Pattern Analysis Joint Meeting
Christian Riess

Worksheet 1: Tuesday April 20 / Friday April 23 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 15 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

## Task 1: Values of a PDF

Which two properties must the value range of a probability density function (PDF) satisfy?

$p(x) \geq 0$

$\int_{-\infty}^{\infty} p(x)\,dx = 1$

## Task 2: A Tiny Sampling Pitfall

The lecture states that the presented sampling algorithm can operate on **arbitrary** PDFs. However, there might be a subtle pitfall in the actual **implementation** of the algorithm on very special PDFs.

What could go wrong?

## Task 3: Lecture Information Processing

Let us "calibrate" the lecture content with your perception.

What do you think: which specific pieces of the lecture are relevant for the examination, and which specific roles do the other parts play?

name of probability.

Pattern Analysis Joint Meeting
Christian Riess

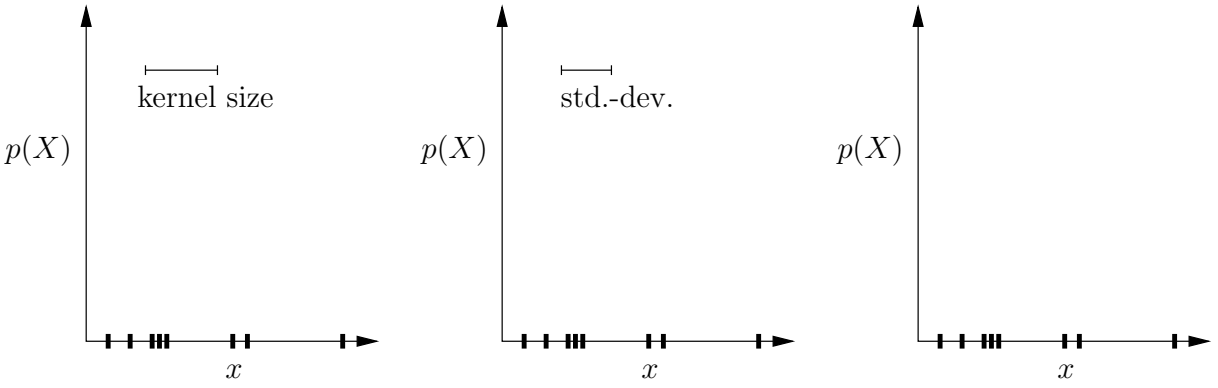
Worksheet 2: Tuesday April 27 / Friday April 30 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 15 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

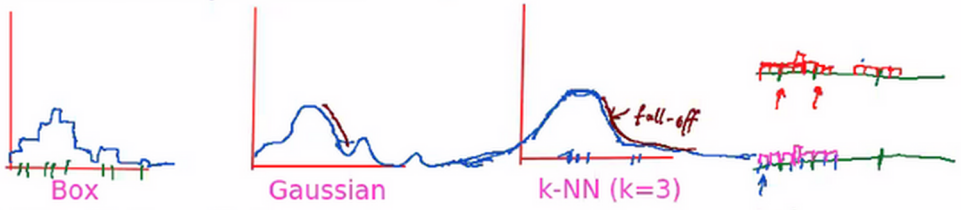
# Task 1: Sketch of Density Estimates

Below are three copies of a 1-D distribution of points along the $x$-axis. The goal is to perform three density estimations on the paper. Use the $y$-axis to indicate the probability mass. Of course, approximate ("sketch") solutions are just fine.




Parzen windows (box kernel) Gaussian kernel k-NN ($k = 3$)

Q1: Sketch of Density Estimates




Q2: "Density Forensics"

kernel window: hyperparam.

Proposal: draw samples, recompute density w/ box, gaussian, k-NN, calculate matching error select method with lowest error

Note that if you have an isolated sample, you directly see the box kernel or Gauss kernel

Proposal: calculate derivative, box will have discrete jumps. Open issue: how to distinguish Gaussian and k-NN?

With k-NN you can never have the value of zero. For Gaussian you could theoretically also have non-zero points, but in practice you always(!) truncate the Gaussian to a limited window size

# Task 3: Computational Cost and Memory Cost

Given a $D$-dimensional sample space, each dimension is scaled to a value range between 0 and 1. We have $N$ samples. We would like use a kernel with compact support[1] to estimate the density.

Let us think about the computational cost and memory cost of these variants:

(a) Discretize the space into a histogram, where each dimension is split into $B$ bins. What is the computational cost of creating such a histogram? What is the memory cost?

