Generative and discriminative classification techniques

Machine Learning and Category Representation 2014-2015

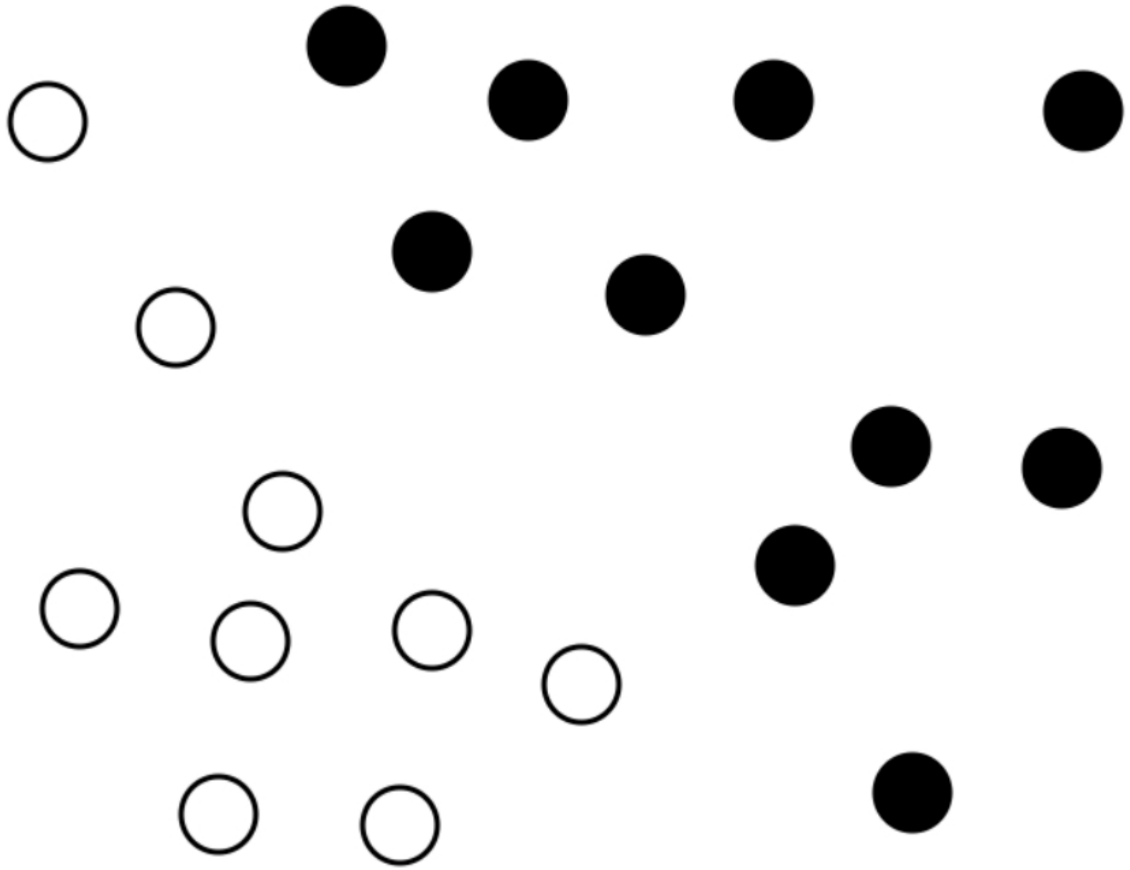
Jakob Verbeek, November 28, 2014

Course website:

http://lear.inrialpes.fr/~verbeek/MLCR.14.15

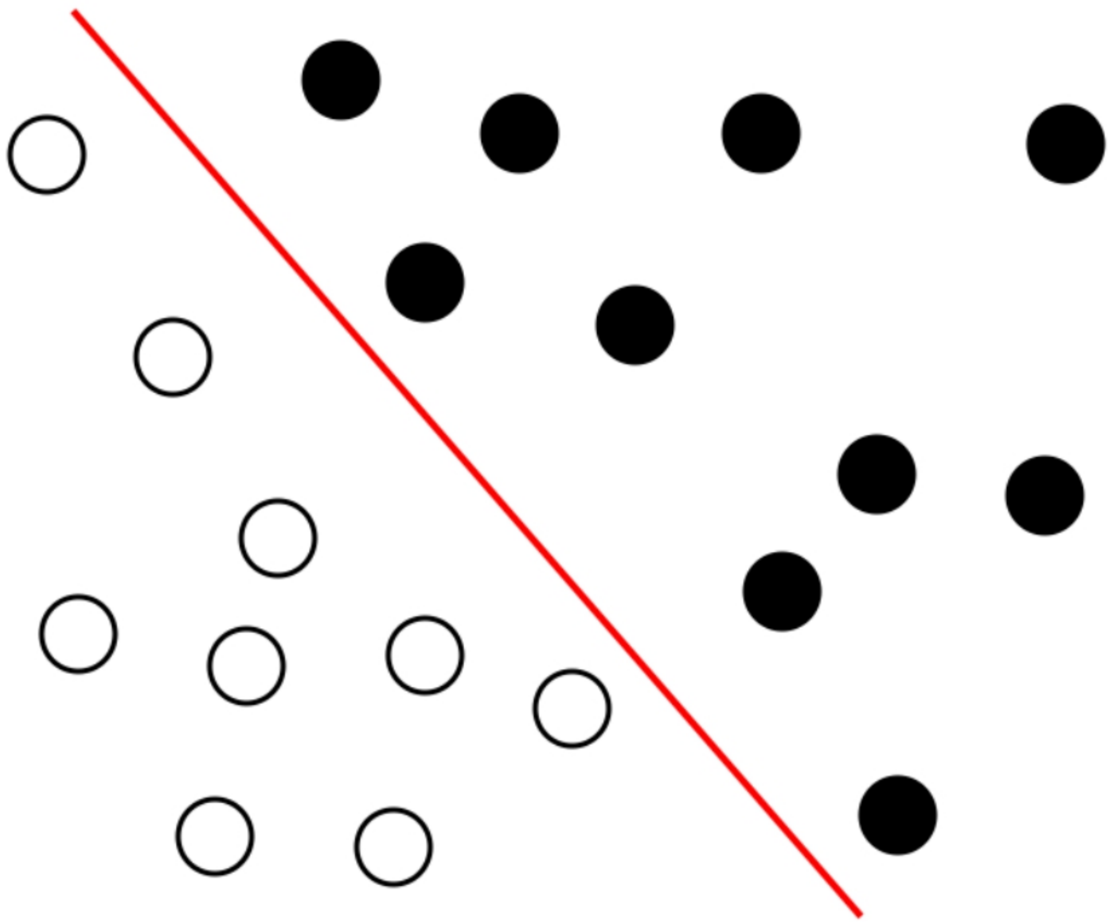
Classification

- Given training data labeled for two or more classes



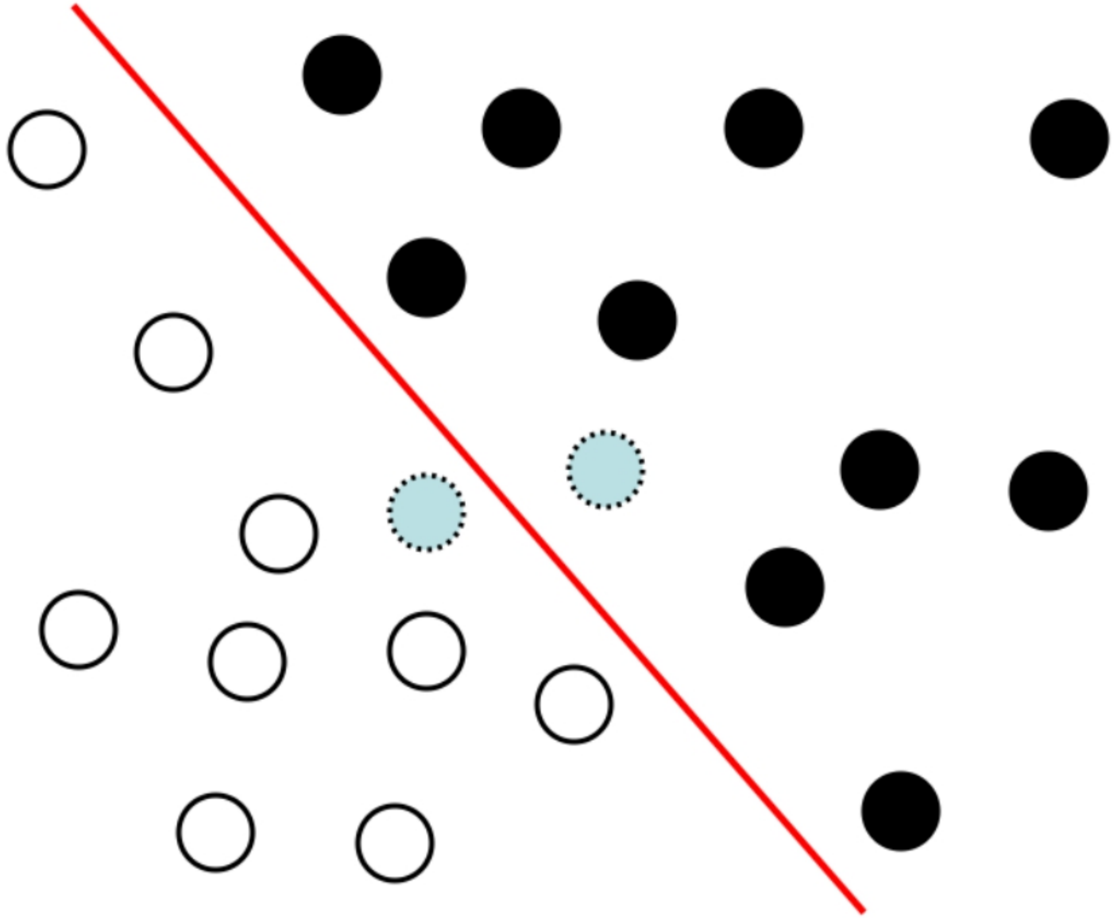
Classification

- Given training data labeled for two or more classes
- Determine a surface that separates those classes



Classification

- Given training data labeled for two or more classes
- Determine a surface that separates those classes
- Use that surface to predict the class membership of new data



Classification examples in category-level recognition

- Image classification: for each of a set of labels, predict if it is relevant or not for a given image.
- For example: Person = yes, TV = yes, car = no, ...

informatics mathematics Inria

Grenoble INP Ensimag

Classification examples in category-level recognition

- Category localization: predict bounding box coordinates.
- Classify each possible bounding box as containing the category or not.
- Report most confidently classified box.

Classification examples in category-level recognition

- Semantic segmentation: classify pixels to categories (multi-class)
- Impose spatial smoothness by Markov random field models.

Classification examples in category-level recognition

- Event recognition: classify video as belonging to a certain category or not.
- Example of “cliff diving” category video recognized by our system.

Classification examples in category-level recognition

- Temporal action localization: find all instances in a movie.
- Enables "fast-forward" to actions of interest, here "drinking"

Classification

- Goal is to predict for a test data input the corresponding class label.
  - **Data input x**, eg. image but could be anything, format may be vector or other
  - **Class label y**, can take one out of at least 2 discrete values, can be more

  - In binary classification we often refer to one class as “positive”, and the other as “negative”

- Classifier: function f(x) that assigns a class to x, or probabilities over the classes.

- Training data: pairs (x,y) of inputs x, and corresponding class label y.

- Learning a classifier: determine function f(x) from some family of functions based on the available training data.

- Classifier partitions the input space into regions where data is assigned to a given class
  - Specific form of these boundaries will depend on the family of classifiers used

Generative classification: principle

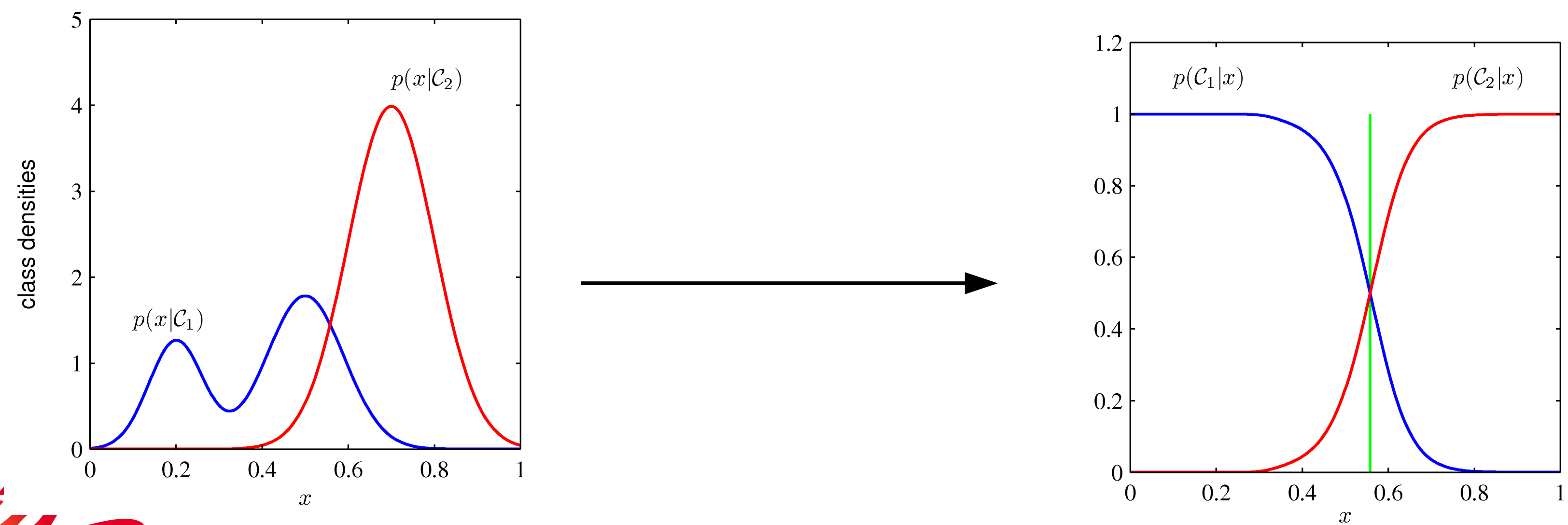
- Model the class conditional distribution over data x for each class y: $p(x|y)$
  - Data of the class can be sampled (generated) from this distribution
- Estimate the a-priori probability that a class will appear $p(y)$

- Infer the probability over classes using Bayes' rule of conditional probability

$$p(y|x)=\frac{p(y)\,p(x|y)}{p(x)}$$

- Unconditional distribution on x is obtained by marginalizing over the class y

$$p(x)=\sum_y p(y)\,p(x|y)$$



Generative classification: practice

- In order to apply Bayes' rule, we need to estimate two distributions.

