INTRODUCTION TO MACHINE LEARNING

Introduction to Machine Learning

Alex Smola and S.V.N. Vishwanathan

Yahoo! Labs
Santa Clara
–and–
Departments of Statistics and Computer Science
Purdue University
–and–
College of Engineering and Computer Science
Australian National University

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011–4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2008

Printed in the United Kingdom at the University Press, Cambridge

Typeface Monotype Times 10/13pt *System* LaTeX 2$_\varepsilon$ [Alexander J. Smola and S.V.N. Vishwanathan]

A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication data available

ISBN 0 521 82583 0 hardback

Author: vishy
Revision: 256
Timestamp: September 28, 2011
URL: svn://repos.stat.purdue.edu/thebook/trunk/Book/thebook.tex

Contents

1

Density Estimation

1.1 Limit Theorems

Assume you are a gambler and go to a casino to play a game of dice. As it happens, it is your unlucky day and among the 100 times you toss the dice, you only see '6' eleven times. For a fair dice we know that each face should occur with equal probability $\frac{1}{6}$. Hence the expected value over 100 draws is $\frac{100}{6} \approx 17$, which is considerably more than the eleven times that we observed. Before crying foul you decide that some mathematical analysis is in order.

The probability of seeing a *particular* sequence of m trials out of which n are a '6' is given by $\frac{1}{6}^n \frac{5}{6}^{m-n}$. Moreover, there are $\binom{m}{n} = \frac{m!}{n!(m-n)!}$ different sequences of '6' and 'not 6' with proportions n and $m-n$ respectively. Hence we may compute the probability of seeing a '6' only 11 or less via

$$\Pr(X \leq 11) = \sum_{i=0}^{11} p(i) = \sum_{i=0}^{11} \binom{100}{i} \left[\frac{1}{6}\right]^i \left[\frac{5}{6}\right]^{100-i} \approx 7.0\% \tag{1.1}$$

After looking at this figure you decide that things are probably reasonable. And, in fact, they are consistent with the convergence behavior of a simulated dice in Figure 1.1. In computing (1.1) we have learned something

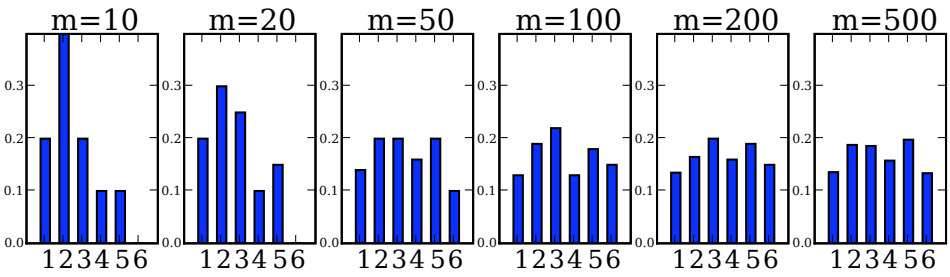

Fig. 1.1. Convergence of empirical means to expectations. From left to right: empirical frequencies of occurrence obtained by casting a dice 10, 20, 50, 100, 200, and 500 times respectively. Note that after 20 throws we still have not observed a single '6', an event which occurs with only $\left[\frac{5}{6}\right]^{20} \approx 2.6\%$ probability.

useful: the expansion is a special case of a *binomial* series. The first term counts the number of configurations in which we could observe i times '6' in a sequence of 100 dice throws. The second and third term are the probabilities of seeing one particular instance of such a sequence.

Note that in general we may not be as lucky, since we may have considerably less information about the setting we are studying. For instance, we might not *know* the actual probabilities for each face of the dice, which would be a likely assumption when gambling at a casino of questionable reputation. Often the outcomes of the system we are dealing with may be continuous valued random variables rather than binary ones, possibly even with unknown range. For instance, when trying to determine the average wage through a questionnaire we need to determine how many people we need to ask in order to obtain a certain level of confidence.

To answer such questions we need to discuss limit theorems. They tell us by how much averages over a set of observations may deviate from the corresponding expectations and how many observations we need to draw to estimate a number of probabilities reliably. For completeness we will present proofs for some of the more fundamental theorems in Section 1.1.2. They are useful albeit non-essential for the understanding of the remainder of the book and may be omitted.

1.1.1 Fundamental Laws

The Law of Large Numbers developed by Bernoulli in 1713 is one of the fundamental building blocks of statistical analysis. It states that averages over a number of observations converge to their expectations given a sufficiently large number of observations and given certain assumptions on the independence of these observations. It comes in two flavors: the weak and the strong law.

Theorem 1.1 (Weak Law of Large Numbers) *Denote by $X_1, \ldots, X_m$ random variables drawn from $p(x)$ with mean $\mu = \mathbf{E}_{X_i}[x_i]$ for all i. Moreover let*

$$\bar{X}_m := \frac{1}{m} \sum_{i=1}^{m} X_i \tag{1.2}$$

be the empirical average over the random variables X_i. Then for any $\epsilon > 0$ the following holds

$$\lim_{m \to \infty} \Pr\left(\left|\bar{X}_m - \mu\right| \leq \epsilon\right) = 1. \tag{1.3}$$

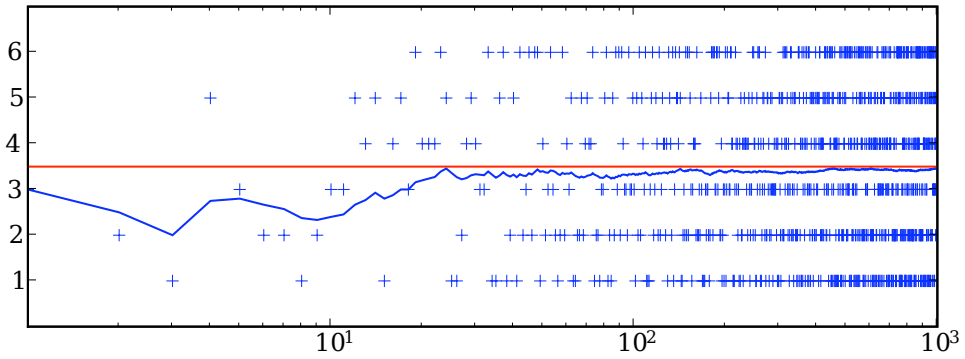

Fig. 1.2. The mean of a number of casts of a dice. The horizontal straight line denotes the mean 3.5. The uneven solid line denotes the actual mean $\bar{X}_n$ as a function of the number of draws, given as a semilogarithmic plot. The crosses denote the outcomes of the dice. Note how $\bar{X}_n$ ever more closely approaches the mean 3.5 are we obtain an increasing number of observations.

This establishes that, indeed, for large enough sample sizes, the average will converge to the expectation. The strong law strengthens this as follows:

Theorem 1.2 (Strong Law of Large Numbers) *Under the conditions of Theorem 1.1 we have* $\Pr\left(\lim_{m\to\infty} \bar{X}_m = \mu\right) = 1$.

The strong law implies that almost surely (in a measure theoretic sense) $\bar{X}_m$ converges to μ, whereas the weak law only states that for every ϵ the random variable $\bar{X}_m$ will be within the interval $[\mu-\epsilon, \mu+\epsilon]$. Clearly the strong implies the weak law since the measure of the events $\bar{X}_m = \mu$ converges to 1, hence any ϵ-ball around μ would capture this.

Both laws justify that we may take sample averages, e.g. over a number of events such as the outcomes of a dice and use the latter to estimate their means, their probabilities (here we treat the indicator variable of the event as a $\{0;1\}$-valued random variable), their variances or related quantities. We postpone a proof until Section 1.1.2, since an effective way of proving Theorem 1.1 relies on the theory of characteristic functions which we will discuss in the next section. For the moment, we only give a pictorial illustration in Figure 1.2.

Once we established that the random variable $\bar{X}_m = m^{-1}\sum_{i=1}^m X_i$ converges to its mean μ, a natural second question is to establish how *quickly* it converges and what the properties of the limiting distribution of $\bar{X}_m - \mu$ are. Note in Figure 1.2 that the initial deviation from the mean is large whereas as we observe more data the empirical mean approaches the true one.

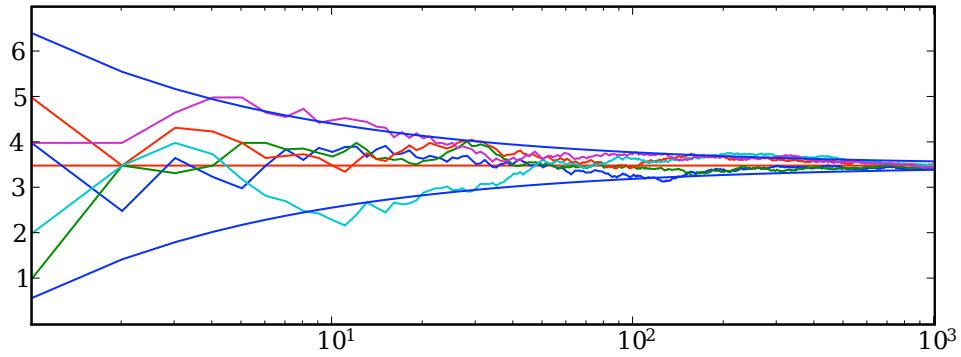

Fig. 1.3. Five instantiations of a running average over outcomes of a toss of a dice. Note that all of them converge to the mean 3.5. Moreover note that they all are well contained within the upper and lower envelopes given by $\mu \pm \sqrt{\mathrm{Var}_X[x]/m}$.

The central limit theorem answers this question exactly by addressing a slightly more general question, namely whether the sum over a number of independent random variables where each of them arises from a *different* distribution might also have a well behaved limiting distribution. This is the case as long as the *variance* of each of the random variables is bounded. The limiting distribution of such a sum is Gaussian. This affirms the pivotal role of the Gaussian distribution.

Theorem 1.3 (Central Limit Theorem) *Denote by X_i independent random variables with means μ_i and standard deviation σ_i. Then*

$$Z_m := \left[\sum_{i=1}^m \sigma_i^2\right]^{-\frac{1}{2}} \left[\sum_{i=1}^m X_i - \mu_i\right] \tag{1.4}$$

converges to a Normal Distribution with zero mean and unit variance.

Note that just like the law of large numbers the central limit theorem (CLT) is an *asymptotic* result. That is, only in the limit of an infinite number of observations will it become exact. That said, it often provides an excellent approximation even for finite numbers of observations, as illustrated in Figure 1.4. In fact, the central limit theorem and related limit theorems build the foundation of what is known as asymptotic statistics.

Example 1.1 (Dice) *If we are interested in computing the mean of the values returned by a dice we may apply the CLT to the sum over m variables*

which have all mean $\mu = 3.5$ *and variance (see Problem 1.1)*

$$\mathrm{Var}_X[x] = \mathbf{E}_X[x^2] - \mathbf{E}_X[x]^2 = (1+4+9+16+25+36)/6 - 3.5^2 \approx 2.92.$$

We now study the random variable $W_m := m^{-1}\sum_{i=1}^m [X_i - 3.5]$. *Since each of the terms in the sum has zero mean, also* W_m*'s mean vanishes. Moreover,* W_m *is a multiple of* Z_m *of (1.4). Hence we have that* W_m *converges to a normal distribution with zero mean and standard deviation* $2.92m^{-\frac{1}{2}}$.

Consequently the average of m *tosses of the dice yields a random variable with mean* 3.5 *and it will approach a normal distribution with variance* $m^{-\frac{1}{2}}2.92$. *In other words, the empirical mean converges to its average at rate* $O(m^{-\frac{1}{2}})$. *Figure 1.3 gives an illustration of the quality of the bounds implied by the CLT.*

One remarkable property of functions of random variables is that in many conditions convergence properties of the random variables are bestowed upon the functions, too. This is manifest in the following two results: a variant of Slutsky's theorem and the so-called delta method. The former deals with limit behavior whereas the latter deals with an extension of the central limit theorem.

Theorem 1.4 (Slutsky's Theorem) *Denote by* X_i, Y_i *sequences of random variables with* $X_i \to X$ *and* $Y_i \to c$ *for* $c \in \mathbb{R}$ *in probability. Moreover, denote by* $g(x,y)$ *a function which is continuous for all* (x,c). *In this case the random variable* $g(X_i, Y_i)$ *converges in probability to* $g(X,c)$.

For a proof see e.g. [Bil68]. Theorem 1.4 is often referred to as the continuous mapping theorem (Slutsky only proved the result for affine functions). It means that for functions of random variables it is possible to pull the limiting procedure *into* the function. Such a device is useful when trying to prove asymptotic normality and in order to obtain characterizations of the limiting distribution.

Theorem 1.5 (Delta Method) *Assume that* $X_n \in \mathbb{R}^d$ *is asymptotically normal with* $a_n^{-2}(X_n - b) \to \mathcal{N}(0,\Sigma)$ *for* $a_n^2 \to 0$. *Moreover, assume that* $g : \mathbb{R}^d \to \mathbb{R}^l$ *is a mapping which is continuously differentiable at* b. *In this case the random variable* $g(X_n)$ *converges*

$$a_n^{-2}\left(g(X_n) - g(b)\right) \to \mathcal{N}(0, [\nabla_x g(b)]\Sigma[\nabla_x g(b)]^\top). \tag{1.5}$$

Proof Via a Taylor expansion we see that

$$a_n^{-2}\left[g(X_n) - g(b)\right] = [\nabla_x g(\xi_n)]^\top a_n^{-2}(X_n - b) \tag{1.6}$$

Here ξ_n lies on the line segment $[b, X_n]$. Since $X_n \to b$ we have that $\xi_n \to b$, too. Since g is continuously differentiable at b we may apply Slutsky's theorem to see that $a_n^{-2}\left[g(X_n) - g(b)\right] \to \left[\nabla_x g(b)\right]^\top a_n^{-2}(X_n - b)$. As a consequence, the transformed random variable is asymptotically normal with covariance $\left[\nabla_x g(b)\right]\Sigma\left[\nabla_x g(b)\right]^\top$. ■

We will use the delta method when it comes to investigating properties of maximum likelihood estimators in exponential families. There g will play the role of a mapping between expectations and the natural parametrization of a distribution.

1.1.2 The Characteristic Function

The Fourier transform plays a crucial role in many areas of mathematical analysis and engineering. This is equally true in statistics. For historic reasons its applications to distributions is called the characteristic function, which we will discuss in this section. At its foundations lie standard tools from functional analysis and signal processing [Rud73, Pap62]. We begin by recalling the basic properties:

Definition 1.6 (Fourier Transform) *Denote by $f : \mathbb{R}^n \to \mathbb{C}$ a function defined on a d-dimensional Euclidean space. Moreover, let $x, \omega \in \mathbb{R}^n$. Then the Fourier transform F and its inverse F^{-1} are given by*

$$F[f](\omega) := (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^n} f(x) \exp(-i \langle \omega, x \rangle) dx \tag{1.7}$$

$$F^{-1}[g](x) := (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^n} g(\omega) \exp(i \langle \omega, x \rangle) d\omega. \tag{1.8}$$

The key insight is that $F^{-1} \circ F = F \circ F^{-1} = \mathrm{Id}$. In other words, F and F^{-1} are inverses to each other for all functions which are L_2 integrable on $\mathbb{R}^d$, which includes probability distributions. One of the key advantages of Fourier transforms is that derivatives and convolutions on f translate into multiplications. That is $F[f \circ g] = (2\pi)^{\frac{d}{2}} F[f] \cdot F[g]$. The same rule applies to the inverse transform, i.e. $F^{-1}[f \circ g] = (2\pi)^{\frac{d}{2}} F^{-1}[f] F^{-1}[g]$.

The benefit for statistical analysis is that often problems are more easily expressed in the Fourier domain and it is easier to prove convergence results there. These results then carry over to the original domain. We will be exploiting this fact in the proof of the law of large numbers and the central limit theorem. Note that the definition of Fourier transforms can be extended to more general domains such as groups. See e.g. [BCR84] for further details.

We next introduce the notion of a *characteristic function* of a distribution.[1]

Definition 1.7 (Characteristic Function) *Denote by $p(x)$ a distribution of a random variable $X \in \mathbb{R}^d$. Then the characteristic function $\phi_X(\omega)$ with $\omega \in \mathbb{R}^d$ is given by*

$$\phi_X(\omega) := (2\pi)^{\frac{d}{2}} F^{-1}[p(x)] = \int \exp(i \langle \omega, x \rangle) dp(x). \tag{1.9}$$

In other words, $\phi_X(\omega)$ is the *inverse Fourier transform* applied to the probability measure $p(x)$. Consequently $\phi_X(\omega)$ *uniquely* characterizes $p(x)$ and moreover, $p(x)$ can be recovered from $\phi_X(\omega)$ via the forward Fourier transform. One of the key utilities of characteristic functions is that they allow us to deal in easy ways with *sums* of random variables.

