (i.e., it satisfies a relation similar to (E.18)) and also satisfies the normalization $g(0) = 1$. Thus up to a normalisation factor, a continuous characteristic function is equivalent to the covariance function of a stationary process. Additional details can be found in [177].

E.3.2 Mercer's Theorem

Mercer's theorem provides the connection between symmetric positive definite kernels and dot products in "feature space". Consider an integral operator $\kappa : L_2 \to L_2$, between two L_2 (square-integrable) spaces, with continuous symmetric kernel K, so that

$$(\kappa f)y = \int K(x,y)f(x)dx. \tag{E.20}$$

Assume that K is also positive definite, i.e.

$$\int f(x)K(x,y)f(y)dxdy > 0 \tag{E.21}$$

if $f \neq 0$. Then there exists an orthonormal set of basis of functions $\xi_i(x)$ such that K can be expanded in the form

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \xi_i(x)\xi_i(y) \tag{E.22}$$

with $\lambda_i \geq 0$, and the scalar product product $(\xi_i \xi_j)_{L_2} = \delta_{ij}$ (orthonormality), for any pair of integers i and j. From (E.20) and the orthonormality condition, we have

$$(\kappa \xi_i)y = \int \sum_{j=1}^{\infty} \lambda_j \xi_j(x)\xi_j(y)\xi_i(x)dx = \lambda_i \xi_i(y). \tag{E.23}$$

In other words, κ is a compact operator with an eigenvector decomposition with eigenvectors ξ_i and nonnegative eigenvalues λ_i. If we define the function $\phi(x)$ by

$$\phi(x) = \sum_{i=1}^{\infty} \sqrt{\lambda_i}\xi_i(x), \tag{E.24}$$

then using the orthonormality conditions again yields

$$K(x, y) = \phi(x)\phi(y), \tag{E.25}$$

which is the decomposition required in (E.8). Conversely, if we start with a continuous embedding $\phi(x)$ of x into a feature space of dimension M, we can then define a continuous kernel $K(x, y)$ using (E.25). The corresponding operator is positive definite since

$$\begin{aligned} \int f(x)K(x,y)f(y)dxdy &= \int f(x)(\phi(x)\phi(y))f(y)dxdy = \\ \sum_{i=1}^{M}\int f(x)\phi_i(x)\phi_i(y)f(y)dxdy &= \sum_{i=1}^{M}(\int f(x)\phi_i(x)dx)^2 \geq 0. \end{aligned} \tag{E.26}$$

This page intentionally left blank

Appendix F

Symbols and Abbreviations

Probabilities

- π: Unscaled degree of confidence or belief
- $\mathbf{P}(P, Q, R \ldots)$: Probability (actual probability distributions)
- **E** ($\mathbf{E}_Q$): Expectation (expectation with respect to Q)
- **Var**: Variance
- **Cov**: Covariance
- X_i, Y_i (x_i, y_i): Propositions or random variables (x_i actual value of X_i)
- $\bar{X}$: Complement or negation of X
- $X \perp Y$ ($X \perp Y | Z$): X and Y are independent (independent conditionally on Z)
- $\mathbf{P}(x_1, \ldots, x_n)$: Probability that $X_1 = x_1, \ldots, X_n = x_n$. When the context is clear, this is also written as $\mathbf{P}(X_1, \ldots, X_n)$. Likewise, for a specific density Q, we write $Q(x_1, \ldots, x_n)$ or $Q(X_1, \ldots, X_n)$
- $\mathbf{P}(X|Y)(\mathbf{E}(X|Y))$: Conditional probability (conditional expectation)
- $\mathcal{N}(\mu, \sigma), \mathcal{N}(\mu, C), \mathcal{N}(\mu, \sigma^2), \mathcal{N}(x; \mu, \sigma^2)$: Normal (or Gaussian) density with mean μ and variance σ^2, or covariance matrix C
- $\Gamma(w|\alpha, \lambda)$: Gamma density with parameters α and λ
- $\mathcal{D}_{\alpha Q}$: Dirichlet distribution with parameters α and Q

- $t(x; \nu, m, \sigma^2)$, $t(\nu, m, \sigma^2)$: Student distribution with ν degrees of freedom, location m, and scale σ
- $\mathcal{I}(x; \nu, \sigma^2)$, $\mathcal{I}(\nu, \sigma^2)$: scaled inverse gamma distribution with ν degrees of freedom and scale σ

Functions

- $\mathcal{E}$: Energy, error, negative log-likelihood or log-posterior (depending on context)
- $\mathcal{E}_T$, $\mathcal{E}_G$, $\mathcal{E}_C$: Training error, generalization error, classification error
- $\mathcal{E}_P$: Parsimony error
- $\mathcal{F}$: Free energy
- $\mathcal{L}$: Lagrangian
- $\mathcal{D}$: Decision function
- $\mathcal{R}$: Risk function
- $\mathcal{R}_K$: Empirical risk function
- $\mathcal{H}(P)$, $\mathcal{H}(X)$: Entropy of the distribution P, or the random variable X/differential entropy in continuous case
- $\mathcal{H}(P, Q)$, $\mathcal{H}(X, Y)$: Relative entropy between the distributions P and Q or between the random variables X and Y
- $\mathcal{I}(P, Q)$, $\mathcal{I}(X, Y)$: Mutual information between the distributions P and Q, or the random variables X and Y
- Z: Partition function or normalizing factor (sometimes also C)
- C: Constant or normalizing factor
- $\delta(x, y)$: Kronecker function equal to 1 if $x = y$ and 0 otherwise
- f, f': Generic function and derivative of f
- $\Gamma(x)$: Gamma function
- $B(\alpha, Q)$: Beta function (appendix D)

- We also use convex (∪) to denote upward convexity (positive second derivative), and convex (∩) to denote downward convexity (negative second derivative), rather than the more confusing "convex" and "concave" expressions

Models, Alphabets, and Sequences

- M ($M = M(w)$): Model (model with parameters w)
- D: Data
- I: Background information
- H: Hidden or latent variables or causes
- $S = \{s_1, s_2, \ldots, s_{|S|}\}$: Set of states of a system
- s: Generic state
- A (X): Alphabet (generic letter)
- $A = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$: DNA alphabet
- $A = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}$: RNA alphabet
- $A = \{\mathsf{A}, \mathsf{C}, \mathsf{D}, \ldots\}$: Amino acid alphabet
- A^*: Set of finite strings over $\mathbf{A}$
- $O = (\mathsf{X}^1 \ldots \mathsf{X}^t \ldots)$: Generic sequence ("O" stands for "observation" or "ordered")
- $\varnothing$: Empty sequence
- $O_1, \ldots, O_K$: Set of training sequences
- O_k^j: jth letter of kth sequence

Graphs and Sets

- $G = (V, E)$: Undirected graph with vertex set V and edge set E
- $G = (V, \vec{E})$: Directed graph with vertex set V and edge set E
- T: Tree
- $N(i)$: Neighbors of vertex i

- $N^+(i)$: Children of vertex i in a directed graph
- $N^-(i)$: Parents of vertex i in a directed graph
- $C^+(i)$: The future, or descendants, of vertex i in a directed graph
- $C^-(i)$: The past, or ancestors, of vertex i in a directed graph
- $N(I)$: Neighbors or boundary of a set I of vertices
- $\mathcal{P}(G)$: Family of probability distributions satisfying the conditional independence assumptions described by G
- G^C: Clique graph of G
- G^M: Moral graph of G
- $\cup$, $\cap$, $\bar{\ }$: Union, intersection, complement of sets
- $\emptyset$: Empty set

Dimensions

- $|A|$: Number of alphabet symbols
- $|S|$: Number of states
- $|H|$: Number of hidden units in HMM/NN hybrid models
- N: Length of sequences (average length)
- K: Number of sequences or examples (e.g., in a training set)
- T: Time horizon (sometimes also temperature when no confusion is possible)

General Parameters

- w : Generic vector of parameters
- t_{ji}: Transition probability from i to j, for instance in a Markov chain
- ^{t} (w_{ij}^t, X^t): Time index, in algorithmic iterations or in sequences
- $^+$, $^-$ (w_{ij}^+): Relative time index, in algorithmic iterations
- * (w_{ij}^*): Optimal solutions

- η: Learning rate

Neural Networks

- w_{ij}: Connection weight from unit j to unit i
- w_i, λ_i: Bias of unit i, gain of unit i
- $D_j = (d_j, t_j)$: Training example; d_j is the input vector and t_j is the corresponding target ouput vector
- $y_i = f_i(x_i)$: Input-output relation for unit i: x_i is the total input into the unit, f_i is the transfer function, and y_i is the output
- $y(d_i)$: Output activity of NN with input vector d_i
- $y_j(d_i)$: Activity of the jth ouput unit of NN with input vector d_i
- $t_j(d_i)$: Target value for the jth ouput unit of NN with input vector d_i

Hidden Markov Models

- m, d, i, h: Main, delete, insert, and anchor states. Most of the time, i is just an index
- *start*, *end:* Start state and end state of an HMM (also denoted S and E in figures)
- E: Set of emitting states of a model
- D: Set of delete (silent) states of a model
- L: In appendix D only, L denotes the set of states in the loop of an HMM loop architecture
- t_{ij} (w_{ij}): Transition probability from state j to state i (normalized exponential representation)
- $e_{i\mathsf{X}}$ ($w_{i\mathsf{X}}$): Emission probability for letter X from state i (normalized exponential representation)
- t_{ij}^D: Silent transition probability from state j to state i
- π: Path variables
- $n(i, \mathsf{X}, \pi, O)$: Number of times the letter X is produced from state i along a path π for a sequence O in a given HMM

- $\alpha_i(t)$: Forward variables
- $\alpha_i^L(t)$: Forward variables in the HMM loop architecture
- $\beta_i(t)$: Backward variables
- $\hat{\alpha}_i(t)$: Scaled forward variables
- $\hat{\beta}_i(t)$: Scaled backward variables
- $\gamma_i(t)$: Probability of being in state i at time t in an HMM for a given observation sequence
- $\gamma_{ji}(t)$: Probability of using the i to j transition at time t in an HMM for a given observation sequence
- $\delta_i(t)$: Variables used in the recursion of the Viterbi algorithms
- κ: Probability of going around an HMM loop silently
- $b(\mathsf{X}, \mathsf{Y}, \mathsf{Z})$: Bendability of triplet XYZ
- $B(i, O)$: Bendability of sequence O at position i
- $B(i)$: Bendability of a family of sequences at position i
- W: Length of averaging window in bendability calculations

Bidirectional Architectures

- W: Total number of parameters
- O_t: Output probability vector
- B_t: Backward context vector
- F_t: Forward context vector
- I_t: Input vector
- $\eta(.)$: Output function
- $\beta(.)$: Backward transition function
- $\phi(.)$: Forward transition function
- n: Typical number of states in the chains
- q: Shift operator

Grammars

- L: Language
- G: Grammar
- $L(G)$: Language generated by grammar G
- R: Production rules of a grammar
- V: Alphabet of variables
- $s = start$: Start variable
- $\alpha \rightarrow \beta$: Grammar production rule: α "produces" or "expands to" β
- $\pi_i(t)$: Derivation variable in grammars
- $n(\beta, u, \pi, O)$: Number of times the rule $u \rightarrow \beta$ is used in the derivation π of a sequence O in a given grammar
- $P_{\alpha\rightarrow\beta}$ ($w_{\alpha\rightarrow\beta}$): Probability of the production rule $\alpha \rightarrow \beta$ in a stochastic grammar (normalized exponential representation)

Phylogenetic Trees

- r: Root node
- X_i: Letter assigned to vertex i
- d_{ji}: Time distance from node i to node j
- $p_{\mathsf{X}_j\mathsf{X}_i}(d_{ji})$: Probability that X_i is substituted by X_j over a time d_{ji}
- $\chi^i(t)$: Random variable associated with letter at position i in a sequence at time t
- $p^i_{\mathsf{YX}}(t)$: Probability that X is substituted by Y over a time t at position i in a sequence
- $P(t) = (p_{\mathsf{YX}}(t))$: Matrix of substitution probabilities for time t
- $Q = (q_{\mathsf{YX}})$: Derivative matrix of P at time 0 ($Q = P'(0)$)
- $p = (p_{\mathsf{X}})$: Stationary distribution
- χ_i: Random variable associated with letter at node i in a tree

- I: Set of internal nodes of a tree
- $O^+(i)$: Evidence contained in subtree rooted at note i

Microarrays

- n (n_c, n_t): Number of expression measurements of a gene (in the control and treatment cases)
- $x_1^c, \ldots, x_{n_c}^c$ ($x_1^t, \ldots, x_{n_t}^t$): Expression measurements of a gene in the control case (and treatment case)
- m (m_c, m_t): Empirical means of measurements of a gene (in the control and treatment cases)
- s^2 (s_c^2, s_t^2): Empirical variances of measurements of a gene (in the control and treatment cases)
- $d_1, \ldots, d_N$: N data points to be clustered
- K: Number of clusters

Kernel Methods and Support Vector Machines

- w: Vector of model parameters
- λ_i: Weights
- ξ_i: Slack variables
- $K_{ij} = K(x_i, x_j)$: Kernel function
- F: Fisher information matrix
- $\phi(x)$: Feature vector
- $U(x)$: Gradient vector of the log-likelihood with respect to model parameters
- h: VC dimension

Abbreviations

- CFG: Context-free grammar
- CSG: Context-sensitive grammar

- BIOHMM: Bidirectional IOHMM
- BRNN: Bidirectional RNN
- EM: Expectation maximization
- HMM: Hidden Markov model
- IOHMM: Input-output HMM
- LMS: Least mean square
- MAP: Maximum a posteriori
- MaxEnt: Maximum entropy
- MCMC: Markov chain Monte Carlo
- ML: Maximum likelihood
- MLP: Multilayer perceptron
- MP: Mean posterior
- NN: Neural network
- RNN: Recursive NN
- RG: Regular grammar
- REG: Recursively enumerable grammar
- SG: Stochastic grammar
- SCFG: Stochastic context-free grammar
- SS: Secondary structure
- SVM: Support vector machine
- VC: Vapnik-Chervonenkis

This page intentionally left blank

References

[1] Y. Abu-Mustafa. Machines that learn from hints. *Sci. American*, 272:64–69, 1995.