(b) How does the computational and memory cost grow when each dimension is discretized into $2B$ bins?

(c) Assume that we implement a kernel density estimator with a box kernel of size $K^D$. If we naively implement this kernel density estimator, what is the computational cost and memory cost?

(d) Open-ended question: can we do something smarter than the options stated above?
Note 1: This question has *many* possible answers, and is beyond the contents of the lecture. It aims to challenge your algorithmic thinking. Will you pick up the challenge?
Note 2: If a sketch helps to explain your idea, prepare one for the joint meeting.

Q3: Computational Cost & Memory Cost

a) Histogram D dim., B bins per dim:
Memory: O(B^D) (-> query is O(1))
or store only bin-ID per sample: O(N*D*log(B)) (-> query is O(N*D)
I would hypothesize that we can get significantly better than that with, e.g. locality sensitivy hashing, or with a quad tree or so to expedite nearest neighbor search)
Computational Cost: O(B^D * K^D) if we do this really naively

[1] *Compact support* means that only finitely many entries are non-zero, i.e., we can use the notion of a window size.

Q1: Bias and Variance in Density Estimation

small kernel support: low bias, high variance
large kernel support: high bias, low variance

Remember for the bias-variance tradeoff:

Simple models are inherently bound by the bias
If we need to reduce the bias, we need more complex models, i.e., models with more parameters (roughly stated)
On the other hand, more complex models increase the variance, so we have to search for a sweet spot somewhere in the middle

## Task 1: Bias and Variance in Density Estimation

Discuss the bias-variance tradeoff for density estimation. Specific sub-questions can be:

- When is the bias too large, when is the variance too large?
- Why are we not guaranteed to find a good tradeoff with our density estimators?
- What can we do if we do not find a good tradeoff?

Illustrate your findings with exercise results.

Q2: Parameters and Hyperparameters in Tree-based methods

Parameters: leaf node data or classification ratios or regression target, split location & dimension at an internal node
Hyperparameters: tree depth, number of trees, randomization parameters: how many splitting locations to sample, bagging ratio, number of sample space dimensions to try in an internal node

Optimization of parameters: training
Optimization of hyperparameters: typically cross-validation, or some other non-reference metrics

## Task 2: Parameters and Hyperparameters in Tree-based Methods

模型参数是根据数据自动估算的。但模型超参数是手动设置的，

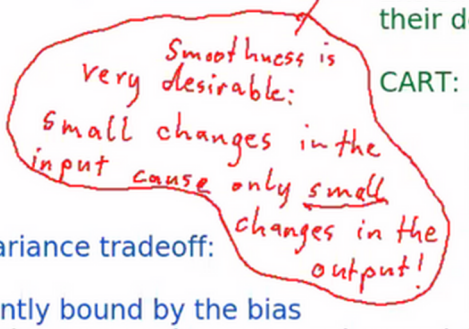
- What are the parameters, what are the hyperparameters of CARTs and Random Forests?
- How are the parameters optimized?
- How are the hyperparameters optimized?

## Task 2: Smoothness of Tree-based Methods

- Why are decision boundaries of random forests smooth?
- Why are decision boundaries of what python calls “extremely randomized forests” even smoother?
- Thinking of smoothness, why has CART a tendency to overfit?

Q1: Bias and Variance in Density Estimation

small kernel support: low bias, high variance
large kernel support: high bias, low variance




Remember for the bias-variance tradeoff:

Simple models are inherently bound by the bias
If we need to reduce the bias, we need more complex models, i.e., models with more parameters (roughly stated)
On the other hand, more complex models increase the variance, so we have to search for a sweet spot somewhere in the middle

Smoothness comes from averaging different decision boundaries (because each tree is different from the other)

