Pattern Recognition

Discriminative Learning

Discriminative models

There exists a joint probability distribution $p(x, k; \theta)$ (observation, class; parameter). The task is to learn θ

However (see the “Bayesian Decision theory”),

$$R(d) = \sum_k p(k|x) \cdot C(d, k) \rightarrow \min_d$$

→ i.e. only the posterior $p(k|x)$ is relevant for the recognition.

The Idea: decompose the joint probability distribution into

$$p(x, k; \theta) = p(x) \cdot p(k|x; \theta)$$

with an **arbitrary** $p(x)$ and a parameterized posterior.

→ learn the parameters of the **posterior** probability distribution.

Maximum Likelihood

Let the training data $L = \left((x^1, k^1), (x^2, k^2) \ldots (x^l, k^l)\right)$ be given

$$p(L; \Theta) = \prod_l \left[p(x^l) \cdot p(k^l | x^l; \Theta) \right]$$

$$\ln p(L; \Theta) = \sum_l \ln p(x^l) + \sum_l \ln p(k^l | x^l; \Theta)$$

The first term can be omitted as we are not interested in $p(x)$

The second term is often called the **conditional likelihood**.

ML, Example – Gaussians

$$p(x,k) = p(k) \cdot \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\left[-\frac{\|x-\mu^k\|^2}{2\sigma^2}\right]$$

Derive the posterior from the joint probability distribution:

$$p(k=1|x) = \frac{p(1)p(x|1)}{p(1)p(x|1)+p(2)p(x|2)} = \frac{1}{1+\frac{p(2)p(x|2)}{p(1)p(x|1)}} =$$

$$= \frac{1}{1+\exp\left[-\frac{\|x-\mu^2\|^2}{2\sigma^2}+\frac{\|x-\mu^1\|^2}{2\sigma^2}+\ln p(2)-\ln p(1)\right]} =$$

$$= \frac{1}{1+\exp\big(\langle w,x\rangle + b\big)}$$

Logistic regression model.

ML, Example – Gaussians

$$CL \text{ (conditional likelihood)} = \sum_l \ln p(k^l|x^l; w, b) =$$

$$= \sum_{l:k^l=1} -\ln\bigl(1+\exp(wx^l+b)\bigr) + \sum_{l:k^l=2} \Bigl[wx^l + b - \ln\bigl(1+\exp(wx^l+b)\bigr)\Bigr] =$$

$$= w \sum_{l:k^l=2} x^l + n_2 \cdot b - \sum_l \ln\bigl(1+\exp(wx^l+b)\bigr)$$

Derivations:

$$\frac{\partial\, CL}{\partial a} = \sum_{l:k^l=2} x^l - \sum_l \frac{\exp(wx^l+b)}{1+\exp(wx^l+b)} x^l =$$

$$= \sum_{l:k^l=2} x^l - \sum_l p(k{=}2|x^l; w, b) x^l$$

$$\frac{\partial\, CL}{\partial b} = n_2 - \sum_l p(k{=}2|x^l; w, b)$$

It is not possible to resolve it analytically. Note: the subject is convex
→ Gradient-method leads to the global solution.

Discriminative models

No unsupervised learning ☹

For an incomplete training set $L = (x^1, x^2 \dots x^l)$

$$\ln p(L;\Theta) = \sum_l \ln \sum_k p(x^l, k) =$$

$$= \sum_l \ln \sum_k \left[p(x^l) \cdot p(k|x^l;\Theta) \right] = \sum_l \ln p(x^l)$$

→ does not depend on the parameter at all.

This were **discriminative** models learned **generatively** (i.e. ML).

Discriminant functions

- Let a parameterized family of probability distributions be given.
- Each particular p.d. leads to a classifier.
- The final goal is the classification (applying the classifier.)

Generative approach:

1. Learn the parameters of the probability distribution (e.g. ML)
2. Derive the corresponding classifier (e.g. Bayes)
3. Apply the classifier for test data

Discriminative approach:

1. Learn the unknown parameters of the classifier directly
2. Apply the classifier for test data

If the family of classifiers is “well parameterized”, it is not necessary to consider the underlying probability distribution at all !!!

Example – Gaussians

Two classes, Gaussians of equal variance as conditional p.d.-s.

→ Classifier is a hyperplane
→ search for a "good" hyperplane $\langle w, x \rangle < b$
(that "fits" the training set)

Compare: $2 \cdot n + 2$ free parameters of the probability distribution, only n free parameters of the classifier.

→ one classifier corresponds to many probability distributions.

For Gaussians: the location of the hyperplane does not depend on σ ,
Centers μ^1 and μ^2 are not relevant, but their difference $\mu^1 - \mu^2$
(see the board).

Empirical Risk

How to find a good classifier ?

Bayesian risk:

$$R_b(e) = \sum_x \sum_k p(x,k)\, C\Big(e(x), k\Big) \to \min_e$$

But now it can not be computed because there is no p.d. !!!

We have only the training set $L = \Big((x^l, k^l) \ldots\Big)$

The **Bayesian** risk is replaced by the **Empirical** one – average loss over the training set instead of over the whole space:

$$R_e(e) = \sum_l C\Big(e(x^l), k^l\Big) \to \min_{e \in \mathcal{E}}$$

With a pre-defined classifier family $\mathcal{E}$.

