InfoScale: Information-Based Scaling

PhD Thesis Proposal

Leejay Wu

May 31, 2002

Overview

- Task, motivation and thesis
- State of the art
- My current work
- Proposed work

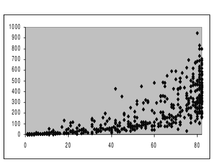
Task, Motivation and Thesis

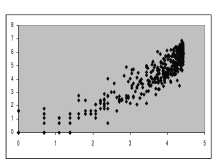
- What is the problem?
- Why should anybody care?
- What do I propose?

Choosing a Distance Function

- Given
  - unlabeled vector data (unsupervised)
- Produce
  - An appropriate distance function for data mining (isomorphic: better scales)
- Critical subtask
  - Define and quantify "appropriate"

Which is better?





- Same data, different scales
- Is one more appropriate than the other?

Importance

- Distances define
  - Neighborhoods (*k* nearest objects, objects within radius *r*)
  - Similarity (inverse relationship)
- End-user applications include
  - Clusters
  - Principal components
  - Relevance judgements

The InfoScale Approach: Make every bit count!

- Information theory provides a mathematically and intuitively satisfying definition for propriety of scaling.
- Both nonlinear scaling and attribute selection can be guided by this metric to form an improved distance function.

Overview

- Task, motivation and thesis
- State of the art
- My current work
- Proposed work

Taxonomy of Other Approaches

- Linear projection methods (PCA, Johnson-Lindenstrauss, ICA, perceptrons...)
- Manifold methods (Roweis, Tanenbaum)
- Mapping methods (Kohonen SOMs)
- Nonlinear scaling methods (kernel PCA, principal curves, autoassociative NNs)

Main Weaknesses

- Linear projection, manifold and mapping seek to *preserve* (not improve) structure
- Nonlinear scaling methods generally expensive, difficult to use
  - Choice of kernel
  - Choice of network design, parameters
  - Difficult to interpret

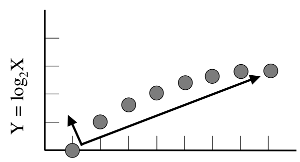
Linear Projection Methods

- Find reduced-dimensionality basis
- Minimize global errors in Euclidean distance
- Examples
  - Principal components analysis (PCA)
  - Johnson-Lindenstrauss methods: random projection approximation of PCA
  - Independent component analysis (ICA)
  - Perceptrons

Linear Projection Methods

- Weaknesses
  - Tries to preserve all structure, good AND bad
  - Linear model is inflexible
- Sample bad case
  - Curves

One or two components?



Nonlinear dependencies not explained or exploited by PCA

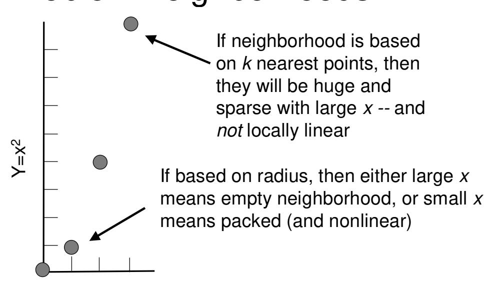
Manifold Methods

- Embed in manifold (curve, surface, et al) of lesser dimensionality
- Focus on preserving local distances
- Example
  - Locally linear embedding (use eigenvectors based on local covariance)

Manifold Methods

- Weaknesses
  - New points require interpolation (no basis)
  - Very dependent on good definition of neighborhood
  - Therefore very dependent on good scaling and parameter definitions
- Sample bad case
  - Polynomial data (highly variable density)

Problem neighborhoods



X=uniform data (0,1,2,3), Y=x^2 (0,1,4,9)

Mapping methods

- Map into lower-dimensional space based on a distance matrix
- Preserve distances (some local, some global)
- Examples
  - Kruskal's multidimensional scaling (global)
  - Kohonen self-organizing maps (local)

Mapping methods

- Weaknesses
  - New points require interpolation (no basis)
  - Do not exploit dimensions
  - Dependent on parameter definitions (dimensionality of embedding space)

Nonlinear Scaling Methods

- Nonlinear transformations permitted
- Error metric varies
- Examples
  - Principal curves
  - Auto-associative neural networks
  - Nonlinear kernel + PCA

Nonlinear Scaling Methods

- Weaknesses
  - May be *ad hoc*; choosing a kernel is difficult, important
  - May be expensive; $O(n^2)$ for principal curves, arbitrary for neural networks
  - May be complex and difficult to interpret

Overview

- Task, motivation and thesis
- State of the art
- My current work
- Proposed work

InfoScale Approach

- Foci
  - Skew
  - Redundancy
- Information theory
  - How to measure skew, redundancy?
- InfoScale
  - How to counter skew, redundancy?