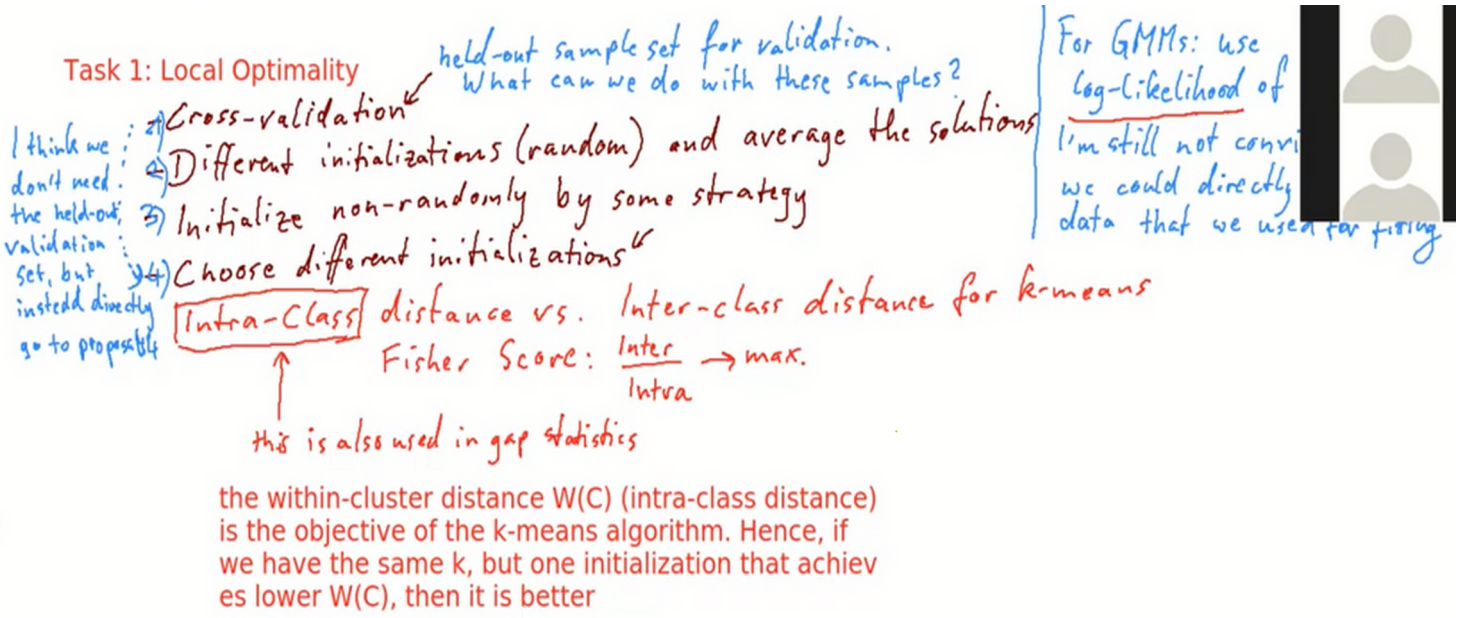
Extremely randomized forests are even less correlated, i.e., their decision boundaries differ more -> smoother

CART: a single tree can not average out sharp decision boundaries. So what shall we do? Make a very flat tree with high bias: this is by definition smooth :-) but not a good classifier/regressor
Or make it really deep and hope that we reach that way a smooth transition. However, what we obtain then is a high-variance classifier/regressor instead, i.e., it is again non-smooth

Q2: Parameters and Hyperparameters in Tree-based methods

Parameters: leaf node data or classification ratios or regression target, split location & dimension at an internal node
Hyperparameters: tree depth, number of trees, randomization parameters: how many splitting locations to sample, bagging ratio, number of sample space dimensions to try in an internal node

Optimization of parameters: training
Optimization of hyperparameters: typically cross-validation, or some other non-reference metrics like BIC (Bayesian Information Criterion)




## Task 1: Local Optimality

Both k-means and GMM fitting are only locally optimal, i.e., they find a solution that is only optimal w.r.t. the starting parameters. Assume that we have a customer who really wants to use k-means or GMMs, but local optima should be avoided.

How can we increase our chances to find a solution close to the global optimum? Can we propose a metric to determine whether solution $A$ is closer to the global optimum than solution $B$?

## Task 2: Clustering Forensics

K-means and GMMs may result in different clusterings. Think about the differences between the k-Means and GMM objective functions, and construct an input where they should give easily distinguishable results (for "reasonable" starting parameters).

## Task 3: GMMs for Density Estimation

Since the GMM estimates a PDF $p(\mathbf{X})$, it could also be used for density estimation.

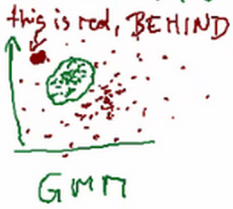
- How does the GMM density estimator relate to our other methods?
- For which sample distributions might a GMM density estimator be preferable?