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[3] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers. Principles, Techniques, and Tools.* Addison-Wesley, Reading, MA, 1986.

[4] S. M. Aji and R. J. McEliece. The generalized distributive law. Technical Report, Department of Electrical Engineering, California Institute of Technology, 1997.

[5] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Aut. Control*, 19:716–723, 1974.

[6] C. Alff-Steinberger. The genetic code and error transmission. *Proc. Natl. Acad. Sci. USA*, 64:584–591, 1969.

[7] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96:6745–6750, 1999.

[8] S. F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.*, 219:555–565, 1991.

[9] S. F. Altschul, M. S. Boguski, W. Gish, and J. C. Wootton. Issues in searching molecular sequence databases. *Nat. Genet.*, 6:119–129, 1994.

[10] S. F. Altschul, R. Carrol, and D. J. Lipman. Weights for data related by a tree. *J. Mol. Biol.*, 207:647–653, 1989.

[11] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[12] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and L. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.*, 25:3389–3402, 1997.

[13] S.F. Altschul. A protein alignment scoring system sensitive at all evolutionary distances. *J. Mol. Evol.*, 36:290–300, 1993.

[14] S.F. Altschul. Local alignment statistics. *Meth. Enzymol.*, 274:460–480, 1996.

[15] S. Amari. Natural gradient works efficiently in learning. *Neural Comp.*, 10:251–276, 1998.

[16] S. Amari and N. Murata. Statistical theory of learning curves under entropic loss criterion. *Neural Comp.*, 5:140–153, 1993.

[17] C. A. F. Andersen and S. Brunak. Amino acid subalphabets can improve protein structure prediction. *Submitted*, 2001.

[18] M. A. Andrade, G. Casari, C. Sander, and A. Valencia. Classification of protein families and detection of the determinant residues with an improved self-organizing map. *Biol. Cybern.*, 76:441–450, 1997.

[19] S. M. Arfin, A. D. Long, E. T. Ito, L. Tolleri, M. M. Riehle, E. S. Paegle, and G. W. Hatfield. Global gene expression profiling in *escherichia coli* K12: the effects of integration host factor. *J. Biol. Chem.*, 275:29672–29684, 2000.

[20] B. C. Arnold and S. J. Press. Compatible conditional distributions. *J. Amer. Statist. Assn.*, 84:152–156, 1989.

[21] M. Ashburner. On the representation of gene function in genetic databases. *ISMB*, 6, 1998.

[22] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.

[23] A. Bairoch. The PROSITE dictionary of sites and patterns in proteins, its current status. *Nucl. Acids Res.*, 21:3097–3103, 1993.

[24] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence data bank and its supplement TrEMBL. *Nucl. Acids Res.*, 25:31–36, 1997.

[25] J. K. Baker. Trainable grammars for speech recognition. In J. J. Wolf and D. H. Klat, editors, *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550, 1979.

[26] P. Baldi. Gradient descent learning algorithms overview: A general dynamical systems perspective. *IEEE Trans. on Neural Networks*, 6:182–195, 1995.

[27] P. Baldi. Substitution matrices and hidden Markov models. *J. Comput. Biol.*, 2:497–501, 1995.

[28] P. Baldi. On the convergence of a clustering algorithm for protein-coding regions in microbial genomes. *Bioinformatics*, 16:367–371, 2000.

[29] P. Baldi. *The Shattered Self–the End of Natural Evolution*. MIT Press, Cambridge, MA, 2001.

[30] P. Baldi and Pierre-Francois Baisnee. Sequence analysis by additive scales: DNA structure for sequences and repeats of all lengths. *Bioinformatics*, 16:865–889, 2000.

[31] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16:412–424, 2000.

[32] P. Baldi, S. Brunak, Y. Chauvin, J. Engelbrecht, and A. Krogh. Hidden Markov models for human genes. In G. Tesauro J. D. Cowan and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 761–768. Morgan Kaufmann, San Francisco, 1994.

[33] P. Baldi, S. Brunak, Y. Chauvin, J. Engelbrecht, and A. Krogh. Periodic sequence patterns in human exons. In *Proceedings of the 1995 Conference on Intelligent Systems for Molecular Biology (ISMB95)*. AAAI Press, Menlo Park, CA, 1995.

[34] P. Baldi, S. Brunak, Y. Chauvin, and A. Krogh. Naturally occurring nucleosome positioning signals in human exons and introns. *J. Mol. Biol.*, 263:503–510, 1996.

[35] P. Baldi, S. Brunak, Y. Chauvin, and A. Krogh. Hidden Markov models for human genes: periodic patterns in exon sequences. In S. Suhai, editor, *Theoretical and Computational Methods in Genome Research*, pages 15–32, New York, 1997. Plenum Press.

[36] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Bidirectional dynamics for protein secondary structure prediction. In R. Sun and C. L. Giles, editors, *Sequence Learning: Paradigms, Algorithms, and Applications*, pages 99–120. Springer Verlag, New York, 2000.

[37] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15:937–946, 1999.

[38] P. Baldi and Y. Chauvin. Hidden Markov models of the G-protein-coupled receptor family. *J. Comput. Biol.*, 1:311–335, 1994.

[39] P. Baldi and Y. Chauvin. Smooth on-line learning algorithms for hidden Markov models. *Neural Comp.*, 6:305–316, 1994.

[40] P. Baldi and Y. Chauvin. Hybrid modeling, HMM/NN architectures, and protein applications. *Neural Comp.*, 8:1541–1565, 1996.

[41] P. Baldi, Y. Chauvin, T. Hunkapillar, and M. McClure. Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA*, 91:1059–1063, 1994.

[42] P. Baldi, Y. Chauvin, F. Tobin, and A. Williams. Mining data bases of partial protein sequences using hidden Markov models. Net-ID/SmithKline Beecham Technical Report, 1996.

[43] P. Baldi and G. Wesley Hatfield. *Microarrays and Gene Expression*. Cambridge University Press, Cambridge, UK, 2001.

[44] P. Baldi and A. D. Long. A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 17(6):509–519, 2001.

[45] P. Baldi, G. Pollastri, C. A. F. Andersen, and S. Brunak. Matching protein β-sheet partners by feedforward and recurrent neural networks. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 25–36. AAAI Press, Menlo Park, CA, 2000.

[46] P. Baldi, G. Pollastri, C. A. F. Andersen, and S. Brunak. Matching protein beta-sheet partners by feedforward and recurrent neural networks. *ISMB*, 8:25–36, 2000.

[47] J. M. Baldwin. The probable arrangement of the helices in G protein coupled receptors. *EMBO J.*, 12:1693–1703, 1993.

[48] F. G. Ball and J. A. Rice. Stochastic models for ion channels: Introduction and bibliography. *Mathemat. Biosci.*, 112:189–206, 1992.

[49] N. Barkai, H. S. Seung, and H. Sompolinsky. Local and global convergence of online learning. *Phys. Rev. L*, 75:1415–1418, 1995.

[50] N. Barkai and H. Sompolinsky. Statistical mechanics of the maximum-likelihood density-estimation. *Phys. Rev. E*, 50:1766–1769, 1994.

[51] V. Barnett. *Comparative Statistical Inference.* John Wiley, New York, 1982.

[52] G. J. Barton. Protein multiple sequence alignment and flexible pattern matching. *Meth. Enzymol.*, 183:403–427, 1990.

[53] E. B. Baum. Toward a model of mind as a laissez-faire economy of idiots. Preprint, 1997.

[54] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[55] A. A. Beaudry and G. F. Joyce. Directed evolution of an RNA enzyme. *Science*, 257:635–641, 1992.

[56] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text Compression.* Prentice-Hall, Englewood Cliffs, NJ, 1990.

[57] Y. Bengio, Y. Le Cunn, and D. Henderson. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks and hidden Markov models. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 937–944. Morgan Kaufmann, San Francisco, CA, 1994.

[58] Y. Bengio and P. Frasconi. An input-output HMM architecture. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 427–434. Morgan Kaufmann, San Francisco, 1995.

[59] R. Benne. RNA editing. The long and the short of it. *Nature*, 380:391–392, 1996.

[60] S. A. Benner. Patterns of divergence in homologous proteins as indicators of tertiary and quaternary structure. *Adv. Enzyme Regul.*, 28:219–236, 1989.

[61] S. A. Benner. Predicting the conformation of proteins from sequences. Progress and future progress. *J. Mol. Recog.*, 8:9–28, 1995.

[62] D. A. Benson, M. S. Boguski, D. J. Lipman, and J. Ostell. GenBank. *Nucl. Acids Res.*, 25:1–6, 1997.

[63] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.

[64] A. L. Berman, E. Kolker, and E. N. Trifonov. Underlying order in protein sequence organization. *Proc. Natl. Acad. Sci. USA.*, 91:4044–4047, 1994.

[65] G. Bernardi. The human genome: Organization and evolutionary history. *Ann. Rev. Genetics*, 29:445–476, 1995.

[66] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 1995.

[67] D. Bertsimas and J. Tsitsiklis. Simulated annealing. *Statis. Sci.*, 8:10–15, 1993.

[68] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. R. Statis. Soc. B*, 36:192–225, 1974.

[69] J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems. *Statis. Sci.*, 10:3–66, 1995.

[70] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[71] R. E. Blahut. *Principles and Practice of Information Theory*. Addison-Wesley, Reading, MA, 1987.

[72] M. Blatt, S. Wiseman, and E. Domany. Super-paramagnetic clustering of data. *Phys. Review Lett.*, 76:3251–3254, 1996.

[73] G. Blobel. Intracellular membrane topogenesis. *Proc. Natl. Acad. Sci. USA*, 77:1496, 1980.

[74] N. Blom, S. Gammeltoft, and S. Brunak. Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. *J. Mol. Biol.*, 294:1351–1362, 1999.

[75] N. Blom, J. Hansen, D. Blaas, and S. Brunak. Cleavage site analysis in picornaviral polyproteins by neural networks. *Protein Sci.*, 5:2203–2216, 1996.

[76] M. Bloom and O. G. Mouritsen. The evolution of membranes. In R. Lipowsky and E. Sackmann, editors, *Handbook of Biological Physics vol. 1*, pages 65–95, Amsterdam, 1995. Elsevier Science.

[77] G. Bohm. New approaches in molecular structure prediction. *Biophys. Chem.*, 59:1–32, 1996.

[78] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, B. Lautrup, L. Nørskov, O. H. Olsen, and S. B. Petersen. Protein secondary structures and homology by neural networks: The α-helices in rhodopsin. *FEBS Letters*, 241:223–228, 1988.

[79] P. Bork, C. Ouzounis, and C. Sander. From genome sequences to protein function. *Curr. Opin. Struct. Biol.*, 4:393–403, 1994.

[80] P. Bork, C. Ouzounis, C. Sander, M. Scharf, R. Schneider, and E. Sonnhammer. Comprehensive sequence analysis of the 182 predicted open reading frames of yeast chromosome iii. *Protein Sci.*, 1:1677–1690, 1992.

[81] M. Borodovsky and J. McIninch. Genmark: Parallel gene recognition for both DNA strands. *Computers Chem.*, 17:123–133, 1993.

[82] M. Borodovsky, J. D. McIninch, E. V. Koonin, K. E. Rudd, C. Medigue, and A. Danchin. Detection of new genes in a bacterial genome using Markov models for three gene classes. *Nucl. Acids Res.*, 23:3554–3562, 1995.

[83] M. Borodovsky, K. E. Rudd, and E. V. Koonin. Intrinsic and extrinsic approaches for detecting genes in a bacterial genome. *Nucl. Acids Res.*, 22:4756–4767, 1994.

[84] H. Bourlard and N. Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic, Boston, 1994.