Theorem 1.8 (Sums of random variables and convolutions) *Denote by $X, Y \in \mathbb{R}$ two independent random variables. Moreover, denote by $Z := X + Y$ the sum of both random variables. Then the distribution over Z satisfies $p(z) = p(x) \circ p(y)$. Moreover, the characteristic function yields:*

$$\phi_Z(\omega) = \phi_X(\omega)\phi_Y(\omega). \tag{1.10}$$

Proof Z is given by $Z = X + Y$. Hence, for a given $Z = z$ we have the freedom to choose $X = x$ freely provided that $Y = z - x$. In terms of distributions this means that the joint distribution $p(z, x)$ is given by

$$\begin{aligned} p(z, x) &= p(Y = z - x)p(x) \\ \text{and hence } p(z) &= \int p(Y = z - x) dp(x) = [p(x) \circ p(y)](z). \end{aligned}$$

The result for characteristic functions follows form the property of the Fourier transform. ∎

For sums of several random variables the characteristic function is the product of the individual characteristic functions. This allows us to prove both the weak law of large numbers and the central limit theorem (see Figure 1.4 for an illustration) by proving convergence in the Fourier domain.

Proof [Weak Law of Large Numbers] At the heart of our analysis lies a Taylor expansion of the exponential into

$$\begin{aligned} \exp(iwx) &= 1 + i \langle w, x \rangle + o(|w|) \\ \text{and hence } \phi_X(\omega) &= 1 + iw\mathbf{E}_X[x] + o(|w|). \end{aligned}$$

[1] In Chapter **??** we will discuss more general descriptions of distributions of which ϕ_X is a special case. In particular, we will replace the exponential $\exp(i \langle \omega, x \rangle)$ by a kernel function $k(x, x')$.

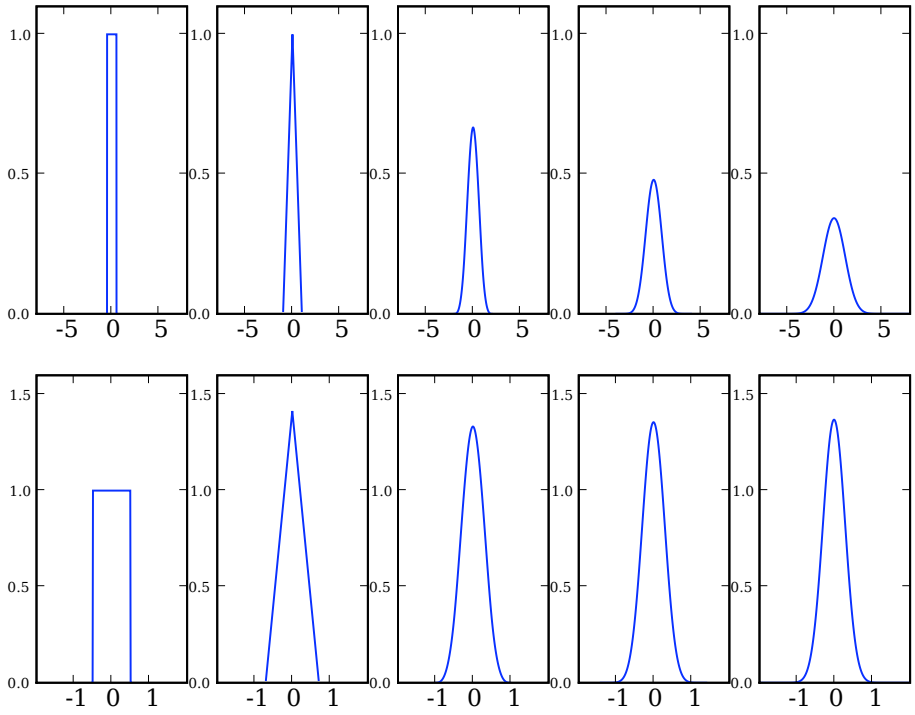

Fig. 1.4. A working example of the central limit theorem. The top row contains distributions of sums of uniformly distributed random variables on the interval $[0.5, 0.5]$. From left to right we have sums of $1, 2, 4, 8$ and 16 random variables. The bottom row contains the same distribution with the means rescaled by $\sqrt{m}$, where m is the number of observations. Note how the distribution converges increasingly to the normal distribution.

Given m random variables X_i with mean $\mathbf{E}_X[x] = \mu$ this means that their average $\bar{X}_m := \frac{1}{m}\sum_{i=1}^{m} X_i$ has the characteristic function

$$\phi_{\bar{X}_m}(\omega) = \left(1 + \frac{i}{m} w\mu + o(m^{-1}\,|w|)\right)^m \tag{1.11}$$

In the limit of $m \to \infty$ this converges to $\exp(iw\mu)$, the characteristic function of the constant distribution with mean μ. This proves the claim that in the large sample limit $\bar{X}_m$ is essentially constant with mean μ. ■

Proof [Central Limit Theorem] We use the same idea as above to prove the CLT. The main difference, though, is that we need to assume that the

second moments of the random variables X_i *exist*. To avoid clutter we only prove the case of constant mean $\mathbf{E}_{X_i}[x_i] = \mu$ and variance $\mathrm{Var}_{X_i}[x_i] = \sigma^2$.

Let $Z_m := \frac{1}{\sqrt{m\sigma^2}} \sum_{i=1}^{m} (X_i - \mu)$. Our proof relies on showing convergence of the characteristic function of Z_m, i.e. ϕ_{Z_m} to that of a normally distributed random variable W with zero mean and unit variance. Expanding the exponential to second order yields:

$$\exp(iwx) = 1 + iwx - \frac{1}{2}w^2x^2 + o(|w|^2)$$
$$\text{and hence } \phi_X(\omega) = 1 + iw\mathbf{E}_X[x] - \frac{1}{2}w^2\mathrm{Var}_X[x] + o(|w|^2)$$

Since the mean of Z_m vanishes by centering $(X_i - \mu)$ and the variance per variable is m^{-1} we may write the characteristic function of Z_m via

$$\phi_{Z_m}(\omega) = \left(1 - \frac{1}{2m}w^2 + o(m^{-1}\,|w|^2)\right)^m$$

As before, taking limits $m \to \infty$ yields the exponential function. We have that $\lim_{m\to\infty} \phi_{Z_m}(\omega) = \exp(-\frac{1}{2}\omega^2)$ which is the characteristic function of the normal distribution with zero mean and variance 1. Since the characteristic function transform is injective this proves our claim. ■

Note that the characteristic function has a number of useful properties. For instance, it can also be used as moment generating function via the identity:

$$\nabla^n_\omega \phi_X(0) = i^{-n}\mathbf{E}_X[x^n]. \tag{1.12}$$

Its proof is left as an exercise. See Problem 1.2 for details. This connection also implies (subject to regularity conditions) that if we know the moments of a distribution we are able to reconstruct it directly since it allows us to reconstruct its characteristic function. This idea has been exploited in density estimation [Cra46] in the form of Edgeworth and Gram-Charlier expansions [Hal92].

1.1.3 Tail Bounds

In practice we never have access to an *infinite* number of observations. Hence the central limit theorem does not apply but is just an approximation to the real situation. For instance, in the case of the dice, we might want to state *worst case bounds* for *finite* sums of random variables to determine by how much the empirical mean may deviate from its expectation. Those bounds will not only be useful for simple averages but to quantify the behavior of more sophisticated estimators based on a set of observations.

The bounds we discuss below differ in the amount of knowledge they assume about the random variables in question. For instance, we might only know their mean. This leads to the Gauss-Markov inequality. If we know their mean and their variance we are able to state a stronger bound, the Chebyshev inequality. For an even stronger setting, when we know that each variable has bounded range, we will be able to state a Chernoff bound. Those bounds are progressively more tight and also more difficult to prove. We state them in order of technical sophistication.

Theorem 1.9 (Gauss-Markov) *Denote by $X \geq 0$ a random variable and let μ be its mean. Then for any $\epsilon > 0$ we have*

$$\Pr(X \geq \epsilon) \leq \frac{\mu}{\epsilon}. \tag{1.13}$$

Proof We use the fact that for nonnegative random variables

$$\Pr(X \geq \epsilon) = \int_{\epsilon}^{\infty} dp(x) \leq \int_{\epsilon}^{\infty} \frac{x}{\epsilon} dp(x) \leq \epsilon^{-1} \int_{0}^{\infty} x dp(x) = \frac{\mu}{\epsilon}.$$

This means that for random variables with a small mean, the proportion of samples with large value has to be small. ■

Consequently deviations from the mean are $O(\epsilon^{-1})$. However, note that this bound does *not* depend on the number of observations. A useful application of the Gauss-Markov inequality is Chebyshev's inequality. It is a statement on the range of random variables using its variance.

Theorem 1.10 (Chebyshev) *Denote by X a random variable with mean μ and variance σ^2. Then the following holds for $\epsilon > 0$:*

$$\Pr(|x - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}. \tag{1.14}$$

Proof Denote by $Y := |X - \mu|^2$ the random variable quantifying the deviation of X from its mean μ. By construction we know that $\mathbf{E}_Y[y] = \sigma^2$. Next let $\gamma := \epsilon^2$. Applying Theorem 1.9 to Y and γ yields $\Pr(Y > \gamma) \leq \sigma^2/\gamma$ which proves the claim. ■

Note the improvement to the Gauss-Markov inequality. Where before we had bounds whose confidence improved with $O(\epsilon^{-1})$ we can now state $O(\epsilon^{-2})$ bounds for deviations from the mean.

Example 1.2 (Chebyshev bound) *Assume that $\bar{X}_m := m^{-1} \sum_{i=1}^{m} X_i$ is the average over m random variables with mean μ and variance σ^2. Hence*

$\bar{X}_m$ also has mean μ. Its variance is given by

$$\mathrm{Var}_{\bar{X}_m}[\bar{x}_m] = \sum_{i=1}^{m} m^{-2}\mathrm{Var}_{X_i}[x_i] = m^{-1}\sigma^2.$$

Applying Chebyshev's inequality yields that the probability of a deviation of ϵ from the mean μ is bounded by $\frac{\sigma^2}{m\epsilon^2}$. For fixed failure probability $\delta = \Pr(|\bar{X}_m - \mu| > \epsilon)$ we have

$$\delta \leq \sigma^2 m^{-1}\epsilon^{-2} \text{ and equivalently } \epsilon \leq \sigma/\sqrt{m\delta}.$$

This bound is quite reasonable for large δ but it means that for high levels of confidence we need a huge number of observations.

Much stronger results can be obtained if we are able to bound the *range* of the random variables. Using the latter, we reap an exponential improvement in the quality of the bounds in the form of the McDiarmid [McD89] inequality. We state the latter without proof:

Theorem 1.11 (McDiarmid) *Denote by $f : \mathcal{X}^m \to \mathbb{R}$ a function on $\mathcal{X}$ and let X_i be independent random variables. In this case the following holds:*

$$\Pr\left(|f(x_1, \ldots, x_m) - \mathbf{E}_{X_1,\ldots,X_m}[f(x_1, \ldots, x_m)]| > \epsilon\right) \leq 2\exp\left(-2\epsilon^2 C^{-2}\right).$$

Here the constant C^2 is given by $C^2 = \sum_{i=1}^m c_i^2$ where

$$\left|f(x_1, \ldots, x_i, \ldots, x_m) - f(x_1, \ldots, x_i', \ldots, x_m)\right| \leq c_i$$

for all $x_1, \ldots, x_m, x_i'$ and for all i.

This bound can be used for averages of a number of observations when they are computed according to some algorithm as long as the latter can be encoded in f. In particular, we have the following bound [Hoe63]:

Theorem 1.12 (Hoeffding) *Denote by X_i iid random variables with bounded range $X_i \in [a, b]$ and mean μ. Let $\bar{X}_m := m^{-1}\sum_{i=1}^m X_i$ be their average. Then the following bound holds:*

$$\Pr\left(\left|\bar{X}_m - \mu\right| > \epsilon\right) \leq 2\exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right). \tag{1.15}$$

Proof This is a corollary of Theorem 1.11. In $\bar{X}_m$ each individual random variable has range $[a/m, b/m]$ and we set $f(X_1, \ldots, X_m) := \bar{X}_m$. Straightforward algebra shows that $C^2 = m^{-2}(b-a)^2$. Plugging this back into McDiarmid's theorem proves the claim. ■

Note that (1.15) is *exponentially* better than the previous bounds. With increasing sample size the confidence level also increases exponentially.

Example 1.3 (Hoeffding bound) *As in example 1.2 assume that X_i are iid random variables and let $\bar{X}_m$ be their average. Moreover, assume that $X_i \in [a, b]$ for all i. As before we want to obtain guarantees on the probability that $|\bar{X}_m - \mu| > \epsilon$. For a given level of confidence $1 - \delta$ we need to solve*

$$\delta \leq 2 \exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right) \tag{1.16}$$

for ϵ. Straightforward algebra shows that in this case ϵ needs to satisfy

$$\epsilon \geq |b - a| \sqrt{\left[\log 2 - \log \delta\right] / 2m} \tag{1.17}$$

In other words, while the confidence level only enters logarithmically into the inequality, the sample size m improves our confidence only with $\epsilon = O(m^{-\frac{1}{2}})$. That is, in order to improve our confidence interval from $\epsilon = 0.1$ to $\epsilon = 0.01$ we need 100 times as many observations.

While this bound is tight (see Problem 1.5 for details), it is possible to obtain better bounds if we know *additional* information. In particular knowing a bound on the *variance* of a random variable in addition to knowing that it has bounded range would allow us to strengthen the statement considerably. The Bernstein inequality captures this connection. For details see [BBL05] or works on empirical process theory [vdVW96, SW86, Vap82].

1.1.4 An Example

It is probably easiest to illustrate the various bounds using a concrete example. In a semiconductor fab processors are produced on a wafer. A typical 300mm wafer holds about 400 chips. A large number of processing steps are required to produce a finished microprocessor and often it is impossible to assess the effect of a design decision until the finished product has been produced.

Assume that the production manager wants to change some step from process 'A' to some other process 'B'. The goal is to increase the yield of the process, that is, the number of chips of the 400 potential chips on the wafer which can be sold. Unfortunately this number is a random variable, i.e. the number of working chips per wafer can vary widely between different wafers. Since process 'A' has been running in the factory for a very long time we may assume that the yield is well known, say it is $\mu_A = 350$ out of 400 processors on average. It is our goal to determine whether process 'B' is better and what its yield may be. Obviously, since production runs are expensive we want to be able to determine this number as quickly as possible, i.e. using as few wafers as possible. The production manager is risk

averse and wants to ensure that the new process is really better. Hence he requires a confidence level of 95% before he will change the production.

A first step is to formalize the problem. Since we know process 'A' exactly we only need to concern ourselves with 'B'. We associate the random variable X_i with wafer i. A reasonable (and somewhat simplifying) assumption is to posit that all X_i are independent and identically distributed where all X_i have the mean μ_B. Obviously we do not know μ_B — otherwise there would be no reason for testing! We denote by $\bar{X}_m$ the average of the yields of m wafers using process 'B'. What we are interested in is the accuracy ϵ for which the probability

$$\delta = \Pr(|\bar{X}_m - \mu_B| > \epsilon) \text{ satisfies } \delta \leq 0.05.$$

Let us now discuss how the various bounds behave. For the sake of the argument assume that $\mu_B - \mu_A = 20$, i.e. the new process produces on average 20 additional usable chips.

Chebyshev In order to apply the Chebyshev inequality we need to bound the variance of the random variables X_i. The worst possible variance would occur if $X_i \in \{0; 400\}$ where both events occur with equal probability. In other words, with equal probability the wafer if fully usable or it is entirely broken. This amounts to $\sigma^2 = 0.5(200-0)^2 + 0.5(200-400)^2 = 40,000$. Since for Chebyshev bounds we have

$$\delta \leq \sigma^2 m^{-1} \epsilon^{-2} \tag{1.18}$$

we can solve for $m = \sigma^2/\delta\epsilon^2 = 40,000/(0.05 \cdot 400) = 20,000$. In other words, we would typically need 20,000 wafers to assess with reasonable confidence whether process 'B' is better than process 'A'. This is completely unrealistic.

Slightly better bounds can be obtained if we are able to make better assumptions on the variance. For instance, if we can be sure that the yield of process 'B' is at least 300, then the largest possible variance is $0.25(300-0)^2 + 0.75(300-400)^2 = 30,000$, leading to a minimum of 15,000 wafers which is not much better.

Hoeffding Since the yields are in the interval $\{0, \ldots, 400\}$ we have an explicit bound on the range of observations. Recall the inequality (1.16) which bounds the failure probably $\delta = 0.05$ by an exponential term. Solving this for m yields

$$m \geq 0.5|b-a|^2 \epsilon^{-2} \log(2/\delta) \approx 737.8 \tag{1.19}$$

In other words, we need at lest 738 wafers to determine whether process 'B' is better. While this is a significant improvement of almost two orders of magnitude, it still seems wasteful and we would like to do better.

