Solution sketch to problems of 2024 exam

Problem 1: Monte Carlo

Approximation: $\mu_h \approx m$, since we have $E(m) = \mu_h$

Estimate for uncertainty: $\text{Std}(m) \approx \frac{\sqrt{v}}{10}$, or $\text{Var}(m) \approx \frac{v}{100}$

For this to be valid we need to have $\text{Var}_f\big(h(X)\big) < \infty$, or $E_f\ (h(X)^2) < \infty$, where subscript f indicate that the quantities are computed under the distribution of f.

The dimension of x does not impact the result directly, but off course if the dimension of x is high this might influence the variance: $\text{Var}_f\big(h(X)\big)$.

To decide whether the number is sufficient we need to know which accuracy that is required in the answer. The standard deviation above can be used to get a handle on this. If we increase the number of samples the error will go as: $\text{Std}_{\text{B}}(m) = \frac{\sqrt{v}}{\sqrt{B}}$. If noting else is said we could use a rule of the thumb that the standard deviation due to Monte Carlo variability should be a fraction of the standard deviation of $h(X)$, e.g. 5% => 400 samples 1% => 10 000 samples.

Problem 2: EM-Algorithm

Utilizing that the data are independent, we get:

$$L(\theta) = \prod_{i=1}^{T} (1-p)\phi(x_i|\mu,\sigma^2) + pf_c(x_i) \tag{1}$$

For a single observation we have:

$$p(x_i|C_i = 0) = \phi(x_i|\mu,\sigma^2), \qquad p(x_i|C_i = 1) = f_c(x_i)$$

$$p(x_i, C_i) = p(C_i = c)p(x_i|C_i = c) = \prod_{c=0}^{1} [p(C_i = c)p(x_i|C_i = c)]^{I(C_i=c)}$$

Utilizing that $p(C_i = 1) = p$ and $p(C_i = 0) = (1-p)$, and that all data are independent we find that:

$$p(x, C|\theta) = \prod_{i=1}^{T} [(1-p)\phi(x_i|\mu,\sigma^2)]^{I(C_i=0)}[pf_c(x_i)]^{I(C_i=1)}$$

Taking the logarithm and inserting the normal density we get:

$$l(x, C|\theta) =$$

$$\sum_{i=1}^{T} I(C_i = 0) \cdot \frac{1}{2}\left(2\ln(1-p) - \ln 2\pi - \ln\sigma^2 - \frac{(x_i-\mu)^2}{\sigma^2}\right)$$
$$+ I(C_i = 1)(\ln p + \ln f_c(x_i)\,)$$

b) The $Q(\theta|\theta^{(t)})$ function is the expectation of the log-likelihood for the complete data, given the current estimate of the parameters and the observed data.

$$Q(\theta|\theta^{(t)}) = E(l(x,C|\theta)|x,\theta^{(t)})$$

In our situation we have that the only random part in the likelihood of the complete data is the indicator functions, which turn into probabilities for the indicator event:

$$Q(\theta|\theta^{(t)})$$
$$= \sum_{i=1}^{T} P(C_i = 0|x_i,\theta^{(t)}) \cdot \frac{1}{2}(2\ln(1-p) - \ln\phi(x_i|\mu,\sigma^2))$$
$$+ P(C_i = 1|x_i,\theta^{(t)})\,[\ln \mathrm{p} + \ln f_c(x_i)]$$
$$= \sum_{i=1}^{T} P(C_i = 0|x_i,\theta^{(t)}) \cdot \frac{1}{2}\left(2\ln(1-p) - \ln 2\pi - \ln\sigma^2 - \frac{(x_i-\mu)^2}{\sigma^2}\right)$$
$$+ P(C_i = 1|x_i,\theta^{(t)})\,[\ln \mathrm{p} + \ln f_c(x_i)]$$