[85] J. M. Bower and D. Beeman. *The Book of Genesis: Exploring Realistic Neural Models with the GEneral NEural SImulations System*. Telos/Springer-Verlag, New York, 1995.

[86] G. E. P. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison-Wesley, Reading, MA, 1973.

[87] A. Brack and L. E. Orgel. Beta structures of alternating polypeptides and their possible prebiotic significance. *Nature*, 256:383–387, 1975.

[88] D. Bray. Protein molecules as computational elements in living cells. *Nature*, 376:307–312, 1995.

[89] A. Brazma, I. J. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res.*, 8:1202–1215, 1998.

[90] L. Breiman. Discussion of neural networks and related methods for classification. *J. R. Statis. Soc. B*, 56:409–456, 1994.

[91] V. Brendel and H. G. Busse. Genome structure described by formal languages. *Nucl. Acids Res.*, 12:2561–2568, 1984.

[92] S. Brenner, G. Elgar, R. Sandford, A. Macrae, B. Venkatesh, and S. Aparicio. Characterization of the pufferfish (Fugu) genome as a compact model vertebrate genome. *Nature*, 366:265–268, 1993.

[93] S. E. Brenner, C. Chothia, and T. J. P. Hubbard. Population statistics of protein structures: lessons from structural classification. *Curr. Opin. Struct. Biol.*, 7:369–376, 1997.

[94] L. D. Brown. *Fundamentals of Statistical Exponential Families*. Institute of Mathematical Statistics, Hayward, CA, 1986.

[95] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. Walsh Sugnet, T. S. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97:262–267, 2000.

[96] I. Brukner, R. Sánchez, D. Suck, and S. Pongor. Sequence-dependent bending propensity of DNA as revealed by DNase I: parameters for trinucleotides. *EMBO J.*, 14:1812–1818, 1995.

[97] S. Brunak. Non-linearities in training sets identified by inspecting the order in which neural networks learn. In O. Benhar, C. Bosio, P. Del Giudice, and E. Tabet, editors, *Neural Networks: From Biology to High Energy Physics*, pages 277–288, Pisa, 1991. ETS Editrice.

[98] S. Brunak. Doing sequence analysis by inspecting the order in which neural networks learn. In D. M. Soumpasis and T. M. Jovin, editors, *Computation of Biomolecular Structures — Achievements, Problems and Perspectives*, pages 43–54, Berlin, 1993. Springer–Verlag.

[99] S. Brunak and J. Engelbrecht. Correlation between protein secondary structure and the mRNA nucleotide sequence. *Proteins*, 25:237–252, 1996.

[100] S. Brunak, J. Engelbrecht, and S. Knudsen. Cleaning up gene databases. *Nature*, 343:123, 1990.

[101] S. Brunak, J. Engelbrecht, and S. Knudsen. Neural network detects errors in the assignment of pre-mRNA splice site. *Nucl. Acids Res.*, 18:4797–4801, 1990.

[102] S. Brunak, J. Engelbrecht, and S. Knudsen. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *J. Mol. Biol.*, 220:49–65, 1991.

[103] S. Brunak and B. Lautrup. *Neural Networks—Computers with Intuition*. World Scientific Pub., Singapore, 1990.

[104] J. Buhmann and H. Kuhnel. Vector quantization with complexity costs. *IEEE Trans. Information Theory*, 39:1133–1145, 1993.

[105] C. J. Bult, O. White, G. J. Olsen, L. Zhou, R. D. Fleischmann, G. G. Sutton, J. A. Blake, L. M. FitzGerald, R. A. Clayton, J. D. Gocayne, A. R. Kerlavage, B. A. Dougherty, J. F. Tomb, M. D. Adams, C. I. Reich, R. Overbeek, E. F. Kirkness, K. G. Weinstock, J. M. Merrick, A. Glodek, J. L. Scott, N. S. M. Geoghagen, and J. C. Venter. Complete genome sequence of the methanogenic archaeon, *Methanococcus jannaschii*. *Science*, 273:1058–1073, 1996.

[106] W. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Trans. Knowledge Data Eng.*, 8:195–210, 1996.

[107] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic dna. *J. Mol. Biol.*, 268:78–94, 1997.

[108] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[109] J. M. Burke and A. Berzal-Herranz. In vitro selection and evolution of RNA: Applications for catalytic RNA, molecular recognition, and drug discovery. *Faseb J.*, 7:106–112, 1993.

[110] M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34:353–367, 1996.

[111] H. J. Bussemaker, H. Li, and E. D. Siggia. Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis. *Proc. Natl. Acad. Sci. USA*, 97:10096–10100, 2000.

[112] C. R. Calladine and H. R. Drew. *Understanding DNA—The Molecule and How it Works.* Academic Press, London, 1992.

[113] L. R. Cardon and G. D. Stormo. Expectation-maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *J. Mol. Biol.*, 223:159–170, 1992.

[114] C. R. Carlson and A. B. Kolsto. A small (2.4 mb) bacillus cereus chromosome corresponds to a conserved region of a larger (5.3 mb) bacillus cereus chromosome. *Mol. Microbiol.*, 13:161–169, 1994.

[115] R. Caruana. Learning many related tasks at the same time with backpropagation. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 7*, pages 657–664, San Mateo, CA, 1995. Morgan Kaufmann.

[116] T. Cavalier-Smith. Introduction: The evolutionary significance of genome size. In T. Cavalier-Smith, editor, *The Evolution of Genome Size*, pages 1–36. John Wiley & Sons, Chichester, UK, 1985.

[117] T. Cavalier-Smith. The origin of cells: A symbiosis between genes, catalysts, and membranes. *Cold Spring Harbor Symp. Quant. Biol.*, 52:805–824, 1987.

[118] J. M. Chandonia and M. Karplus. New methods for accurate prediction of protein secondary structure. *Proteins*, 35:293–306, 1999.

[119] E. Chargaff. Structure and function of nucleic acids as cell constituents. *Fed. Proc.*, 10:654–659, 1951.

[120] E. Chargaff. How genetics got a chemical education. *Ann. N. Y. Acad. Sci.*, 325:345–360, 1979.

[121] E. Charniak. Bayesian networks without tears. *AI Mag.*, 12:50–63, 1991.

[122] P. Cheeseman. An inquiry into computer understanding. *Comput. Intell.*, 4:57–142, 1988. With discussion.

[123] R. O. Chen, R. Felciano, and R. B. Altman. RIBOWEB: linking structural computations to a knowledge base of published experimental data. *ISMB*, 5:84–87, 1997.

[124] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 93–103. AAAI Press, Menlo Park, CA, 2000.

[125] G. I. Chipens, Y. U. I. Balodis, and L. E. Gnilomedova. Polarity and hydropathic properties of natural amino acids. *Ukrain. Biokhim. Zh.*, 63:20–29, 1991.

[126] Sung-Bae Cho and Jin H. Kim. An HMM/MLP architecture for sequence recognition. *Neural Comp.*, 7:358–369, 1995.

[127] C. Chotia. One thousand families for the molecular biologist. *Nature*, 357:543–544, 1992.

[128] P. Y. Chou and G. D. Fasman. Empirical predictions of protein conformations. *Ann. Rev. Biochem.*, 47:251–276, 1978.

[129] P.Y. Chou and G.D. Fasman. Prediction of the secondary structure of proteins from their amino acid sequence. *Adv. Enzymol. Relat. Areas Mol. Biol.*, 47:45–148, 1978.

[130] G. A. Churchill. Stochastic models for heterogeneous DNA sequences. *Bull. Mathem. Biol.*, 51:79–94, 1989.

[131] M. G. Claros, S. Brunak, and G. von Heijne. Prediction of n-terminal protein sorting signals. *Curr. Opin. Struct. Biol.*, 7:394–398, 1997.

[132] J-M. Claverie. What if there are only 30,000 human genes. *Science*, 291:1255–1257, 2001.

[133] N. Colloc'h and F. E. Cohen. Beta-breakers: An aperiodic secondary structure. *J. Mol. Biol.*, 221:603–613, 1991.

[134] Int. Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[135] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Art. Intell.*, 42:393–405, 1990.

[136] J. L. Cornette, K. B. Cease, H. Margalit, J. L. Spouge, J. A. Berzofsky, and C. DeLisi. Hydrophobicity scales and computational techniques for detecting amphiphatic structures in proteins. *J. Mol. Biol.*, 195:659–685, 1987.

[137] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1991.

[138] R. T. Cox. Probability, frequency and reasonable expectation. *Am. J. Phys.*, 14:1–13, 1964.

[139] I. P. Crawford, T. Niermann, and K. Kirschner. Prediction of secondary structure by evolutionary comparison: application to the alpha subunit of tryptophan synthase. *Proteins*, 2:118–129, 1987.

[140] F. H. C. Crick. The origin of the genetic code. *J. Mol. Biol.*, 38:367–379, 1968.

[141] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[142] L. Croft, S. Schandorff, F. Clark, K. Burrage, P. Arctander, and J. S. Mattick. ISIS, the intron information system, reveals the high frequency of alternative splicing in the human genome. *Nature Genet.*, 24:340–341, 2000.

[143] S. Dalal, S. Balasubramanian, and L. Regan. Protein alchemy: Changing beta-sheet into alpha-helix. *Nat. Struct. Biol.*, 4:548–552, 1997.

[144] S. Das, L. Yu, C. Gaitatzes, R. Rogers, J. Freeman, J. Bienkowska, R. M. Adams, T. F. Smith, and J. Lindelien. Biology's new Rosetta Stone. *Nature*, 385:29–30, 1997.

[145] A. P. Dawid. Applications of a general propagation algorithm for probabilistic expert systems. *Stat. Comp.*, 2:25–36, 1992.

[146] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. *Neural Comp.*, 7:889–904, 1995.

[147] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statis. Soc.*, B39:1–22, 1977.

[148] J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.

[149] D. Devos and A. Valencia. Practical limits of function prediction. *Proteins*, 41:98–107, 2000.

[150] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Appl. Prob.*, 1:36–61, 1991.

[151] K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan. Principles of protein folding—a perspective from simple exact models. *Protein Sci.*, 4:561–602, 1995.

[152] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195:216–222, 1987.

[153] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[154] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, Cambridge, 1998.

[155] S. R. Eddy. Hidden Markov models. *Curr. Opin. Struct. Biol.*, 6:361–365, 1996.

[156] S. R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucl. Acids Res.*, 22:2079–2088, 1994.

[157] S. R. Eddy, G. Mitchinson, and R. Durbin. Maximum discrimination hidden Markov models of sequence consensus. *J. Comp. Biol.*, 2:9–23, 1995.

[158] H. Ehrig, M. Korff, and M. Lowe. Tutorial introduction to the algebraic approach of graph grammars based on double and single pushouts. In H. Ehrig, H. J. Kreowski, and G. Rozenberg, editors, *Lecture Notes in Computer Science*, volume 532, pages 24–37. Springer-Verlag, 1991.

[159] M. Eigen. The hypercycle. A principle of natural self-organization. Part A: Emergence of the hypercycle. *Naturwissenschaften*, 64:541–565, 1977.

[160] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci USA*, 95:14863–14868, 1998.

[161] D. Eisenberg. Into the black night. *Nat. Struct. Biol.*, 4:95–97, 1997.

[162] D. Eisenberg, E. M. Marcotte, and I. Xenarios T. O. Yeates. Protein function in the post-genomic era. *Nature*, 405:823–826, 2000.

[163] G. Elgar, R. Sandford, S. Aparicio, A. Macrae, B. Venkatesh, and S. Brenner. Small is beautiful: Comparative genomics with the pufferfish (*Fugu rubripes*). *Trends Genet.*, 12:145–150, 1996.

[164] J. Engelbrecht, S. Knudsen, and S. Brunak. G/C rich tract in the 5' end of human introns. *J. Mol. Biol.*, 227:108–113, 1992.

[165] J. Engelfriet and G. Rozenberg. Graph grammars based on node rewriting: An introduction to NLC graph grammars. In H. Ehrig, H. J. Kreowski, and G. Rozenberg, editors, *Lecture Notes in Computer Science*, volume 532, pages 12–23. Springer-Verlag, 1991.

[166] D. M. Engelman, T. A. Steitz, and A. Goldman. Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Ann. Rev. Biophys. Biophys. Chem.*, 15:321–353, 1986.

[167] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. Protein interaction maps for complete genomes based on gene fusion events. *Nature*, 402:86–90, 1999.

[168] C. J. Epstein. Role of the amino-acid "code" and of selection for conformation in the evolution of proteins. *Nature*, 210:25–28, 1966.

[169] D. T. Ross et al. Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Genet.*, 24:227–235, 2000.

[170] J. C. Venter et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001.

[171] U. Scherf et al. A gene expression database for the molecular pharmacology of cancer. *Nat. Genet.*, 24:236–244, 2000.

[172] B. S. Everitt. *An Introduction to Latent Variable Models*. Chapman and Hall, London, 1984.