Task 2: k-means: each point has a unique cluster (→ ML afer GMM also gives that on the other hand, you could also simulate probabilities in k-means)

k-means clusters can not overlap

Cluster boundary: GMM: "ellipsoid" more or less
k-means: Voronoi tesselation: lines bound the cluster

Isolated samples / substructures / other special things:


Task 3: Gaussian of course ☺
Gaps between clusters are filled
k-NN, kernel DE, Density Forests

GMM ⟷ Gaussian kernel
no objective function
↓
requires mean, cov.

separation of components
⇓
Estimates mean, covariance

Note some similarity in the resulting models of GMMs and Dens. Forests:
Both distribute a couple of Gaussians in the space

GMM: parametric, number of components needed

k-NN, Parzen wind.: Non-parametric "less prior knowledge required"

Density Forest: Parametric or non-parametric?

Each tree of height h gives ~ $2^h$ Gaussians

You can in principle let the trees grow to infinity, and prune them back to some "natural" size

Pattern Analysis Joint Meeting
Christian Riess

Worksheet 5: Tuesday May 18 / Friday May 21 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 25 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

## Task 1: Mean Shift Clusters vs. k-Means/GMM Clusters

Following up on Task 2 from last week: on which inputs do you expect Mean Shift to produce noticeably different results from the k-Means and GMM algorithm? Assume that all three algorithms are parameterized such that the number of resulting clusters is approximately the same.

## Task 2: Optimality Criterion in the Gap Statistics

Think about and explain in your own words what the optimality criterion (Eqn. (1) in Lecture 9) aims to achieve:

(a) What structural cue is hidden in the criterion $G(k) \leq G(k+1)$?

(b) Why the argmin?

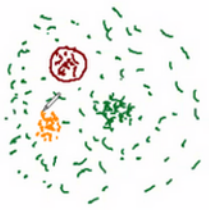
(c) Let us also try a long shot, and (loosely) relate the optimality criterion to the training of a Density Forest

## Task 3: Gap Statistics also for Mean Shift and GMMs?

It might be tempting to re-use the Gap Statistics for the Mean Shift and the GMM clustering to select the kernel size and the number of GMM components. However, I think that this is suboptimal. Which concerns could be raised here?

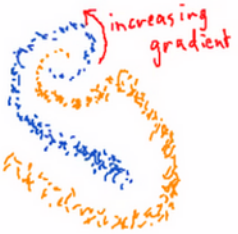
Task 1: Mean Shift Clusters vs. k-Means/GMM Clusters

- samples that are far away from others will be allocated in their own cluster. k-Means would assign them to the nearest bigger cluster



Cluster shapes: we saw that k-means always does Voronoi cells, GMMs can generate non-convex clusters and also clusters that are surrounded by another cluster.

Mean Shift: a sample moves "upwards" on the gradient within the kernel window
⇒ we can obtain almost arbitrary cluster shapes, as long as there is always an increasing path on the gradient




(this is a theoretical consideration of course, in practice you will probably not have such wild clusters)

Task 2: Optimality Criterion in the Gap Statistics

How to find the optimal value of k in k-means

argmin_k { G(k) >= G(k+1) - s_(k+1) }

Generally, less clusters are preferred

Consider the uniform distribution as the worst distribution (structure-wise) to obtain a k-means clustering

The gap is a relative measure of how well we summarize the structure in our distribution compared to that worst-case distribution

We expect the gap to increase as long as we are "on the right track", i.e., as long as we summarize more structure better with adding clusters

If we add a cluster beyond the optimum number k, we "lose ground" with respect to the worst case distribution, i.e., the gap decreases again

Hence, we choose k* as that k before the gap decreases minus some safety margin expressed by the standard deviation

On c): This is maybe remotely similar to the Density Forest, where we split the density with growing tree depth. However, one difference is that we don't necessarily stop to grow the tree, but instead prune unnecessary splits back in a second step

Pattern Analysis Joint Meeting
Christian Riess

Worksheet 6: Tuesday June 1 / Friday June 4 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 20 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

# Task 1: Take-Home Messages from the GMM Model Selection

Several elements of the lecture on GMM Model Selection have a tutorial-like character, with little relevance for the exam.

Let us ask the reverse question: which information from the lecture can be useful?

# Task 2: Distribution of Distances in High-Dimensional Spaces

We have two equations in the lecture on the Curse of Dimensionality that demonstrate a counter-intuitive effect: in higher-dimensional spaces, most samples are located at the boundary.