Central Limit Theorem The central limit theorem is an *approximation*. This means that our reasoning is not accurate any more. That said, for large enough sample sizes, the approximation is good enough to use it for practical predictions. Assume for the moment that we knew the variance σ^2 exactly. In this case we know that $\bar{X}_m$ is approximately normal with mean μ_B and variance $m^{-1}\sigma^2$. We are interested in the interval $[\mu-\epsilon, \mu+\epsilon]$ which contains 95% of the probability mass of a normal distribution. That is, we need to solve the integral

$$\frac{1}{2\pi\sigma^2}\int_{\mu-\epsilon}^{\mu+\epsilon} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = 0.95 \tag{1.20}$$

This can be solved efficiently using the cumulative distribution function of a normal distribution (see Problem 1.3 for more details). One can check that (1.20) is solved for $\epsilon = 2.96\sigma$. In other words, an interval of $\pm 2.96\sigma$ contains 95% of the probability mass of a normal distribution. The number of observations is therefore determined by

$$\epsilon = 2.96\sigma/\sqrt{m} \text{ and hence } m = 8.76\frac{\sigma^2}{\epsilon^2} \tag{1.21}$$

Again, our problem is that we do *not* know the variance of the distribution. Using the worst-case bound on the variance, i.e. $\sigma^2 = 40,000$ would lead to a requirement of at least $m = 876$ wafers for testing. However, while we do not *know* the variance, we may estimate it along with the mean and use the empirical estimate, possibly plus some small constant to ensure we do not underestimate the variance, instead of the upper bound.

Assuming that fluctuations turn out to be in the order of 50 processors, i.e. $\sigma^2 = 2500$, we are able to reduce our requirement to approximately 55 wafers. This is probably an acceptable number for a practical test.

Rates and Constants The astute reader will have noticed that all three confidence bounds had scaling behavior $m = O(\epsilon^{-2})$. That is, in all cases the number of observations was a fairly ill behaved function of the amount of confidence required. If we were just interested in convergence per se, a statement like that of the Chebyshev inequality would have been entirely sufficient. The various laws and bounds can often be used to obtain considerably better *constants* for statistical confidence guarantees. For more

complex estimators, such as methods to classify, rank, or annotate data, a reasoning such as the one above can become highly nontrivial. See e.g. [MYA94, Vap98] for further details.

1.2 Parzen Windows

1.2.1 Discrete Density Estimation

The convergence theorems discussed so far mean that we can use empirical observations for the purpose of density estimation. Recall the case of the Naive Bayes classifier of Section **??**. One of the key ingredients was the ability to use information about word counts for different document classes to estimate the probability $p(w^j|y)$, where w^j denoted the number of occurrences of word j in document x, given that it was labeled y. In the following we discuss an extremely simple and crude method for estimating probabilities. It relies on the fact that for random variables X_i drawn from distribution $p(x)$ with discrete values $X_i \in \mathcal{X}$ we have

$$\lim_{m\to\infty} \hat{p}_X(x) = p(x) \tag{1.22}$$

$$\text{where } \hat{p}_X(x) := m^{-1} \sum_{i=1}^{m} \{x_i = x\} \text{ for all } x \in \mathcal{X}. \tag{1.23}$$

Let us discuss a concrete case. We assume that we have 12 documents and would like to estimate the probability of occurrence of the word 'dog' from it. As raw data we have:

Document ID	1	2	3	4	5	6	7	8	9	10	11	12
Occurrences of 'dog'	1	0	2	0	4	6	3	0	6	2	0	1

This means that the word 'dog' occurs the following number of times:

Occurrences of 'dog'	0	1	2	3	4	5	6
Number of documents	4	2	2	1	1	0	2

Something unusual is happening here: for some reason we never observed 5 instances of the word dog in our documents, only 4 and less, or alternatively 6 times. So what about 5 times? It is reasonable to assume that the corresponding value should not be 0 either. Maybe we did not sample enough. One possible strategy is to add pseudo-counts to the observations. This amounts to the following estimate:

$$\hat{p}_X(x) := (m + |\mathcal{X}|)^{-1} \Big[1 + \sum_{i=1}^{m} \{x_i = x\} = p(x)\Big] \tag{1.24}$$

Clearly the limit for $m \to \infty$ is still $p(x)$. Hence, asymptotically we do not lose anything. This prescription is what we used in Algorithm **??** used a method called Laplace smoothing. Below we contrast the two methods:

Occurrences of 'dog'	0	1	2	3	4	5	6
Number of documents	4	2	2	1	1	0	2
Frequency of occurrence	0.33	0.17	0.17	0.083	0.083	0	0.17
Laplace smoothing	0.26	0.16	0.16	0.11	0.11	0.05	0.16

The problem with this method is that as $|\mathcal{X}|$ increases we need increasingly more observations to obtain even a modicum of precision. On average, we will need at least one observation for every $x \in \mathcal{X}$. This can be infeasible for large domains as the following example shows.

Example 1.4 (Curse of Dimensionality) *Assume that $\mathcal{X} = \{0, 1\}^d$, i.e. x consists of binary bit vectors of dimensionality d. As d increases the size of $\mathcal{X}$ increases exponentially, requiring an exponential number of observations to perform density estimation. For instance, if we work with images, a 100×100 black and white picture would require in the order of 10^{3010} observations to model such fairly low-resolution images accurately. This is clearly utterly infeasible — the number of particles in the known universe is in the order of 10^{80}. Bellman [Bel61] was one of the first to formalize this dilemma by coining the term 'curse of dimensionality'.*

This example clearly shows that we need better tools to deal with high-dimensional data. We will present one of such tools in the next section.

1.2.2 Smoothing Kernel

We now proceed to proper density estimation. Assume that we want to estimate the distribution of weights of a population. Sample data from a population might look as follows: $X = \{57, 88, 54, 84, 83, 59, 56, 43, 70, 63, 90, 98, 102, 97, 106, 99, 103, 112\}$. We could use this to perform a density estimate by placing discrete components at the locations $x_i \in X$ with weight $1/|X|$ as what is done in Figure 1.5. There is no reason to believe that weights are quantized in kilograms, or grams, or miligrams (or pounds and stones). And even if it were, we would expect that similar weights would have similar densities associated with it. Indeed, as the right diagram of Figure 1.5 shows, the corresponding density is continuous.

The key question arising is how we may transform X into a realistic estimate of the density $p(x)$. Starting with a 'density estimate' with only

discrete terms

$$\hat{p}(x) = \frac{1}{m} \sum_{i=1}^{m} \delta(x - x_i) \tag{1.25}$$

we may choose to smooth it out by a smoothing kernel $h(x)$ such that the probability mass becomes somewhat more spread out. For a density estimate on $\mathcal{X} \subseteq \mathbb{R}^d$ this is achieved by

$$\hat{p}(x) = \frac{1}{m} \sum_{i=1}^{m} r^{-d} h\left(\tfrac{x-x_i}{r}\right). \tag{1.26}$$

This expansion is commonly known as the *Parzen windows* estimate. Note that obviously h must be chosen such that $h(x) \geq 0$ for all $x \in \mathcal{X}$ and moreover that $\int h(x)dx = 1$ in order to ensure that (1.26) is a proper probability distribution. We now formally justify this smoothing. Let R be a small region such that

$$q = \int_R p(x)\, dx.$$

Out of the m samples drawn from $p(x)$, the probability that k of them fall in region R is given by the binomial distribution

$$\binom{m}{k} q^k (1-q)^{m-k}.$$

The expected fraction of points falling inside the region can easily be computed from the expected value of the Binomial distribution: $\mathbb{E}[k/m] = q$. Similarly, the variance can be computed as $\mathrm{Var}[k/m] = q(1-q)/m$. As $m \to \infty$ the variance goes to 0 and hence the estimate peaks around the expectation. We can therefore set

$$k \approx mq.$$

If we assume that R is so small that $p(x)$ is constant over R, then

$$q \approx p(x) \cdot V,$$

where V is the volume of R. Rearranging we obtain

$$p(x) \approx \frac{k}{mV}. \tag{1.27}$$

Let us now set R to be a cube with side length r, and define a function

$$h(u) = \begin{cases} 1 \text{ if } |u_i| \leq \frac{1}{2} \\ 0 \text{ otherwise.} \end{cases}$$

Observe that $h\left(\frac{x-x_i}{r}\right)$ is 1 if and only if x_i lies inside a cube of size r centered around x. If we let

$$k = \sum_{i=1}^{m} h\left(\frac{x - x_i}{r}\right),$$

then one can use (1.27) to estimate p via

$$\hat{p}(x) = \frac{1}{m} \sum_{i=1}^{m} r^{-d} h\left(\frac{x - x_i}{r}\right),$$

where r^d is the volume of the hypercube of size r in d dimensions. By symmetry, we can interpret this equation as the sum over m cubes centered around m data points x_n. If we replace the cube by any smooth kernel function $h(\cdot)$ this recovers (1.26).

There exists a large variety of different kernels which can be used for the kernel density estimate. [Sil86] has a detailed description of the properties of a number of kernels. Popular choices are

$$h(x) = (2\pi)^{-\frac{1}{2}} e^{-\frac{1}{2}x^2} \quad \text{Gaussian kernel} \tag{1.28}$$

$$h(x) = \tfrac{1}{2} e^{-|x|} \quad \text{Laplace kernel} \tag{1.29}$$

$$h(x) = \tfrac{3}{4} \max(0, 1 - x^2) \quad \text{Epanechnikov kernel} \tag{1.30}$$

$$h(x) = \tfrac{1}{2} \chi_{[-1,1]}(x) \quad \text{Uniform kernel} \tag{1.31}$$

$$h(x) = \max(0, 1 - |x|) \quad \text{Triangle kernel.} \tag{1.32}$$

Further kernels are the triweight and the quartic kernel which are basically powers of the Epanechnikov kernel. For practical purposes the Gaussian kernel (1.28) or the Epanechnikov kernel (1.30) are most suitable. In particular, the latter has the attractive property of compact support. This means that for any given density estimate at location x we will only need to evaluate terms $h(x_i - x)$ for which the distance $\|x_i - x\|$ is less than r. Such expansions are computationally much cheaper, in particular when we make use of fast nearest neighbor search algorithms [GIM99, IM98]. Figure 1.7 has some examples of kernels.

1.2.3 Parameter Estimation

So far we have not discussed the issue of parameter selection. It should be evident from Figure 1.6, though, that it is quite crucial to choose a good kernel width. Clearly, a kernel that is overly wide will oversmooth any fine detail that there might be in the density. On the other hand, a very narrow

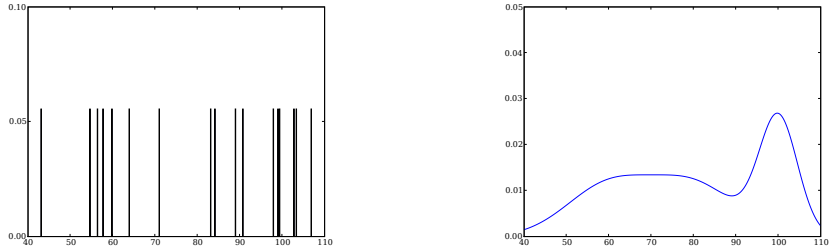

Fig. 1.5. Left: a naive density estimate given a sample of the weight of 18 persons. Right: the underlying weight distribution.

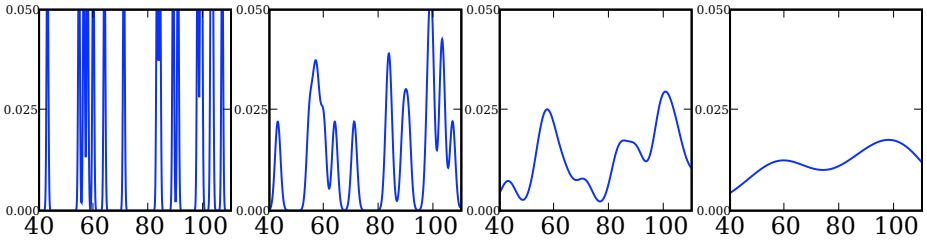

Fig. 1.6. Parzen windows density estimate associated with the 18 observations of the Figure above. From left to right: Gaussian kernel density estimate with kernel of width $0.3, 1, 3$, and 10 respectively.

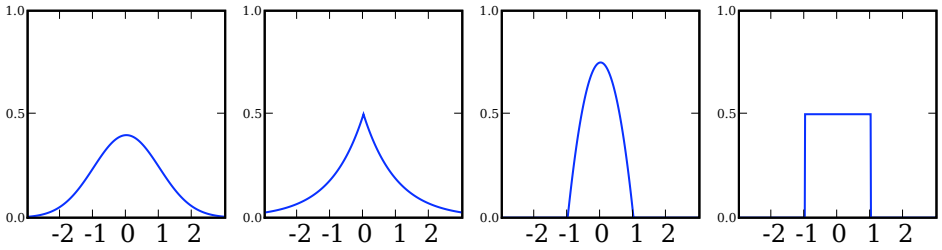

Fig. 1.7. Some kernels for Parzen windows density estimation. From left to right: Gaussian kernel, Laplace kernel, Epanechikov kernel, and uniform density.

kernel will not be very useful, since it will be able to make statements only about the locations where we actually observed data.

Moreover, there is the issue of choosing a suitable kernel function. The fact that a large variety of them exists might suggest that this is a crucial issue. In practice, this turns out not to be the case and instead, the choice

of a suitable kernel width is much more vital for good estimates. In other words, size matters, shape is secondary.

The problem is that we do not know which kernel width is best for the data. If the problem is one-dimensional, we might hope to be able to eyeball the size of r. Obviously, in higher dimensions this approach fails. A second option would be to choose r such that the log-likelihood of the data is maximized. It is given by

$$\log \prod_{i=1}^{m} p(x_i) = -m \log m + \sum_{i=1}^{m} \log \sum_{j=1}^{m} r^{-d} h\left(\tfrac{x_i - x_j}{r}\right) \tag{1.33}$$

Remark 1.13 (Log-likelihood) *We consider the logarithm of the likelihood for reasons of computational stability to prevent numerical underflow. While each term $p(x_i)$ might be within a suitable range, say 10^{-2}, the product of 1000 of such terms will easily exceed the exponent of floating point representations on a computer. Summing over the logarithm, on the other hand, is perfectly feasible even for large numbers of observations.*

Unfortunately computing the log-likelihood is equally infeasible: for decreasing r the only surviving terms in (1.33) are the functions $h((x_i - x_i)/r) = h(0)$, since the arguments of all other kernel functions diverge. In other words, the log-likelihood is maximized when $p(x)$ is peaked exactly at the locations where we observed the data. The graph on the left of Figure 1.6 shows what happens in such a situation.

What we just experienced is a case of *overfitting* where our model is too flexible. This led to a situation where our model was able to explain the observed data "unreasonably well", simply because we were able to adjust our parameters given the data. We will encounter this situation throughout the book. There exist a number of ways to address this problem.

Validation Set: We could use a subset of our set of observations as an *estimate* of the log-likelihood. That is, we could partition the observations into $\mathbf{X} := \{x_1, \ldots, x_n\}$ and $\mathbf{X}' := \{x_{n+1}, \ldots, x_m\}$ and use the second part for a likelihood score according to (1.33). The second set is typically called a *validation set*.

n-fold Cross-validation: Taking this idea further, note that there is no particular reason why any given x_i should belong to $\mathbf{X}$ or $\mathbf{X}'$ respectively. In fact, we could use all splits of the observations into sets $\mathbf{X}$ and $\mathbf{X}'$ to infer the quality of our estimate. While this is computationally infeasible, we could decide to split the observations into n equally sized subsets, say $\mathbf{X}_1, \ldots, \mathbf{X}_n$ and use each of them as a

validation set at a time while the remainder is used to generate a density estimate.

Typically n is chosen to be 10, in which case this procedure is referred to as *10-fold cross-validation*. It is a computationally attractive procedure insofar as it does not require us to change the basic estimation algorithm. Nonetheless, computation can be costly.

Leave-one-out Estimator: At the extreme end of cross-validation we could choose $n = m$. That is, we only remove a single observation at a time and use the remainder of the data for the estimate. Using the average over the likelihood scores provides us with an even more fine-grained estimate. Denote by $p_i(x)$ the density estimate obtained by using $\mathbf{X} := \{x_1, \ldots, x_m\}$ without x_i. For a Parzen windows estimate this is given by

$$p_i(x_i) = (m-1)^{-1} \sum_{j \neq i} r^{-d} h\left(\tfrac{x_i - x_j}{r}\right) = \tfrac{m}{m-1} \left[p(x_i) - r^{-d} h(0)\right]. \tag{1.34}$$

Note that this is precisely the term $r^{-d}h(0)$ that is removed from the estimate. It is this term which led to divergent estimates for $r \to 0$. This means that the leave-one-out log-likelihood estimate can be computed easily via

$$L(\mathbf{X}) = m \log \tfrac{m}{m-1} + \sum_{i=1}^{m} \log \left[p(x_i) - r^{-d} h(0)\right]. \tag{1.35}$$

We then choose r such that $L(\mathbf{X})$ is maximized. This strategy is very robust and whenever it can be implemented in a computationally efficient manner, it is very reliable in performing model selection.

An alternative, probably more of theoretical interest, is to choose the scale r *a priori* based on the amount of data we have at our disposition. Intuitively, we need a scheme which ensures that $r \to 0$ as the number of observations increases $m \to \infty$. However, we need to ensure that this happens slowly enough that the number of observations within range r keeps on increasing in order to ensure good statistical performance. For details we refer the reader to [Sil86]. Chapter **??** discusses issues of model selection for estimators in general in considerably more detail.

1.2.4 Silverman's Rule

Assume you are an aspiring demographer who wishes to estimate the population density of a country, say Australia. You might have access to a limited census which, for a random portion of the population determines where they live. As a consequence you will obtain a relatively high number of samples of city dwellers, whereas the number of people living in the countryside is likely to be very small.