[173] B. S. Everitt and D. J. Hand. *Finite Mixture Distributions*. Chapman and Hall, London and New York, 1981.

[174] P. Fariselli and R. Casadio. Prediction of the number of residue contacts in proteins. *ISMB*, 8:146–151, 19.

[175] B. A. Fedorov. Long-range order in globular proteins. *FEBS Lett.*, 62:139–141, 1976.

[176] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley & Sons, New York, 3rd edition, 1968.

[177] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 2. John Wiley & Sons, New York, 2nd edition, 1971.

[178] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.

[179] E. A. Ferran, B. Pflugfelder, and P. Ferrara. Self-organized neural maps of human protein sequences. *Protein Sci.*, 3:507–521, 1994.

[180] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains with an application to an exclusion process. *Ann. Appl. Prob.*, 1:62–87, 1991.

[181] W. M. Fitch. Towards defining the course of evolution: Minimum change for a specific tree topology. *Syst. Zool.*, 20:406–416, 1971.

[182] W. M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155:279–284, 1967.

[183] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J. F. Tomb, B. A. Dougherty, and J. M. Merrick. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*, 269:496–512, 1995.

[184] M. L. Forcada and R. C. Carrasco. Learning the initial state of a second-order recurrent neural network during regular-language inference. *Neural Comp.*, 7:923–930, 1995.

[185] D. R. Forsdyke. Relative roles of primary sequence and (G+C)% in determining the hierarchy of frequencies of complementary trinucleotide pairs in dnas of different species. *J. Mol. Evol.*, 41:573–581, 1995.

[186] G. E. Fox and C. R. Woese. The architecture of 5S rRNA and its relation to function. *J. Mol. Evol.*, 6:61–76, 1975.

[187] V. Di Francesco, J. Garnier, and P. J. Munson. Protein topology recognition from secondary structure sequences—Applications of the hidden Markov models to the alpha class proteins. *J. Mol. Biol.*, 267:446–463, 1997.

[188] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Trans. Neural Networks*, 9:768–786, 1998.

[189] C. M. Fraser, J. D. Gocayne, O. White, M. D. Adams, R. A. Clayton, R. D. Fleischmann, C. J. Bult, A. R. Kerlavage, G. Sutton, and J. M. Kelley. The minimal gene complement of *Mycoplasma genitalium*. *Science*, 270:397–403, 1995.

[190] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J. Comp. Biol.*, 7:601–620, 2000.

[191] A. Frigessi, P. Di Stefano, C. R. Hwang, and S. J. Sheu. Convergence rate of the Gibbs sampler, the Metropolis algorithm and other single-site updating dynamics. *J. R. Stat. Soc.*, 55:205–219, 1993.

[192] D. Frishman and P. Argos. Knowledge-based secondary structure assignment. *Proteins*, 23:566–579, 1995.

[193] Y. Fujiwara, M. Asogawa, and A. Konagaya. Stochastic motif extraction using hidden Markov models. In *Proceedings of Second International Conference on Intelligent Systems for Molecular Biology*, pages 138–146, Menlo Park, CA, 1994. AAAI/MIT Press.

[194] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16:906–914, 2000.

[195] G. Gamow. Possible relation between deoxyribonucleic acid and protein structures. *Nature*, 173:318, 1954.

[196] J. Garnier, D. J. Osguthorpe, and B. Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J. Mol. Biol.*, 120:97–120, 1978.

[197] R. A. Garrett. Genomes: *Methanococcus jannaschii* and the golden fleece. *Curr. Biol.*, 6:1376–1377, 1996.

[198] A. Gelman and T. P. Speed. Characterizing a joint probability distribution by conditionals. *J. R. Statis. Soc. B*, 55:185–188, 1993.

[199] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741, 1984.

[200] D. Gerhold and C. T. Caskey. It's the genes! EST access to human genome content. *Bioessays*, 18:973–981, 1996.

[201] M. Gerstein, E. Sonnhammer, and C. Chotia. Volume changes in protein evolution. *J. Mol. Biol.*, 236:1067–1078, 1994.

[202] C. J. Geyer. Practical Markov chain Monte Carlo. *Statis. Sci.*, 7:473–511, 1992.

[203] Z. Ghahramani. Learning dynamic Bayesian networks. *Adap. Proc. Seq. Data Struct.*, 1387:168–197, 1998.

[204] Z. Ghahramani. Learning dynamic Bayesian networks. In M. Gori and C. L. Giles, editors, *Adaptive Processing of Temporal Information. Lecture Notes in Artifical Intelligence.* Springer Verlag, Heidelberg, 1998.

[205] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 1997.

[206] M. Gibbs and D. J. C. MacKay. Efficient implementation of Gaussian processes. Technical report, Cavendish Laboratory, Cambridge, UK, 1997.

[207] L. M. Gierasch. Signal sequences. *Biochemistry*, 28:923–930, 1989.

[208] C. L. Giles, C. B. Miller, D. Chen, H. H. Chen, G. Z. Sun, and Y. C. Lee. Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Comp.*, 4:393–405, 1992.

[209] W. R. Gilks, D. G. Clayton, D. J. Spiegelhalter, N. G. Best, A. J. McNeil, L. D. Sharples, and A. J. Kirby. Modelling complexity: Applications of Gibbs sampling in medicine. *J. R. Statis. Soc.*, 55:39–52, 1993.

[210] W. R. Gilks, A. Thomas, and D. J. Spiegelhalter. A language and program for complex Bayesian modelling. *The Statistician*, 43:69–78, 1994.

[211] P. Gill, P. L. Ivanov, C. Kimpton, R. Piercy, N. Benson, G. Tully, I. Evett, E. Hagelberg, and K. Sullivan. Identification of the remains of the Romanov family by DNA analysis. *Nat. Genet.*, 6:130–135, 1994.

[212] P. Gill, C. Kimpton, R. Aliston-Greiner, K. Sullivan, M. Stoneking, T. Melton, J. Nott, S. Barritt, R. Roby, and M. Holland. Establishing the identity of Anna Anderson Manahan. *Nat. Genet.*, 9:9–10, 1995.

[213] V. Giudicelli and M.-P. Lefranc. Ontology for Immunogenetics: IMGT-ONTOLOGY. *Bioinformatics*, 12:1047–1054, 1999.

[214] U. Gobel, C. Sander, R. Schneider, and A. Valencia. Correlated mutations and residue contacts in proteins. *Proteins*, 18:309–317, 1994.

[215] A. Goffeau. Life with 6000 genes. *Science*, 274:546, 1996.

[216] A. L. Goldberg and R. E. Wittes. Genetic code: Aspects of organization. *Science*, 153:420–424, 1966.

[217] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[218] D. S. Goodsell and R. E. Dickerson. Bending and curvature calculations in B-DNA. *Nucl. Acids Res.*, 22:5497–5503, 1994.

[219] J. Gorodkin, L. J. Heyer, S. Brunak, and G. D. Stormo. Displaying the information contents of structural RNA alignments: the structure logos. *CABIOS*, 13:583–586, 1997.

[220] J. Gorodkin, L. J. Heyer, and G. D. Stormo. Finding common sequence and structure motifs in a set of RNA sequences. In T. Gaasterland, P. Karp, K. Karplus, C. Ouzounis, C. Sander, and A. Valencia, editors, *Proceedings of the Fifth International Conference on Intellelligent Systems for Molecular Biology*, pages 120–123, Menlo Park, California, 1997. AAAI/MIT Press.

[221] J. Gorodkin, L. J. Heyer, and G. D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucl. Acids Res.*, 25:3724–3732, 1997.

[222] M. Gouy, P. Marliere, C. Papanicolaou, and J. Ninio. Prediction of secondary structures of nucleic acids: Algorithmic and physical aspects. *Biochimie*, 67:523–531, 1985.

[223] C. W. J. Granger. Combining forecasts—twenty years later. *J. Forecasting*, 8:167–173, 1989.

[224] P. Green, D. Lipman, L. Hillier, R. Waterson, D. States, and J. M. Claverie. Ancient conserved regions in new gene sequences and the protein databases. *Science*, 259:1711–1716, 1993.

[225] P. C. Gregory and T. J. Loredo. A new method for the detection of a periodic signal of unknown shape and period. *Astrophys. J.*, 398:146–168, 1992.

[226] M. Gribskov, A. D. McLachlan, and D. Eisenberg. Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, 84:4355–4358, 1987.

[227] T. Gudermann, T. Schoneberg, and G. Schultz. Functional and structural complexity of signal transduction via g-protein-coupled receptors. *Annu. Rev. Neurosci.*, 20:399–427, 1997.

[228] S. F. Gull. Bayesian inductive inference and maximum entropy. In G. J. Erickson and C. R. Smith, editors, *Maximum entropy and Bayesian methods in science and engineering*, pages 53–74. Kluwer, Dordrecht, 1988.

[229] S.F. Gull. Developments in maximum entropy data analysis. In J. Skilling, editor, *Maximum entropy and Bayesian methods*, pages 53–71. Kluwer, Dordrecht, 1989.

[230] B. Hajeck. Cooling schedules for optimal annealing. *Math. of Operation Res.*, 13:311–329, 1988.

[231] S. Hampson, P. Baldi, D. Kibler, and S. Sandmeyer. Analysis of yeast's ORFs upstream regions by parallel processing, microarrays, and computational methods. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 190–201. AAAI Press, Menlo Park, CA, 2000.

[232] S. Hampson, D. Kibler, and P. Baldi. Distribution patterns of locally over-represented k-mers in non-coding yeast DNA. 2001. Submitted.

[233] S. Handley. Classifying nucleic acid sub-sequences as introns or exons using genetic programming. In C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, editors, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 162–169. AAAI Press, Menlo Park, CA, 1995.

[234] J. Hanke, D. Brett, I. Zastrow, A. Aydin, S. Delbruck, G. Lehmann, F. Luft, J. Reich, and P. Bork. Alternative splicing of human genes: more the rule than the exception? *Trends Genet.*, 15:389–390, 1999.

[235] J. E. Hansen, O. Lund, J. Engelbrecht, H. Bohr, J. O. Nielsen, J. E.-S. Hansen, and S. Brunak. Prediction of O-glycosylation of mammalian proteins: Specificity patterns of UDP-GalNAc:polypeptide n-acetylgalactosaminyltransferase. *J. Biochem. Biol.*, 307:801–813, 1995.

[236] J. E. Hansen, O. Lund, N. Tolstrup, A. A. Gooley, K. L. Williams, and S. Brunak. NetOglyc: prediction of mucin type O-glycosylation sites based on sequence context and surface accessibility. *Glycocon. J.*, 15:115–130, 1998.

[237] L. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. and Machine Intell.*, 12:993–1001, 1990.

[238] J. C. Harsanyi. *Rational behaviour and bargaining equilibrium in games and social situations.* Cambridge University Press, Cambridge, UK, 1977.

[239] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402, Supp.:C47–C52, 1999.

[240] M. Hasegawa and T. Miyata. On the antisymmetry of the amino acid code table. *Orig. Life*, 10:265–270, 1980.

[241] M. A. El Hassan and C. R. Calladine. Propeller-twisting of base-pairs and the conformational mobility of dinucleotide steps in DNA. *J. Mol. Biol.*, 259:95–103, 1996.

[242] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[243] J. Hasty, J. Pradines, M. Dolnik, and J. J. Collins. Noise-based switches and amplifiers for gene expression. *Proc. Natl. Acad. Sci. USA*, 97:2075–2080, 2000.

[244] S. Hayward and J. F. Collins. Limits on α-helix prediction with neural network models. *Proteins*, 14:372–381, 1992.

[245] S. M. Hebsgaard, P. G. Korning, N. Tolstrup, J. Engelbrecht, P. Rouzé, and S. Brunak. Splice site prediction in *Arabidopsis thaliana* pre-mRNA by combining local and global sequence information. *Nucl. Acids Res.*, 24:3439–3452, 1996.

[246] D. Heckerman. Bayesian networks for data mining. *Data Mining and Knowl. Discov.*, 1:79–119, 1997.

[247] J. Hein. Unified approach to alignment and phylogenies. *Meth. Enzymol.*, 183:626–645, 1990.

[248] R. Henderson, J. M. Baldwin, T. A. Ceska, F. Zemlin, E. Beckmann, and K. H. Downing. Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *J. Mol. Biol.*, 213:899–929, 1990.

[249] S. Henikoff and J. Henikoff. Position-based sequence weights. *J. Mol. Biol.*, 243:574–578, 1994.

[250] S. Henikoff and J. G. Henikoff. Protein family classification based on searching a database of blocks. *Genomics*, 19:97–107, 1994.

[251] B. Hermann and S. Hummel, editors. *Ancient DNA*. Springer-Verlag, New York, 1994.

[252] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the theory of neural computation.* Addison-Wesley, Redwood City, CA, 1991.

[253] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of co-expressed genes. *Genome Res.*, 9:1106–1115, 1999.

[254] R. Hinegardner. Evolution of cellular DNA content in teleost fishes. *Am. Nat.*, 102:517–523, 1968.