- A-priori class distribution
  - In some cases the class prior probabilities are known in advance.
  - If the frequencies in the training data set are representative for the true class probabilities, then estimate the prior by these frequencies.
  - More elaborate methods exist, but not discussed here.

- Class conditional data distributions
  - Select a class of density models
    - Parametric model, e.g. Gaussian, Bernoulli, …
    - Semi-parametric models: mixtures of Gaussian, Bernoulli, ...
    - Non-parametric models: histograms, nearest-neighbor method, …
    - Or more structured models taking problem knowledge into account.
  - Estimate the parameters of the model using the data in the training set associated with that class.

Estimation of the class conditional model

- Given a set of n samples from a certain class, and a family of distributions.

  $X = \{x_{1,} \dots, x_n\}$ $\qquad$ $P = \{p_\theta(x); \theta \in \Theta\}$
- Question how do we quantify the fit of a certain model to the data, and how do we find the best model defined in this sense?

- Maximum a-posteriori (MAP) estimation: use Bayes' rule again as follows:
  - Assume a prior distribution over the parameters of the model $p(\theta)$
  - Then the posterior likelihood of the model given the data is

    $$p(\theta|X) = p(x|\theta)\, p(\theta) / p(X)$$
  - Find the most likely model given the observed data

    $$\hat{\theta} = \operatorname{argmax}_\theta\, p(\theta|X) = \operatorname{argmax}_\theta \{\ln p(\theta) + \ln p(X|\theta)\}$$

- Maximum likelihood parameter estimation: assume prior over parameters is uniform (for bounded parameter spaces), or “near uniform” so that its effect is negligible for the posterior on the parameters.
  - In this case the MAP estimator is given by $\hat{\theta} = \operatorname{argmax}_\theta\, p(X|\theta)$
  - For i.id. samples:

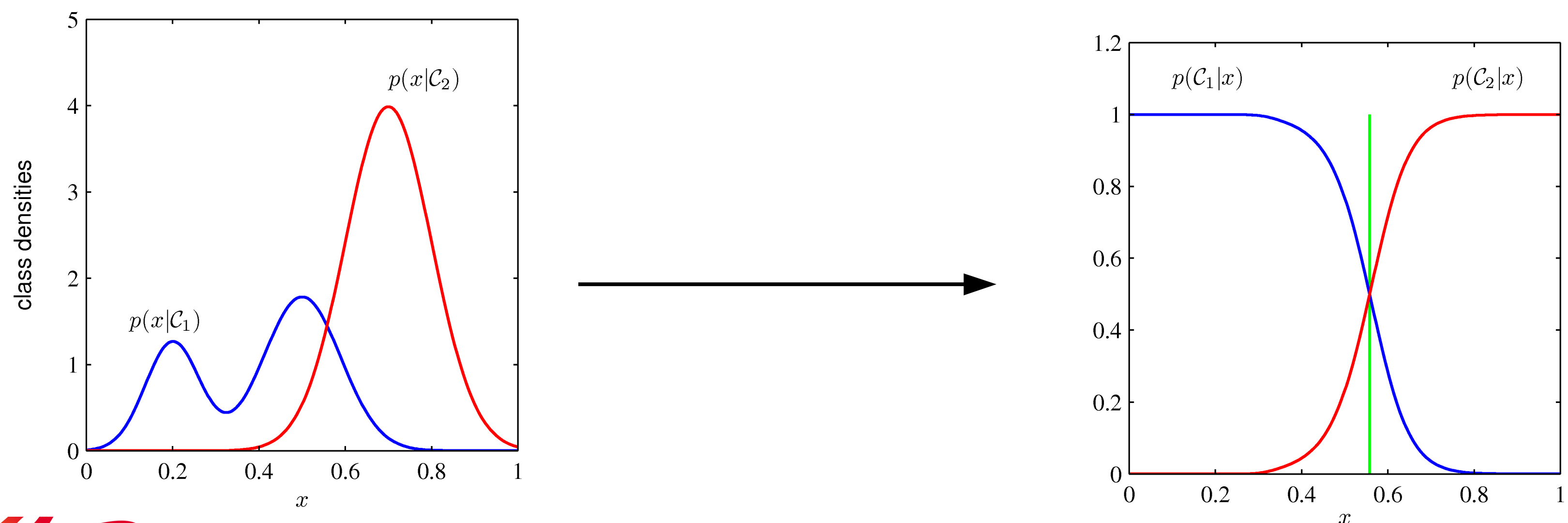
    $$\hat{\theta} = \operatorname{argmax}_\theta \prod_{i=1}^{n} p(x_i|\theta) = \operatorname{argmax}_\theta \sum_{i=1}^{n} \ln p(x_i|\theta)$$

Generative classification methods

- Generative probabilistic methods use Bayes' rule for prediction
  - Problem is reformulated as one of parameter/density estimation

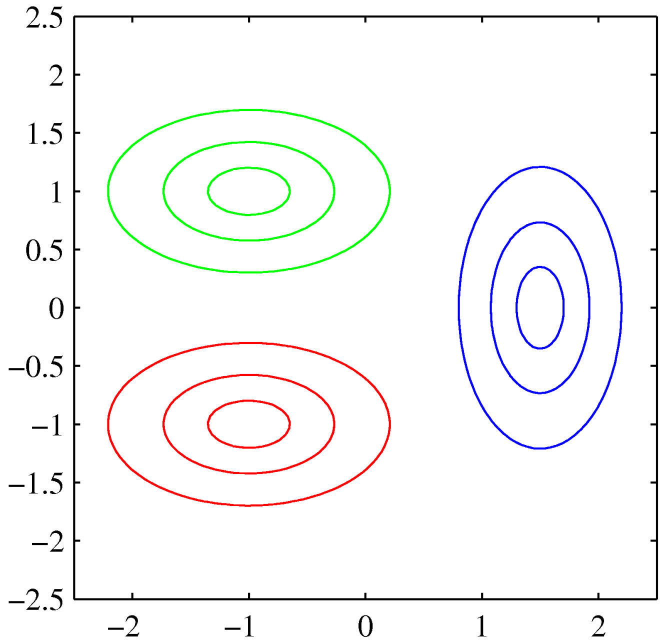
$$p(y|x) = \frac{p(y)\,p(x|y)}{p(x)} \qquad\qquad p(x) = \sum_y p(y)\,p(x|y)$$