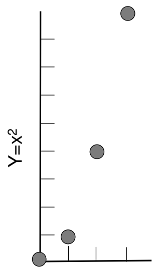
Foci

- *Skew* = uneven distribution
  - Difficult to define neighborhoods
  - Difficult to find rules
  - Generally difficult to use
- *Redundancy* = non-novel data
  - Waste of time, space
  - Impairs data mining accuracy

$y=x^2$ versus $y=x$

- Uniform x, dependent $y=x^2$
- Semantically identical nonlinear principal component (can represent with just x and one equation)
- $y=x^2$ has skewed distribution
- $y=x$ is easier to use

Skew and Neighborhoods



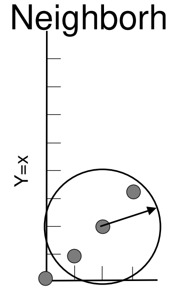
$Y=x^2$ with uniform x

Skewed Y axis --

neighborhoods tough to define (neither *k* nor radius easy to choose)

X=uniform data (0,1,2,3), $Y=x^2$ (0,1,4,9)

Uniform Data and Neighborhoods



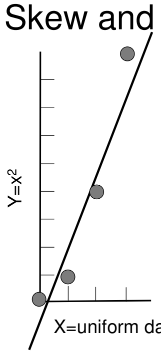
Y=x with uniform x

Same semantic content (can be explained purely via *x*)

Neighborhoods obviously much easier to define

X=uniform data (0,1,2,3), Y=x (0,1,2,3)

Skew and Rules



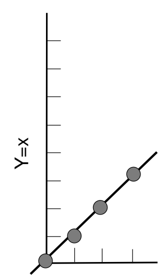
$Y=x^2$ with uniform x

Skewed Y axis, nonlinear relationship --

linear regression unreliable

X=uniform data (0,1,2,3), $Y=x^2$ (0,1,4,9)

Uniform Data and Rules



Y=x with uniform x

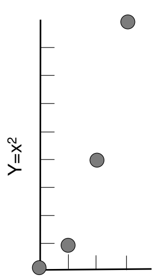
Same semantic content (can be explained purely via *x*)

Linear regression = trivial

Skew = harder to find rules

X=uniform data (0,1,2,3), Y=x (0,1,2,3)

Skew and Efficiency



$Y=x^2$ with uniform x

Skewed Y axis = Wider range (0..9)

Requires numerous bits to encode that range.

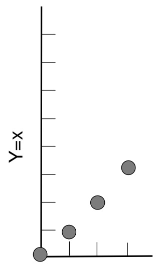
$2\lceil \log_2(n+1) \rceil + 1$ self-delimiting code: 20 bits for Y axis, 14 for X axis

X=uniform data (0,1,2,3), $Y=x^2$ (0,1,4,9)

A Self-Delimiting Code

- Store *m*-bit string S as 1^m0S (2m+1 bits)
- Unambiguous length -- no additional delimiters needed

Uniform Data and Efficiency



X=uniform data (0,1,2,3), Y=x (0,1,2,3)

Y=x with uniform x

Only one attribute needed -- obvious to even simple selection algorithm

Both axes uniform = 14 bits each with self-delimiting code

Redundancy = Undesirable

- Data takes space to store
- Data takes time to process
- Redundancy *hurts accuracy* by overemphasizing factors.

Ex: Financial Data

- 365 daily balances
- 365 daily total-loans-outstanding
- Total value of prior loan defaults
- Balances (highly dependent on each other) overwhelm default total

Choosing a Distance Function

- Given
  - unlabeled vector data (unsupervised)
- Produce
  - An distance function for data mining (isomorphic: better scales) with less skew and less redundancy
- Critical subtask
  - Quantify skew, redundancy

InfoScale Approach

- Foci
  - Skew
  - Redundancy
- Information theory
  - How to measure skew, redundancy?
- InfoScale
  - How to counter skew, redundancy?