Where:

$$P(C_i = 1|x_i,\theta^{(t)}) = \frac{P(C_i = 1, x_i|\theta^{(t)})}{p(x_i|\theta^{(t)})} = \frac{pf_c(x_i)}{(1-p)\phi(x_i|\mu,\sigma^2) + pf_c(x_i)},$$

and $P(C_i = 0|x_i,\theta^{(t)}) = 1 - P(C_i = 1|x_i,\theta^{(t)})$

To derive the estimates, we find:

$$\frac{\partial Q(\theta|\theta^{(t)})}{\partial p} = \sum_{i=1}^{T} P(C_i = 0|x_i,\theta^{(t)}) \cdot \left(\frac{1}{1-p}\right) + P(C_i = 1|x_i,\theta^{(t)})\frac{1}{p}$$

$$\frac{\partial Q(\theta|\theta^{(t)})}{\partial p} = 0 \implies p = \frac{1}{T}\sum_{i=1}^{T} P(C_i = 1|x_i,\theta^{(t)})$$

$$\frac{\partial Q(\theta|\theta^{(t)})}{\partial \mu} = \sum_{i=1}^{T} P(C_i = 0|x_i,\theta^{(t)}) \cdot \frac{1}{2}\left(\frac{2(x_i-\mu)}{\sigma^2}\right)$$

$$\frac{\partial Q(\theta|\theta^{(t)})}{\partial \mu} = 0 \implies \mu = \frac{\sum_{i=1}^{T} P(C_i = 0|x_i,\theta^{(t)})x_i}{\sum_{i=1}^{T} P(C_i = 0|x_i,\theta^{(t)})}$$

$$\frac{\partial Q(\theta|\theta^{(t)})}{\partial \sigma^2} = \sum_{i=1}^{T} P(C_i = 0|x_i, \theta^{(t)}) \cdot \frac{1}{2}\left(-\frac{1}{\sigma^2} + \frac{(x_i - \mu)^2}{(\sigma^2)^2}\right)$$

$$\frac{\partial Q(\theta|\theta^{(t)})}{\partial \sigma^2} = 0 \Rightarrow \sigma^2 = \frac{\sum_{i=1}^{T} P(C_i = 0|x_i, \theta^{(t)})(x_i - \mu)^2}{\sum_{i=1}^{T} P(C_i = 0|x_i, \theta^{(t)})}$$

c) In a parametric bootstrap we can quantify the uncertainty by generating new datasets from a parameterized model, where the parameters are estimated from original dataset. In the context of the contamination model above, $\theta = (p, \mu, \sigma^2)$, can be estimated using formulas from b, giving $\hat{\theta} = (\hat{p}, \hat{\mu}, \widehat{\sigma^2})$, and $f_c(x)$ is known.

The algorithm for the parametric bootstrap is then:

1) Repeat B times:

    1) Sample T samples from $N(\hat{\mu}, \widehat{\sigma^2})$, $\boldsymbol{X} = (X_1, X_2, \dots, X_T)$

    2) Sample number of contaminated samples, $N_C \sim \text{binominal}(T, \hat{p})$,

    3) Sample N_C samples from $f_C(x)$, $Y_1, Y_2, \dots, Y_{N_C}$

    4) Replace N_C samples from $\boldsymbol{X}$, $\boldsymbol{X_C} = (Y_1, Y_2, \dots, Y_{N_C}, X_{N_C+1}, \dots, X_T)$

    5) Estimate $\widehat{\mu_k}, \widehat{\sigma_k^2}$, using $\boldsymbol{X}$ and $\widehat{\mu_{k,C}}, \widehat{\sigma_{k,C}^2}$ using $\boldsymbol{X_C}$.

2) Compare the statistics of $\left(\widehat{\mu_k}, \widehat{\sigma_k^2}\right)_{k=1}^{B}$ and $\left(\widehat{\mu_{k,C}}, \widehat{\sigma_{k,C}^2}\right)_{k=1}^{B}$, i.e. compute mean and covariance of the two samples, compute the bias and difference in uncertainty.

Note: by using the algorithm above we utilize common random numbers to reduce the Monte Carlo variability in the comparison due to a finite (B) number of samples. It is possible also to do comparison with two independent samples, or with theoretical results for the Gaussian case.

Problem 3: SGD

It is known from theory that convergence of the algorithm requires the learning rate to approach zero in the limit, at a rate such that:

$$\sum_{i=1}^{\infty} \alpha_t = \infty, \text{and} \sum_{i=1}^{\infty} \alpha_t^2 < \infty$$

Although the first part is fulfilled with a constant learning rate, the second is not. In the figure both cases oscillate randomly around 2. The largest learning rate converges faster but has larger amplitude in the oscillations. This can roughly be interpreted like having a large learning rate makes the first sum grow fast which finds the level fast, but also the second sum grows fast and give a large variability around the true value.

