pertinence
simple solutions for complex processes

www.pertinence.com

Clustering and Prediction: some thoughts

Olivier Bousquet
olivier.bousquet@pertinence.com

July 4th, 2005

> Goal of this talk

- **Not a presentation of research results**
- **But some ideas and questions, in order to stimulate discussions**

Outline

- What is clustering?
- What is the quality of a clustering?
- What is the quality of an algorithm for a problem?
- What is the quality of an algorithm in general?
- How to choose the number of clusters?
- Conclusion

pertinence
simple solutions for complex processes

What is the task of clustering?

- "Extract hidden structure in the data"
- What structure?
  - What is the form of the result?
  - How to measure its quality?
- How to extract it?
  - Statistical issues (sampling)
  - Computational issues
- How many clusters?
  - Model order selection

pertinence
simple solutions for complex processes

Model

- **Usual statistical model (same as Shai or Ule's talks)**
  - P an unknown distribution
  - the data has been generated i.i.d. from P
- **Given this sample, we want to infer information about P itself**
  - Ben-David: "Get a simple yet meaningful description of the distribution"
- **Two questions when analyzing a clustering algorithm**
  - What would it give in the limit of large samples?
  - How does it approach this limit?

pertinence

What is a clustering?

Consider the whole space X

- **Partition** $f : \mathrm{X} \to \{1,\ldots,K\}$

with permutation invariance

- **Quantization** $f : \mathrm{X} \to \{x_1,\ldots,x_K\}$
- **Partitions and quantizations (on X) are equivalent**
- **Soft partition: non-deterministic map of the above type**

$$f(x) = (p_1,\ldots,p_K) \text{ with } \sum_{k=1}^{K} p_k = 1$$

- **Soft partitions + density = mixture models**
- **Hierarchical model: a collection of (nested)partitions for each K in N**

pertinence

> Extension

- **Some algorithms work directly on the dataset**
- **They need to be extended to the whole space**
- **So specifying a clustering algorithm should mean specifying**
  - How to label the sample points
  - How to extend this labeling to the whole space
- **Examples**
  - k-means can be extended with 1-NN (but anything else as well)
  - mixture models can be directly extended

pertinence
simple solutions for complex processes

> What should be measured?

- **Quality of a clustering**
  - Empirical quality (typically the criterion optimized by the algorithm)
  - True quality (requires extension and knowledge about the distribution, or can be estimated by CV or bounds)
- **Quality of an algorithm for a given problem**
  - For a given distribution P, estimate how “good” is the structure extracted from P by the algorithm on average
  - Not for a specific clustering
  - Can be used for model order selection
- **Quality of an algorithm in general**
  - Assess quality on other problems
  - Cannot be used for model order selection

pertinence
simple solutions for complex processes

Outline

- What is clustering?
- What is the quality of a clustering?
- What is the quality of an algorithm for a problem?
- What is the quality of an algorithm in general?
- How to choose the number of clusters?
- Conclusion

pertinence

Just a remark

- Ule's talk: goal is to estimate d(C(Pn),C*(P)) but this cannot be done directly
- Could one use Yatracos' trick for density estimation (Devroye and Lugosi 2000)
- Goal is to find the best density in a class with respect to

$$\|f_n - f\| = \int |f_n - f|$$

- Cannot be done directly, but the following trick gets within a factor of 3 of the best in the class !

$$\arg\min_{f \in F} \sup_{A \in \mathrm{A}} \left| \int_A f - \mu_n(A) \right|$$

pertinence

What is a good clustering?

- X random variable distributed according to P
- Y random variable distributed according to

$$P(Y = k|X) = f(X)_k$$

- X and Y should have “maximum relationship”
- Need a measure of structural relationship

Possible information measures

- Mutual information
  - Non-structural (=H(Y) for deterministic partitions)
  - But can be estimated in a structural way: build a smoothed density estimator of X
  - Structure can be brought in by extra information (IB)
- Bayes error (from X to Y)
  - Non-structural (0 for deterministic partitions)
  - But can be estimated in a structural way: build a classifier
- Structure is a subjective notion

Measuring structural information

- **Consistent algorithms will have error converging to the Bayes error, but how fast depends on the structure**
- **The limit value is not informative**
  - structure is lost for infinite samples
  - unless algorithm is not consistent
  - the learning curve is more informative?
- **However, it is not clear what property of this learning curve should be used.**

- **Examples:**
  - Expected classification error of a 1NN with n samples
  - Can be extended to (conditional) density estimation: randomized prediction

pertinence

Measuring structural information (proposal)

- **Structural relationship: how does X help to predict Y?**

- **Need assumptions**
  - Depends on a given loss function (risk functional)
  - Depends on a given prediction algorithm
  - Depends on the sample size