Background: Information Theory

- Entropy
  - Mathematical definition
  - Intuition
- Relevance
  - Uniformity and Independence
  - Marginal Information Criterion (MIC)

Information Theory

- Shannon information
  - (entropy) $H(X) = -\Sigma p(x=x_i)\log_2 p(x=x_i)$
  - (differential entropy; requires pdf) $H(X) = -\int f(\varsigma)\log_2 f(\varsigma)d\varsigma$
- Intuitive meaning
  - $H(X)$ = expected bits needed to identify instance of random variable X given knowledge of distribution

Information Efficiency

- *n* bits of **input** does not mean *n* bits of **content**
- Fixed dimensionality + fixed precision = fixed bits of input
- If inefficient
  - Drop attributes (reducing bits in)
  - Transform (increasing bits of content)

Adding information?

- info + transformation = more information?
  - Might seem illogical or arbitrary -- transformation is something user provides, not data
  - Consider it EXPRESSION of information. Same semantic content, just easier to use. Pointless given perfect machine learners, but those don't exist.

Entropy Examples: Uniformity

- Fair coin: 50/50 heads/tails = 1 bit
- Rigged: 100/0 heads/tails = 0 bits

- Uniform distribution = less predictability = more information

Entropy Examples: Independence

- Two independent fair coins: 2 bits
- Flip both; if match, reverse one with equal probability: 1 bit

- Independence = less redundancy = more information

Independence and Uniformity

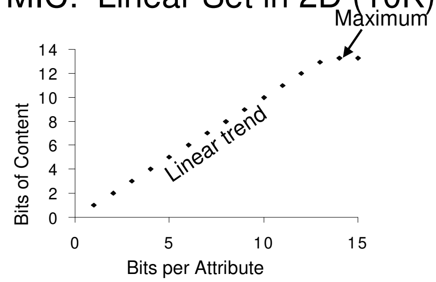
- Independence = opposite of redundancy.
- Uniformity = opposite of skew (skew = relative sparsity)

- Redundancy and skew are serious problems facing distance functions.

MIC: Marginal Information Content

- MIC = information contained per bit of input per attribute
  - High MIC = attributes are informative
  - Low MIC = inefficient (skew or redundancy)
- Meaningful for self-similar sets

MIC: Linear Set in 2D (10K)



MIC (slope) = 1 (= intrinsic dimensionality)

Intrinsic Dimensionality

- What: intuitive definitions
- How: estimation procedure
- Why: connection to Shannon information and InfoScale

Intrinsic Dimensionality: What

- Machine Learning: Number of components required to model data
- Intuitive definitions
  - Degrees of freedom
  - Number of latent factors

Intrinsic Dimensionality: Why

- D_1 estimable in O(mn), n=cardinality, m=dimensionality for self-similar sets
- D_1 = <u>marginal information content</u> (MIC)
  - In: 1 additional bit per attribute
  - Out: D_1 additional bits of content (within limits)
- More robust than entropy

Why MIC over Entropy?

<u>Entropy</u>	<u>MIC</u>
Parameter sensitive	Parameter robust
Hard to compare between scales	*Rate* of information robust enough for comparison
Not self-verifying	If it doesn't apply, you'll know

Intrinsic Dimensionality: How

- Self-similar ⇒ information linearly proportional to bits of precision
- *r*, discretization hypergrid width
- $H_r(X)$, information of *X* discretized via *r*
- Information fractal dimension

$$D_1(X) = \frac{-\partial H_r(X)}{\partial \log_2 r}$$

Choosing a Distance Function

- Given
  - unlabeled vector data (unsupervised)
- Produce
  - An distance function for data mining (isomorphic: better scales) with higher marginal information content
- Critical subtask
  - Produce better scales

InfoScale Approach

- Foci
  - Skew
  - Redundancy
- Information theory
  - How to measure skew, redundancy?
- InfoScale
  - How to counter skew, redundancy?