If we attempt to perform density estimation using Parzen windows, we will encounter an interesting dilemma: in regions of high density (i.e. the cities) we will want to choose a narrow kernel width to allow us to model the variations in population density accurately. Conversely, in the outback, a very wide kernel is preferable, since the population there is very low. Unfortunately, this information is exactly what a density estimator itself could tell us. In other words we have a chicken and egg situation where having a good density estimate seems to be necessary to come up with a good density estimate.

Fortunately this situation can be addressed by realizing that we do not actually need to know the *density* but rather a rough estimate of the latter. This can be obtained by using information about the average distance of the k nearest neighbors of a point. One of Silverman's rules of thumb [Sil86] is to choose r_i as

$$r_i = \frac{c}{k} \sum_{x \in kNN(x_i)} \|x - x_i\| . \tag{1.36}$$

Typically c is chosen to be 0.5 and k is small, e.g. $k = 9$ to ensure that the estimate is computationally efficient. The density estimate is then given by

$$p(x) = \frac{1}{m} \sum_{i=1}^{m} r_i^{-d} h\left(\tfrac{x - x_i}{r_i}\right) . \tag{1.37}$$

Figure 1.8 shows an example of such a density estimate. It is clear that a locality dependent kernel width is better than choosing a uniformly constant kernel density estimate. However, note that this increases the computational complexity of performing a density estimate, since first the k nearest neighbors need to be found before the density estimate can be carried out.

1.2.5 Watson-Nadaraya Estimator

Now that we are able to perform density estimation we may use it to perform classification and regression. This leads us to an effective method for non-parametric data analysis, the Watson-Nadaraya estimator [Wat64, Nad65].

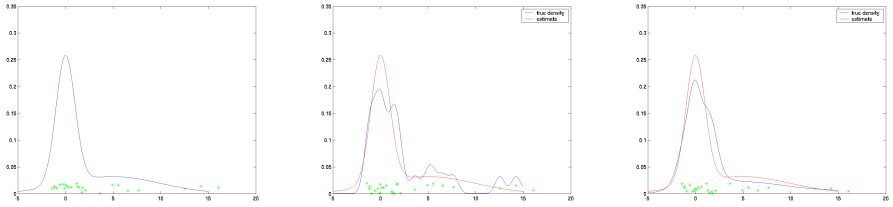

Fig. 1.8. Nonuniform density. Left: original density with samples drawn from the distribution. Middle: density estimate with a uniform kernel. Right: density estimate using Silverman's adjustment.

The basic idea is very simple: assume that we have a binary classification problem, i.e. we need to distinguish between two classes. Provided that we are able to compute density estimates $p(x)$ given a set of observations $\mathbf{X}$ we could appeal to Bayes rule to obtain

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{\frac{m_y}{m} \cdot \frac{1}{m_y} \sum_{i:y_i=y} r^{-d} h\left(\frac{x_i - x}{r}\right)}{\frac{1}{m} \sum_{i=1}^{m} r^{-d} h\left(\frac{x_i - x}{r}\right)}. \tag{1.38}$$

Here we only take the sum over all x_i with label $y_i = y$ in the numerator. The advantage of this approach is that it is very cheap to design such an estimator. After all, we only need to compute sums. The downside, similar to that of the k-nearest neighbor classifier is that it may require sums (or search) over a large number of observations. That is, evaluation of (1.38) is potentially an $O(m)$ operation. Fast tree based representations can be used to accelerate this [BKL06, KM00], however their behavior depends significantly on the dimensionality of the data. We will encounter computationally more attractive methods at a later stage.

For binary classification (1.38) can be simplified considerably. Assume that $y \in \{\pm 1\}$. For $p(y = 1|x) > 0.5$ we will choose that we should estimate $y = 1$ and in the converse case we would estimate $y = -1$. Taking the difference between twice the numerator and the denominator we can see that the function

$$f(x) = \frac{\sum_i y_i h\left(\frac{x_i - x}{r}\right)}{\sum_i h\left(\frac{x_i - x}{r}\right)} = \sum_i y_i \frac{h\left(\frac{x_i - x}{r}\right)}{\sum_i h\left(\frac{x_i - x}{r}\right)} =: \sum_i y_i w_i(x) \tag{1.39}$$

can be used to achieve the same goal since $f(x) > 0 \iff p(y = 1|x) > 0.5$. Note that $f(x)$ is a weighted combination of the labels y_i associated with weights $w_i(x)$ which depend on the proximity of x to an observation x_i. In other words, (1.39) is a smoothed-out version of the k-nearest neighbor classifier of Section **??**. Instead of drawing a hard boundary at the k closest

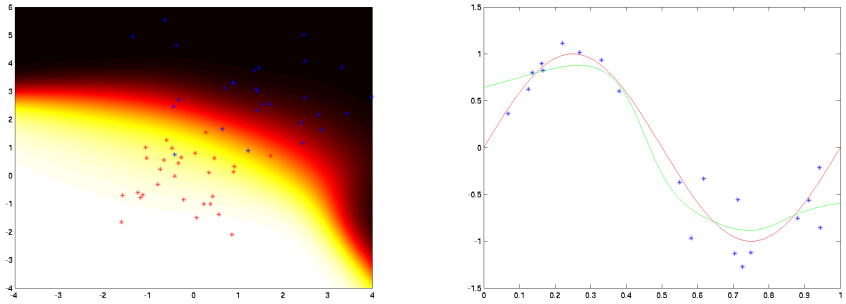

Fig. 1.9. Watson Nadaraya estimate. Left: a binary classifier. The optimal solution would be a straight line since both classes were drawn from a normal distribution with the same variance. Right: a regression estimator. The data was generated from a sinusoid with additive noise. The regression tracks the sinusoid reasonably well.

observation we use a soft weighting scheme with weights $w_i(x)$ depending on which observations are closest.

Note furthermore that the numerator of (1.39) is very similar to the simple classifier of Section **??**. In fact, for kernels $k(x, x')$ such as the Gaussian RBF kernel, which are also kernels in the sense of a Parzen windows density estimate, i.e. $k(x, x') = r^{-d}h\left(\frac{x-x'}{r}\right)$ the two terms are identical. This means that the Watson Nadaraya estimator provides us with an alternative explanation as to why (**??**) leads to a usable classifier.

In the same fashion as the Watson Nadaraya classifier extends the k-nearest neighbor classifier we also may construct a Watson Nadaraya regression estimator by replacing the binary labels y_i by real-valued values $y_i \in \mathbb{R}$ to obtain the regression estimator $\sum_i y_i w_i(x)$. Figure 1.9 has an example of the workings of both a regression estimator and a classifier. They are easy to use and they work well for moderately dimensional data.

1.3 Exponential Families

Distributions from the exponential family are some of the most versatile tools for statistical inference. Gaussians, Poisson, Gamma and Wishart distributions all form part of the exponential family. They play a key role in dealing with graphical models, classification, regression and conditional random fields which we will encounter in later parts of this book. Some of the reasons for their popularity are that they lead to convex optimization problems and that they allow us to describe probability distributions by linear models.

1.3.1 Basics

Densities from the exponential family are defined by

$$p(x;\theta) := p_0(x)\exp\left(\langle\phi(x),\theta\rangle - g(\theta)\right). \tag{1.40}$$

Here $p_0(x)$ is a density on $\mathcal{X}$ and is often called the base measure, $\phi(x)$ is a map from x to the sufficient statistics $\phi(x)$. θ is commonly referred to as the *natural* parameter. It lives in the space dual to $\phi(x)$. Moreover, $g(\theta)$ is a normalization constant which ensures that $p(x)$ is properly normalized. g is often referred to as the log-partition function. The name stems from physics where $Z = e^{g(\theta)}$ denotes the number of states of a physical ensemble. g can be computed as follows:

$$g(\theta) = \log \int_{\mathcal{X}} \exp\left(\langle\phi(x),\theta\rangle\right) dx. \tag{1.41}$$

Example 1.5 (Binary Model) *Assume that $\mathcal{X} = \{0;1\}$ and that $\phi(x) = x$. In this case we have $g(\theta) = \log\left[e^0 + e^\theta\right] = \log\left[1 + e^\theta\right]$. It follows that $p(x=0;\theta) = \frac{1}{1+e^\theta}$ and $p(x=1;\theta) = \frac{e^\theta}{1+e^\theta}$. In other words, by choosing different values of θ one can recover different Bernoulli distributions.*

One of the convenient properties of exponential families is that the log-partition function g can be used to generate moments of the distribution itself simply by taking derivatives.

Theorem 1.14 (Log partition function) *The function $g(\theta)$ is convex. Moreover, the distribution $p(x;\theta)$ satisfies*

$$\nabla_\theta g(\theta) = \mathbf{E}_x\left[\phi(x)\right] \text{ and } \nabla^2_\theta g(\theta) = \mathrm{Var}_x\left[\phi(x)\right]. \tag{1.42}$$

Proof Note that $\nabla^2_\theta g(\theta) = \mathrm{Var}_x\left[\phi(x)\right]$ implies that g is convex, since the covariance matrix is positive semidefinite. To show (1.42) we expand

$$\nabla_\theta g(\theta) = \frac{\int_{\mathcal{X}} \phi(x)\exp\langle\phi(x),\theta\rangle\, dx}{\int_{\mathcal{X}} \exp\langle\phi(x),\theta\rangle} = \int \phi(x)p(x;\theta)dx = \mathbf{E}_x\left[\phi(x)\right]. \tag{1.43}$$

Next we take the second derivative to obtain

$$\nabla^2_\theta g(\theta) = \int_{\mathcal{X}} \phi(x)\left[\phi(x) - \nabla_\theta g(\theta)\right]^\top p(x;\theta)dx \tag{1.44}$$

$$= \mathbf{E}_x\left[\phi(x)\phi(x)^\top\right] - \mathbf{E}_x\left[\phi(x)\right]\mathbf{E}_x\left[\phi(x)\right]^\top \tag{1.45}$$

which proves the claim. For the first equality we used (1.43). For the second line we used the definition of the variance. ■

One may show that higher derivatives $\nabla_\theta^n g(\theta)$ generate higher order cumulants of $\phi(x)$ under $p(x;\theta)$. This is why g is often also referred as the cumulant-generating function. Note that in general, computation of $g(\theta)$ is nontrivial since it involves solving a highdimensional integral. For many cases, in fact, the computation is NP hard, for instance when $\mathcal{X}$ is the domain of permutations [FJ95]. Throughout the book we will discuss a number of approximation techniques which can be applied in such a case.

Let us briefly illustrate (1.43) using the binary model of Example 1.5. We have that $\nabla_\theta = \frac{e^\theta}{1+e^\theta}$ and $\nabla_\theta^2 = \frac{e^\theta}{(1+e^\theta)^2}$. This is exactly what we would have obtained from direct computation of the mean $p(x=1;\theta)$ and variance $p(x=1;\theta) - p(x=1;\theta)^2$ subject to the distribution $p(x;\theta)$.

1.3.2 Examples

A large number of densities are members of the exponential family. Note, however, that in statistics it is not common to express them in the dot product formulation for historic reasons and for reasons of notational compactness. We discuss a number of common densities below and show why they can be written in terms of an exponential family. A detailed description of the most commonly occurring types are given in a table.

Gaussian Let $x, \mu \in \mathbb{R}^d$ and let $\Sigma \in \mathbb{R}^{d\times d}$ where $\Sigma \succ 0$, that is, Σ is a positive definite matrix. In this case the normal distribution can be expressed via

$$\begin{aligned} p(x) &= (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \\ &= \exp\left(x^\top \left[\Sigma^{-1}\mu\right] + \operatorname{tr}\left(\left[-\frac{1}{2}xx^\top\right]\left[\Sigma^{-1}\right]\right) - c(\mu,\Sigma)\right) \end{aligned} \tag{1.46}$$

where $c(\mu,\Sigma) = \frac{1}{2}\mu^\top\Sigma^{-1}\mu + \frac{d}{2}\log 2\pi + \frac{1}{2}\log|\Sigma|$. By combining the terms in x into $\phi(x) := (x, -\frac{1}{2}xx^\top)$ we obtain the sufficient statistics of x. The corresponding linear coefficients $(\Sigma^{-1}\mu, \Sigma^{-1})$ constitute the natural parameter θ. All that remains to be done to express $p(x)$ in terms of (1.40) is to rewrite $g(\theta)$ in terms of $c(\mu,\Sigma)$. The summary table on the following page contains details.

Multinomial Another popular distribution is one over k discrete events. In this case $\mathcal{X} = \{1,\ldots,k\}$ and we have in completely generic terms $p(x) = \pi_x$ where $\pi_x \geq 0$ and $\sum_x \pi_x = 1$. Now denote by $e_x \in \mathbb{R}^k$ the x-th unit vector of the canonical basis, that is $\langle e_x, e_{x'}\rangle = 1$ if $x = x'$

and 0 otherwise. In this case we may rewrite $p(x)$ via

$$p(x) = \pi_x = \exp\left(\langle e_x, \log \pi \rangle\right) \tag{1.47}$$

where $\log \pi = (\log \pi_1, \ldots, \log \pi_k)$. In other words, we have succeeded in rewriting the distribution as a member of the exponential family where $\phi(x) = e_x$ and where $\theta = \log \pi$. Note that in this definition θ is restricted to a $k-1$ dimensional manifold (the k dimensional probability simplex). If we relax those constraints we need to ensure that $p(x)$ remains normalized. Details are given in the summary table.

Poisson This distribution is often used to model distributions over discrete events. For instance, the number of raindrops which fall on a given surface area in a given amount of time, the number of stars in a given volume of space, or the number of Prussian soldiers killed by horse-kicks in the Prussian cavalry all follow this distribution. It is given by

$$p(x) = \frac{e^{-\lambda}\lambda^x}{x!} = \frac{1}{x!}\exp\left(x \log \lambda - \lambda\right) \text{ where } x \in \mathbb{N}_0\,. \tag{1.48}$$

By defining $\phi(x) = x$ we obtain an exponential families model. Note that things are a bit less trivial here since $\frac{1}{x!}$ is the nonuniform counting *measure* on $\mathbb{N}_0$. The case of the uniform measure which leads to the exponential distribution is discussed in Problem 1.16.

The reason why many discrete processes follow the Poisson distribution is that it can be seen as the limit over the average of a large number of Bernoulli draws: denote by $z \in \{0, 1\}$ a random variable with $p(z = 1) = \alpha$. Moreover, denote by z_n the sum over n draws from this random variable. In this case z_n follows the multinomial distribution with $p(z_n = k) = \binom{n}{k}\alpha^k(1-\alpha)^{n-k}$. Now assume that we let $n \to \infty$ such that the expected value of z_n remains constant. That is, we rescale $\alpha = \frac{\lambda}{n}$. In this case we have

$$\begin{aligned} p(z_n = k) &= \frac{n!}{(n-k)!k!}\frac{\lambda^k}{n^k}\left(1 - \frac{\lambda}{n}\right)^{n-k} \\ &= \frac{\lambda^k}{k!}\left(1 - \frac{\lambda}{n}\right)^n \left[\frac{n!}{n^k(n-k)!}\left(1 - \frac{\lambda}{n}\right)^k\right] \end{aligned} \tag{1.49}$$

For $n \to \infty$ the second term converges to $e^{-\lambda}$. The third term converges to 1, since we have a product of only $2k$ terms, each of which converge to 1. Using the exponential families notation we may check that $\mathbf{E}[x] = \lambda$ and that moreover also $\mathrm{Var}[x] = \lambda$.

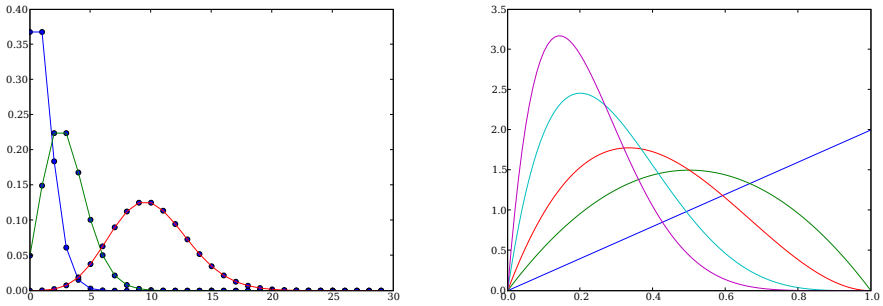

Fig. 1.10. Left: Poisson distributions with $\lambda = \{1, 3, 10\}$. Right: Beta distributions with $a = 2$ and $b \in \{1, 2, 3, 5, 7\}$. Note how with increasing b the distribution becomes more peaked close to the origin.

Beta This is a distribution on the unit interval $\mathcal{X} = [0, 1]$ which is very versatile when it comes to modelling unimodal and bimodal distributions. It is given by

$$p(x) = x^{a-1}(1-x)^{b-1}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}. \tag{1.50}$$

Taking logarithms we see that this, too, is an exponential families distribution, since $p(x) = \exp((a-1)\log x + (b-1)\log(1-x) + \log\Gamma(a+b) - \log\Gamma(a) - \log\Gamma(b))$.