[255] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

[256] H. Le Hir, M. J. Moore, and L. E. Maquat. Pre-mrna splicing alters mRNP composition: evidence for stable association of proteins at exon-exon junctions. *Genes Dev.*, 14:1098–1108, 2000.

[257] R. Hirata, Y. Ohsumk, A. Nakano, H. Kawasaki, K. Suzuki, and Y. Anraku. Molecular structure of a gene, vma1, encoding the catalytic subunit of H(+)- translocating adenosine triphosphatase from vacuolar membranes of *Saccharomyces cerevisiae*. *J. Biol. Chem.*, 265:6726–6733, 1990.

[258] W. S. Hlavacek and M. S. Savageau. Completely uncoupled and perfectly coupled gene expression in repressible systems. *J. Mol. Biol.*, 266:538–558, 1997.

[259] U. Hobohm, M. Scharf, R. Schneider, and C. Sander. Selection of representative protein data sets. *Protein Sci.*, 1:409–417, 1992.

[260] I. L. Hofacker, W. Fontana, P. F. Stadler, S. Bonherffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, 125:167–188, 1994.

[261] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* MIT Press, Cambridge, MA, 1992.

[262] L. H. Holley and M. Karplus. Protein secondary structure prediction with a neural network. *Proc. Nat. Acad. Sci. USA*, 86:152–156, 1989.

[263] F. C. P. Holstege, E. G. Jennings, J. J. Wyrick, T. I. Lee, C. J. Hengartner, M. R. Green, T. R. Golub, E. S. Lander, and R. A. Young. Dissecting the regulatory circuitry of a eukaryotic genome. *Cell*, 95:717–728, 1998.

[264] K. Hornik, M. Stinchcombe, and H. White. Universal approximation of an unknown function and its derivatives using multilayer feedforward networks. *Neural Networks*, 3:551–560, 1990.

[265] K. Hornik, M. Stinchcombe, H. White, and P. Auer. Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. *Neural Comp.*, 6:1262–1275, 1994.

[266] Z. Huang, S. B. Prusiner, and F. E. Cohen. Scrapie prions: A three-dimensional model of an infectious fragment. *Folding & Design*, 1:13–19, 1996.

[267] Z. Huang, S. B. Prusiner, and F. E. Cohen. Structures of prion proteins and conformational models for prion diseases. *Curr. Top. Microbiol. Immunol.*, 207:49–67, 1996.

[268] T. J. Hubbard and J. Park. Fold recognition and ab initio structure predictions using hidden Markov models and beta-strand pair potentials. *Proteins*, 25:398–402, 1995.

[269] J. P. Huelsenbeck and B. Rannala. Phylogenetic methods come of age: Testing hypotheses in an evoutionary context. *Science*, 276:227–232, 1997.

[270] J. D. Hughes, P. W. Estep, S. Tavazole, and G. M. Church. Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *saccharomyces cerevisiae*. *J. Mol. Biol.*, 296:1205–1214, 2000.

[271] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, K. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[272] V. Isham. An introduction to spatial point processes and Markov random fields. *Internat. Statist. Rev.*, 49:21–43, 1981.

[273] O. C. Ivanov and B. Förtsch. Universal regularities in protein primary structure: preference in bonding and periodicity. *Orig. Life Evol. Biosph.*, 17:35–49, 1986.

[274] P. L. Ivanov, M. J. Wadhams, R. K. Roby, M. M. Holland, V. W. Weedn, and T. J. Parsons. Mitochondrial DNA sequence heteroplasmy in the grand duke of Russia Georgij Romanov establishes the authenticity of the remains of Tsar Nicholas II. *Nat. Genet.*, 12:417–420, 1996.

[275] T. S. Jaakkola, M. Diekhans, and D. Haussler. Using the Fisher kernel method to detect remore protein homologies. In T. Lengauer, R. Schneider, P. Bork, D. Brutlag, J. Glasgow, H. W. Mewes, and R. Zimmer, editors, *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB99)*, pages 149–155. AAAI Press, Menlo Park, CA, 1999.

[276] T. S. Jaakkola and I. Jordan. Recursive algorithms for approximating probabilities in graphical models. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 487–493. MIT Press, Cambridge, MA, 1997.

[277] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Comp.*, 3:79–87, 1991.

[278] B. D. James, G. J. Olsen, and N. R. Pace. Phylogenetic comparative analysis of RNA secondary structure. *Meth. Enzymol.*, 180:227–239, 1989.

[279] P. G. Jansen. *Exploring the exon universe using neural networks.* PhD thesis, The Technical University of Denmark, 1993.

[280] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, 1957.

[281] E. T. Jaynes. Information theory and statistical mechanics. II. *Phys. Rev.*, 108:171–190, 1957.

[282] E. T. Jaynes. Prior probabilities. *IEEE Trans. Systems Sci. Cybernet.*, 4:227–241, 1968.

[283] E. T. Jaynes. Bayesian methods: General background. In J. H. Justice, editor, *Maximum entropy and Bayesian methods in statistics*, pages 1–25. Cambridge University Press, Cambridge, 1986.

[284] E. T. Jaynes. Probability theory: The logic of science. Unpublished., 1994.

[285] W. H. Jeffreys and J. O. Berger. Ockham's razor and Bayesian analysis. *Am. Sci.*, 80:64–72, 1992.

[286] F. V. Jensen. *An Introduction to Bayesian Networks.* Springer Verlag, New York, 1996.

[287] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Comput. Statist. Quart.*, 4:269–282, 1990.

[288] L. J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, C. Workman, C. A. Andersen, K. Rapacki, H.H. Stærfelt, A. Krogh, S. Knudsen, A. Valencia, and S. Brunak. Using posttranslational modifications to predict orphan protein function for the human genome. *Submitted*, 2001.

[289] H. Jeong, B. Tomber, R. Albert, Z.N. Oltvai, and A.-L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000. in press.

[290] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, 292:195–202, 1999.

[291] D. T. Jones, C. M. Moody, J. Uppenbrink, J. H. Viles, P. M. Doyle, C. J. Harris, L. H. Pearl, P. J. Sadler, and J. M. Thornton. Towards meeting the Paracelsus challenge: The design, synthesis, and characterization of paracelsin-43, an alpha-helical protein with over 50% sequence identity to an all-beta protein. *Proteins*, 24:502–513, 1996.

[292] M. I. Jordan. *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.

[293] M. I. Jordan, Z. Ghahramani, and L. K. Saul. Hidden Markov decision trees. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 501–507. MIT Press, Cambridge, MA, 1997.

[294] T. H. Jukes. Possibilities for the evolution of the genetic code from a preceding form. *Nature*, 246:22–26, 1973.

[295] T. H. Jukes and C. R. Cantor. Evolution of protein molecules. In *Mammalian Protein Metabolism*, pages 21–132. Academic Press, New York, 1969.

[296] B. Jungnickel, T.A. Rapoport, and E. Hartmann. Protein translocation: Common themes from bacteria to man. *FEBS Lett.*, 346:73–77, 1994.

[297] W. Kabsch and C. Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen–bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.

[298] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *J. Art. Intell. Res.*, 4:237–285, 1996.

[299] P. Kahn. From genome to proteome: Looking at a cell's proteins. *Science*, 270:369–370, 1995.

[300] D. Kaiser and R. Losick. How and why bacteria talk to each other. *Cell*, 73:873–885, 1993.

[301] P. M. Kane, C. T. Yamashiro, D. F. Wolczyk, N. Neff, M. Goebl, and T. H. Stevens. Protein splicing converts the yeast tfp1 gene product to the 69-kd subunit of the vacuolar h(+)-adenosine triphosphatase. *Science*, 250:651–657, 1990.

[302] N. Kaplan and C. H. Langley. A new estimate of sequence divergence of mitochondrial DNA using restriction endonuclease mappings. *J. Mol. Evol.*, 13:295–304, 1979.

[303] J. D. Karkas, R. Rudner, and E. Chargaff. Separation of *B. subtilis* DNA into complementary strands, II. Template functions and composition as determined by transcription with RNA polymerase. *Proc. Natl. Acad. Sci. USA*, 60:915–920, 1968.

[304] S. Karlin, B. E. Blaisdell, and P. Bucher. Quantile distributions of amino acid usage in protein classes. *Prot. Eng.*, 5:729–738, 1992.

[305] S. Karlin and J. Mrazek. What drives codon choices in human genes. *J. Mol. Biol.*, 262:459–472, 1996.

[306] S. Karlin, F. Ost, and B. E. Blaisdell. Patterns in DNA and amino acid sequences and their statistical significance. In M.S. Waterman, editor, *Mathematical methods for DNA sequences*, pages 133–157, Boca Raton, Fla., 1989. CRC Press.

[307] P. Karp, M. Riley, S. Paley, A. Pellegrini-Toole, and M. Krummenacker. EcoCyc: Electronic encyclopedia of *e. coli* genes and metabolism. *Nucl. Acids Res.*, 27:55–59, 1999.

[308] P. D. Karp. An ontology for biological function based on molecular interactions. *Bioinformatics*, 16:269–285, 2000.

[309] P. D. Karp, M. Krummenacker, S. Paley, and J. Wagg. Integrated pathway-genome databases and their role in drug discovery. *Trends Biotech.*, 17:275–281, 1999.

[310] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.

[311] S. A. Kauffman. The large scale structure and dynamics of gene control circuits: an ensemble approach. *J. Theor. Biol.*, 44:167–190, 1974.

[312] S. A. Kauffman. Requirements for evolvability in complex systems: orderly dynamics and frozen components. *Physica D*, 42:135–152, 1990.

[313] T. Kawabata and J. Doi. Improvement of protein secondary structure prediction using binary word encoding. *Proteins*, 27:36–46, 1997.

[314] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

[315] W. J. Kent and A. M. Zahler. The Intronerator: exploring introns and alternative splicing in caenorhabditis elegans. *Nucl. Acids Res.*, 28:91–93, 2000.

[316] D. H. Kenyon and G. Steinman. *Biochemical Predestinations*. McGraw-Hill, New York, 1969.

[317] H. G. Khorana. Bacteriorhodopsin, a membrane protein that uses light to translocate protons. *J. Biol. Chem.*, 263:7439–7442, 1988.

[318] H. G. Khorana, G. E. Gerber, W. C. Herlihy, C. P. Gray, R. J. Anderegg, K. Nihei, and K. Biemann. Amino acid sequence of bacteriorhodopsin. *Proc. Natl. Acad. Sci.*, 76:5046–5050, 1979.

[319] J. L. King and T. H. Jukes. Non-Darwinian evolution. *Science*, 164:788–798, 1969.

[320] R. D. King and M. J. Sternberg. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Prot. Sci.*, 5:2298–2310, 1996.

[321] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[322] T. M. Klingler and D. L. Brutlag. Discovering side-chain correlation in alpha-helices. In R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, editors, *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 236–243. AAAI Press, Menlo Park, CA, 1994.

[323] D. G. Kneller, F. E. Cohen, and R. Langridge. Improvements in protein secondary structure prediction by an enhanced neural network. *J. Mol. Biol.*, 214:171–182, 1990.

[324] P. Koehl and M. Levitt. A brighter future for protein structure prediction. *Nat. Struct. Biol.*, 6:108–111, 1999.

[325] L. F. Kolakowski. GCRDb: A G-protein-coupled receptor database. *Receptors Channels*, 2:1–7, 1994.

[326] A. K. Konopka. Sequences and codes: Fundamentals of biomolecular cryptology. In D. W. Smith, editor, *Biocomputing—Informatics and Genome Projects*, pages 119–174, San Diego, 1994. Academic Press.

[327] P. G. Korning, S. M. Hebsgaard, P. Rouze, and S. Brunak. Cleaning the GenBank *Arabidopsis thaliana* data set. *Nucl. Acids Res.*, 24:316–320, 1996.

[328] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge MA, 1992.

[329] J. R. Koza. Evolution of a computer program for classifying protein segments as transmembrane domains using genetic programming. In R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, editors, *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 244–252. AAAI Press, Menlo Park, CA, 1994.

[330] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Press, Cambridge MA, 1994.

[331] A. Kreegipuu, N. Blom, S. Brunak, and J. Jarv. Statistical analysis of protein kinase specificity determinants. *FEBS Lett.*, 430:45–50, 1998.

[332] J. K. Kristensen. Analysis of cis alternatively spliced mammalian genes. Master Thesis, University of Copenhagen, 2000.

[333] A. Krogh. Two methods for improving performance of an HMM and their application for gene finding. In T. Gaasterland et al., editor, *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 179–186. AAAI Press, Menlo Park, CA, 1997.

[334] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, 235:1501–1531, 1994.

[335] A. Krogh, B. Larsson, G. von Heijne, and E. L. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *J. Mol. Biol.*, 305:567–580, 2001.

[336] A. Krogh, I. S. Mian, and D. Haussler. A hidden Markov model that finds genes in *E. coli* DNA. *Nucl. Acids Res.*, 22:4768–4778, 1994.