Computations:

$$g(x) = \frac{1}{2}\,(x\ -\ b)^2, \qquad \nabla g(x) = (x - b)$$

Starting with (6) insert expression for z_t, with a constant learning rate, and subtract b from both sides gives:

$$x_{t+1} - b = \ x_t - b\ -\ \alpha\ \cdot (x_t - b) - \alpha\varepsilon_t$$
$$x_{t+1} - b = (1 - \alpha\,)(x_t - b) - \alpha\varepsilon_t$$

Re-inserting this relation for t, we get:

$$x_{t+1} - b = (1 - \alpha\,)\big((1 - \alpha\,)(x_{t-1} - b) - \alpha\varepsilon_{t-1}\big) - \alpha\varepsilon_t$$
$$x_{t+1} - b = (1 - \alpha\,)^2(x_{t-1} - b) - \alpha(1 - \alpha\,)\varepsilon_{t-1} - \alpha\varepsilon_t$$

Which gives the result.

Generalizing this, we see that as we move back k steps in time in time, the first term will evolve as $(1 - \alpha\,)^k\ (x_{t-k+1} - b)$,..., $(1 - \alpha\,)^{t+1}\ (x_0 - b)$. The second term will evolve as:

$$\varepsilon_t^k = \sum_{t=0}^{k-1}(1 - \alpha\,)^k\varepsilon_{t-k}$$

Thus, the variance will be $\mathrm{Var}\big(\varepsilon_t^k\big) = \sum_{t=0}^{k-1}(1 - \alpha\,)^{2k},$ such that:

$$k \to \infty \Rightarrow \mathrm{Var}\big(\varepsilon_t^k\big) \to \frac{1}{1 - (1 - \alpha)^2} = \frac{1}{2\alpha - \alpha^2} \Rightarrow \mathrm{Var}\big(\alpha\ \varepsilon_t^k\big) \to \frac{\alpha}{2 - \alpha} \approx \frac{\alpha}{2}$$

Here we see that for the deterministic part we will have exponential convergence, whereas for the stochastic part the variance will scale with the learning rate. This is the type of behavior observed in the figure, where we see an exponential form initially when the first term is dominating, and random fluctuations in the latter part.

Problem 4: Slice sampler

a) In the Gibbs Sampler, we sample the joint distribution using iterations. In each step we update one variable by sampling from the conditional distribution of this variable given the other variables. The other variables are fixed at their current value in the iterations. There are different versions of the Gibbs sampler, we can do a systematic scan or a random scan.

In an example with two variables (u, x) the algorithm goes:

```
1) Initialize $x = x_0$
2) For $i$ from 1 to $B$:
    a. Sample $u_i$ from $f(u|x_{i-1})$
    b. Sample $x_i$ from $f(x|u_i)$
```

The algorithm below samples from the "gray area" using the Gibbs-sampler, updating $f(u|x_{i-1}) = \mathrm{Unif}(0, f(x_{i-1}\,))$, and $f(x|u_i) = \mathrm{Unif}(\{x\colon f(x) > u\,\})$. The boundaries of the region of latter can be found solving the equation $\phi(L) = u$. Which gives the limits in step 3.

b) In a McMC step using M-H we have that we propose with probability $g(x_p|x)$, and accept the proposal with probability , $w$here:

$$R(x_p|x) = \frac{f(x_p)g(x|x_p)}{f(x)g(x_p|x)}$$

In our case: $f(x) = f(x|u_i) = \text{Unif}(\{x: f(x) > u\}) \propto I(f(x) > u)$, and

$$g(x_p|x) = \begin{cases} \frac{1}{2r} & x - r < x_p < x + r \\ 0 & \text{else} \end{cases}$$

Thus: $R(x_p|x) = \frac{I(f(x_p)>u)}{I(f(x)>u)} \propto I(f(x_p) > u)$. If at some point $f(x) < u$, then the ratio is undefined. However, this will only occur if we accept a proposal x_p for which $f(x_p) > u$, since this probability is zero, we just need to be careful with our starting point. The step in the algorithm becomes:

A) Sample $x_p \sim g(x_p|x)$
B) Accept x_p if $f(x_p) > u$

c) For a McMC to sample from a target distribution we need: The chain to be irreducible, aperiodic, and recurrent for the limit distribution to be the stationary point of the Markov chain. Next we need the transition kernel $p(y|x)$ fulfill:

$$f(y) = \int f(x)p(y|x)dx$$

The chain is aperiodic, since the algorithm as it is constructed will propose points outside the valid region, we have a probability of staying in the same location for one or more steps, and thus avoid cycles. It is not obvious that the chain will not be irreducible/recurrent since the support of $\phi(x)$ is infinite and the support of the proposal distribution is finite.

d) In the table we see that there is a monotone trend for acceptance probabilities which decrease as r increases, whereas for effective sample size and GR statistics there is an extreme point $r = 10$. The values of effective sample size and GR statisticsis preferable for $r = 10$. If we look at the acceptance rate, we would expect the optimal value r to have acceptance rate in the range 0.3-0.5, thus it is likely that an even better value could have been found for r values between 1 and 10. If we look at the sample path, we see that the high acceptance rate for low r values is because there is too high correlation in the sample path. This effect is also seen in the cumsum-statistics which have large values and a smooth appearance for low values of r. We see that the correlation decreases as r increases, but for $r = 100$ the acceptance rate becomes too small, this is seen in the sample path which have long periods of constant values. In addition to the values and plots shown, the corresponding values for u and f(x) would increase the trust in the data. Summary statistics for $f(x, u)$ is not required since this is constant. As seen from answer to 1a) we want more than 400 samples to have a reasonable reduction in the uncertainty. Thus case 3 and 4 are ok.

Problem 5: Simulated annealing

Note it is not expected that all expressions should be understood, (2-opt, TSP, Graph coloring, etc) but comments should be given to those that are in the curriculum. In general, the answer from the LLM generates much text that was not requested, and have a mistake with respect to the effect of restricting the neighborhood. See specific comments below.

Intro: Just a rephrase of question.

Part 1: Text is ok, but it would have been nice to be clearer text e.g. possible solutions vs valid solutions.

Part 2: Text is ok, but it should be clearer that a particular neighborhood is a *choice*. This choice will *define* which cells that are the adjacent ones.

Part 3: In this element b) is not correct, by restricting the neighborhood we increase the number of local optima, i.e. we make the problem worse with respect to "being stuck". By extending size of the neighborhood, we might increase the probability to escape from what would be considered local optima according to a narrower neighborhood definition. In c) this is details that where not relevant to the question asked LLM, but when it is brought up, it would have been nice to have the formula as well.

Part 4: This statement actually gets right what is wrong in 3b.

Part 5: In 5c, this is a too general statement and thus misleading, if we set up SA correct all options are possible, and we can even give guarantee of convergence if the cooling schedule (defined below) is logarithmic. However, given a limited time we only explore a subset of solutions. Which subset that is investigated (within a limited time) is determined by randomness.

Part 6: In 6b, the examples are slightly random.

Summary: It is true that the neighborhood guides the SA, but the neighborhood also creates the local minima, which SA escapes by the randomness.

A direct answer to the question could have been: "In discrete optimization problems, the neighborhood limits the search space for "improved solutions" to a set adjacent to the current solution. This makes the search space manageable in terms of compute time in each iteration, but also creates a set of local minima. A local minimum is such that even though it is impossible to improve the solution in the current neighborhood, there might still exist a is a better solution among all possibilities.

It should also be mentioned missing in (answer by LLM) that the neighborhoods must communicate, such that it is possible to visit any solution in a finite number of steps.

SA for maximization of $J(x)$:

1) Initialize x_0
2) Propose a new value x_p by randomly sampling the neighborhood of x_i
3) Accept the new proposal $(x_{i+1} = x_p)$ with probability $\min(1, \alpha)$, where $\alpha = \exp\left(\left(J(x_p) - J(x_i)\right)/\tau_j\right)$, otherwise keep the old value $(x_{i+1} = x_i)$
4) Update τ_j according to the cooling schedule.
5) repeat from 2)

A cooling schedule is defined by setting a sequence of pairs (τ_j, m_j), such that you keep the level τ_j for m_j iterations, before you update to next τ_j, i.e. use τ_j when $\sum_{k=0}^{j-1} m_k < i < \sum_{k=0}^{j} m_k, m_0 = 0$. In the cooling schedule τ_j converges monotonously towards zero as j increases. i.e. $\tau_1 > \tau_2 > \cdots > 0$.