(a) Think of a (maybe halfways intuitive) explanation for this effect.

(b) Let us double-check the statement that distances become less discriminative. Never mind about this question if your reply to question 1) already covers it.

b)
维度趋于无穷大的情况下，这N个样本之间，所有点和其它的点基本上彼此均匀地相距，此时点之间的距离是没有意义的因为所有点到其它点的距离都趋于相同

# Task 3: Exercise Solutions

Nothing to do for you in the internal discussion: This pseudo task just reserves some time for us to look through and discuss your exercise solutions.

June 1, 2021:
Worksheet 6:

Task 1:

Important:

Bayesian model w/ Priors, ... the exact inference is not possible:

→ Variational Inference

→ MCMC

Not so important:

Specific equations for various priors (Dirichlet, Gaussian-Wishart)

⇒ Variational solution is again an EM algorithm

⇒ Assumption in the Variational Framework: $q(\mathbf{Z})$ and $q(\pi, \Sigma, \mu)$ are independent

($q(\mathbf{Z})$: Expect. step; $q(\pi, \Sigma, \mu)$: Maximization step; independence: here we approximate)

Important but not explained:

- Bayesian vs. frequentist approach

Task 2: June 1, 2021, Worksheet 6:

Curse of Dimensionality

Visualize 1-D (line), 2-D (circle), 3-D (sphere), look how ratio of samples increases for increasing dim.

Consider a D-dimensional vector with random entries between 0 and 1.
This is a hypercube, not a sphere, but never mind.


This brings the whole vector at the boundary of the space

⇒ With increasing dimensionality of the feature vector, we have an increasing chance of hitting the boundary in one of those dimensions

Pattern Analysis Joint Meeting
Christian Riess

Worksheet 7: Tuesday June 8 / Friday June 11 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 20 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

## Task 1: Mapping Unseen Data to the Manifold

Consider a setup where data is first mapped onto a lower-dimensional manifold, and then fed to a classifier. During training, the manifold is learned and also the classifier. For testing, we need to map unseen data to the lower-dimensional manifold prior to classification. Brainstorm: how can this be achieved for PCA, MDS, ISOMAP, and LE?

## Task 2: Model Selection for PCA, MDS, ISOMAP, LE

Consider a data manifold with an intrinsic dimensionality of $d'$ that happens to reside in a $d$-dimensional space. Unfortunately, we do not know the value of $d'$.

Brainstorm: how can we find the "right" number of dimensions for PCA, MDS, ISOMAP, LE?

## Task 3 (only Friday group): Exercise Solutions

Nothing to do for you in the internal discussion: This pseudo task just reserves some time for us to look through and discuss your exercise solutions.

## Task 1: Mapping Unseen Data to the Manifold

- PCA: Transform is a linear projection that is obtained during PCA calculation ⇒ just apply it to new data
- MDS: complicated.., does the sample come with absolute coordinates? ⇒ PCA-style mapping
  if we assume that we obtain a sample also by relations to other points: ? Probably the same mapping holds, but we would need to think/calculate more here
- ISOMAP:
  - They also use eigenvalue decompositions, but a weird neighborhood relationship before that ⇒ this must be taken into account
- LE: ⇒ Shotton/Criminisi/Konukoglu (⇒ Random Forest Paper on studon) propose on page 18? a linear interpolation of nearby points in the lower-dimensional space using affinities calculated in the higher dim.-space. Note that this is an approximation, it would probably not work well for outlier points.

Note: all methods use an eigenvalue decomposition. Can we use the eigenvectors for the projection?

Remark on computational complexity: MDS matrix size is $N \times N$
PCA matrix size is $N \times d$
⇒ if $d \gg N$, then MDS is cheaper

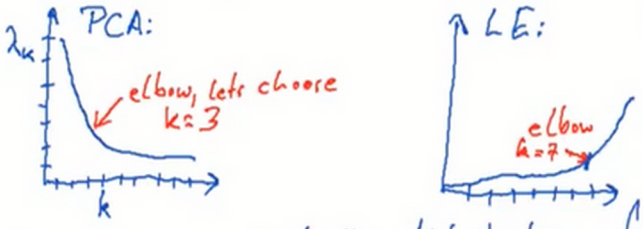
PA Joint Meeting
Worksheet 7

## Task 2: Model Selection for our Manifold Learning Methods

Pearson Correlation Coefficient between inverse projection $d' \to d$ and the original data

In the lecture: set a threshold on the sum of eigenvalues.