[337] A. Krogh and G. Mitchinson. Maximum entropy weighting of aligned sequences of proteins of DNA. In C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, editors, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 215–221. AAAI Press, Menlo Park, CA, 1995.

[338] A. Krogh and S. K. Riis. Prediction of beta sheets in proteins. In M. C. Mozer S. Touretzky and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 917–923. MIT Press, Boston, MA, 1996.

[339] A. Krogh and P. Sollich. Statistical mechanics of ensemble learning. *Phys. Rev. E*, 55:811–825, 1997.

[340] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation and active learning. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. MIT Press, Cambridge, MA, 1995.

[341] S. Kullback. *Information Theory and Statistics.* Dover Publications, New York, 1959.

[342] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22:79, 1986.

[343] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized hidden Markov model for the recognition of human genes in dna. *ISMB*, 4:134–142, 1996.

[344] A. Lapedes, C. Barnes, C. Burks, R. Farber, and K. Sirotkin. Application of neural networks and other machine learning algorithms to dna sequence analysis. In G. I. Bell and T. G. Marr, editors, *Computers in DNA. The Proceedings of the Interface Between Computation Science and Nucleic Acid Sequencing Workshop.*, volume VII, pages 157–182. Addison Wesley, Redwood City, CA, 1988.

[345] K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Lang.*, 4:35–36, 1990.

[346] N. Larsen, G. J. Olsen, B. L. Maidak, M. J. McCaughey, R. Overbeek, T. J. Macke, T. L. Marsh, and C. R. Woese. The ribosomal database project. *Nucl. Acids Res.*, 21:3021–3023, 1993.

[347] E. E. Lattman and G. D. Rose. Protein folding–what's the question? *Proc. Natl. Acad. Sci. USA*, 90:439–441, 1993.

[348] S. L. Lauritzen. *Graphical Models.* Oxford University Press, Oxford, 1996.

[349] S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H. G. Leimer. Independence properties of directed Markov fields. *Networks*, 20:491–505, 1990.

[350] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. R. Statis. Soc. B*, 50:157–224, 1988.

[351] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[352] C. E. Lawrence and A. A. Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7:41–51, 1990.

[353] J. R. Lawton, F. A. Martinez, and C. Burks. Overview of the LiMB database. *Nucl. Acids Res.*, 17:5885–5899, 1989.

[354] C. Lee, R. G. Klopp, R. Weindruch, and T. A. Prolla. Gene expression profile of aging and its retardation by caloric restriction. *Science*, 285:1390–1393, 1999.

[355] M. T. Lee, F. C. Kuo, G. A. Whitmore, and J. Sklar. Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. *Proc. Natl. Acad. Sci. USA*, 97:9834–9839, 2000.

[356] N. Lehman and G. F. Joyce. Evolution in vitro of an RNA enzyme with altered metal dependence. *Nature*, 361:182–185, 1993.

[357] E. Levin and R. Pieraccini. Planar hidden Markov modeling: From speech to optical character recognition. In S. J. Hanson, J. D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 731–738. Morgan Kaufmann, San Mateo, CA, 1993.

[358] J. Levin, S. Pascarella, P. Argos, and J. Garnier. Quantification of secondary structure prediction improvement using multiple alignments. *Prot. Eng.*, 6:849–854, 1993.

[359] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Syst. Tech. J.*, 62:1035–1074, 1983.

[360] S. Lewis, M. Ashburner, and M. G. Reese. Annotating eukaryote genomes. *Curr. Opin. Struct. Biol.*, 10:349–354, 2000.

[361] F. Liang, I. Holt, G. Pertea, S. Karamycheva, S.L. Salzberg, and J. Quackenbush. Gene index analysis of the human genome estimates approximately 120, 000 genes. *Nat. Genetics*, 25:239–240, 200.

[362] V. I. Lim. Algorithms for prediction of α-helical and β-structural regions in globular proteins. *J. Mol. Biol.*, 88:873–894, 1974.

[363] S. Lin and A. D. Riggs. The general affinity of lac repressor for *E. coli* DNA: Implications for gene regulation in procaryotes and eucaryotes. *Cell*, 4:107–111, 1975.

[364] T. Lin, B. G. Horne, P. Tiño, and C. L. Giles. Learning long-term dependencies in NARX recurrent neural networks. *IEEE Trans. Neural Networks*, 7:1329–1338, 1996.

[365] H. Lodish, D. Baltimore, A. Berk, S. L. Zipursky, and P. Matsudairas. *Molecular cell biology*. Scientific American Books, New York, 3rd edition, 1995.

[366] A. D. Long, H. J. Mangalam, B. Y. Chan, L. Tolleri, G. W. Hatfield, and P. Baldi. Global gene expression profiling in *escherichia coli* K12: Improved statistical inference from DNA microarray data using analysis of variance and a Bayesian statistical framework. *J. Biol. Chem.*, 276:19937–19944, 2001.

[367] A. V. Lukashin and M. Borodovsky. GeneMark.hmm: new solutions for gene finding. *Nucl. Acids Res.*, 26:1107–1115, 1998.

[368] O. Lund, K. Frimand, J. Gorodkin, H. Bohr, J. Bohr, J. Hansen, and S. Brunak. Protein distance constraints predicted by neural networks and probability density functions. *Prot. Eng.*, 25:1241–1248, 1997.

[369] D. H. Ly, D. J. Lockhart, R. A. Lerner, and P. G. Schultz. Mitotic misregulation and human aging. *Science*, 287:2486–2492, 2000.

[370] M. J. MacGregor, T. P. Flores, and M. J. E. Sternberg. Prediction of beta-turns in proteins using neural networks. *Prot. Eng.*, 2:521–526, 1989.

[371] A. L. Mackay. Optimization of the genetic code. *Nature*, 216:159–160, 1967.

[372] D. J. C. MacKay. Bayesian interpolation. *Neural Comp.*, 4:415–447, 1992.

[373] D. J. C. MacKay. A practical Bayesian framework for back-propagation networks. *Neural Comp.*, 4:448–472, 1992.

[374] D. J. C. MacKay. Density networks and their application to protein modelling. In J. Skilling and S. Sibisi, editors, *Maximum Entropy and Bayesian Methods*, pages 259–268, Dordrecht, 1996. Kluwer.

[375] D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Comp.*, 11:1035–1068, 1999.

[376] D. J. C. MacKay and L. C. Bauman Peto. A hierarchical Dirichlet language model. *Nat. Lang. Eng.*, 1:1–19, 1995.

[377] R. Maclin and J. Shavlik. Using knowledge-based neural networks to improve algorithms: Refining the Chou–Fasman algorithm for protein folding. *Machine Learning*, 11:195–215, 1993.

[378] E. M. Marcotte. Computational genetics: finding protein function by nonhomology methods. *Curr. Opin. Struct. Biol.*, 10:359–365, 2000.

[379] E. M. Marcotte, M. Pellegrini, H. L. Ng, D. L. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285:751–753, 1999.

[380] E. M. Marcotte, M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402:83–86, 1999.

[381] E. Marinari and G. Parisi. Simulated tempering: A new Monte Carlo scheme. *Europhys. Lett.*, 19:451–458, 1992.

[382] B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, 405:442–451, 1975.

[383] H. H. McAdams and A. Arkin. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Genet.*, 15:65–69, 1999.

[384] P. McCullagh and J. A. Nelder. *Generalized linear models.* Chapman and Hall, London, 1989.

[385] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE J. Sel. Areas Commun.*, 16:140–152, 1998.

[386] L. J. McGuffin, K. Bryson, and J. T. Jones. The PSIPRED protein structure prediction server. *Bioinformatics*, 16:404–405, 2000.

[387] X. L. Meng and D. B. Rubin. Recent extensions to the EM algorithm. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian statistics*, volume 4, pages 307–320. Oxford University Press, Oxford, 1992.

[388] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[389] F. Miescher. Uber die chemische Zusammensetzung der Eiterzellen. In F. Hoppe-Seyler, editor, *Medicinisch-chemische Untersuchungen*, pages 441–460, Berlin, 1871. August Hirschwald.

[390] G. L. G. Miklos and G. M. Rubin. The role of the genome project in determining gene function: Insights from model organisms. *Cell*, 86:521–529, 1996.

[391] E. Mjolsness, D. H. Sharp, and J. Reinitz. A connectionist model of development. *J. Theor. Biol.*, 152:429–453, 1991.

[392] J. M. Modestino and J. Zhang. A Markov random field model-based approach to image interpretation. *IEEE Trans. Pattern Anal. Machine Intell.*, 14:606–615, 1992.

[393] R. N. Moll, M. A. Arbib, and A. J. Koufry. *An Introduction to Formal Language Theory*. Springer-Verlag, New York, 1988.

[394] J. C. Mullikin, S. E. Hunt, C. G. Cole, B. J. Mortimore, C. M. Rice, J. Burton, L. H. Matthews, R. Pavitt, R. W. Plumb, S. K. Sims, R. M. Ainscough, J. Attwood, J. M. Bailey, K. Barlow, R. M. Bruskiewich, P. N. Butcher, N. P. Carter, Y. Chen, C. M. Clee, P. C. Coggill, J. Davies, R. M. Davies, E. Dawson, M.D. Francis, A. A. Joy, R. G. Lamble, C. F. Langford, J. Macarthy, V. Mall, A. Moreland, E. K. Overton-Larty, M. T. Ross, L. C. Smith, C. A. Steward, J. E. Sulston, E. J. Tinsley, K. J. Turney, D. L. Willey, G. D. Wilson, A. A. McMurray, I. Dunham, J. Rogers, and D. R. Bentley. An SNP map of human chromosome 22. *Nature*, 407:516–520, 2000.

[395] R. M. Neal. Connectionist learning of belief networks. *Art. Intell.*, 56:71–113, 1992.

[396] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report. Department of Computer Science, University of Toronto, 1993.

[397] R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, Department of Computer Science, University of Toronto, 1995.

[398] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, New York, 1996.

[399] R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report no. 9702. Department of Statistics, University of Toronto, 1997.

[400] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. Technical Report, Department of Computer Science, University of Toronto, Canada, 1993.

[401] S. B. Needleman and C. D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[402] E. J. Neer. G proteins: Critical control points for transmembrane signals. *Prot. Sci.*, 3:3–14, 1994.

[403] M. A. Newton, C. M. Kendziorski, C. S. Richmond, F. R. Blattner, and K. W. Tsui. On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *J. Comp. Biol.*, 8:37–52, 2001.

[404] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Prot. Eng.*, 10:1–6, 1997.

[405] H. Nielsen, J. Engelbrecht, G. von Heijne, and S. Brunak. Defining a similarity threshold for a functional protein sequence pattern: The signal peptide cleavage site. *Proteins*, 24:316–320, 1996.

[406] H. Nielsen and A. Krogh. Prediction of signal peptides and signal anchors by a hidden Markov model. *ISMB*, 6:122–130, 1998.

[407] M. W. Nirenberg, O. W. Jones, P. Leder, B. F. C. Clark, W. S. Sly, and S. Pestka. On the coding of genetic information. *Cold Spring Harbor Symp. Quant. Biol.*, 28:549–557, 1963.

[408] R. Nowak. Entering the postgenome era. *Science*, 270:368–371, 1995.

[409] L. E. Orgel. A possible step in the origin of the genetic code. *Isr. J. Chem.*, 10:287–292, 1972.

[410] R. L. Ornstein, R. Rein, D. L. Breen, and R. D. MacElroy. An optimised potential function for the calculation of nucleic acid interaction energies. I. Base stacking. *Biopolymers*, 17:2341–2360, 1978.

[411] Y. A. Ovchinnikov, N. G. Abdulaev, M. Y. Feigina, A. V. Kiselev, and N. A. Lobanov. The structural basis of the functioning of bacteriorhodopsin: An overview. *FEBS Lett.*, 100:219–234, 1979.

[412] M. Pagel and R. A. Johnstone. Variation across species in the size of the nuclear genome supports the junk-DNA explanation for the c-value paradox. *Proc. R. Soc. Lond. (Biol.)*, 249:119–124, 1992.

[413] A. Pandey and M. Mann. Proteomics to study genes and genomes. *Nature*, 405:837–846, 2000.

[414] L. Pardo, J. A. Ballesteros, R. Osman, and H. Weinstein. On the use of the transmembrane domain of bacteriorhodopsin as a template for modeling the three-dimensional structure of guanine nucleotide-binding regulatory protein-coupled receptors. *Proc. Natl. Acad. Sci. USA*, 89:4009–4012, 1992.

[415] R. Parsons and M. E. Johnson. DNA sequence assembly and genetic programming—new results and puzzling insights. In C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, editors, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 277–284. AAAI Press, Menlo Park, CA, 1995.

[416] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, CA, 1988.

[417] W. R. Pearson. Rapid and sensitive sequence comparison with FASTP and FASTA. *Meth. Enzymol.*, 183:63–98, 1990.

[418] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proc. Nat. Acad. Sci. USA*, 85:2444–2448, 1988.

