Statistical Classification

UW Computational Linguistics, February 11 2005

presented by Jeremy G. Kahn

The Classification Problem (vs. Prediction)

Prediction requires a value in a space — off by how much?

Classification needs a hard decision. $x \in \{A, B, C\}$?

Error metrics are not equivalent.

Prediction can be used for classification, but vice versa is more difficult.

Bayesian decisions

Suppose we know *priors* and *distributions* (both *shape* and *parameters*) of the two classes (A and B) that we're interested in.

$p(A|X)$ and $p(B|X)$

given $x \in X$ — predict A or B.

How to do classification? Bayesian logic:

$$d(x) = \left\{ \begin{array}{ll} A & p(B|x) < p(A|x) \\ B & p(A|x) < p(B|x) \end{array} \right\}$$

But note the assumptions here.

Decision models

If we know lots about the priors and distributions, the decision function may be arbitrarily complex.

(illustration: multimodal function)

(illustration: higher dimensional variables: x need not be a scalar. . .)

But in the usual case: **we don't have perfect knowledge**. We're *modeling*, which always discards some information, and we have *limited training data*, which limits our approximation.

Decision model complexity

(Under some circumstances,) we'd like our models to be highly constrained:

- stronger statement about their predictions
- search space smaller
- increases *bias*, decreases *variance*

Vapnik-Chernovenkis Dimension

There's a measure for how (un)constrained a model is:

The **Vapnik-Chernovenkis (VC) Dimension** describes the number of (ordinary-placement) points that can be separated by a given class of decision function.

VC dim of:

- a point in a 1-d input space?
- a sum of sines in 1-d input space?
- a line in a 2-d space?
- a plane in a 3-d space?
- a n-plane in an $n+1$ space?

(does anybody feel like we're in the Baxter Building? The Fortress of Solitude?)

Modeling approaches

Nearest Neighbor “models” compare the input point to (all?) training data and vote.

Generative models compare the input point to the *best approximation of the generating function* and select the model that assigns the highest likelihood.

Discriminative models compare the input point to the *best approximation of the boundary between the classes* and decide which side it lies on.

k-Nearest Neighbor "modeling"

- Not modeling the distribution (all modeling assumptions are in the dimensionality of the space)
- Voting model, based on k nearest neighbors' vote. This requires some model of "nearest" in the input vector X.
- votes may be weighted; likewise, k may be "all those within a certain range"; these design decisions are called *kernel functions*. (Strictly speaking, kernel methods are a superclass of k-NN methods.)

Question: equal-weight voting and $k = N$ is what kind of decision function?

Problems with kernel methods

- Dimensions may not be easily intraconvertible. [Consider input dims for phonetic voicing as a category!]
- What happens as the number of dimensions increases without increasing the input data?
- What is the k neighborhood? How many items in the training set must be examined to classify a new input point?
- What is the VC dimension of the k-NN model?

Generative models

Assume or discover the *shape* of the distribution:

- Gaussian (normal)
- exponential
- uniform
- etc...

Usually, we pick a well-behaved, unimodal distribution. This decision is not always theoretically well-founded.

Statistical t-tests rely on assumption of Gaussian.

Learning generative models

- a closed-form solution exists under certain distributions
- estimating the parameters can be very tricky; need enough data to get good guesses for:
  - priors
  - model parameters (for Gaussian: mean, variance)
- what about multimodal classes?

What is the VC dimension of two Gaussians in 1-d:

- with equal variance and priors?
- with equal variance and *different* priors?
- with *different* variance and equal priors?

Problems with generative models

- working out the right generative model: Gaussian assumption may not hold
- breakdowns in high dimension
- mixture models not closed-form

Discriminative models

- computing all the distributions just to get to a separation decision can be a lot of (fragile) work
- Not modeling distributions: modeling *boundary between* distributions
- a "bad" attitude: focuses on difficult cases. (Begs the question — how do we define the difficult cases?)

VC dimension is ultimately the number of classes. (2 for now).

Math for "edge"

The learned edge in this case is (a generalization of) *Vapnik's optimal separating hyperplane*[a].

Kernel-like: We want to determine a collection of *support vectors* that have as much distance as possible to the hyperplane.

best plane p is the one that maximizes the distance to the closest i points. Each of the i points is a *support vector*:

$$\hat{p} = \arg\max_p \sum_{x \in \arg\min_i d(p,i)} d(p,x)$$

(Illustration: focuses on edge points)

[a] If I remember the *Player's Handbook* correctly, VOSH is a 9th level illusionist spell.

Support vector machines (SVMs)

if idea of "distance" is sufficiently general, doesn't suffer from dimensionality problems.

No magic bullets: if the data are not separable in the model, this *will* fail.

Still can be vulnerable to high dimensionality, especially if there's redundancy. (Much more robust than (e.g.) k-NN, though.)

Additional directions

- multi-class SVMs, fancy kernels, etc.:
  `http://svmlight.joachims.org/`
- complex structures?
- other discriminative models