Figure 1.10 has a graphical description of the Poisson distribution and the Beta distribution. For a more comprehensive list of exponential family distributions see the table below and [Fel71, FT94, MN83]. In principle any map $\phi(x)$, domain $\mathcal{X}$ with underlying measure μ are suitable, as long as the log-partition function $g(\theta)$ can be computed efficiently.

Theorem 1.15 (Convex feasible domain) *The domain of definition Θ of $g(\theta)$ is convex.*

Proof By construction g is convex and differentiable everywhere. Hence the below-sets for all values c with $\{x|g(x) \leq c\}$ exist. Consequently the domain of definition is convex. ■

Having a convex function is very valuable when it comes to parameter inference since convex minimization problems have unique minimum values and global minima. We will discuss this notion in more detail when designing maximum likelihood estimators.

Name	Domain $\mathcal{X}$	Measure	$\phi(x)$	$g(\theta)$	Domain Θ
Bernoulli	$\{0,1\}$	Counting	x	$\log\left(1+e^{\theta}\right)$	$\mathbb{R}$
Multinomial	$\{1..N\}$	Counting	e_x	$\log\sum_{i=1}^{N} e^{\theta_i}$	$\mathbb{R}^N$
Exponential	$\mathbb{N}_0^+$	Counting	x	$-\log\left(1-e^{\theta}\right)$	$(-\infty,0)$
Poisson	$\mathbb{N}_0^+$	$\frac{1}{x!}$	x	e^{θ}	$\mathbb{R}$
Laplace	$[0,\infty)$	Lebesgue	x	$\log\theta$	$(-\infty,0)$
Gaussian	$\mathbb{R}$	Lebesgue	$\left(x,-\frac{1}{2}x^2\right)$	$\frac{1}{2}\log 2\pi-\frac{1}{2}\log\theta_2+\frac{1}{2}\frac{\theta_1^2}{\theta_2}$	$\mathbb{R}\times(0,\infty)$
	$\mathbb{R}^n$	Lebesgue	$\left(x,-\frac{1}{2}xx^{\top}\right)$	$\frac{n}{2}\log 2\pi-\frac{1}{2}\log\lvert\theta_2\rvert+\frac{1}{2}\theta_1^{\top}\theta_2^{-1}\theta_1$	$\mathbb{R}^n\times\mathfrak{C}_n$
Inverse Normal	$[0,\infty)$	$x^{-\frac{3}{2}}$	$\left(-x,-\frac{1}{x}\right)$	$\frac{1}{2}\log\pi-2\sqrt{\theta_1\theta_2}-\frac{1}{2}\log\theta_2$	$(0,\infty)^2$
Beta	$[0,1]$	$\frac{1}{x(1-x)}$	$(\log x,\log(1-x))$	$\log\frac{\Gamma(\theta_1)\Gamma(\theta_2)}{\Gamma(\theta_1+\theta_2)}$	$\mathbb{R}^2$
Gamma	$[0,\infty)$	$\frac{1}{x}$	$(\log x,-x)$	$\log\Gamma(\theta_1)-\theta_1\log\theta_2$	$(0,\infty)^2$
Wishart	$\mathfrak{C}_n$	$\lvert X\rvert^{-\frac{n+1}{2}}$	$\left(\log\lvert x\rvert,-\frac{1}{2}x\right)$	$-\theta_1\log\lvert\theta_2\rvert+\theta_1 n\log 2$ $+\sum_{i=1}^{n}\log\Gamma\left(\theta_1+\frac{1-i}{2}\right)$	$\mathbb{R}\times\mathfrak{C}_n$
Dirichlet	S_n	$\left(\prod_{i=1}^{n}x_i\right)^{-1}$	$(\log x_1,\ldots,\log x_n)$	$\sum_{i=1}^{n}\log\Gamma(\theta_i)-\log\Gamma\left(\sum_{i=1}^{n}\theta_i\right)$	$(\mathbb{R}^+)^n$
Inverse χ^2	$\mathbb{R}^+$	$e^{-\frac{1}{2x}}$	$-\log x$	$(\theta-1)\log 2+\log(\theta-1)$	$(0,\infty)$
Logarithmic	$\mathbb{N}$	$\frac{1}{x}$	x	$\log(-\log(1-e^{\theta}))$	$(-\infty,0)$
Conjugate	Θ	Lebesgue	$(\theta,-g(\theta))$	generic	

$\mathfrak{S}_n$ denotes the probability simplex in n dimensions. $\mathfrak{C}_n$ is the cone of positive semidefinite matrices in $\mathbb{R}^{n\times n}$.

1.4 Estimation

In many statistical problems the challenge is to estimate parameters of interest. For instance, in the context of exponential families, we may want to estimate a parameter $\hat{\theta}$ such that it is close to the "true" parameter θ^* in the distribution. While the problem is fully general, we will describe the relevant steps in obtaining estimates for the special case of the exponential family. This is done for two reasons — firstly, exponential families are an important special case and we will encounter slightly more complex variants on the reasoning in later chapters of the book. Secondly, they are of a sufficiently simple form that we are able to show a *range* of different techniques. In more advanced applications only a small subset of those methods may be practically feasible. Hence exponential families provide us with a working example based on which we can compare the consequences of a number of different techniques.

1.4.1 Maximum Likelihood Estimation

Whenever we have a distribution $p(x;\theta)$ parametrized by some parameter θ we may use data to find a value of θ which maximizes the *likelihood* that the data would have been generated by a distribution with this choice of parameter.

For instance, assume that we observe a set of temperature measurements $\mathbf{X} = \{x_1, \ldots, x_m\}$. In this case, we could try finding a normal distribution such that the likelihood $p(\mathbf{X};\theta)$ of the data under the assumption of a normal distribution is maximized. Note that this does *not* imply in any way that the temperature measurements are actually drawn from a normal distribution. Instead, it means that we are attempting to find the Gaussian which fits the data in the best fashion.

While this distinction may appear subtle, it is critical: we do *not* assume that our model accurately reflects reality. Instead, we simply try doing the best possible job at modeling the data given a specified model class. Later we will encounter alternative approaches at estimation, namely Bayesian methods, which *make* the assumption that our model ought to be able to describe the data accurately.

Definition 1.16 (Maximum Likelihood Estimator) *For a model $p(\cdot;\theta)$ parametrized by θ and observations $\mathbf{X}$ the maximum likelihood estimator (MLE) is*

$$\hat{\theta}_{\mathrm{ML}}[\mathbf{X}] := \operatorname*{argmax}_{\theta} p(\mathbf{X};\theta). \tag{1.51}$$

In the context of exponential families this leads to the following procedure: given m observations drawn iid from some distribution, we can express the joint likelihood as

$$p(\mathbf{X};\theta) = \prod_{i=1}^{m} p(x_i;\theta) = \prod_{i=1}^{m} \exp\left(\langle \phi(x_i), \theta\rangle - g(\theta)\right) \tag{1.52}$$

$$= \exp\left(m\left(\langle \mu[\mathbf{X}], \theta\rangle - g(\theta)\right)\right) \tag{1.53}$$

$$\text{where } \mu[\mathbf{X}] := \frac{1}{m}\sum_{i=1}^{m} \phi(x_i). \tag{1.54}$$

Here $\mu[\mathbf{X}]$ is the empirical average of the map $\phi(x)$. Maximization of $p(\mathbf{X};\theta)$ is equivalent to minimizing the negative log-likelihood $-\log p(\mathbf{X};\theta)$. The latter is a common practical choice since for independently drawn data, the product of probabilities decomposes into the sum of the logarithms of individual likelihoods. This leads to the following objective function to be minimized

$$-\log p(\mathbf{X};\theta) = m\left[g(\theta) - \langle \theta, \mu[\mathbf{X}]\rangle\right] \tag{1.55}$$

Since $g(\theta)$ is convex and $\langle \theta, \mu[\mathbf{X}]\rangle$ is linear in θ, it follows that minimization of (1.55) is a convex optimization problem. Using Theorem 1.14 and the first order optimality condition $\nabla_\theta g(\theta) = \mu[\mathbf{X}]$ for (1.55) implies that

$$\theta = \left[\nabla_\theta g\right]^{-1}\left(\mu[\mathbf{X}]\right) \text{ or equivalently } \mathbf{E}_{x\sim p(x;\theta)}[\phi(x)] = \nabla_\theta g(\theta) = \mu[\mathbf{X}]. \tag{1.56}$$

Put another way, the above conditions state that we aim to find the distribution $p(x;\theta)$ which has the same expected value of $\phi(x)$ as what we observed empirically via $\mu[\mathbf{X}]$. Under very mild technical conditions a solution to (1.56) exists.

In general, (1.56) cannot be solved analytically. In certain special cases, though, this is easily possible. We discuss two such choices in the following: Multinomial and Poisson distributions.

Example 1.6 (Poisson Distribution) *For the Poisson distribution*[1] *where* $p(x;\theta) = \frac{1}{x!}\exp(\theta x - e^\theta)$ *it follows that* $g(\theta) = e^\theta$ *and* $\phi(x) = x$*. This allows*

[1] Often the Poisson distribution is specified using $\lambda := \log\theta$ as its rate parameter. In this case we have $p(x;\lambda) = \lambda^x e^{-\lambda}/x!$ as its parametrization. The advantage of the *natural* parametrization using θ is that we can directly take advantage of the properties of the log-partition function as generating the cumulants of x.

us to solve (1.56) in closed form using

$$\nabla_\theta g(\theta) = e^\theta = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ and hence } \theta = \log \sum_{i=1}^{m} x_i - \log m. \tag{1.57}$$

Example 1.7 (Multinomial Distribution) *For the multinomial distribution the log-partition function is given by* $g(\theta) = \log \sum_{i=1}^{N} e^{\theta_i}$*, hence we have that*

$$\nabla_i g(\theta) = \frac{e^{\theta_i}}{\sum_{j=1}^{N} e^{\theta_j}} = \frac{1}{m}\sum_{j=1}^{m} \{x_j = i\}\,. \tag{1.58}$$

It is easy to check that (1.58) is satisfied for $e^{\theta_i} = \sum_{j=1}^{m} \{x_j = i\}$*. In other words, the MLE for a discrete distribution simply given by the empirical frequencies of occurrence.*

The multinomial setting also exhibits two rather important aspects of exponential families: firstly, choosing $\theta_i = c + \log \sum_{i=1}^{m} \{x_j = i\}$ for any $c \in \mathbb{R}$ will lead to an equivalent distribution. This is the case since the sufficient statistic $\phi(x)$ is not minimal. In our context this means that the coordinates of $\phi(x)$ are linearly dependent — for any x we have that $\sum_j [\phi(x)]_j = 1$, hence we could eliminate one dimension. This is precisely the additional degree of freedom which is reflected in the scaling freedom in θ.

Secondly, for data where some events do not occur at all, the expression $\log \left[\sum_{j=1}^{m} \{x_j = i\}\right] = \log 0$ is ill defined. This is due to the fact that this particular set of counts occurs on the boundary of the convex set within which the natural parameters θ are well defined. We will see how different types of priors can alleviate the issue.

Using the MLE is not without problems. As we saw in Figure 1.1, convergence can be slow, since we are not using any side information. The latter can provide us with problems which are both numerically better conditioned and which show better convergence, *provided that our assumptions are accurate*. Before discussing a Bayesian approach to estimation, let us discuss basic statistical properties of the estimator.

1.4.2 Bias, Variance and Consistency

When designing any estimator $\hat{\theta}(\mathbf{X})$ we would like to obtain a number of desirable properties: in general it should not be biased towards a particular solution unless we have good reason to believe that this solution should be preferred. Instead, we would like the estimator to recover, at least on

average, the "correct" parameter, should it exist. This can be formalized in the notion of an *unbiased* estimator.

Secondly, we would like that, even if no correct parameter can be found, e.g. when we are trying to fit a Gaussian distribution to data which is not normally distributed, that we will converge to the best possible parameter choice as we obtain more data. This is what is understood by *consistency*.

Finally, we would like the estimator to achieve low bias and near-optimal estimates as quickly as possible. The latter is measured by the *efficiency* of an estimator. In this context we will encounter the Cramér-Rao bound which controls the best possible rate at which an estimator can achieve this goal. Figure 1.11 gives a pictorial description.

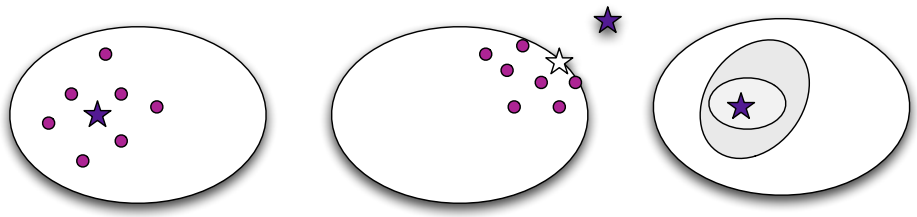

Fig. 1.11. Left: unbiased estimator; the estimates, denoted by circles have as mean the true parameter, as denoted by a star. Middle: consistent estimator. While the true model is not within the class we consider (as denoted by the ellipsoid), the estimates converge to the white star which is the best model within the class that approximates the true model, denoted by the solid star. Right: different estimators have different regions of uncertainty, as made explicit by the ellipses around the true parameter (solid star).

Definition 1.17 (Unbiased Estimator) *An estimator* $\hat{\theta}[\mathbf{X}]$ *is unbiased if for all* θ *where* $\mathbf{X} \sim p(\mathbf{X}; \theta)$ *we have* $\mathbf{E}_{\mathbf{X}}[\hat{\theta}[\mathbf{X}]] = \theta$.

In other words, in expectation the parameter estimate matches the true parameter. Note that this only makes sense if a true parameter actually *exists*. For instance, if the data is Poisson distributed and we attempt modeling it by a Gaussian we will obviously not obtain unbiased estimates.

For finite sample sizes MLE is often *biased*. For instance, for the normal distribution the variance estimates carry bias $O(m^{-1})$. See problem 1.19 for details. In general, under fairly mild conditions, MLE is asymptotically unbiased [DGL96]. We prove this for exponential families. For more general settings the proof depends on the dimensionality and smoothness of the family of densities that we have at our disposition.

Theorem 1.18 (MLE for Exponential Families) *Assume that* $\mathbf{X}$ *is an* m*-sample drawn iid from* $p(x;\theta)$*. The estimate* $\hat{\theta}[\mathbf{X}] = g^{-1}(\mu[\mathbf{X}])$ *is asymptotically normal with*

$$m^{-\frac{1}{2}}[\hat{\theta}[\mathbf{X}] - \theta] \to \mathcal{N}(0, \left[\nabla_\theta^2 g(\theta)\right]^{-1}). \tag{1.59}$$

In other words, the estimate $\hat{\theta}[\mathbf{X}]$ is asymptotically normal, it converges to the true parameter θ, and moreover, the variance at the correct parameter is given by the inverse of the covariance matrix of the data, as given by the second derivative of the log-partition function $\nabla_\theta^2 g(\theta)$.

Proof Denote by $\mu = \nabla_\theta g(\theta)$ the true mean. Moreover, note that $\nabla_\theta^2 g(\theta)$ is the covariance of the data drawn from $p(x;\theta)$. By the central limit theorem (Theorem 1.3) we have that $n^{-\frac{1}{2}}[\mu[\mathbf{X}] - \mu] \to \mathcal{N}(0, \nabla_\theta^2 g(\theta))$.

Now note that $\hat{\theta}[\mathbf{X}] = [\nabla_\theta g]^{-1}(\mu[\mathbf{X}])$. Therefore, by the delta method (Theorem 1.5) we know that $\hat{\theta}[\mathbf{X}]$ is also asymptotically normal. Moreover, by the inverse function theorem the Jacobian of g^{-1} satisfies $\nabla_\mu [\nabla_\theta g]^{-1}(\mu) = \left[\nabla_\theta^2 g(\theta)\right]^{-1}$. Applying Slutsky's theorem (Theorem 1.4) proves the claim. ■

Now that we established the asymptotic properties of the MLE for exponential families it is only natural to ask how much variation one may expect in $\hat{\theta}[\mathbf{X}]$ when performing estimation. The Cramer-Rao bound governs this.

Theorem 1.19 (Cramér and Rao [Rao73]) *Assume that* $\mathbf{X}$ *is drawn from* $p(\mathbf{X};\theta)$ *and let* $\hat{\theta}[\mathbf{X}]$ *be an asymptotically unbiased estimator. Denote by* I *the Fisher information matrix and by* B *the variance of* $\hat{\theta}[\mathbf{X}]$ *where*

$$I := \mathrm{Cov}\left[\nabla_\theta \log p(\mathbf{X};\theta)\right] \text{ and } B := \mathrm{Var}\left[\hat{\theta}[\mathbf{X}]\right]. \tag{1.60}$$

In this case $\det IB \geq 1$ *for all estimators* $\hat{\theta}[\mathbf{X}]$.