Empirical Risk for linear discriminant functions

1. The family of classifiers: all linear classifiers $\langle w, x \rangle < b$ with unknown parameters $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$

2. $\mathbb{1}(k \neq k')$ as the loss-function

3. Assumption: there exist classifier that reaches zero loss (the training set is separable)

→ The Perceptron Algorithm.

Vapnik-Chervonenkis Dimension

Is the learning good (enough) ?

A reasonable measure would be the reached Bayesian risk. However, it can not be computed since there is no probability model. However, one can compute the Empirical risk.

→ The question: how fast (and whether at all) does the Empirical risk converges to the Bayesian one with the increase of the training set ?

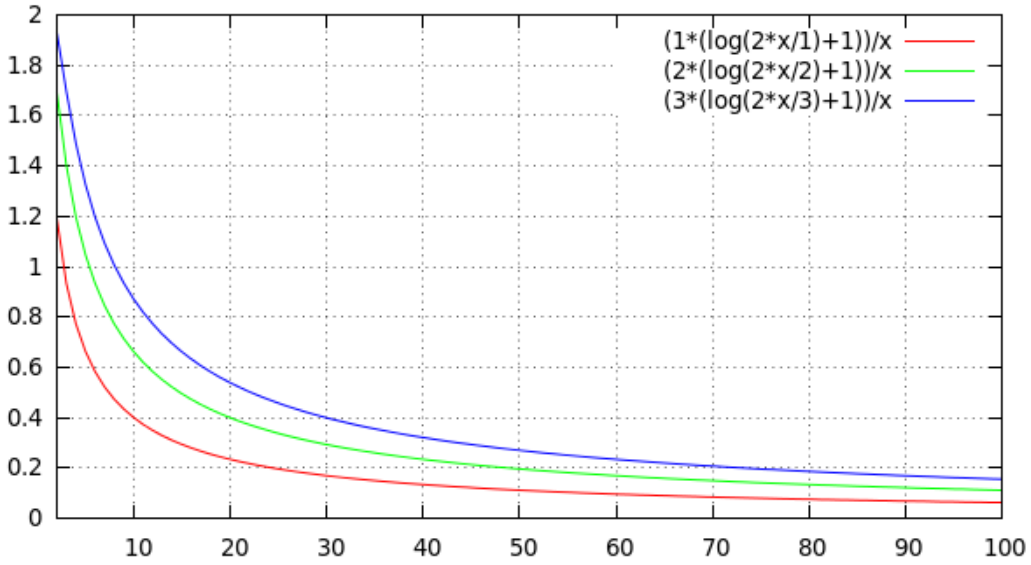
Upper bound for the difference (Vapnik, Chervonenkis, 1968):

$$P\left\{ |R_b - R_e| < \sqrt{\frac{h\left(\log(2N/h)+1\right) - \log(\delta/4)}{N}} \right\} > 1 - \delta$$

The probability (over all training sets) that the considered difference is less then something is greater as something".

Vapnik-Chervonenkis Dimension

$$P\left\{|R_b - R_e| < \sqrt{\frac{h\left(\log(2N/h)+1\right) - \log(\delta/4)}{N}}\right\} > 1-\delta$$



The convergence speed depends on a constant h , which is called Vapnik-Chervonenkis Dimension. It reflects the "power" of the classifier family. The greater VC the worse the **generalization capabilities** of the classifier family.

VC-Dimension

A constructive definition:

A classifier family **shatters** the set of data points if, for all classifications of these points, there exists a classifier such that the model makes no errors when evaluating that set of data points.

The VC-Dimension of the family is the maximal number of points that can be arranged so that the family shatters them.

Alternative: The VC-Dimension is the smallest number of data points so that for any arrangement there exists a classification that can not be re-produced by the family.

Example: for linear classifiers in $\mathbb{R}^n$ the VC-dimension is $\mathrm{VC}{=}n+1$ (see the board).

VC-Dimension

The VC-dimension is often related to the number of free parameters (but not always, example – sinus, one free parameter, infinite VC)

The lower is VC the more robust is the family of classifiers.

Dilemma: complex data → complex classifiers (to reach **good recognition** rate) → many free parameters (high VC) → **bad generalization** capabilities.

Overfitting: the classifier specializes to a particular training set.

Generative vs. discriminative

Families of classifiers are usually “simpler” compared to the corresponding families of probability distributions (lower dimensions, less restricted etc.)

Often it is not necessary to care about the model consistency (such as e.g. normalization) → algorithms become simpler.

It is possible to use more complex decision strategies, i.e. to reach better recognition results.

However:

Large classified training sets are usually necessary, unsupervised learning is not possible at all.

Worse generalization capabilities, overfitting.

Conclusion

A “hierarchy of abstraction”:

1. **Generative models** (joint probability distributions) represent the **entire** “world”. At the learning stage (ML) the probability of the training set is maximized, no loss function.

2. **Discriminative models** represent posterior probability distributions, i.e. only what is needed for recognition. At the learning stage (ML) the **conditional likelihood** is maximized, no loss function.

3. **Discriminant functions**: no probability distribution, decision strategy is learned directly, the **Empirical risk** is minimized.