[419] W.R. Pearson. Effective protein sequence comparison. *Meth. Enzymol.*, 266:227–258, 1996.

[420] E. Pebay-Peyroula, G. Rummel, J. P. Rosenbusch, and E. M. Landau. X-ray structure of bacteriorhodopsin at 2.5 angstroms from microcrystals grown in lipidic cubic phases. *Science*, 277:1676–1681, 1997.

[421] A. G. Pedersen, P. F. Baldi, Y. Chauvin, and S. Brunak. DNA structure in human polymerase II promoters. *J. Mol. Biol.*, 281:663–673, 1998.

[422] A. G. Pedersen and H. Nielsen. Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 226–233, Menlo Park, CA., 1997. AAAI Press.

[423] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T.O Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 38:667–677, 1999.

[424] M. D. Perlwitz, C. Burks, and M. S. Waterman. Pattern analysis of the genetic code. *Advan. Appl. Math.*, 9:7–21, 1988.

[425] C. M. Perou, S. S. Jeffrey, M. van de Rijn, C. A. Rees, M. B. Eisen, D. T. Ross, A. Pergamenschikov, C. F. Williams, S. X. Zhu, J. C. Lee, D. Lashkari, D. Shalon, P. O. Brown, and D. Botstein. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl. Acad. Sci. USA*, 96:9212–9217, 1999.

[426] M. P. Perrone and L. N. Cooper. When networks disagree: ensemble method for neural networks. In R. J. Mammone, editor, *Neural networks for speech and image processing*, chapter 10. Chapman and Hall, London, 1994.

[427] T. N. Petersen, C. Lundegaard, M. Nielsen, H. Bohr, J. Bohr, S. Brunak, G. P. Gippert, and O. Lund. Prediction of protein secondary structure at 80% accuracy. *Proteins*, 41:17–20, 2000.

[428] P. A. Pevzner. *Computational Molecular Biology—An Algorithmic Approach.* MIT Press, Cambridge, MA, 2000.

[429] G. Pollastri, P. Baldi, P. Fariselli, and R. Casadio. Improved prediction of the number of residue contacts in proteins by recurrent neural networks. *Bioinformatics*, 2001. Proceedings of the ISMB 2001 Conference.

[430] V. V. Prabhu. Symmetry observations in long nucleotide sequences. *Nucl. Acids Res.*, 21:2797–2800, 1993.

[431] J. W. Pratt, H. Raiffa, and R. Schlaifer. *Introduction to Statistical Decision Theory.* MIT Press, Cambridge, MA, 1995.

[432] S. R. Presnell and F. E. Cohen. Artificial neural networks for pattern recognition in biochemical sequences. *Ann. Rev. Biophys. Biomol. Struct.*, 22:283–298, 1993.

[433] S. J. Press. *Bayesian Statistics: Principles, Models, and Applications.* John Wiley, New York, 1989.

[434] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes in C.* Cambridge University Press, Cambridge, 1988.

[435] L. G. Presta and G. D. Rose. Helix signals in proteins. *Science*, 240:1632–1641, 1988.

[436] W. C. Probst, L. A. Snyder, D. I. Schuster, J. Brosius, and S. C. Sealfon. Sequence alignment of the G-protein coupled receptor superfamily. *DNA and Cell Biol.*, 11:1–20, 1992.

[437] N. Qian and T. J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, 202:865–884, 1988.

[438] M. B. Qumsiyeh. Evolution of number and morphology of mammalian chromosomes. *J. Hered.*, 85:455–465, 1994.

[439] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[440] T. A. Rapoport. Transport of proteins across the endoplasmic reticulum membrane. *Science*, 258:931–936, 1992.

[441] M. G. Reese, D. Kulp, H. Tammana, and D. Haussler. Genie–gene finding in *drosophila melanogaster*. *Genome Res.*, 10:529–538, 2000.

[442] F. M. Richards and C. E. Kundrot. Identification of structural motifs from protein coordinate data: Secondary structure and first-level supersecondary structure. *Proteins*, 3:71–84, 1988.

[443] D. S. Riddle, J. V. Santiago, S. T. Bray-Hall, N. Doshi, V.P. Grantcharova, Q. Yi, and D. Baker. Functional rapidly folding proteins from simplified alphabets. *Nat. Struct. Biol.*, 4:805–809, 1997.

[444] R. Riek, S. Hornemann, G. Wider, M. Billeter, R. Glockshuber, and K. Wuthrich. NMR structure of the mouse prion protein domain PrP(121-321). *Nature*, 382:180–182, 1996.

[445] S. K. Riis and A. Krogh. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *J. Comput. Biol.*, 3:163–183, 1996.

[446] J. J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[447] É. Rivals, M. Dauchet, J. P. Delahaye, and O. Delgrange. Compression and genetic sequence analysis. *Biochimie*, 78:315–322, 1996.

[448] D. Ron, Y. Singer, and N. Tishby. The power of amnesia. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 176–183. Morgan Kaufmann, San Francisco, CA, 1994.

[449] G. D. Rose. Protein folding and the Paracelsus challenge. *Nat. Struct. Biol.*, 4:512–514, 1997.

[450] G. D. Rose and T. P. Creamer. Protein folding: predicting predicting. *Proteins*, 19:1–3, 1994.

[451] B. Rost and C. Sander. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Nat. Acad. Sci. USA*, 90:7558–7562, 1993.

[452] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.

[453] B. Rost and C. Sander. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins*, 19:55–72, 1994.

[454] B. Rost, C. Sander, and R. Schneider. Redefining the goals of protein secondary structure prediction. *J. Mol. Biol.*, 235:13–26, 1994.

[455] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin. Backpropagation: The basic theory. In *Backpropagation: Theory, Architectures and Applications*, pages 1–34. Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.

[456] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, editors, *Parallel distributed processing: Explorations in the microstructure of cognition.*, volume 1: Foundations, pages 318–362, Cambridge, MA., 1986. MIT Press.

[457] R. Russell and G. Barton. The limits of protein secondary structure prediction accuracy from multiple sequence alignment. *J. Mol. Biol.*, 234:951–957, 1993.

[458] J. R. Saffran, R. N. Aslin, and E. L. Newport. Statistical learning by 8-month-old infants. *Science*, 274:1926–1928, 1996.

[459] Y. Sakakibara. Efficient learning of context-free grammars from positive structural examples. *Info. Comput.*, 97:23–60, 1992.

[460] Y. Sakakibara, M. Brown, R. Hughey, I. S. Mian, K. Sjölander, R. C. Underwood, and D. Haussler. Stochastic context-free grammars for tRNA modeling. *Nucl. Acids Res.*, 22:5112–5120, 1994.

[461] S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucl. Acids Res.*, 26:544–548, 1998.

[462] C. Sander and R. Schneider. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9:56–68, 1991.

[463] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, C. A. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith. Nucleotide sequence of bacteriophage phi X174 DNA. *Nature*, 265:687–695, 1977.

[464] D. Sankoff and P. Rousseau. Locating the vertices of a Steiner tree in an arbitrary metric space. *Math. Prog.*, 9:240–246, 1975.

[465] L. K. Saul and M. I. Jordan. Exploiting tractable substructures in intractable networks. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 486–492. MIT Press, Cambridge, MA, 1996.

[466] M. A. Savageau. Power-law formalism: a canonical nonlinear approach to modeling and analysis. In V. Lakshmikantham, editor, *World Congress of Nonlinear Analysts 92*, volume 4, pages 3323–3334. Walter de Gruyter Publishers, Berlin, 1996.

[467] R. D. Schachter. Probabilistic inference and influence diagrams. *Operation Res.*, 36:589–604, 1988.

[468] R. D. Schachter, S. K. Anderson, and P. Szolovits. Global conditioning for probabilistic inference in belief networks. In *Proceedings of the Uncertainty in AI Conference*, pages 514–522, San Francisco, CA, 1994. Morgan Kaufmann.

[469] D. Schneider, C. Tuerk, and L. Gold. Selection of high affinity RNA ligands to the bacteriophage r17 coat protein. *J. Mol. Biol.*, 228:862–869, 1992.

[470] F. Schneider. Die funktion des arginins in den enzymen. *Naturwissenschaften*, 65:376–381, 1978.

[471] R. Schneider, A. de Daruvar, and C. Sander. The HSSP database of protein structure-sequence alignments. *Nucleic Acids Res.*, 25:226–230, 1997.

[472] T. D. Schneider. Reading of DNA sequence logos: Prediction of major groove binding by information theory. *Meth. Enzymol.*, 274:445–455, 1996.

[473] T. D. Schneider and R. M. Stephens. Sequence logos: A new way to display consensus sequences. *Nucl. Acids Res.*, 18:6097–6100, 1990.

[474] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. Information content of binding sites on nucleotide sequences. *J. Mol. Biol.*, 188:415–431, 1986.

[475] B. Scholkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings First International Conference on Knowledge Discovery and Data Mining.* AAAI Press, Menlo Park, CA, 1995.

[476] H. P. Schwefel and R. Manner, editors. *Parallel Problem Solving from Nature*, Berlin, 1991. Springer-Verlag.

[477] R. R. Schweitzer. Anastasia and Anna Anderson. *Nat. Genet.*, 9:345, 1995.

[478] W. Schwemmler. *Reconstruction of Cell Evolution: A Periodic System of Cells.* CRC Press, Boca Raton, FL, 1994.

[479] D. B. Searls. Linguistics approaches to biological sequences. *CABIOS*, 13:333–344, 1997.

[480] T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Syst.*, 1:145–168, 1987.

[481] P. H. Sellers. On the theory and computation of evolutionary distances. *SIAM J. Appl. Math.*, 26:787–793, 1974.

[482] H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Phys. Rev. A*, 45:6056–6091, 1992.

[483] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 623–656, 1948.

[484] R. Sharan and R. Shamir. CLICK: a clustering algorithm with applications to gene expression analysis. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 307–316. AAAI Press, Menlo Park, CA, 2000.

[485] I. N. Shindyalov, N. A. Kolchanov, and C. Sander. Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Prot. Eng.*, 7:349–358, 1994.

[486] J. E. Shore and R. W. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Trans. Info. Theory*, 26:26–37, 1980.

[487] P. Sibbald and P. Argos. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *J. Mol. Biol.*, 216:813–818, 1990.

[488] R. R. Sinden. *DNA Structure and Function.* Academic Press, San Diego, 1994.

[489] K. Sjölander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *CABIOS*, 12:327–345, 1996.

[490] A. F. Smith and G. O. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *J. R. Statis. Soc.*, 55:3–23, 1993.

[491] A. F. M. Smith. Bayesian computational methods. *Phil. Trans. R. Soc. London A*, 337:369–386, 1991.

[492] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

[493] P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Comp.*, 9:227–267, 1997.

[494] E. E. Snyder and G. D. Stormo. Identification of protein coding regions in genomic DNA. *J. Mol. Biol.*, 248:1–18, 1995.

[495] V. V. Solovyev, A. A. Salamov, and C. B. Lawrence. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucl. Acids Res.*, 22:5156–5153, 1994.

[496] V. V. Solovyev, A. A. Salamov, and C. B. Lawrence. Prediction of human gene structure using linear discriminant functions and dynamic programming. In C. Rawling, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, editors, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 367–375, Cambridge, 1995. AAAI Press.

[497] E. L. L. Sonnhammer, S. R. Eddy, and R. Durbin. Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins*, 28:405–420, 1997.

[498] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

[499] D. J. Spiegelhalter, A. P. Dawid, S. L. Lauritzen, and R. G. Cowell. Bayesian analysis in expert systems. *Stat. Sci.*, 8:219–283, 1993.

[500] F. Spitzer. Markov random fields and Gibbs ensembles. *Am. Math. Monthly*, 78:142–154, 1971.

[501] S. Stamm, M. Q. Zhang, T. G. Marr, and D. M. Helfman. A sequence compilation and comparison of exons that are alternatively spliced in neurons. *Nucl. Acids Res.*, 22:1515–1526, 1994.

[502] S. Steinberg, A. Misch, and M. Sprinzl. Compilation of tRNA sequences and sequences of tRNA genes. *Nucl. Acids Res.*, 21:3011–3015, 1993.

[503] G. Stoesser, P. Sterk, M. A. Tull, P. J. Stoehr, and G. N. Cameron. The EMBL nucleotide sequence database. *Nucl. Acids Res.*, 25:7–13, 1997.

[504] A. Stolcke and S. Omohundro. Hidden Markov model induction by Bayesian model merging. In S. J. Hanson, J. D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 11–18. Morgan Kaufmann, San Mateo, CA, 1993.

[505] P. Stolorz, A. Lapedes, and Y. Xia. Predicting protein secondary structure using neural net and statistical methods. *J. Mol. Biol.*, 225:363–377, 1992.

[506] G. D. Stormo, T. D. Schneider, L. Gold, and A. Ehrenfeucht. Use of the “perceptron” algorithm to distinguish translational initiation sites in *e. coli*. *Nucl. Acids Res.*, 10:2997–3011, 1982.