Proof We prove the claim for the scalar case. The extension to matrices is straightforward. Using the Cauchy-Schwarz inequality we have

$$\mathrm{Cov}^2\left[\nabla_\theta \log p(\mathbf{X};\theta), \hat{\theta}[\mathbf{X}]\right] \leq \mathrm{Var}\left[\nabla_\theta \log p(\mathbf{X};\theta)\right] \mathrm{Var}\left[\hat{\theta}[\mathbf{X}]\right] = IB. \tag{1.61}$$

Note that at the true parameter the expected log-likelihood score vanishes

$$\mathbf{E}_{\mathbf{X}}[\nabla_\theta \log p(\mathbf{X};\theta)] = \nabla_\theta \int p(\mathbf{X};\theta) d\mathbf{X} = \nabla_\theta 1 = 0. \tag{1.62}$$

Hence we may simplify the covariance formula by dropping the means via

$$\begin{aligned}\operatorname{Cov}\left[\nabla_\theta \log p(\mathbf{X};\theta), \hat{\theta}[\mathbf{X}]\right] &= \mathbf{E}_{\mathbf{X}}\left[\nabla_\theta \log p(\mathbf{X};\theta)\hat{\theta}[\mathbf{X}]\right] \\ &= \int p(\mathbf{X};\theta)\hat{\theta}(\mathbf{X})\nabla_\theta \log p(\mathbf{X};\theta)d\theta \\ &= \nabla_\theta \int p(\mathbf{X};\theta)\hat{\theta}(\mathbf{X})d\mathbf{X} = \nabla_\theta\theta = 1.\end{aligned}$$

Here the last equality follows since we may interchange integration by $\mathbf{X}$ and the derivative with respect to θ. ■

The Cramér-Rao theorem implies that there is a limit to how well we may estimate a parameter given finite amounts of data. It is also a yardstick by which we may measure how efficiently an estimator uses data. Formally, we define the efficiency as the quotient between actual performance and the Cramér-Rao bound via

$$e := 1/\det IB. \tag{1.63}$$

The closer e is to 1, the lower the variance of the corresponding estimator $\hat{\theta}(\mathbf{X})$. Theorem 1.18 implies that for exponential families MLE is asymptotically efficient. It turns out to be generally true.

Theorem 1.20 (Efficiency of MLE [Cra46, GW92, Ber85]) *The maximum likelihood estimator is asymptotically efficient ($e = 1$).*

So far we only discussed the behavior of $\hat{\theta}[\mathbf{X}]$ whenever there *exists* a true θ generating $p(\theta;\mathbf{X})$. If this is not true, we need to settle for less: how well $\hat{\theta}[\mathbf{X}]$ approaches the best possible choice of within the given model class. Such behavior is referred to as consistency. Note that it is not possible to define consistency *per se*. For instance, we may ask whether $\hat{\theta}$ converges to the optimal parameter θ^*, or whether $p(x;\hat{\theta})$ converges to the optimal density $p(x;\theta^*)$, and with respect to which norm. Under fairly general conditions this turns out to be true for finite-dimensional parameters and smoothly parametrized densities. See [DGL96, vdG00] for proofs and further details.

1.4.3 A Bayesian Approach

The analysis of the Maximum Likelihood method might suggest that inference is a solved problem. After all, in the limit, MLE is unbiased and it exhibits as small variance as possible. Empirical results using a *finite* amount of data, as present in Figure 1.1 indicate otherwise.

While not making any assumptions can lead to interesting and general

theorems, it ignores the fact that in practice we almost always have some idea about what to expect of our solution. It would be foolish to ignore such additional information. For instance, when trying to determine the voltage of a battery, it is reasonable to expect a measurement in the order of 1.5V or less. Consequently such *prior* knowledge should be incorporated into the estimation process. In fact, the use of side information to guide estimation turns out to be *the* tool to building estimators which work well in high dimensions.

Recall Bayes' rule (**??**) which states that $p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$. In our context this means that if we are interested in the posterior probability of θ assuming a particular value, we may obtain this using the likelihood (often referred to as evidence) of x having been generated by θ via $p(x|\theta)$ and our prior belief $p(\theta)$ that θ might be chosen in the distribution generating x. Observe the subtle but important difference to MLE: instead of treating θ as a parameter of a density model, we treat θ as an unobserved random variable which we may attempt to infer given the observations $\mathbf{X}$.

This can be done for a number of different purposes: we might want to infer the most likely value of the parameter given the posterior distribution $p(\theta|\mathbf{X})$. This is achieved by

$$\hat{\theta}_{\mathrm{MAP}}(\mathbf{X}) := \operatorname*{argmax}_{\theta} p(\theta|\mathbf{X}) = \operatorname*{argmin}_{\theta} -\log p(\mathbf{X}|\theta) - \log p(\theta). \tag{1.64}$$

The second equality follows since $p(\mathbf{X})$ does not depend on θ. This estimator is also referred to as the *Maximum a Posteriori*, or MAP estimator. It differs from the maximum likelihood estimator by adding the negative log-prior to the optimization problem. For this reason it is sometimes also referred to as Penalized MLE. Effectively we are penalizing unlikely choices θ via $-\log p(\theta)$.

Note that using $\hat{\theta}_{\mathrm{MAP}}(\mathbf{X})$ as the parameter of choice is not quite accurate. After all, we can only infer a distribution over θ and in general there is no guarantee that the posterior is indeed concentrated around its mode. A more accurate treatment is to use the *distribution* $p(\theta|\mathbf{X})$ directly via

$$p(x|\mathbf{X}) = \int p(x|\theta)p(\theta|\mathbf{X})d\theta. \tag{1.65}$$

In other words, we integrate out the unknown parameter θ and obtain the density estimate directly. As we will see, it is generally impossible to solve (1.65) exactly, an important exception being conjugate priors. In the other cases one may resort to sampling from the posterior distribution to approximate the integral.

While it is possible to design a wide variety of prior distributions, this book

focuses on two important families: norm-constrained prior and conjugate priors. We will encounter them throughout, the former sometimes in the guise of regularization and Gaussian Processes, the latter in the context of exchangeable models such as the Dirichlet Process.

Norm-constrained priors take on the form

$$p(\theta) \propto \exp(-\lambda \left\|\theta - \theta_0\right\|_p^d) \text{ for } p, d \geq 1 \text{ and } \lambda > 0. \tag{1.66}$$

That is, they restrict the deviation of the parameter value θ from some guess θ_0. The intuition is that extreme values of θ are much less likely than more moderate choices of θ which will lead to more smooth and even distributions $p(x|\theta)$.

A popular choice is the Gaussian prior which we obtain for $p = d = 1$ and $\lambda = 1/2\sigma^2$. Typically one sets $\theta_0 = 0$ in this case. Note that in (1.66) we did not spell out the normalization of $p(\theta)$ — in the context of MAP estimation this is not needed since it simply becomes a constant offset in the optimization problem (1.64). We have

$$\hat{\theta}_{\text{MAP}}[\mathbf{X}] = \operatorname*{argmin}_{\theta} m\left[g(\theta) - \langle \theta, \mu[\mathbf{X}] \rangle\right] + \lambda \left\|\theta - \theta_0\right\|_p^d \tag{1.67}$$

For $d, p \geq 1$ and $\lambda \geq 0$ the resulting optimization problem is *convex* and it has a unique solution. Moreover, very efficient algorithms exist to solve this problem. We will discuss this in detail in Chapter **??**. Figure 1.12 shows the regions of equal prior probability for a range of different norm-constrained priors.

As can be seen from the diagram, the choice of the norm can have profound consequences on the solution. That said, as we will show in Chapter **??**, the estimate $\hat{\theta}_{\text{MAP}}$ is well concentrated and converges to the optimal solution under fairly general conditions.

An alternative to norm-constrained priors are *conjugate* priors. They are designed such that the posterior $p(\theta|\mathbf{X})$ has the same functional form as the prior $p(\theta)$. In exponential families such priors are defined via

$$p(\theta|n, \nu) = \exp\left(\langle n\nu, \theta \rangle - ng(\theta) - h(\nu, n)\right) \text{ where} \tag{1.68}$$

$$h(\nu, n) = \log \int \exp\left(\langle n\nu, \theta \rangle - ng(\theta)\right) d\theta. \tag{1.69}$$

Note that $p(\theta|n, \nu)$ itself is a member of the exponential family with the feature map $\phi(\theta) = (\theta, -g(\theta))$. Hence $h(\nu, n)$ is *convex* in $(n\nu, n)$. Moreover, the posterior distribution has the form

$$p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta|n, \nu) \propto \exp\left(\langle m\mu[\mathbf{X}] + n\nu, \theta \rangle - (m + n)g(\theta)\right). \tag{1.70}$$

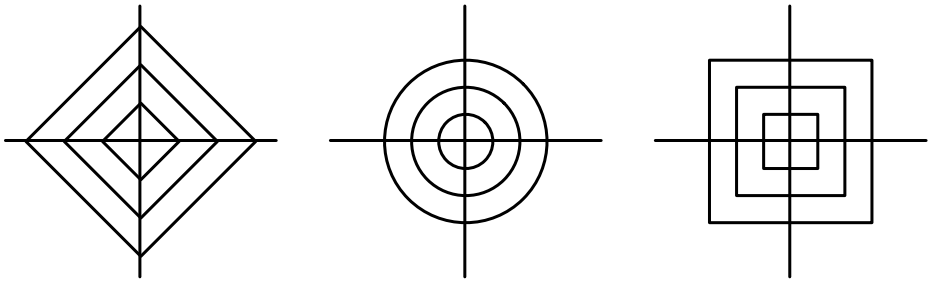

Fig. 1.12. From left to right: regions of equal prior probability in $\mathbb{R}^2$ for priors using the ℓ_1, ℓ_2 and ℓ_∞ norm. Note that only the ℓ_2 norm is invariant with regard to the coordinate system. As we shall see later, the ℓ_1 norm prior leads to solutions where only a small number of coordinates is nonzero.

That is, the posterior distribution has the same form as a conjugate prior with parameters $\frac{m\mu[\mathbf{X}]+n\nu}{m+n}$ and $m+n$. In other words, n acts like a phantom sample size and ν is the corresponding mean parameter. Such an interpretation is reasonable given our desire to design a prior which, when combined with the likelihood remains in the same model class: we treat prior knowledge as having observed virtual data beforehand which is then added to the actual set of observations. In this sense data and prior become completely equivalent — we obtain our knowledge either from actual observations or from virtual observations which describe our belief into how the data generation process is supposed to behave.

Eq. (1.70) has the added benefit of allowing us to provide an exact normalized version of the posterior. Using (1.68) we obtain that

$$p(\theta|\mathbf{X}) = \exp\left(\langle m\mu[\mathbf{X}] + n\nu, \theta\rangle - (m+n)g(\theta) - h\left(\tfrac{m\mu[\mathbf{X}]+n\nu}{m+n}, m+n\right)\right).$$

The main remaining challenge is to compute the normalization h for a range of important conjugate distributions. The table on the following page provides details. Besides attractive algebraic properties, conjugate priors also have a second advantage — the integral (1.65) can be solved exactly:

$$p(x|\mathbf{X}) = \int \exp\left(\langle\phi(x),\theta\rangle - g(\theta)\right) \times \exp\left(\langle m\mu[\mathbf{X}] + n\nu, \theta\rangle - (m+n)g(\theta) - h\left(\tfrac{m\mu[\mathbf{X}]+n\nu}{m+n}, m+n\right)\right) d\theta$$

Combining terms one may check that the integrand amounts to the normal-

ization in the conjugate distribution, albeit $\phi(x)$ added. This yields

$$p(x|\mathbf{X}) = \exp\left(h\left(\frac{m\mu[\mathbf{X}]+n\nu+\phi(x)}{m+n+1}, m+n+1\right) - h\left(\frac{m\mu[\mathbf{X}]+n\nu}{m+n}, m+n\right)\right)$$

Such an expansion is very useful whenever we would like to draw x from $p(x|\mathbf{X})$ without the need to obtain an instantiation of the latent variable θ. We provide explicit expansions in appendix **??**. [GS04] use the fact that θ can be integrated out to obtain what is called a collapsed Gibbs sampler for topic models [BNJ03].

1.4.4 An Example

Assume we would like to build a language model based on available documents. For instance, a linguist might be interested in estimating the frequency of words in Shakespeare's collected works, or one might want to compare the change with respect to a collection of webpages. While models describing documents by treating them as bags of words which all have been obtained independently of each other are exceedingly simple, they are valuable for quick-and-dirty content filtering and categorization, e.g. a spam filter on a mail server or a content filter for webpages.

Hence we model a document d as a multinomial distribution: denote by w_i for $i \in \{1, \ldots, m_d\}$ the words in d. Moreover, denote by $p(w|\theta)$ the probability of occurrence of word w, then under the assumption that the words are independently drawn, we have

$$p(d|\theta) = \prod_{i=1}^{m_d} p(w_i|\theta). \tag{1.71}$$

It is our goal to find parameters θ such that $p(d|\theta)$ is accurate. For a given collection D of documents denote by m_w the number of counts for word w in the entire collection. Moreover, denote by m the total number of words in the entire collection. In this case we have

$$p(D|\theta) = \prod_i p(d_i|\theta) = \prod_w p(w|\theta)^{m_w}. \tag{1.72}$$

Finding suitable parameters θ given D proceeds as follows: In a maximum likelihood model we set

$$p(w|\theta) = \frac{m_w}{m}. \tag{1.73}$$

In other words, we use the empirical frequency of occurrence as our best guess and the sufficient statistic of D is $\phi(w) = e_w$, where e_w denotes the unit vector which is nonzero only for the "coordinate" w. Hence $\mu[D]_w = \frac{m_w}{m}$.

We know that the conjugate prior of the multinomial model is a Dirichlet model. It follows from (1.70) that the posterior mode is obtained by replacing $\mu[D]$ by $\frac{m\mu[D]+n\nu}{m+n}$. Denote by $n_w := \nu_w \cdot n$ the pseudo-counts arising from the conjugate prior with parameters (ν, n). In this case we will estimate the probability of the word w as

$$p(w|\theta) = \frac{m_w + n_w}{m+n} = \frac{m_w + n\nu_w}{m+n}. \tag{1.74}$$

In other words, we add the pseudo counts n_w to the actual word counts m_w. This is particularly useful when the document we are dealing with is brief, that is, whenever we have little data: it is quite unreasonable to infer from a webpage of approximately 1000 words that words not occurring in this page have zero probability. This is exactly what is mitigated by means of the conjugate prior (ν, n).

Finally, let us consider norm-constrained priors of the form (1.66). In this case, the integral required for

$$\begin{aligned} p(D) &= \int p(D|\theta)p(\theta)d\theta \\ &\propto \int \exp\left(-\lambda \left\|\theta - \theta_0\right\|_p^d + m\left\langle \mu[D], \theta\right\rangle - mg(\theta)\right) d\theta \end{aligned}$$

is *intractable* and we need to resort to an approximation. A popular choice is to replace the integral by $p(D|\theta^*)$ where θ^* maximizes the integrand. This is precisely the MAP approximation of (1.64). Hence, in order to perform estimation we need to solve

$$\underset{\theta}{\text{minimize}}\, g(\theta) - \left\langle \mu[D], \theta\right\rangle + \frac{\lambda}{m}\left\|\theta - \theta_0\right\|_p^d. \tag{1.75}$$

A very simple strategy for minimizing (1.75) is gradient descent. That is for a given value of θ we compute the gradient of the objective function and take a fixed step towards its minimum. For simplicity assume that $d = p = 2$ and $\lambda = 1/2\sigma^2$, that is, we assume that θ is normally distributed with variance σ^2 and mean θ_0. The gradient is given by

$$\nabla_\theta\left[-\log p(D,\theta)\right] = \mathbf{E}_{x\sim p(x|\theta)}[\phi(x)] - \mu[D] + \frac{1}{m\sigma^2}[\theta - \theta_0] \tag{1.76}$$

In other words, it depends on the discrepancy between the mean of $\phi(x)$ with respect to our current model and the empirical average $\mu[X]$, and the difference between θ and the prior mean θ_0.

Unfortunately, convergence of the procedure $\theta \leftarrow \theta - \eta\nabla_\theta\left[\ldots\right]$ is usually very slow, even if we adjust the steplength η efficiently. The reason is that the gradient need not point towards the minimum as the space is most likely

distorted. A better strategy is to use Newton's method (see Chapter **??** for a detailed discussion and a convergence proof). It relies on a second order Taylor approximation

$$-\log p(D, \theta + \delta) \approx -\log p(D, \theta) + \langle \delta, G \rangle + \frac{1}{2} \delta^\top H \delta \tag{1.77}$$

where G and H are the first and second derivatives of $-\log p(D, \theta)$ with respect to θ. The quadratic expression can be minimized with respect to δ by choosing $\delta = -H^{-1}G$ and we can fashion an update algorithm from this by letting $\theta \leftarrow \theta - H^{-1}G$. One may show (see Chapter **??**) that Algorithm 1 is quadratically convergent. Note that the prior on θ ensures that H is well conditioned even in the case where the variance of $\phi(x)$ is not. In practice this means that the prior ensures fast convergence of the optimization algorithm.