- Adding new classes to the model is easy:
  - Existing class conditional models stay as they are
  - Estimate p(x|new class) from training examples of new class
  - Re-estimate class prior probabilities

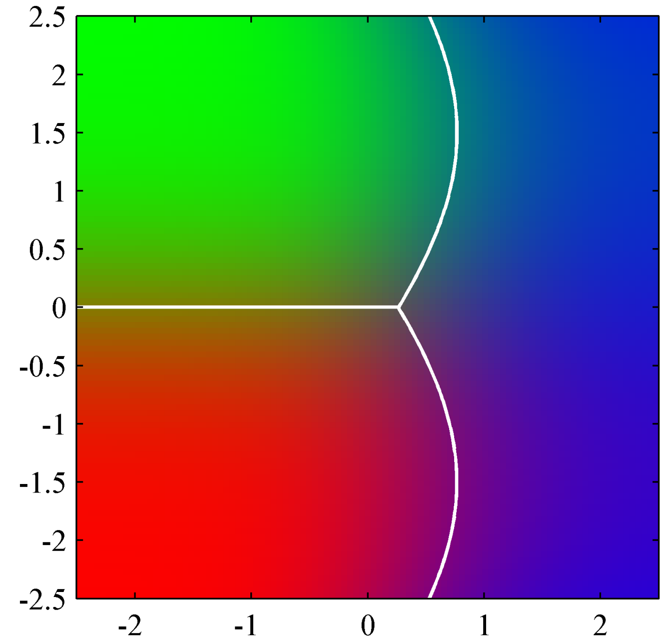


Example of generative classification

- Three-class example in 2D with parametric model
  - Single Gaussian model per class, uniform class prior
  - Exercise 1: how is this model related to the Gaussian mixture model we looked at last week for clustering ?
  - Exercise 2: characterize surface of equal class probability when the covariance matrices are the same for all classes



$p(x|y)$



$$p(y|x) = \frac{p(y)\,p(x|y)}{p(x)}$$

Density estimation, e.g. for class-conditional models

- Any type of data distribution may be used, preferably one that is modeling the data well, so that we can hope for accurate classification results.

- If we do not have a clear understanding of the data generating process, we can use a generic approach,

  - Gaussian distribution, or other reasonable parametric model
    - Estimation in closed form, otherwise often relatively simple estimation

  - Mixtures of XX
    - Estimation using EM algorithm, not more complicated than single XX

  - Non-parametric models can adapt to any data distribution given enough data for estimation. Examples: (multi-dimensional) histograms, and nearest neighbors.
    - Estimation often trivial, given a single smoothing parameter.

Histogram density estimation

- Suppose we have *N* data points use a histogram with *C* cells
- Consider maximum likelihood estimator

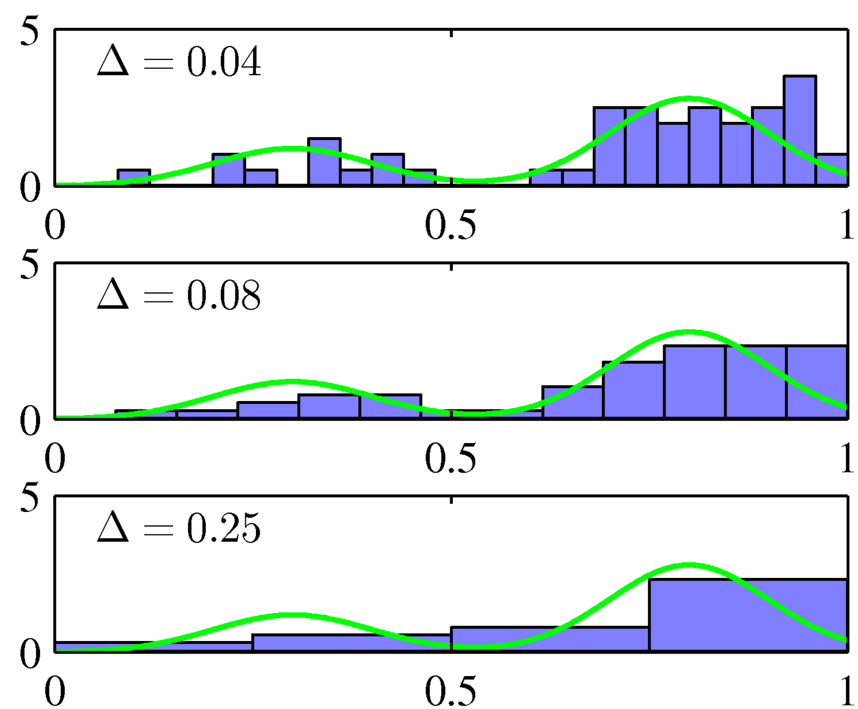
$$\hat{\theta} = \operatorname{argmax}_\theta \sum_{i=1}^{n} p_\theta(x_i) = \operatorname{argmax}_\theta \sum_{c=1}^{C} n_c \ln \theta_c$$

- Take into account constraint that density should integrate to one

$$\theta_C := 1 - \left(\sum_{k=1}^{C-1} v_k \theta_k \right) / v_C$$

- Exercise: derive maximum likelihood estimator

- Some observations:
  - Discontinuous density estimate
  - Cell size determines smoothness
  - Number of cells scales exponentially with the dimension of the data



informatics mathematics Inria

The Naive Bayes model

- Histogram estimation, and other methods, scale poorly with data dimension
  - Fine division of each dimension: many empty bins
  - Rough division of each dimension: poor density model
    - Even for one cut per dimension: 2^D cells

- The number of parameters can be made linear in the data dimensionality by assuming independence between the dimensions

$$p(x) = \prod_{d=1}^{D} p(x(d))$$

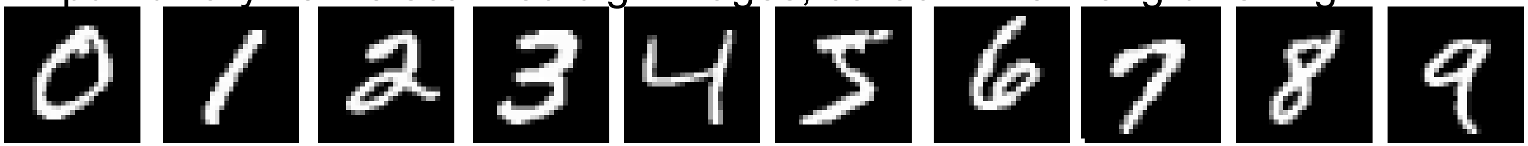
- For example, for histogram model: we estimate a histogram per dimension
  - Still C^D cells, but only D x C parameters to estimate, instead of C^D

- Independence assumption can be (very) unrealistic for high dimensional data
  - But classification performance may still be good using the derived p(y|x)
  - Partial independence, e.g. using graphical models, relaxes this problem.