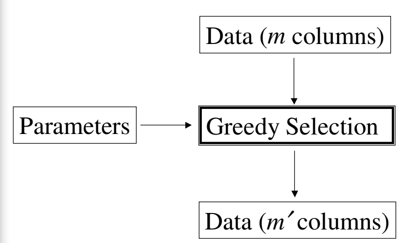
Exploiting MIC

- Transform data (isomorphic to transforming distance function)
- If more uniform, more novel -- more information and greater MIC
- Principle: Make every bit count!

Version 1: (SBBD 2000) Selection

- Objective:
  - Detect and reduce redundancy
  - Dimensionality reduction (only): $\Re^m \Rightarrow \Re^{m'}$
- Method:
  - Use fractal dimension (~ MIC) to guide greedy attribute selection

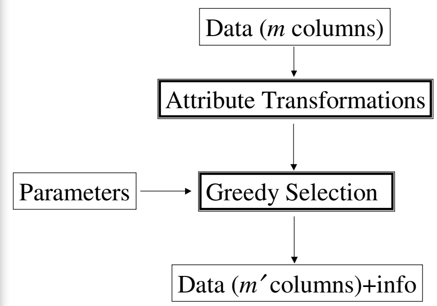
Version 1: Selection



Version 2: (KDD 2002)

- Objective:
  - Detect and reduce skew, redundancy
  - Dimensionality reduction + nonlinear scaling $\Re^m \Rightarrow \Re^{m'}$
- Method:
  - Transform attributes to reduce skew
  - Select from transformed versions

Version 2: Scaling+Selection



Greedy Selection

- Inputs: Data, termination criteria (max attributes and/or minimum MIC gain)
- Procedure: Iterative greedy selection
  - State: Set of retained attributes
  - Can add one attribute per iteration
  - (Version 2 only) Can exchange attributes
- Uninformative $\Rightarrow$ low priority

Attribute Scaling

- Apply nonlinear transformations to fix skew
- Apply normalization to fix disparate ranges
- All* transformed versions of all attributes are candidates for selection (*subject to change)

Nonlinear scaling

- Counter skew with nonlinear scaling
  - Affine transformations preserve single-attribute distribution; useless
- Transform individual attributes
- Choice of transformations important
  - Applicability
  - Computational cost

Exploit Common Distributions

- CDF_X(data from distribution X) = uniform mapping to [0,1]
- Examples used in current work

Gamma	Pareto
Gaussian	Uniform
Lognormal	

Empirical tests

- Impact on dimensionality, MIC
- MIC changes reflect redundancy
- Effects on skew, rule-finding

Impact on Dimensionality

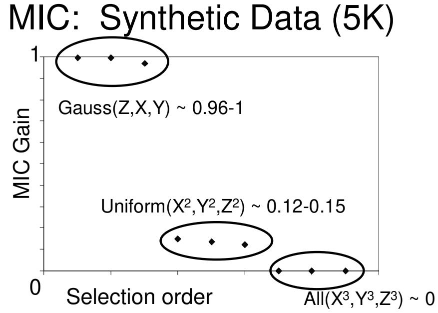
Data set	Before dims/MIC	After dims/MIC
baseball	17/2.05	6/3.70
basketball	45/1.63	8/4.82
CIA-2001	2/0.259	2/1.57
page-blocks	11/1.67	5/3.29

After scaling = more efficient storage, more accessible data

Redundancy Detection Example

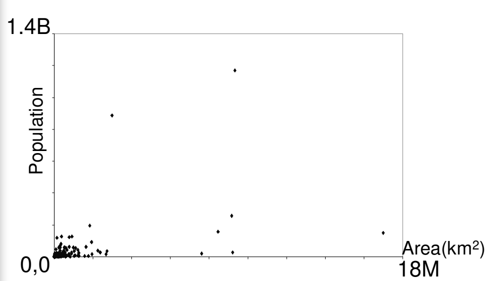
- Nine attributes, 5K points
  - X, Y, Z: IID normal, μ=0, σ=1
  - X^2, Y^2, Z^2, X^3, Y^3, Z^3
- Version 2 nonlinear scaling + selection with no thresholds (select as much as possible)
- Dependent $\Rightarrow$ redundant $\Rightarrow$ less MIC gain

MIC: Synthetic Data (5K)