- **Possible Formulas**

$$S_n(Y|X) = \frac{E(\ell(y_n, Y)) - E(\ell(g_n(X), Y))}{E(\ell(y_n, Y))}$$

$$S_n(X, Y) = \frac{E(\ell(y_n, Y)) + E(\ell'(x_n, X)) - E(\ell(g_n(X), Y)) - E(\ell'(h_n(Y), X))}{E(\ell(y_n, Y)) + E(\ell'(x_n, X))}$$

pertinence

Second-order structure

- **First-order structure: how X can help predict Y, when given a labeled sample**
  - How smooth is Y with respect to X?
  - How extrapolated value match true values?
- **Second-order structure: how X can help predict Y, when given an unlabeled sample**
  - How smooth is Y with respect to P(X)?
  - How one can extrapolate based on the distribution of X only?
- **Combining both**
  - Need a semi-supervised algorithm
  - Take average error given n labeled and m unlabeled points

pertinence
simple solutions for complex processes

Application to clustering quality

- **"Structural information" should be retained by the clustering**
- **First option: from Y to X**
  - "How much information did we loose replacing X by the labels Y?"
  - Measured by the ability of predicting X from Y
  - Deterministic predictor, loss given by d(X,X'), h maps each cluster to a centroid

$$E\left(d\left(X, h_n\left(Y\right)\right)\right)$$

  - Typically the kind of measure used in centroid-based clustering
- **Second option: from X to Y (or even symmetrically)**
  - For example, compute the CV error of a predictor on the labeled sample
  - The algorithm that is used encodes the regularity assumptions

pertinence
simple solutions for complex processes

Extension and structural information

- **If one has a finite clustering (on the sample)**
- **Two options**
  - Use the clustering as a labeled training sample for the predictor and assess its error (i.e. the error of the model built by the predictor)
  - Extend the clustering to the whole space and measure the error of the predictor (under some distribution to be chosen, or for resampled datasets)
- **It is natural to use the extension operator for measuring the quality as well**

pertinence
simple solutions for complex processes

Comments

- **Needs assumptions**
  - Yes but any quality measure does
  - And there is no universal notion of “structure”, just like there is no universal notion of “regularity” in supervised learning
  - One can take the prediction algorithm that is consistent with the extension operator
- **Not well defined in terms of sample size**
  - For practical purposes, use the given sample size
  - But this needs investigation in order to compute the limiting value

pertinence
simple solutions for complex processes

Outline

- What is clustering?
- What is the quality of a clustering?
- What is the quality of an algorithm for a problem?
- What is the quality of an algorithm in general?
- How to choose the number of clusters?
- Conclusion

pertinence
simple solutions for complex processes

Quality assessment

- Quality measures of a clutering: measure the “fit”
- Quality measures of an algorithm
  - not geared towards a sample but towards the distribution
  - can be estimated on a sample
  - aimed at selecting the “right” number of clusters
- Examples
  - Penalty term (BIC, MDL): arbitrary choice
  - Cross-validated quality measure: requires extension (hence prediction)
  - Gap statistic
  - Stability

pertinence
simple solutions for complex processes

Stability

- Several versions: Elisseeff & Ben-Hur, Lange et al, Ben-David & Schaefer…
- If the clustering is stable with respect to small changes in the dataset, it captures relevant structure
- Need to compare clusterings
- Need to resample

pertinence

Comparing clusterings

- Same set
  - Plenty of existing measures
  - Based on membership only / Based on distance
  - Using permutations for comparing labels
- Different sets
  - Require extension operator: i.e. a classification algorithm
  - Possibly semi-supervised extension
- Prediction interpretation
  - predict Y' from Y or from (X,Y)
  - advantage: it can be defined even for different numbers of clusters
  - prediction instead of permutations: arbitrary maps rather than bijective ones. Best map can be computed in O(k^2) time

pertinence

Stability and prediction

- **Lange et al. 2002**
  - Cluster the first half
  - Cluster the second half and extend it to the first half
  - Compare the labels
- **Ben-David & Schaefer 2005**
  - Cluster the first half
  - Cluster the second half
  - Compare both extensions (to the union)
- **Quality measure based on prediction**
  - Cluster the whole data
  - Extend half of the labels to the other half
  - Compare the labels
- **Combining with stability?**

pertinence
simple solutions for complex processes

Stability: supervised vs unsupervised

- **Distance between clusters or between losses?**
- **Supervised stability**
  - stable algorithm satisfy some sort of Lipschitz condition
  - need to assess variability in loss only (estimation error)
- **Unsupervised**
  - quality of a clustering is not only measured by a “quality measure” (no unique goal)
  - assumption: for large sample sizes, the clustering has converged to an “optimal one”, stability measures how far we are from this

pertinence
simple solutions for complex processes

Issues with stability

- **Does not capture the "fit" (Iris example in Lange et al 2002)**
  - Which quality measure to use? (Shai: distance to a random clustering)
  - How to relate it to stability?
  - How to trade both?
- **Stability measures several effects**
  - Sampling sensitivity of the algorithm
  - Degeneracy of the quality measure (different clusterings of the same sample may have the same quality)
  - Stability of the algorithm itself (for stochastic algorithms, even for a fixed sample, there may be different clusterings)
- **Hence stability is not only a correction for estimation error, it also detects instability of the algorithm/objective !!**
  - Is this ok?
  - What if the quality measure is degenerate ?