What is also possible is to search for an "elbow" in the distribution of eigenvalues (→ see k-means discussion)







I would assume that the distribution of eigenvalues also permits further reasoning.

Pattern Analysis Joint Meeting
Christian Riess

Worksheet 8: Tuesday June 15 / Friday June 18 2021

Please watch the video prior to the lecture, and think about the questions below. **This time, let us limit the internal discussion to 10 minutes**, since we have relatively few technical content.

You can print this sheet and use the space below for your notes.

# Task 1: Drawing Graphical Models

Translate these products of probabilities into graphical models:

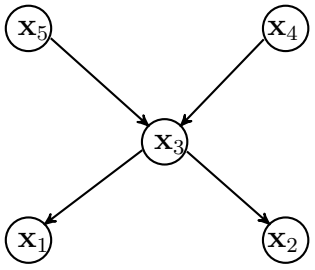
(a) $p(\mathbf{a}|\mathbf{b})p(\mathbf{c}|\mathbf{b})p(\mathbf{b}|\mathbf{d})$

(b) $p(\mathbf{a},\mathbf{b})p(\mathbf{b},\mathbf{c})p(\mathbf{c},\mathbf{d})p(\mathbf{d},\mathbf{a})$



# Task 2: Reading Graphical Models

Write down the factorized density for this graphical model:




p(x4)·p(x5)·p(x3|x4,x5)·p(x1|x3)·p(x2|x3)
或者其他写法？？？？

# Task 3: Clarifications on the Exam

I hope that the video on exam preparation is helpful. Are there further questions?

Pattern Analysis Joint Meeting
Christian Riess

Worksheet 9: Monday June 21 / Friday June 25 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 25 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

# Task 1: Optimization of HMMs

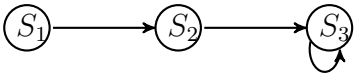
Two questions on the optimization:

(a) HMM training is only locally optimal. What can we do to try to avoid bad local optima?

(b) What is the HMM model selection problem, and how can we address it?

# Task 2: "Super-Fast HMM"

Tony files a patent for his variation of the HMM to recognize speech: the "Blazing Fast HMM". It consists of $N$ states. Any state can be starting state, and the transitions are shown here for $N = 3$:




(a) Does the proposed method deserve its name? What is the computational complexity?

(b) There are also some disadvantages of the proposed design. Which ones?

Task 1: Many algorithms only find local optima (k-means, GMMs, also HMMs, also (beyond PA) neural networks)

This means that the quality of the solution depends on the initialization of the parameters

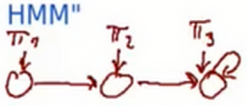
Let's do this → Hence, to increase chances to find a good solution, try different initializations, and either select the best performing model on a held-out validation set, or perform some form of model averaging

What is the HMM model selection problem?

- How many states are used?
- Connection structure of the HMM (left-right HMM, fully connected)

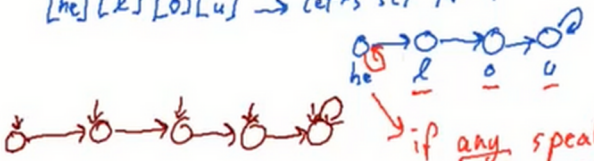
How to optimize these parameters? Cross-validation

Task 2: "Super-Fast HMM"




/hello/

[he] [l] [o] [u] → let's set N=4




if any speaker stretches the "he", then we certainly need a self-loop, otherwise the later states have to deal with the remaining time slices from "he"

a) How fast is it? (for N states)

b) What might be a disadvantage of this design?
⇒ for timing variations in speech, we do need these self-loops.

Q: What is the computational complexity of the forward algorithm?

$O(N^2T)$

↑ # states ↖ speech segment

Reason for $N^2$: when going from $t \to t+1$, we have to consider all outgoing states at $t$ and all incoming edges at $t+1$

In the super-fast HMM, there is always just one outgoing and incoming state ⇒ the cost is $O(1 \cdot T) = O(T)$

⇒ Analogously, all other costs for state transitions vanish

⇒ This effectively removes the hidden state, the model falls back to a simple first-order Markov model.

## Task 3: Small HMMs versus Large HMMs

Tony has a second idea, with the goal of saving on the total number of parameters. Instead of training a single HMM for each word, he could train one large HMM with different paths for different words. That way, he could share representations of identical phonemes in the states. To find out which word was spoken, he could use the Viterbi algorithm to reconstruct the path through the HMM.