[507] G. D. Stormo, T. D. Schneider, and L. M. Gold. Characterization of translational initiation sites in *e. coli*. *Nucl. Acids Res.*, 10:2971–2996, 1982.

[508] C. D. Strader, T. M. Fong, M. R. Tota, and D. Underwood. Structure and function of G protein-coupled receptors. *Ann. Rev. Biochem.*, 63:101–132, 1994.

[509] R. Swanson. A unifying concept for the amino acid code. *Bull. Math. Biol.*, 46:187–203, 1984.

[510] R. H. Swendsen and J. S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.*, 58:86–88, 1987.

[511] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, 96:2907–2912, 1999.

[512] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test the chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Computing*, 13:566–579, 1984.

[513] R. L. Tatusov and E. V. Koonin D. J. Lipman. A genomic perspective on protein families. *Science*, 278:631–637, 1997.

[514] F. J. R. Taylor and D. Coates. The code within the codons. *Biosystems*, 22:177–187, 1989.

[515] W. R. Taylor and K. Hatrick. Compensating changes in protein multiple sequence alignments. *Prot. Eng.*, 7:341–348, 1994.

[516] T. A. Thanaraj. A clean data set of EST-confirmed splice sites from Homo sapiens and standards for clean-up procedures. *Nucl. Acids Res.*, 27:2627–2637, 1999.

[517] H. H. Thodberg. A review of Bayesian neural networks with an application to near infrared spectroscopy. *IEEE Trans. Neural Networks*, 7:56–72, 1996.

[518] C. A. Thomas. The genetic organization of chromosomes. *Ann. Rev. Genet.*, 5:237–256, 1971.

[519] J. L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 33:114–124, 1991.

[520] L. Tierney. Markov chains for exploring posterior distributions. *Ann. Statis.*, 22:1701–1762, 1994.

[521] I. Tinoco, Jr., O. C. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[522] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, New York, 1985.

[523] N. Tolstrup, C. V. Sensen, R. A. Garrett, and I. G. Clausen. Two different and highly organized mechanisms of translation initiation in the archaeon sulfolobus solfataricus. *Extremophiles*, 4:175–179, 2000.

[524] N. Tolstrup, J. Toftgard, J. Engelbrecht, and S. Brunak. Neural network model of the genetic code is strongly correlated to the GES scale of amino-acid transfer free-energies. *J. Mol. Biol.*, 243:816–820, 1994.

[525] E. N. Trifonov. Translation framing code and frame–monitoring mechanism as suggested by the analysis of mRNA and 16S rRNA nucleotide sequences. *J. Mol. Biol.*, 194:643–652, 1987.

[526] M. K. Trower, S. M. Orton, I. J. Purvis, P. Sanseau, J. Riley, C. Christodoulou, D. Burt, C. G. See, G. Elgar, R. Sherrington, E. I. Rogaev, P. St George-Hyslop, S. Brenner, and C. W. Dykes. Conservation of synteny between the genome of the pufferfish (*Fugu rubripes*) and the region on human chromosome 14 (14q24.3) associated with familial Alzheimer disease (AD3 locus). *Proc. Natl. Acad. Sci. USA*, 93:1366–1369, 1996.

[527] D. H. Turner and N. Sugimoto. RNA structure prediction. *Ann. Rev. Biophys. Biophys. Chem.*, 17:167–192, 1988.

[528] E. C. Uberbacher and R. J. Mural. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc. Natl. Acad. Sci. USA*, 88:11261–11265, 1991.

[529] E. C. Uberbacher, Ying Xu, and R. J. Mural. Discovering and understanding genes in human DNA sequence using GRAIL. *Meth. Enzymol.*, 266:259–281, 1996.

[530] J. van Helden, B. Andre, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.*, 281:827–842, 1998.

[531] J. van Helden, M. del Olmo, and J. E. Perez-Ortin. Statistical analysis of yeast genomic downstream sequences reveals putative polyadenylation signals. *Nucl. Acids Res.*, 28:1000–1010, 2000.

[532] E. P. van Someren, L. F. A. Wessels, and M. J. T. Reinders. Linear modeling of genetic networks from experimental data. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 355–366. AAAI Press, Menlo Park, CA, 2000.

[533] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[534] B. Venkatesh, B. H. Tay, G. Elgar, and S. Brenner. Isolation, characterization and evolution of nine pufferfish (*Fugu rubripes*) actin genes. *J. Mol. Biol.*, 259:655–665, 1996.

[535] J. Vilo and A. Brazma. Mining for putative regulatory elements in the yeast genome using gene expression data. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 384–394. AAAI Press, Menlo Park, CA, 2000.

[536] M. Vingron and P. Argos. A fast and sensitive multiple sequence alignment algorithm. *CABIOS*, 5:115–121, 1989.

[537] E. O. Voit. *Canonical Nonlinear Modeling*. Van Nostrand and Reinhold, New York, 1991.

[538] M. V. Volkenstein. The genetic coding of protein structure. *Biochim. Biophys. Acta*, 119:418–420, 1966.

[539] G. von Heijne. A new method for predicting signal sequence cleavage sites. *Nucl. Acids Res.*, 14:4683–4690, 1986.

[540] G. von Heijne. *Sequence Analysis in Molecular Biology: Treasure Trove or Trivial Pursuit?* Academic Press, London, 1987.

[541] G. von Heijne. Transcending the impenetrable: How proteins come to terms with membranes. *Biochim. Biophys. Acta*, 947:307–333, 1988.

[542] G. von Heijne. The signal peptide. *J. Membrane Biol.*, 115:195–201, 1990.

[543] G. von Heijne and C. Blomberg. The beta structure: Inter-strand correlations. *J. Mol. Biol.*, 117:821–824, 1977.

[544] P. H. von Hippel. Molecular databases of the specificity of interaction of transcriptional proteins with genome DNA. In R.F. Goldberger, editor, *Gene expression. Biological regulation and Development, vol. 1*, pages 279–347, New York, 1979. Plenum Press.

[545] S. S. Wachtel and T. R. Tiersch. Variations in genome mass. *Comp. Biochem. Physiol. B*, 104:207–213, 1993.

[546] G. Wahba. *Spline Models of Observational Oata*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[547] J. Wang and R. H. Swendsen. Cluster Monte Carlo algorithms. *Physica A*, 167:565–579, 1990.

[548] J. Wang and W. Wang. A computational approach to simplifying the protein folding alphabet. *Nat. Struct. Biol.*, 6:1033–1038, 1999.

[549] Z. X. Wang. Assessing the accuracy of protein secondary structure. *Nat. Struct. Biol.*, 1:145–146, 1994.

[550] M. S. Waterman. *Introduction to Computational Biology*. Chapman and Hall, London, 1995.

[551] T. A. Welch. A technique for high performance data compression. *IEEE Computer*, 17:8–19, 1984.

[552] J. Wess. G-protein-coupled receptors: molecular mechanisms involved in receptor activation and selectivity of g-protein recognition. *FASEB J.*, 11:346–354, 1997.

[553] J. V. White, C. M. Stultz, and T. F. Smith. Protein classification by stochastich modeling and optimal filtering of amino-acid sequences. *Mathem. Biosci.*, 119:35–75, 1994.

[554] K. P. White, S. A. Rifkin, P. Hurban, and D. S. Hogness. Microarray analysis of *drosophila* development during metamorphosis. *Science*, 286:2179–2184, 1999.

[555] S. H. White. Global statistics of protein sequences: Implications for the origin, evolution, and prediction of structure. *Ann. Rev. Biophys. Biomol. Struct.*, 23:407–439, 1994.

[556] S. H. White and R. E. Jacobs. The evolution of proteins from random amino acid sequences. I. Evidence from the lengthwise distribution of amino acids in modern protein sequences. *J. Mol. Evol.*, 36:79–95, 1993.

[557] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons, New York, 1990.

[558] B. L. Wiens. When log-normal and gamma models give different results: a case study. *The American Statistician*, 53:89–93, 1999.

[559] K. L. Williams, A. A. Gooley, and N. H. Packer. Proteome: Not just a made-up name. *Today's Life Sciences*, June:16–21, 1996.

[560] E. Wingender, X. Chen, R. Hehl, H. Karas, I. liebich, V. Matys, T. Meinhardt, M. Pruss, I. Reuter, and F. Schacherer. TRANSFAC: an integrated system for gene expression regulation. *Nucl. Acids Res.*, 28:316–319, 2000.

[561] H. Winkler. *Verbreitung und Ursache der Parthenogenesis im Pflanzen und Tierreich*. Fischer, Jena, 1920.

[562] C. R. Woese. *The Genetic Code. The Molecular Basis for Genetic Expression*. Harper & Row, New York, 1967.

[563] C. R. Woese, D. H. Dugre, S. A. Dugre, M. Kondo, and W. C. Saxinger. On the fundamental nature and evolution of the genetic code. *Cold Spring Harbor Symp. Quant. Biol.*, 31:723–736, 1966.

[564] C. R. Woese and G. E. Fox. Phylogenetic structure of the prokaryotic domain: The primary kingdoms. *Proc. Natl. Acad. Sci. USA*, 74:5088–5090, 1977.

[565] C. R. Woese, R. R. Gutell, R. Gupta, and H. F. Noller. Detailed analysis of the higher-order structure of 16S-like ribosomal ribonucleic acids. *Microbiol. Rev.*, 47:621–669, 1983.

[566] R. V. Wolfenden, P. M. Cullis, and C. C. F. Southgate. Water, protein folding, and the genetic code. *Science*, 206:575–577, 1979.

[567] T. G. Wolfsberg, A. E. Gabrielian, M. J. Campbell, R. J. Cho, J. L. Spouge, and D. Landsman. Candidate regulatory sequence elements for cell cycle-dependent transcription in *saccharomyces cerevisiae*. *Genome Res.*, 9:775–792, 1999.

[568] D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[569] J. T. Wong. A co-evolution theory of the genetic code. *Proc. Natl. Acad. Sci. USA*, 72:1909–1912, 1975.

[570] F. S. Wouters, M. Markman, P. de Graaf, H. Hauser, H. F. Tabak, K. W. Wirtz, and A. F. Moorman. The immunohistochemical localization of the non-specific lipid transfer protein (sterol carrier protein-2) in rat small intestine enterocytes. *Biochim. Biophys. Acta*, 1259:192–196, 1995.

[571] C. H. Wu. Artificial neural networks for molecular sequence analysis. *Comp. Chem.*, 21:237–256, 1997.

[572] C. H. Wu and J.W. McLarty. *Neural Networks and Genome Informatics*. Elsevier, Amsterdam, 2000.

[573] J. R. Wyatt, J. D. Puglisi, and I Tinoco, Jr. Hybrid system for protein secondary structure prediction. *BioEssays*, 11:100–106, 1989.

[574] L. Xu. A unified learning scheme: Bayesian-Kullback Ying-Yang machine. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, Cambridge, MA, 1996.

[575] M. Ycas. The protein text. In H. P. Yockey, editor, *Symposium on information theory in biology*, pages 70–102, New York, 1958. Pergamon.

[576] T. Yi, Y. Huang, M. I. Simon, and J. Doyle. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proc. Natl. Acad. Sci. USA*, 97:4649–4653, 2000.

[577] H. P. Yockey. *Information Theory and Molecular Biology*. Cambridge University Press, Cambridge, 1992.

[578] J. York. Use of the Gibbs sampler in expert systems. *Artif. Intell.*, 56:115–130, 1992.

[579] C. H. Yuh, H. Bolouri, and E. H. Davidson. Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902, 1998.

[580] A. Zemla, C. Venclovas, K. Fidelis, and B. Rost. A modified definition of SOV, a segment-based measure for protein secondary structure prediction assessment. *Proteins*, 34:220–223, 1999.

[581] M. Q. Zhang. Large-scale gene expression data analysis: a new challenge to computational biologists. *Genome Res.*, 9:681–688, 1999.

[582] X. Zhang, J. Mesirov, and D. Waltz. Hybrid system for protein secondary structure prediction. *J. Mol. Biol.*, 225:1049–1063, 1992.

[583] J. Zhu, J. Liu, and C. Lawrence. Bayesian adaptive alignment and inference. In T. Gaasterland, P. Karp, K. Karplus, C. Ouzounis, C. Sander, and A. Valencia, editors, *Proceedings of Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 358–368. AAAI Press, 1997. Menlo Park, CA.

[584] A. Zien, R. Kuffner, R. Zimmer, and T. Lengauer. Analysis of gene expression data with pathway scores. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 407–417. AAAI Press, Menlo Park, CA, 2000.

[585] M. Zuker. Computer prediction of RNA structure. *Meth. Enzymol.*, 180:262–288, 1989.

[586] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamic and auxiliary information. *Nucl. Acids Res.*, 9:133–148, 1981.

[587] M. Zvelebil, G. Barton, W. Taylor, and M. Sternberg. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J. Mol. Biol.*, 195:957–961, 1987.

This page intentionally left blank

Index