- Principle can be applied to estimation with any type of density estimate

Example of a naïve Bayes model

- Hand-written digit classification
  - Input: binary 28x28 scanned digit images, collect in 784 long bit string



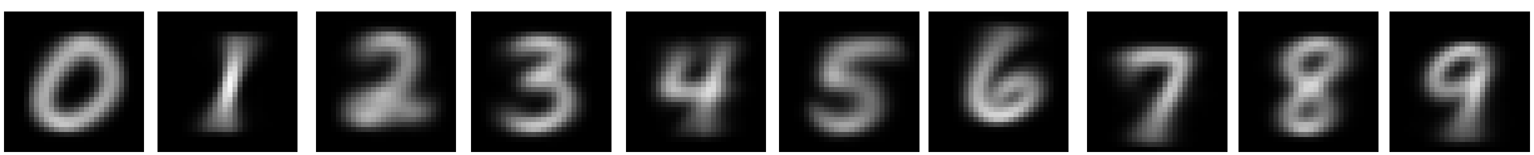
  - Desired output: class label of image

- Generative model over 28 x 28 pixel images: 2^{784} possible images
  - Independent Bernoulli model for each class
  - Probability per pixel per class
  - Maximum likelihood estimator is average value per pixel/bit per class

$$p(x|y=c)=\prod_d p(x^d|y=c)$$

$$p(x^d=1|y=c)=\theta_{cd}$$

$$p(x^d=0|y=c)=1-\theta_{cd}$$



- Classify using Bayes' rule: $p(y|x)=\frac{p(y)\,p(x|y)}{p(x)}$

k-nearest-neighbor density estimation: principle

- **Instead of having fixed cells** as in histogram method,
  - **Center cell** on the test sample for which we evaluate the density.
  - Fix number of samples in the cell, find the corresponding **cell size.**

- Probability to find a point in a sphere ***A*** centered on $\boldsymbol{x_0}$ with volume ***v*** is

$$P(x \in A) = \int_A p(x)\, dx$$

- A smooth density is approximately constant in small region, and thus

$$P(x \in A) = \int_A p(x)\, dx \approx \int_A p(x_0)\, dx = p(x_0)\, v_A$$

- Alternatively: estimate ***P*** from the fraction of training data in ***A****:* $P(x \in A) \approx \frac{k}{N}$
  - Total N data points, k in the sphere ***A***

- Combine the above to obtain estimate $p(x_0) \approx \frac{k}{N v_A}$

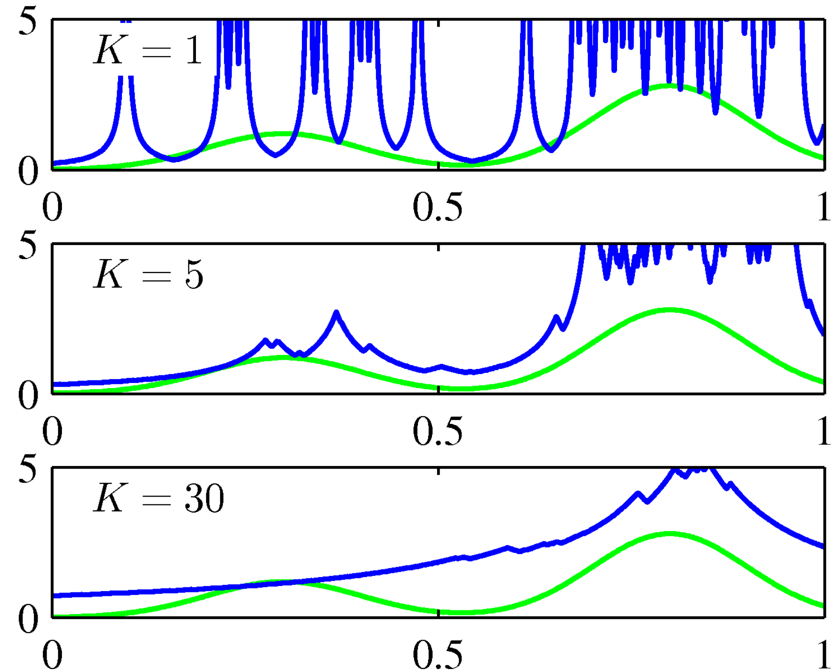
- Note: density estimates not guaranteed to integrate to one!

informatics mathematics Inria

Grenoble INP Ensimag

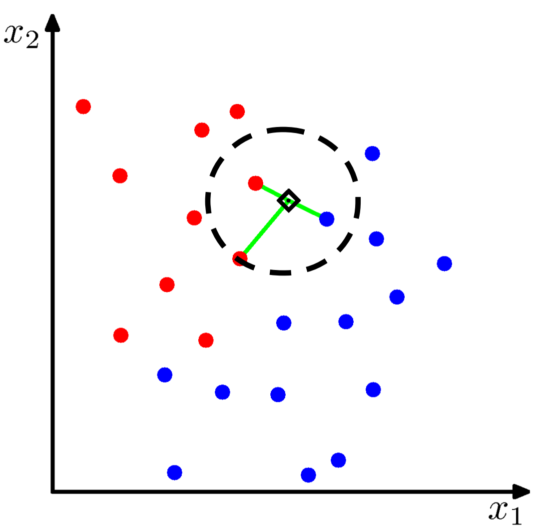
k-nearest-neighbor density estimation: practice

- Procedure in practice:
  - Choose ***k***
  - For given ***x***, compute the volume ***v*** which contain ***k*** samples.
  - Estimate density with $p(x) \approx \frac{k}{Nv}$
- Volume of a sphere with radius ***r*** in ***d*** dimensions is $v(r,d) = \frac{2r^d \pi^{d/2}}{\Gamma(d/2+1)}$
- What effect does ***k*** have?
  - Data sampled from mixture of Gaussians plotted in green
  - Larger ***k***, larger region, smoother estimate
  - Similar role as cell size for histogram estimation



K-nearest-neighbors for classification

- Use Bayes' rule with kNN density estimation for p(x|y)
  - Find sphere volume v to capture ***k*** data points for estimate $p(x)=\frac{k}{N\,v}$
  - Use the same sphere for each class for estimates $p(x|y=c)=\frac{k_c}{N_c\,v}$
  - Estimate class prior probabilities $p(y=c)=\frac{N_c}{N}$
  - Calculate class posterior distribution as fraction of k neighbors in class c