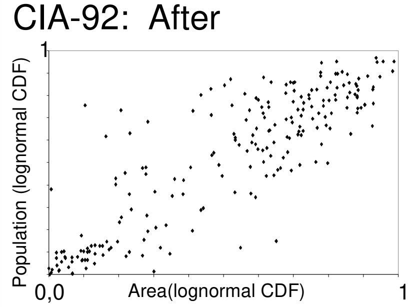
Dependent attributes ARE penalized by MIC

CIA-92: Before



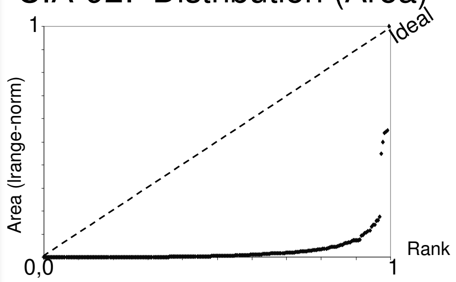
Outliers dominate; correlation=0.4776; unusable

CIA-92: After



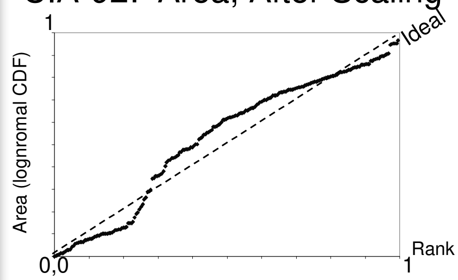
Much more uniform; correlation=0.8636

CIA-92: Distribution (Area)



Most countries' areas within small part of range

CIA-92: Area, After Scaling



Not uniform, but clearly much more so

Overview

- Task, motivation and thesis
- State of the art
- My current work
- Proposed work

Proposed Extensions

- Reconstructing dropped attributes
- MML/MDL assessment
- Selection algorithm improvements
  - Curing myopia
  - Performance improvements
- Comparison versus competitors

Reconstruction

- Why?
  - Rules are goal of data mining
  - Allows evaluation of scales
- Model dropped attributes using available data
- Requires dependency analysis, modeling methods (CART, NN et al)

Minimum Description Length

- Occam's Razor bias: pro-simplicity
- Total cost -- data, parameters, models, error correction
- Answers "How good is this scaling?"

Selection - Curing Myopia

- Greedy selection vulnerable to local maxima
- Possible replacement: genetic programming
  - Fitness function based on total cost?
  - Depends on complexity of cost computation (modeling)

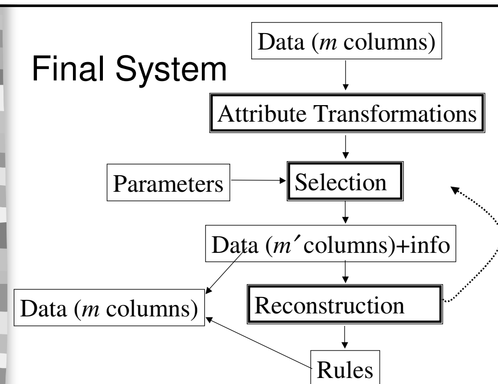
Selection - Performance

- Stage 2 linear in transformations *T*, cardinality *n*, dimensionality *m*; quadratic in iterations *k*
- Eliminate factor of *T* by using *m*, not *mT* candidates for selection
- Should allow for larger *T*, such as including truncated distribution forms

Competitive Evaluation

- Closest system - SPARTAN (Lossy Bayesian-guided selection+CART reconstruction; no scaling, m^4 max-weighted clique)
- Measure total cost

Final System



Schedule

Improve selection speed	T + 2 weeks
Study other transformations	T + 6-8 weeks
Implement basic linear(?) model	T + 10-14 weeks
Design, implement, test GP w/ reconstruction-based fitness	T + 20-26 weeks

Schedule

Implement, test basic NN model	T + 26-34 weeks
Research additional complex models -- CART?	T + 34-42 weeks
Competitive evaluation	T + 42-50 weeks
Case studies -- study utility of rules (secondary effects)	T + 46-58 weeks

Aside: FALCON (VLDB 2000)

- Will be in final thesis document
- Example-driven relevance-feedback system
- Not discussed today
  - Largely finished (negative feedback)
  - Role expected to be advisory