What do you think about this proposal? What might be benefits, what might be challenges with it?

Pattern Analysis Joint Meeting
Christian Riess

Worksheet 10: Monday June 29 / Friday July 2 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 20 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

# Task 1: Slight Variation to the Denoising Task: Input Values

We would like to denoise a binary image, much like the example in the lecture. However, assume that we are forced to work on binary pixel intensities of 0 and 1 instead of $-1$ and 1.

Task 1: We saw a denoising example for pixel values in {-1, 1}

What happens if we have pixel values of {0, 1}?

a) What issue do we run into with our Gibbs sampler?
if $x_i=0$, then it does not matter anymore what $y_i$ is, so we don't have a minimum for $x_i = y_i$

b) How can we fix it?
Define a new function, e.g.

$$E(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

$$E(x_i, y_i) = \| x_i - y_i \|_2^2$$

← unary potential
(pairwise is analogous with $x_j$)

→ Energy calculation:

Potential: $e^{-\left(\sum E(x_i,y_i) + \sum E(y_i,\right.}$

$x=0$:

$E(x_i, y_i) = -\eta x_i y_i$

$E(x_i, x_j) = \beta x_i x_j$

# Task 2: Non-Binary Intensities

Following up on question 1, the use of binary intensities appears quite limiting. What could we do in order to denoise grayscale images, e.g., with 256 different gray values?

Task 2: Non-binary Intensities

Denoising of 256-value grayscale pictures

- median filter, Gauss-filter ... certainly works

⇒ How can we adjust our MRF formulation?

Just represent the 256 grayvalues in a range $[0, \ldots, \text{gray}_{\text{MAX}}]$ and choose potential functions in the style of task 1, i.e., a distance function like $L_1$ or $L_2$ distance,

⇒ We have min. energy for two identical inputs, and increasing energy for diverging inputs.