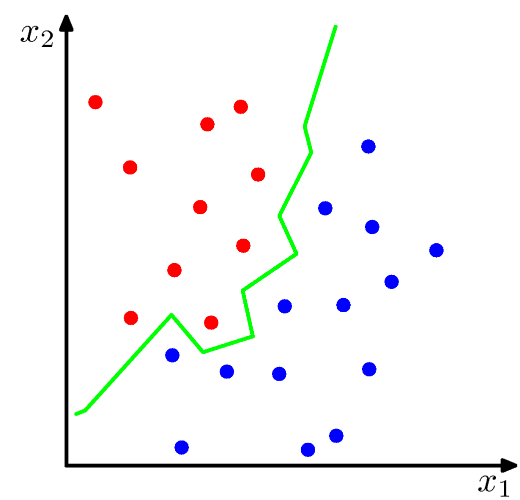


(a)

$$p(y=c|x)=\frac{p(y=c)\,p(x|y=c)}{p(x)}$$

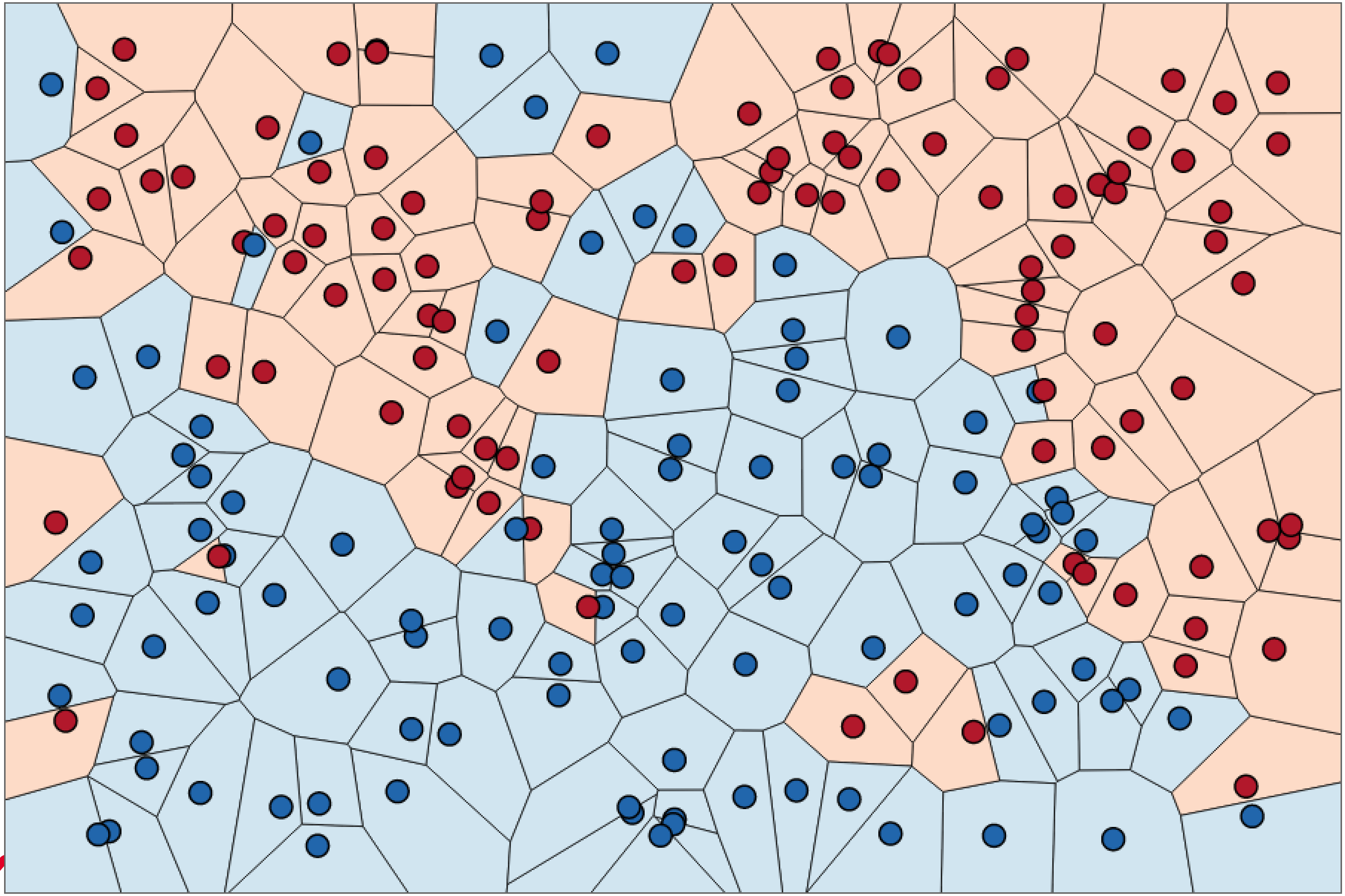
$$=\frac{1}{p(x)}\frac{k_c}{Nv}$$