Algorithm 1 Newton method for MAP estimation

NewtonMAP(D)
  Initialize $\theta = \theta_0$
  while not converged **do**
    Compute $G = \mathbf{E}_{x \sim p(x|\theta)}[\phi(x)] - \mu[D] + \frac{1}{m\sigma^2}[\theta - \theta_0]$
    Compute $H = \mathrm{Var}_{x \sim p(x|\theta)}[\phi(x)] + \frac{1}{m\sigma^2}\,\mathbf{1}$
    Update $\theta \leftarrow \theta - H^{-1}G$
  end while
  return θ

1.5 Sampling

So far we considered the problem of estimating the underlying probability density, given a set of samples drawn from that density. Now let us turn to the converse problem, that is, how to generate random variables given the underlying probability density. In other words, we want to design a random variable generator. This is useful for a number of reasons:

We may encounter probability distributions where optimization over suitable model parameters is essentially impossible and where it is equally impossible to obtain a closed form expression of the distribution. In these cases it may still be possible to perform sampling to draw examples of the kind of data we expect to see from the model. Chapter **??** discusses a number of graphical models where this problem arises.

Secondly, assume that we are interested in testing the performance of a network router under different load conditions. Instead of introducing the under-development router in a live network and wreaking havoc, one could

estimate the probability density of the network traffic under various load conditions and build a model. The behavior of the network can then be simulated by using a probabilistic model. This involves drawing random variables from an estimated probability distribution.

Carrying on, suppose that we generate data packets by sampling and see an anomalous behavior in your router. In order to reproduce and debug this problem one needs access to the same set of random packets which caused the problem in the first place. In other words, it is often convenient if our random variable generator is reproducible; At first blush this seems like a contradiction. After all, our random number generator is supposed to generate random variables. This is less of a contradiction if we consider how random numbers are generated in a computer — given a particular initialization (which typically depends on the state of the system, e.g. time, disk size, bios checksum, etc.) the random number algorithm produces a sequence of numbers which, for all practical purposes, can be treated as iid. A simple method is the linear congruential generator [PTVF94]

$$x_{i+1} = (ax_i + b) \bmod c.$$

The performance of these iterations depends significantly on the choice of the constants a, b, c. For instance, the GNU C compiler uses $a = 1103515245, b = 12345$ and $c = 2^{32}$. In general b and c need to be relatively prime and $a - 1$ needs to be divisible by all prime factors of c and by 4. It is very much advisable *not* to attempt implementing such generators on one's own unless it is absolutely necessary.

Useful desiderata for a pseudo random number generator (PRNG) are that for practical purposes it is statistically indistinguishable from a sequence of iid data. That is, when applying a number of statistical tests, we will accept the null-hypothesis that the random variables are iid. See Chapter **??** for a detailed discussion of statistical testing procedures for random variables. In the following we assume that we have access to a *uniform* RNG $U[0, 1]$ which draws random numbers uniformly from the range $[0, 1]$.

1.5.1 Inverse Transformation

We now consider the scenario where we would like to draw from some distinctively non-uniform distribution. Whenever the latter is relatively simple this can be achieved by applying an inverse transform:

Theorem 1.21 *For $z \sim p(z)$ with $z \in \mathcal{Z}$ and an injective transformation $\phi : \mathcal{Z} \to \mathcal{X}$ with inverse transform ϕ^{-1} on $\phi(\mathcal{Z})$ it follows that the random*

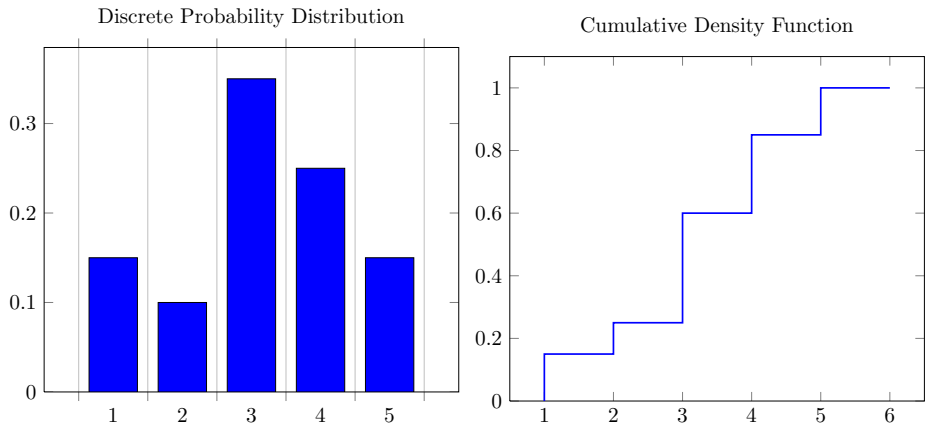

Fig. 1.13. Left: discrete probability distribution over 5 possible outcomes. Right: associated cumulative distribution function. When sampling, we draw x uniformly at random from $U[0,1]$ and compute the inverse of F.

variable $x := \phi(z)$ *is drawn from* $\left|\nabla_x \phi^{-1}(x)\right| \cdot p(\phi^{-1}(x))$. *Here* $\left|\nabla_x \phi^{-1}(x)\right|$ *denotes the determinant of the Jacobian of* ϕ^{-1}.

This follows immediately by applying a variable transformation for a measure, i.e. we change $dp(z)$ to $dp(\phi^{-1}(x))\left|\nabla_x \phi^{-1}(x)\right|$. Such a conversion strategy is particularly useful for univariate distributions.

Corollary 1.22 *Denote by* $p(x)$ *a distribution on* $\mathbb{R}$ *with cumulative distribution function* $F(x') = \int_{-\infty}^{x'} dp(x)$. *Then the transformation* $x = \phi(z) = F^{-1}(z)$ *converts samples* $z \sim U[0,1]$ *to samples drawn from* $p(x)$.

We now apply this strategy to a number of univariate distributions. One of the most common cases is sampling from a discrete distribution.

Example 1.8 (Discrete Distribution) *In the case of a discrete distribution over* $\{1,\ldots,k\}$ *the cumulative distribution function is a step-function with steps at* $\{1,\ldots,k\}$ *where the height of each step is given by the corresponding probability of the event.*

The implementation works as follows: denote by $p \in [0,1]^k$ *the vector of probabilities and denote by* $f \in [0,1]^k$ *with* $f_i = f_{i-1} + p_i$ *and* $f_1 = p_1$ *the steps of the cumulative distribution function. Then for a random variable* z *drawn from* $U[0,1]$ *we obtain* $x = \phi(z) := \operatorname{argmin}_i \{f_i \geq z\}$. *See Figure 1.13 for an example of a distribution over* 5 *events.*

Example 1.9 (Exponential Distribution) *The density of a Exponential-*

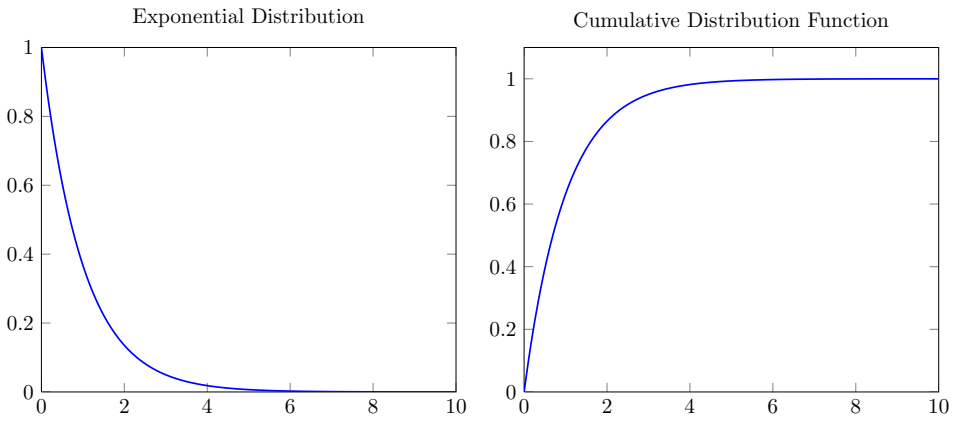

Fig. 1.14. Left: Exponential distribution with $\lambda = 1$. Right: associated cumulative distribution function. When sampling, we draw x uniformly at random from $U[0, 1]$ and compute the inverse.

distributed random variable is given by

$$p(x|\lambda) = \lambda \exp(-\lambda x) \text{ if } \lambda > 0 \text{ and } x \geq 0. \tag{1.78}$$

This allows us to compute its cdf as

$$F(x|\lambda) = 1 - \exp(-\lambda x) \text{if } \lambda > 0 \text{ for } x \geq 0. \tag{1.79}$$

Therefore to generate a Exponential random variable we draw $z \sim U[0, 1]$ and solve $x = \phi(z) = F^{-1}(z|\lambda) = -\lambda^{-1} \log(1 - z)$. Since z and $1 - z$ are drawn from $U[0, 1]$ we can simplify this to $x = -\lambda^{-1} \log z$.

We could apply the same reasoning to the normal distribution in order to draw Gaussian random variables. Unfortunately, the cumulative distribution function of the Gaussian is not available in closed form and we would need resort to rather nontrivial numerical techniques. It turns out that there exists a much more elegant algorithm which has its roots in Gauss' proof of the normalization constant of the Normal distribution. This technique is known as the Box-Müller transform.

Example 1.10 (Box-Müller Transform) *Denote by X, Y independent Gaussian random variables with zero mean and unit variance. We have*

$$p(x, y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)} \tag{1.80}$$

The key observation is that the joint *distribution $p(x, y)$ is radially symmetric, i.e. it only depends on the radius $r^2 = x^2 + y^2$. Hence we may perform*

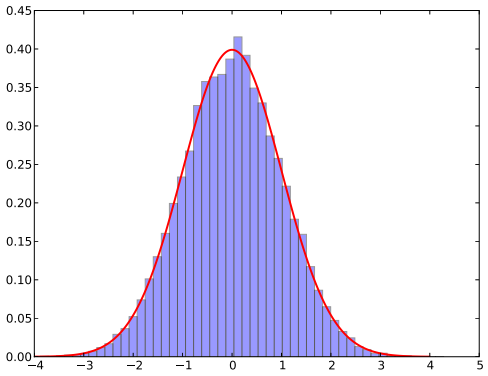

Fig. 1.15. Red: true density of the standard normal distribution (red line) is contrasted with the histogram of 20,000 random variables generated by the Box-Müller transform.

a variable substitution in polar coordinates via the map ϕ *where*

$$x = r\cos\theta \text{ and } y = r\sin\theta \text{ hence } (x, y) = \phi^{-1}(r, \theta). \tag{1.81}$$

This allows us to express the density in terms of (r, θ) *via*

$$p(r,\theta) = p(\phi^{-1}(r,\theta))\left|\nabla_{r,\theta}\phi^{-1}(r,\theta)\right| = \frac{1}{2\pi}e^{-\frac{1}{2}r^2}\left|\begin{bmatrix}\cos\theta & \sin\theta\\ -r\sin\theta & r\cos\theta\end{bmatrix}\right| = \frac{r}{2\pi}e^{-\frac{1}{2}r^2}.$$

The fact that $p(r, \theta)$ *is* constant *in* θ *means that we can easily sample* $\theta \in [0, 2\pi]$ *by drawing a random variable, say* z_θ *from* $U[0, 1]$ *and rescaling it with* 2π*. To obtain a sampler for* r *we need to compute the cumulative distribution function for* $p(r) = re^{-\frac{1}{2}r^2}$*:*

$$F(r') = \int_0^{r'} re^{-\frac{1}{2}r^2}dr = 1 - e^{-\frac{1}{2}r'^2} \text{ and hence } r = F^{-1}(z) = \sqrt{-2\log(1-z)}. \tag{1.82}$$

Observing that $z \sim U[0, 1]$ *implies that* $1 - z \sim U[0, 1]$ *yields the following sampler: draw* $z_\theta, z_r \sim U[0, 1]$ *and compute* x *and* y *by*

$$x = \sqrt{-2\log z_r}\cos 2\pi z_\theta \text{ and } y = \sqrt{-2\log z_r}\sin 2\pi z_\theta.$$

Note that the Box-Müller transform yields two independent *Gaussian random variables. See Figure 1.15 for an example of the sampler.*

Example 1.11 (Uniform distribution on the disc) *A similar strategy*

can be employed when sampling from the unit disc. In this case the closed-form expression of the distribution is simply given by

$$p(x,y) = \begin{cases} \frac{1}{\pi} & \text{if } x^2 + y^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.83}$$

Using the variable transform (1.81) yields

$$p(r,\theta) = p(\phi^{-1}(r,\theta)) \left| \nabla_{r,\theta} \phi^{-1}(r,\theta) \right| = \begin{cases} \frac{r}{\pi} & \text{if } r \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.84}$$

Integrating out θ yields $p(r) = 2r$ for $r \in [0,1]$ with corresponding CDF $F(r) = r^2$ for $r \in [0,1]$. Hence our sampler draws $z_r, z_\theta \sim U[0,1]$ and then computes $x = \sqrt{z_r} \cos 2\pi z_\theta$ and $y = \sqrt{z_r} \sin 2\pi z_\theta$.

1.5.2 Rejection Sampler

All the methods for random variable generation that we looked at so far require intimate knowledge about the pdf of the distribution. We now describe a general purpose method, which can be used to generate samples from an arbitrary distribution. Let us begin with sampling from a set:

Example 1.12 (Rejection Sampler) *Denote by $X \subseteq \mathcal{X}$ a set and let p be a density on $\mathcal{X}$. Then a sampler for drawing from $p_X(x) \propto p(x)$ for $x \in X$ and $p_X(x) = 0$ for $x \notin X$, that is, $p_X(x) = p(x|x \in X)$ is obtained by the procedure:*

repeat
  draw $x \sim p(x)$
until $x \in X$
return x

That is, the algorithm keeps on drawing from p until the random variable is contained in X. The probability that this occurs is clearly $p(X)$. Hence the larger $p(X)$ the higher the efficiency of the sampler. See Figure 1.16.

Example 1.13 (Uniform distribution on a disc) *The procedure works trivially as follows: draw $x, y \sim U[0,1]$. Accept if $(2x-1)^2 + (2y-1)^2 \leq 1$ and return sample $(2x-1, 2y-1)$. This sampler has efficiency $\frac{4}{\pi}$ since this is the surface ratio between the unit square and the unit ball.*

Note that this time we did not need to carry out any sophisticated measure transform. This mathematical convenience came at the expense of a slightly less efficient sampler — about 21% of all samples are rejected.

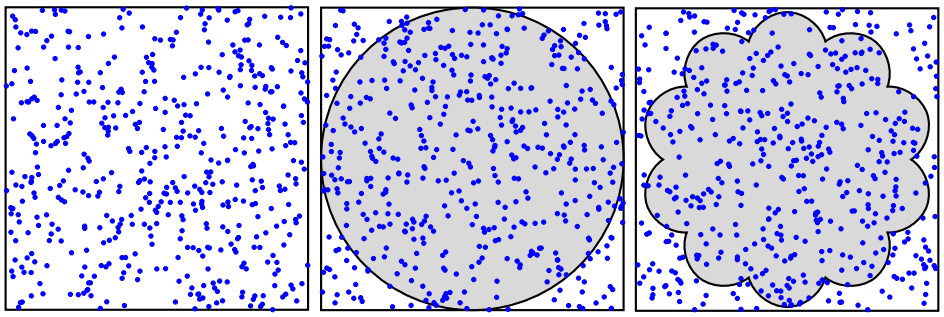

Fig. 1.16. Rejection sampler. Left: samples drawn from the uniform distribution on $[0,1]^2$. Middle: the samples drawn from the uniform distribution on the unit disc are all the points in the grey shaded area. Right: the same procedure allows us to sample uniformly from arbitrary sets.

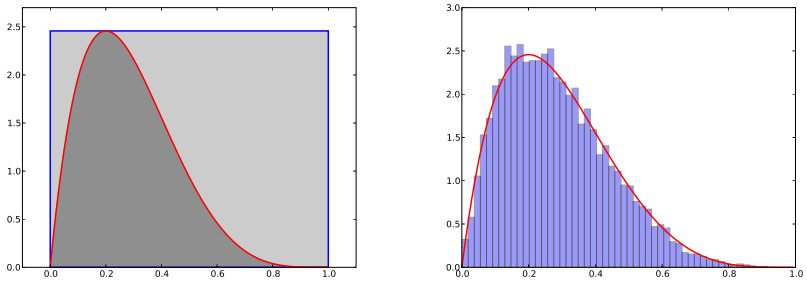

Fig. 1.17. Accept reject sampling for the Beta$(2,5)$ distribution. Left: Samples are generated uniformly from the blue rectangle (shaded area). Only those samples which fall under the red curve of the Beta$(2,5)$ distribution (darkly shaded area) are accepted. Right: The true density of the Beta$(2,5)$ distribution (red line) is contrasted with the histogram of 10,000 samples drawn by the rejection sampler.

The same reasoning that we used to obtain a hard accept/reject procedure can be used for a considerably more sophisticated rejection sampler. The basic idea is that if, for a given distribution p we can find another distribution q which, after rescaling, becomes an upper envelope on p, we can use q to sample from and reject depending on the ratio between q and p.