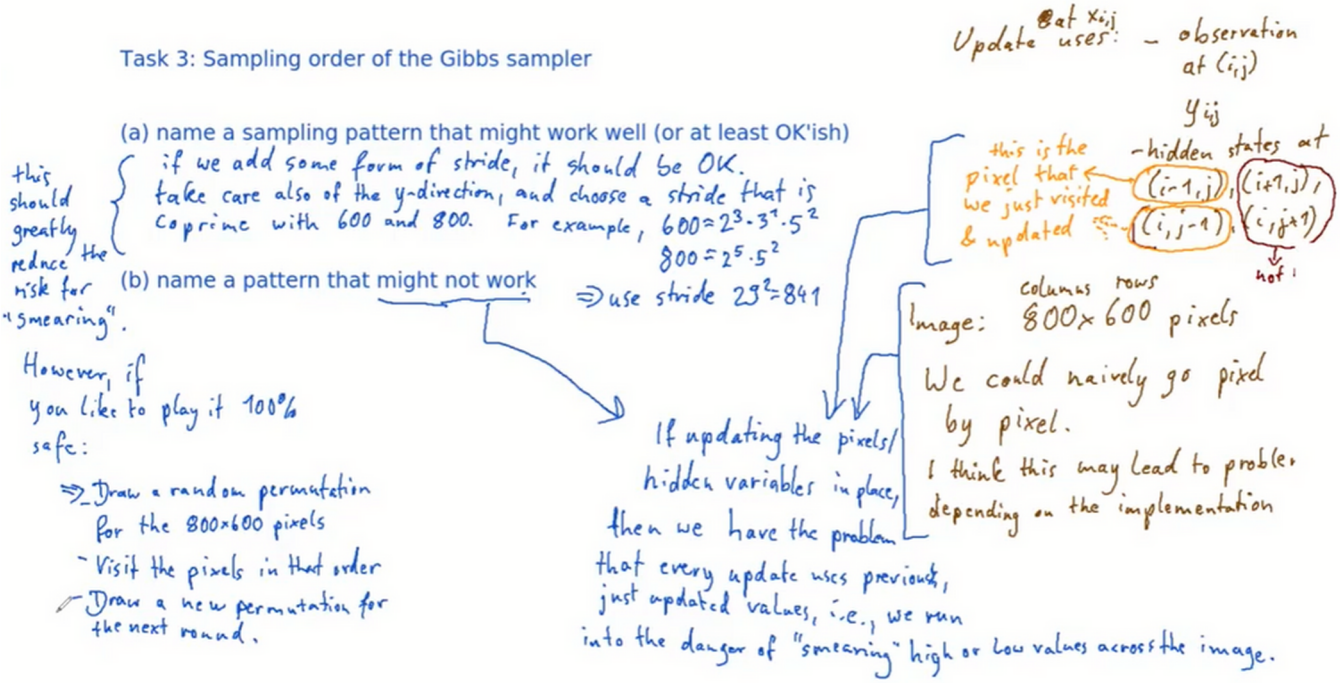
# Task 3: Sampling Order

Tony has the impression that it is inefficient to randomly select points when doing inference via Gibbs sampling: what if one of these points is not selected for a very long time? He proposes to investigate different deterministic sampling patterns that visit all points within a guaranteed number of iterations.

Propose

(a) one deterministic pattern that might potentially work quite well, and

(b) another pattern that will probably lead to bad results?



Pattern Analysis Joint Meeting
Christian Riess

Worksheet 11: Monday July 6 / Friday July 9 2021

Please watch the video prior to the lecture, and think about the questions below. In the joint meeting, you will have 20 minutes time to discuss the questions with your group. Afterwards, we will jointly discuss your solution proposals.

You can print this sheet and use the space below for your notes.

# Task 1: Graph Cut-Solvable Potentials

Check whether each of the MRF setups below can be solved with our min cut transformation. As usual $x_i$, $x_j$ denote hidden variables, and $y_i$ denotes an observation associated with $x_i$.

| | | |
|---|---|---|
| (a) Setup 1: | Task: | Image Smoothing |
| | Labels: | 256 grayscale values, scaled between 0 and 1 |
| | Input: | 256 grayscale values, scaled between 0 and 1 |
| | Connection structure: | grid-like 4-neighborhood (horizontal and vertical neighbors) |
| | unary potentials: | $\exp\left(-3\sqrt{\lvert x_i - y_i \rvert}\right)$ |
| | pairwise potentials: | $\exp\left(-8\lvert x_i - x_j \rvert\right)$ |
| (b) Setup 2: | Task: | Depth Estimation |
| | Labels: | 256 depth values in meters, ranging between 0 and 10 |
| | Input: | Calculations of disparity: Correlations between pairs of shifted input patches |
| | Connection structure: | grid-like 4-neighborhood (horizontal and vertical neighbors) |
| | unary potentials: | $\exp\left(-2\lVert x_i - \text{disparity}(\mathrm{y}_\mathrm{i}^1, \mathrm{y}_\mathrm{i}^2)\rVert_2^2\right)$ |
| | pairwise potentials: | $\exp\left(-2\lVert x_i - x_j \rVert_2^2\right)$ |
| (c) Setup 3: | Task: | Image Sharpening |
| | Labels: | 256 grayscale values, scaled between 0 and 1 |
| | Input: | 256 grayscale values, scaled between 0 and 1 |
| | Connection structure: | grid-like 4-neighborhood (horizontal and vertical neighbors) |
| | unary potentials: | $\exp\left(-(x_i - y_i)\right)$ |
| | pairwise potentials: | $\exp\left(-\frac{1}{\lVert x_i - x_j \rVert_2^2}\right)$ |
| (d) Setup 4: | Task: | Image Smoothing |
| | Labels: | 256 grayscale values, scaled between 0 and 1 |
| | Input: | 256 grayscale values, scaled between 0 and 1 |
| | Connection structure: | grid-like 8-neighborhood (horizontal, vertical, diagonal neighbors) |
| | unary potentials: | $\exp\left(-3\sqrt{\lvert x_i - y_i \rvert}\right)$ |
| | pairwise potentials: | $\exp\left(-8\lvert x_i - x_j \rvert\right)$ |

Remarks on setup 2 (not super-necessary for solving this exercise, but probably good to know): our eyes triangulate depth by looking at the disparity of a point in the scene: the disparity is the lateral shift of a point when seen in the left eye versus seen in the right eye. The closer a point is, the larger is this shift (a finger in front of your nose creates a huge disparity between the left and right eye). Stereo camera systems calculate the disparity by sliding small image patches from the left camera over the image of the right camera. For each position, the correlation of the patches is calculated. The disparity is the shift with the maximum correlation.

## Task 2: Discussion of the Programming Exercises

Let us have a look at the solution to the programming exercise 4!