pertinence
simple solutions for complex processes

Outline

- **What is clustering?**
- **What is the quality of a clustering?**
- **What is the quality of an algorithm for a problem?**
- **What is the quality of an algorithm in general?**
- **How to choose the number of clusters?**
- **Conclusion**

pertinence
simple solutions for complex processes

Another use of prediction

- **Goal: estimate the quality of an algorithm in general or for a class of problems**
- **Consider a classification problem**
- **Determine if clustering helps for classification**
  - Need to choose a classification algorithm
  - Need to determine what is the input of the classifier
  - Not usable for model order selection (but can test a model order selection procedure)

pertinence
simple solutions for complex processes

Possible set-ups

- **Banerjee & Langford**
  - input to the algorithm: X=labels given by the clustering (on a large set), Y=true labels (on a subset)
  - use a semi-supervised algorithm (majority based since it has a one dimensional discrete input)
  - use a bound to estimate the error
- **Candillier et al**
  - class labels for various values of k added to the dataset
  - use a supervised algorithm
  - measure the CV error gain
- **This is fine provided there is indeed a "structural relationship" between X and Y (in the labeled/unlabeled sense)**

pertinence
simple solutions for complex processes

Outline

- What is clustering?
- What is the quality of a clustering?
- What is the quality of an algorithm for a problem?
- What is the quality of an algorithm in general?
- How to choose the number of clusters?
- Conclusion

pertinence
simple solutions for complex processes

Model selection in supervised learning

- Primary goal: estimate expected error
- expected error = empirical error + corrective term
- corrective term: variance estimate (cross-validation, complexity, or instability)
- This can be applied directly to clustering, if the goal is just to find the model with smallest expected quality

pertinence
simple solutions for complex processes

Degeneracy of the quality

- **Is the expected quality a good measure?**
- **Assume full knowledge of P**
  - Does the quality measure have a unique optimum?
  - Is this optimum attained for a finite K?
  - Is there a value of K that can be agreed upon? (consider the case of groups of clusters)
  - Even if P is fully known, is there a unique notion of a "best" decomposition?
  - Given a fixed mixture model, is there a unique solution to the minimal distance problem?
- **A measure of "fit" should go to 0 when K goes to infinity (at least under P)**
  - So there is a need to compensate (arbitrarily?)

pertinence
simple solutions for complex processes

Model order selection

- **What is a good measure of "variance"?**
  - Variance of the fit itself?
  - Variance of the clusterings?
- **Is it possible to avoid assumptions?**
  - No, even with stability (relies on the choice of an extension and a distance between clusterings)
  - Can this impossibility be formalized?

pertinence
simple solutions for complex processes

> Choosing K with structural information?

- **Can we use "structural measures"?**
- **Take the whole distribution**
  - **Predict X from Y will not work (always prefer infinite k)**
  - **Predict Y from X may work**

pertinence
simple solutions for complex processes

> Outline

- **What is clustering?**
- **What is the quality of a clustering?**
- **What is the quality of an algorithm for a problem?**
- **What is the quality of an algorithm in general?**
- **How to choose the number of clusters?**
- **Conclusion**

pertinence
simple solutions for complex processes

> Other ideas from supervised

- **Talk from A.Barron (2005): three questions**
  - **estimation**
  - **approximation**
  - **computation**
- **Estimation is the best understood area**
- **Model selection formalization in supervised learning: what can be gained from it?**
- **Regularization, early stopping: would this help?**
- **Computation: convexity, regularization paths…**

pertinence
simple solutions for complex processes

> Computation?

- **Most algorithms are non-convex**
- **Find efficient convex relaxations?**
  - **Computationally effective**
  - **Close to the solution of the initial problem**
- **Or do as in supervised learning: use convexity as a goal but introduce other aspects**
  - **Relaxation for regularization (e.g. convex clustering shrinkage)**
  - **Use this for stability guarantees**

pertinence
simple solutions for complex processes

> Conclusion/Open questions

- **Is bias unavoidable? If yes, then we just need to clarify it (and not hide it)**
- **Formalize "structural information" measure (especially the sample size issue, both labeled and unlabeled)**
- **Stability**
  - **Bias-variance trade-off: how to measure and trade both?**
  - **Degeneracy of the objective and other sources of instability (regularization may help)**
- **Choosing K: far from being well-posed**