$$=\frac{k_c}{k}$$

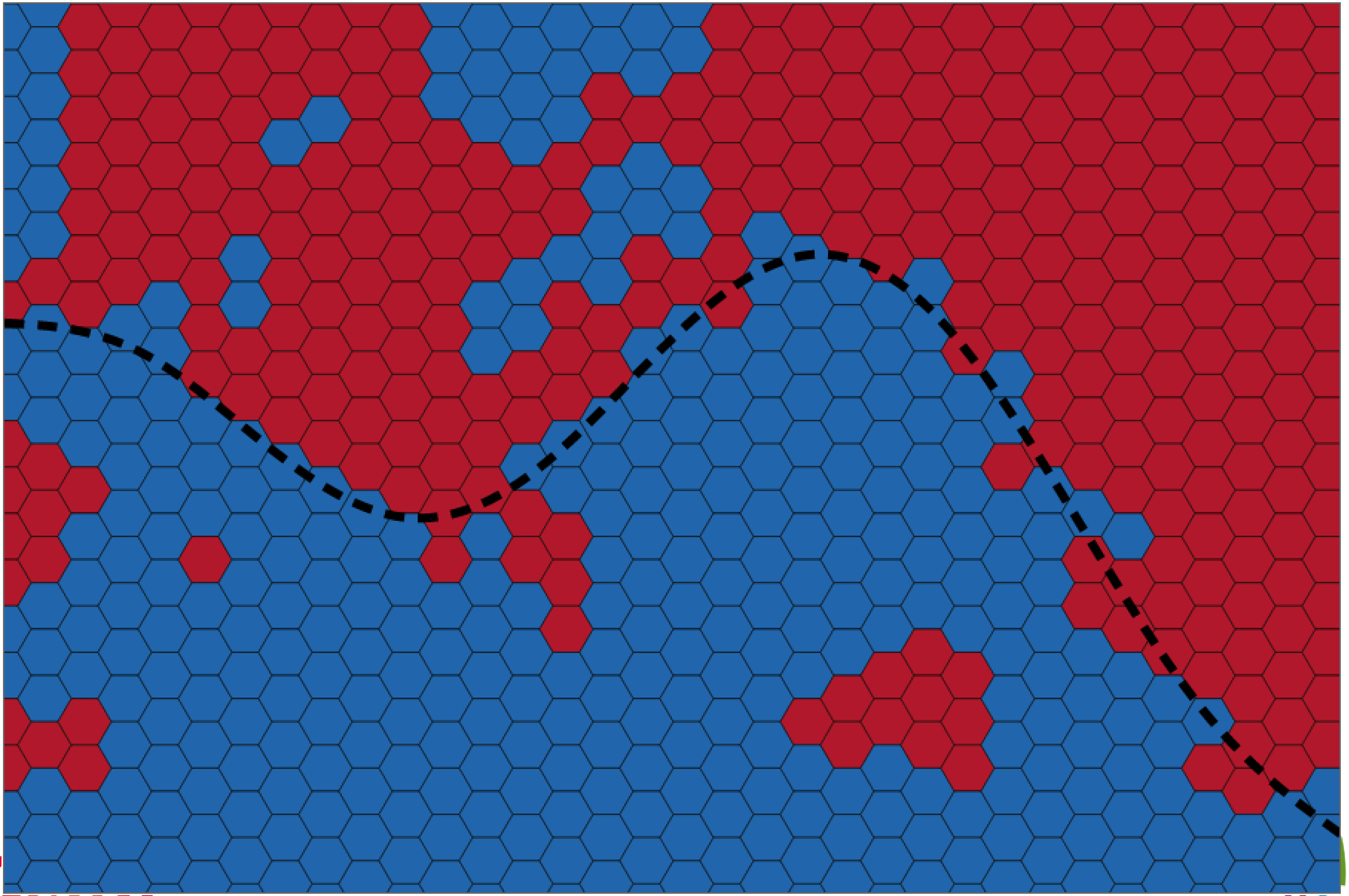


(b)

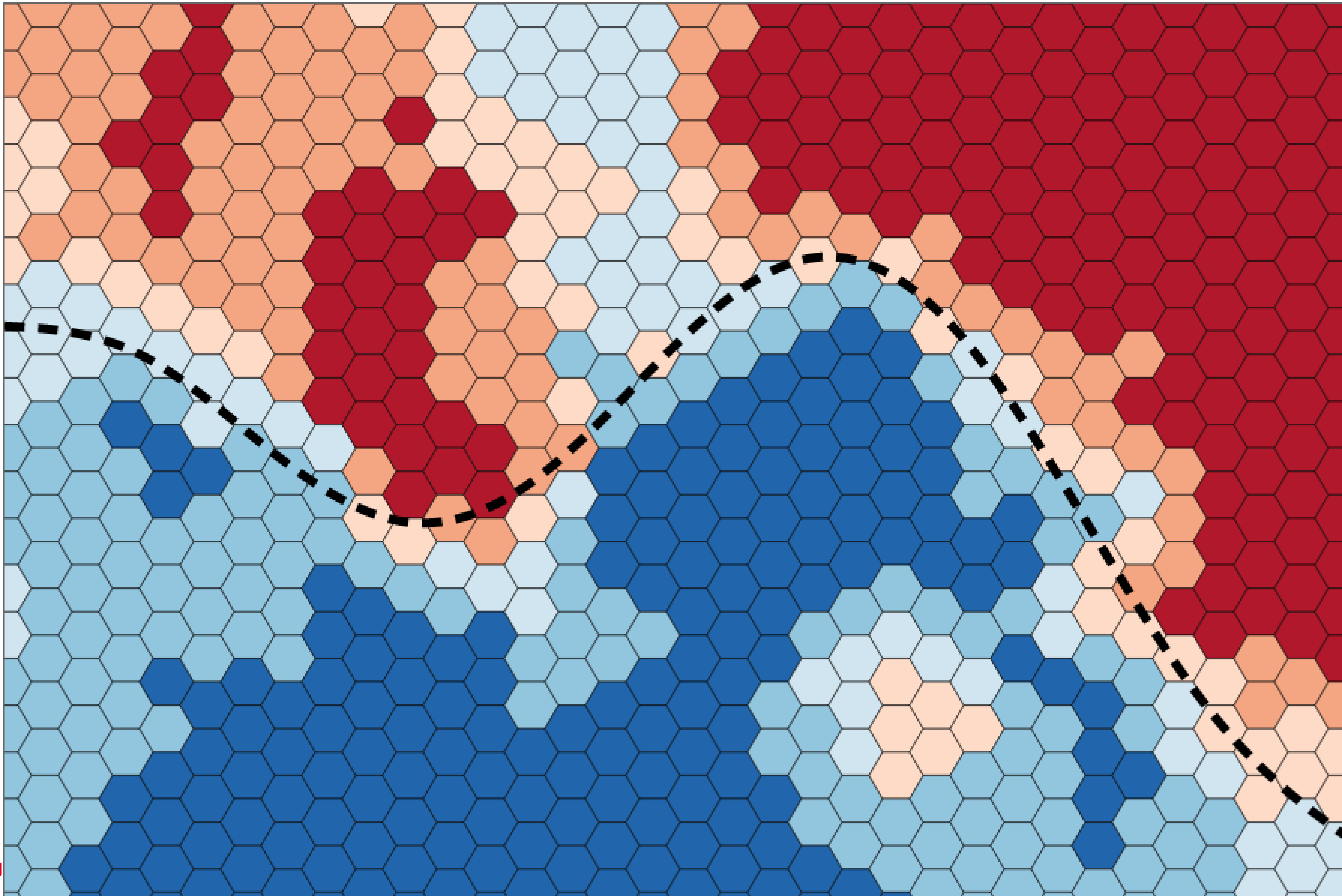
Smoothing effects for large values of k: data set