Theorem 1.23 (Rejection Sampler) *Denote by p and q distributions on $\mathcal{X}$ and let c be a constant such that such that $cq(x) \geq p(x)$ for all $x \in \mathcal{X}$. Then the algorithm below draws from p with acceptance probability c^{-1}.*

repeat
draw $x \sim q(x)$ and $t \sim U[0,1]$

until $ct \leq \frac{p(x)}{q(x)}$
return x

Proof Denote by Z the event that the sample drawn from q is accepted. Then by Bayes rule the probability $\Pr(x|Z)$ can be written as follows

$$\Pr(x|Z) = \frac{\Pr(Z|x)\Pr(x)}{\Pr(Z)} = \frac{\frac{p(x)}{cq(x)} \cdot q(x)}{c^{-1}} = p(x) \tag{1.85}$$

Here we used that $\Pr(Z) = \int \Pr(Z|x)q(x)dx = \int c^{-1}p(x)dx = c^{-1}$. ■

Note that the algorithm of Example 1.12 is a special case of such a rejection sampler — we majorize p_X by the uniform distribution rescaled by $\frac{1}{p(X)}$.

Example 1.14 (Beta distribution) *Recall that the* $\mathrm{Beta}(a, b)$ *distribution, as a member of the Exponential Family with sufficient statistics* $(\log x, \log(1-x))$*, is given by*

$$p(x|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, \tag{1.86}$$

For given (a, b) *one can verify (problem 1.25) that*

$$M := \operatorname*{argmax}_x p(x|a, b) = \frac{a-1}{a+b-2}. \tag{1.87}$$

provided $a > 1$*. Hence, if we use as proposal distribution the uniform distribution* $U[0, 1]$ *with scaling factor* $c = p(M|a, b)$ *we may apply Theorem 1.23. As illustrated in Figure 1.17, to generate a sample from* $\mathrm{Beta}(a, b)$ *we first generate a pair* (x, t)*, uniformly at random from the shaded rectangle. A sample is retained if* $ct \leq p(x|a, b)$*, and rejected otherwise. The acceptance rate of this sampler is* $\frac{1}{c}$*.*

Example 1.15 (Normal distribution) *We may use the Laplace distribution to generate samples from the Normal distribution. That is, we use*

$$q(x|\lambda) = \frac{\lambda}{2} e^{-\lambda|x|} \tag{1.88}$$

as the proposal distribution. For a normal distribution $p = \mathcal{N}(0, 1)$ *with zero mean and unit variance it turns out that choosing* $\lambda = 1$ *yields the most efficient sampling scheme (see Problem 1.27) with*

$$p(x) \leq \sqrt{\frac{2e}{\pi}}\, q(x|\lambda = 1)$$

As illustrated in Figure 1.18, we first generate $x \sim q(x|\lambda = 1)$ *using the inverse transform method (see Example 1.9 and Problem 1.21) and* $t \sim$

$U[0,1]$. *If* $t \leq \sqrt{2e/\pi}p(x)$ *we accept* x*, otherwise we reject it. The efficiency of this scheme is* $\sqrt{\frac{\pi}{2e}}$.

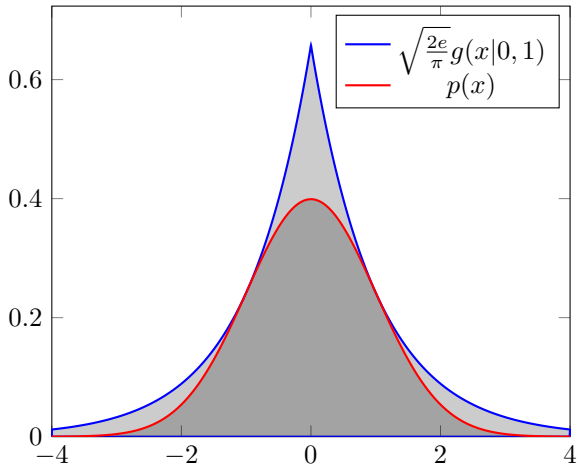

Fig. 1.18. Rejection sampling for the Normal distribution (red curve). Samples are generated uniformly from the Laplace distribution rescaled by $\sqrt{2e/\pi}$. Only those samples which fall under the red curve of the standard normal distribution (darkly shaded area) are accepted.

While rejection sampling is fairly efficient in low dimensions its efficiency is unsatisfactory in high dimensions. This leads us to an instance of the curse of dimensionality [Bel61]: the pdf of a d-dimensional Gaussian random variable centered at 0 with variance $\sigma^2\,\mathbf{1}$ is given by

$$p(x|\sigma^2) = (2\pi)^{-\frac{d}{2}}\sigma^{-d}e^{-\frac{1}{2\sigma^2}\|x\|^2}$$

Now suppose that we want to draw from $p(x|\sigma^2)$ by sampling from another Gaussian q with slightly larger variance $\rho^2 > \sigma^2$. In this case the ratio between both distributions is maximized at 0 and it yields

$$c = \frac{q(0|\sigma^2)}{p(0|\rho^2)} = \left[\frac{\rho}{\sigma}\right]^d$$

If suppose $\frac{\rho}{\sigma} = 1.01$, and $d = 1000$, we find that $c \approx 20960$. In other words, we need to generate approximately 21,000 samples on the average from q to draw a single sample from p. We will discuss a more sophisticated sampling algorithms, namely Gibbs Sampling, in Section **??**. It allows us to draw from rather nontrivial distributions as long as the distributions in small subsets of random variables are simple enough to be tackled directly.

Problems

Problem 1.1 (Bias Variance Decomposition {1}) *Prove that the variance* $\mathrm{Var}_X[x]$ *of a random variable can be written as* $\mathbf{E}_X[x^2] - \mathbf{E}_X[x]^2$.

Problem 1.2 (Moment Generating Function {2}) *Prove that the characteristic function can be used to generate moments as given in (1.12). Hint: use the Taylor expansion of the exponential and apply the differential operator before the expectation.*

Problem 1.3 (Cumulative Error Function {2})

$$\mathrm{erf}(x) = \sqrt{2/\pi} \int_0^x e^{-x^2} dx. \tag{1.89}$$

Problem 1.4 (Weak Law of Large Numbers {2}) *In analogy to the proof of the central limit theorem prove the weak law of large numbers. Hint: use a first order Taylor expansion of* $e^{i\omega t} = 1 + i\omega t + o(t)$ *to compute an approximation of the characteristic function. Next compute the limit* $m \to \infty$ *for* $\phi_{\bar{X}_m}$. *Finally, apply the inverse Fourier transform to associate the constant distribution at the mean* μ *with it.*

Problem 1.5 (Rates and confidence bounds {3}) *Show that the rate of hoeffding is tight — get bound from central limit theorem and compare to the hoeffding rate.*

Problem 1.6 *Why can't we just use each chip on the wafer as a random variable? Give a counterexample. Give bounds if we actually were allowed to do this.*

Problem 1.7 (Union Bound) *Work on many bounds at the same time. We only have logarithmic penalty.*

Problem 1.8 (Randomized Rounding {4}) *Solve the linear system of equations* $Ax = b$ *for integral* x.

Problem 1.9 (Randomized Projections {3}) *Prove that the randomized projections converge.*

Problem 1.10 (The Count-Min Sketch {5}) *Prove the projection trick*

Problem 1.11 (Parzen windows with triangle kernels {1}) *Suppose you are given the following data:* $X = \{2, 3, 3, 5, 5\}$. *Plot the estimated density using a kernel density estimator with the following kernel:*

$$k(u) = \begin{cases} 0.5 - 0.25 * |u| & \text{if } |u| \leq 2 \\ 0 & \text{otherwise.} \end{cases}$$

Problem 1.12 *Gaussian process link with Gaussian prior on natural parameters*

Problem 1.13 *Optimization for Gaussian regularization*

Problem 1.14 *Conjugate prior (student-t and wishart).*

Problem 1.15 (Multivariate Gaussian {1}) *Prove that $\Sigma \succ 0$ is a necessary and sufficient condition for the normal distribution to be well defined.*

Problem 1.16 (Discrete Exponential Distribution {2}) *$\phi(x) = x$ and uniform measure.*

Problem 1.17 *Exponential random graphs.*

Problem 1.18 (Maximum Entropy Distribution) *Show that exponential families arise as the solution of the maximum entropy estimation problem.*

Problem 1.19 (Maximum Likelihood Estimates for Normal Distributions) *Derive the maximum likelihood estimates for a normal distribution, that is, show that they result in*

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ and } \hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu})^2 \tag{1.90}$$

using the exponential families parametrization. Next show that while the mean estimate $\hat{\mu}$ is unbiased, the variance estimate has a slight bias of $O(\frac{1}{m})$. To see this, take the expectation with respect to $\hat{\sigma}^2$.

Problem 1.20 (cdf of Logistic random variable {1}) *Show that the cdf of the Logistic random variable (??) is given by (??).*

Problem 1.21 (Double-exponential (Laplace) distribution {1}) *Use the inverse-transform method to generate a sample from the double-exponential (Laplace) distribution (1.88).*

Problem 1.22 (Normal random variables in polar coordinates {1}) *If X_1 and X_2 are standard normal random variables and let (R, θ) denote the polar coordinates of the pair (X_1, X_2). Show that $R^2 \sim \chi^2_2$ and $\theta \sim \text{Unif}[0, 2\pi]$.*

Problem 1.23 (Monotonically increasing mappings {1}) *A mapping $T : \mathbb{R} \to \mathbb{R}$ is one-to-one if, and only if, T is monotonically increasing, that is, $x > y$ implies that $T(x) > T(y)$.*

Problem 1.24 (Monotonically increasing multi-maps {2}) *Let* $T : \mathbb{R}^n \to \mathbb{R}^n$ *be one-to-one. If* $X \sim p_X(x)$*, then show that the distribution* $p_Y(y)$ *of* $Y = T(X)$ *can be obtained via* (**??**).

Problem 1.25 (Argmax of the $\text{Beta}(a, b)$ **distribution {1})** *Show that the mode of the* $\text{Beta}(a, b)$ *distribution is given by* (1.87).

Problem 1.26 (Accept reject sampling for the unit disk {2}) *Give at least* ***TWO*** *different accept-reject based sampling schemes to generate samples uniformly at random from the unit disk. Compute their efficiency.*

Problem 1.27 (Optimizing Laplace for Standard Normal {1}) *Optimize the ratio* $p(x)/g(x|\mu, \sigma)$*, with respect to* μ *and* σ*, where* $p(x)$ *is the standard normal distribution* (**??**)*, and* $g(x|\mu, \sigma)$ *is the Laplace distribution* (1.88).

Problem 1.28 (Normal Random Variable Generation {2}) *The aim of this problem is to write code to generate standard normal random variables* (**??**) *by using different methods. To do this generate* $U \sim \text{Unif}[0, 1]$ *and apply*

(i) *the Box-Muller transformation outlined in Section* **??**.
(ii) *use the following approximation to the inverse CDF*

$$\Phi^{-1}(\alpha) \approx t - \frac{a_0 + a_1 t}{1 + b_1 t + b_2 t^2}, \tag{1.91}$$

where $t^2 = \log(\alpha^{-2})$ *and*

$$a_0 = 2.30753, a_1 = 0.27061, b_1 = 0.99229, b_2 = 0.04481$$

(iii) *use the method outlined in example 1.15.*

Plot a histogram of the samples you generated to confirm that they are normally distributed. Compare these different methods in terms of the time needed to generate 1000 random variables.

Problem 1.29 (Non-standard Normal random variables {2}) *Describe a scheme based on the Box-Muller transform to generate* d *dimensional normal random variables* $p(x|0, I)$*. How can this be used to generate arbitrary normal random variables* $p(x|\mu, \Sigma)$.

Problem 1.30 (Uniform samples from a disk {2}) *Show how the ideas described in Section* **??** *can be generalized to draw samples uniformly at random from an axis parallel ellipse:* $\{(x, y) : \frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} \leq 1\}$.

Bibliography

[BBL05] O. Bousquet, S. Boucheron, and G. Lugosi, *Theory of classification: a survey of recent advances*, ESAIM: Probab. Stat. **9** (2005), 323– 375.

[BCR84] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic analysis on semigroups*, Springer, New York, 1984.

[Bel61] R. E. Bellman, *Adaptive control processes*, Princeton University Press, Princeton, NJ, 1961.

[Bel05] Alexandre Belloni, *Introduction to bundle methods*, Tech. report, Operation Research Center, M.I.T., 2005.

[Ber85] J. O. Berger, *Statistical decision theory and Bayesian analysis*, Springer, New York, 1985.

[Bil68] Patrick Billingsley, *Convergence of probability measures*, John Wiley and Sons, 1968.

[BKL06] A. Beygelzimer, S. Kakade, and J. Langford, *Cover trees for nearest neighbor*, International Conference on Machine Learning, 2006.

[BNJ03] D. Blei, A. Ng, and M. Jordan, *Latent Dirichlet allocation*, Journal of Machine Learning Research **3** (2003), 993–1022.

[BV04] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, England, 2004.

[Cra46] H. Cramér, *Mathematical methods of statistics*, Princeton University Press, 1946.

[DGL96] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*, Applications of mathematics, vol. 31, Springer, New York, 1996.

[Fel71] W. Feller, *An introduction to probability theory and its applications*, 2 ed., John Wiley and Sons, New York, 1971.

[FJ95] A. Frieze and M. Jerrum, *An analysis of a monte carlo algorithm for estimating the permanent*, Combinatorica **15** (1995), no. 1, 67–83.

[FT94] L. Fahrmeir and G. Tutz, *Multivariate statistical modelling based on generalized linear models*, Springer, 1994.

[GIM99] A. Gionis, P. Indyk, and R. Motwani, *Similarity search in high dimensions via hashing*, Proceedings of the 25th VLDB Conference (Edinburgh, Scotland) (M. P. Atkinson, M. E. Orlowska, P. Valduriez, S. B. Zdonik, and M. L. Brodie, eds.), Morgan Kaufmann, 1999, pp. 518–529.

[GS04] T.L. Griffiths and M. Steyvers, *Finding scientific topics*, Proceedings of the National Academy of Sciences **101** (2004), 5228–5235.

[GW92] P. Groeneboom and J. A. Wellner, *Information bounds and nonparametric maximum likelihood estimation*, DMV, vol. 19, Springer, 1992.

[Hal92] P. Hall, *The bootstrap and edgeworth expansions*, Springer, New York, 1992.

[Hoe63] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, Journal of the American Statistical Association **58** (1963), 13–30.

[HUL93] J.B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms, I and II*, vol. 305 and 306, Springer-Verlag, 1993.

[IM98] P. Indyk and R. Motawani, *Approximate nearest neighbors: Towards remov-*

ing the curse of dimensionality, Proceedings of the 30$^{\text{th}}$ Symposium on Theory of Computing, 1998, pp. 604–613.

[Kel60] J. E. Kelly, *The cutting-plane method for solving convex programs*, Journal of the Society for Industrial and Applied Mathematics **8** (1960), no. 4, 703–712.

[Kiw90] Krzysztof C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable minimization*, Mathematical Programming **46** (1990), 105–122.

[KM00] Paul Komarek and Andrew Moore, *A dynamic adaptation of AD-trees for efficient machine learning on large data sets*, Proc. Intl. Conf. Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 495–502.

[LNN95] Claude Lemaréchal, Arkadii Nemirovskii, and Yurii Nesterov, *New variants of bundle methods*, Mathematical Programming **69** (1995), 111–147.

[LT92] Z. Q. Luo and P. Tseng, *On the convergence of coordinate descent method for convex differentiable minimization*, Journal of Optimization Theory and Applications **72** (1992), no. 1, 7–35.

[Lue84] D. G. Luenberger, *Linear and nonlinear programming*, second ed., Addison-Wesley, Reading, May 1984.

[McD89] C. McDiarmid, *On the method of bounded differences*, Survey in Combinatorics, Cambridge University Press, 1989, pp. 148–188.

[MN83] P. McCullagh and J. A. Nelder, *Generalized linear models*, Chapman and Hall, London, 1983.

[MYA94] N. Murata, S. Yoshizawa, and S. Amari, *Network information criterion — determining the number of hidden units for artificial neural network models*, IEEE Transactions on Neural Networks **5** (1994), 865–872.

[Nad65] E. A. Nadaraya, *On nonparametric estimates of density functions and regression curves*, Theory of Probability and its Applications **10** (1965), 186–190.

[NW99] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer, 1999.

[Pap62] A. Papoulis, *The fourier integral and its applications*, McGraw-Hill, New York, 1962.

[PTVF94] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in c. the art of scientific computation*, Cambridge University Press, Cambridge, UK, 1994.

[Rao73] C. R. Rao, *Linear statistical inference and its applications*, John Wiley and Sons, New York, 1973.

[Rud73] W. Rudin, *Functional analysis*, McGraw-Hill, New York, 1973.

[Sil86] B. W. Silverman, *Density estimation for statistical and data analysis*, Monographs on statistics and applied probability, Chapman and Hall, London, 1986.

[SW86] G.R. Shorack and J.A. Wellner, *Empirical processes with applications to statistics*, Wiley, New York, 1986.

[SZ92] Helga Schramm and Jochem Zowe, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optimization **2** (1992), 121–152.

[Vap82] V. Vapnik, *Estimation of dependences based on empirical data*, Springer, Berlin, 1982.

[Vap98] ———, *Statistical learning theory*, John Wiley and Sons, New York, 1998.

[vdG00] S. van de Geer, *Empirical processes in M-estimation*, Cambridge University Press, 2000.

[vdVW96] A. W. van der Vaart and J. A. Wellner, *Weak convergence and empirical processes*, Springer, 1996.

[Wat64] G. S. Watson, *Smooth regression analysis*, Sankhya A **26** (1964), 359–372.