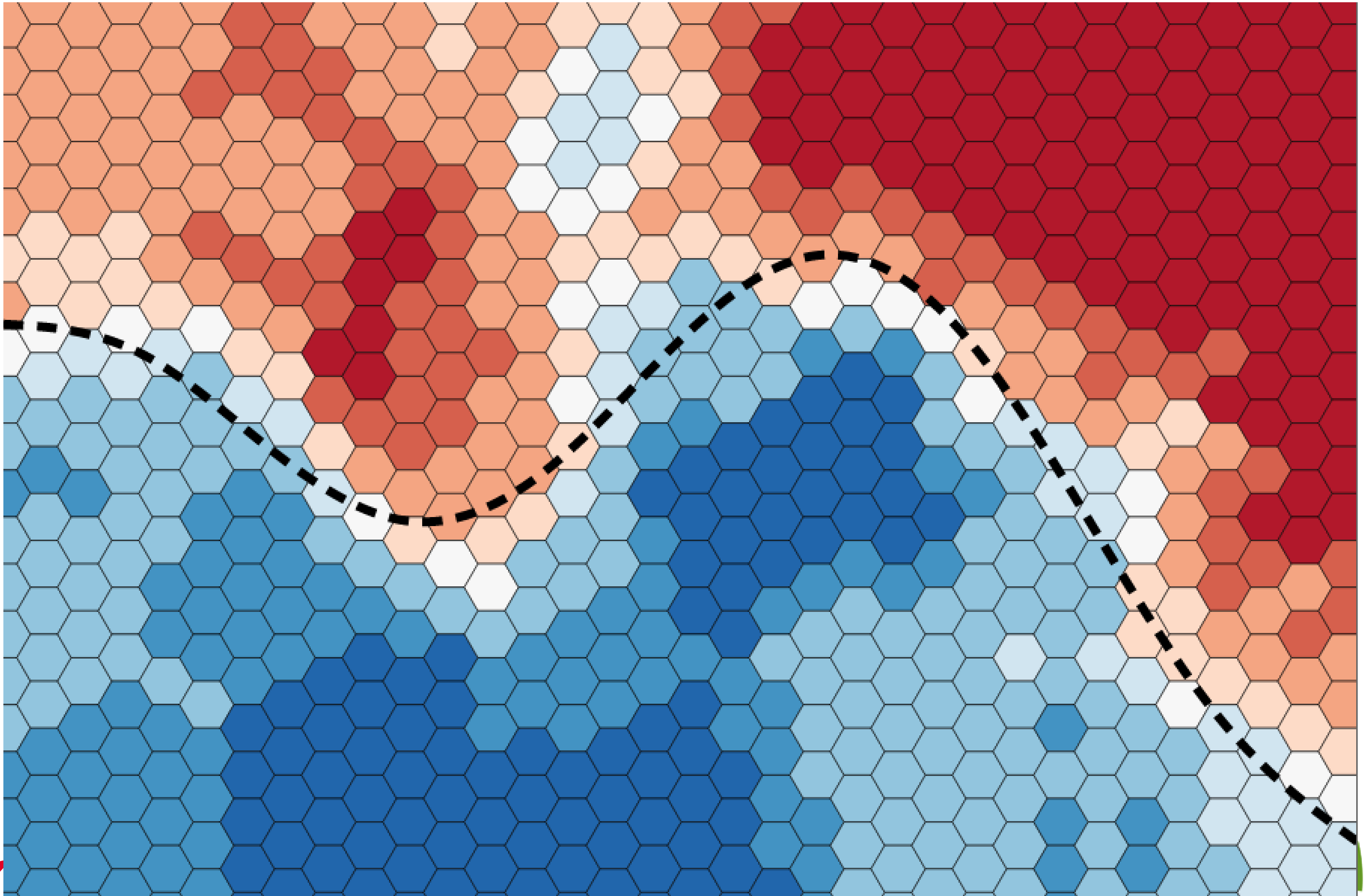
Smoothing effects for large values of k, k=1



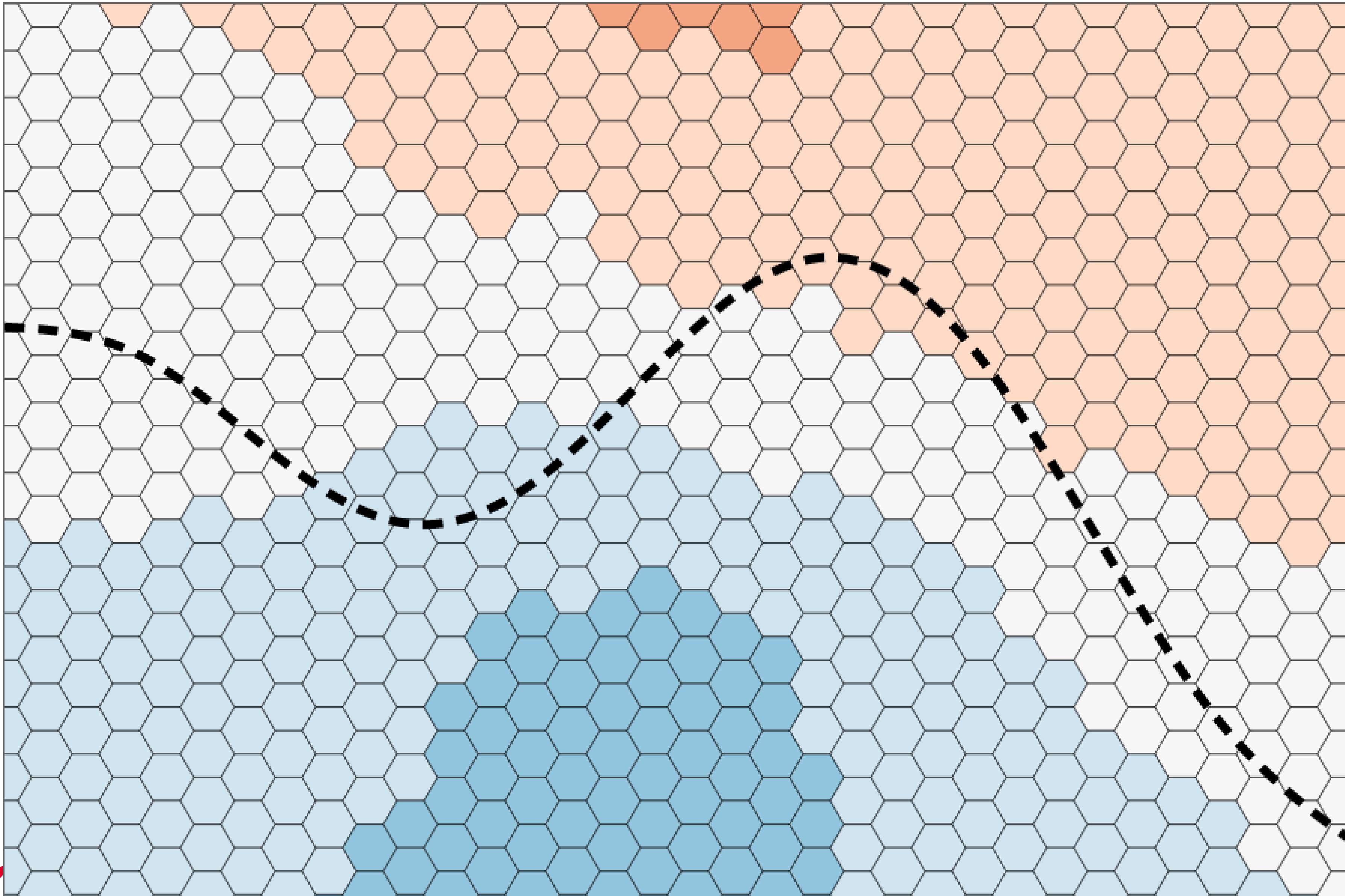
Smoothing effects for large values of k, k=5



Smoothing effects for large values of k, k=10



Smoothing effects for large values of k, k=100



Summary generative classification methods

- (Semi-) Parametric models, e.g. p(x|y) is Gaussian, or mixture of …
  - Pros: no need to store training data, just the class conditional models
  - Cons: may fit the data poorly, and might therefore lead to poor classification result

- Non-parametric models:
  - Pros: flexibility, no assumptions distribution shape, "learning" is trivial. KNN can be used for anything that comes with a distance.
  - Cons of histograms:
    - Only practical in low dimensional data (<5 or so), application in high dimensional data leads to exponentially many and mostly empty cells
    - Naïve Bayes modeling in higher dimensional cases
  - Cons of k-nearest neighbors
    - Need to store all training data (memory cost)
    - Computing nearest neighbors (computational cost)