Distance measures as prior probabilities

Thomas P. Minka

November 15, 2000

Abstract

Many learning algorithms, especially nonparametric ones, use distance measures as a source of prior knowledge about the domain. This paper shows how the work of Baxter and Yianilos provides a formal equivalence between distance measures and prior probability distributions in Bayesian inference. The prior distribution applies either to how the data was generated or to the shape of the discrimination boundary. This perspective is useful for extending distance-based algorithms to new feature spaces and especially for *learning* distance measures on those spaces.

1 The canonical distance measure

The canonical distance measure (CDM), developed by Baxter (1995; 1997) and further generalized here, provides the link explored in this paper between distance measures and prior probability distributions. Other links are possible but not emphasized here. Baxter developed the CDM in the context of "learning how to learn," where there are a series of related tasks that need to be solved. This paper shows that the CDM can be applied in a wider variety of situations, such as when there is only one task to solve. But Baxter's perspective is still the simplest way to understand and derive the CDM, so the paper starts by reviewing his derivation.

Baxter focused on function approximation, which is a perfectly general scenario but may be confusing for those interested in classification. Therefore this section reinterprets Baxter's argument in the form of a thought experiment about building a classifier. What is useful about this thought experiment is that it forces us to convert a prior probability distribution on tasks into a distance measure for nearest-neighbor classification. The solution to the thought experiment necessarily provides a bridge between priors and distances.

The problem:

> You've started a company called "Classifiers R Us" and you've been hired to write a program for 1-nearest-neighbor classification. As you are trying to determine what distance measure to use, your client reveals that the program will always be run on one of 100 classification tasks, where the true classifications are known. Each task uses the same measurement space, say the physical attributes of a person, but one task is about determining gender while another is about determining occupation. Of course, the program gets none of this information: it only gets a set of vectors

labeled with class indices. Without knowing which task your program will face, or what training data it will get, how can you use this information to choose a distance measure?

Suppose the classifier is tested at measurement x_1 and it identifies x_2 as the nearest neighbor, so that it classifies x_1 the same way as x_2. What is the probability of error? Since the true classification function must be one of $f_1, ..., f_{100}$, the probability of error is

$$d(x_1, x_2) = \frac{1}{100} \sum_{i=1}^{100} 1 - \delta(f_i(x_2) - f_i(x_1)) \tag{1}$$

where $f_i(x)$ is the true class of x for task i. This is simply the fraction of times that x_1 and x_2 belong to different classes. What distance measure should the classifier use? Why, $d(x_1, x_2)$ itself! That way the classifier always chooses x_2 to minimize the probability of error. Any other distance measure must have higher probability of error. (Note that even if there was only one task, the error rate might be nonzero. This is because we are limited to a 1-NN classifier.)

What is fascinating about this result is that no other information is required to determine the distance measure. Once you have a list of the possible classifier functions, the details of the measurement space are irrelevant: the measurement could be weight in pounds, a photograph, or a sound sample of laughter. The known classifier functions automatically embed the relevant information about the measurement space, such as invariances to lighting and background noise.

We can generalize this result to the case when performance is measured by a loss function $R(a|b)$, which is the cost of classifying something as a when it really is b. In the previous case, $R(a|b)$ was $1 - \delta(a - b)$. If the classifier picks x_2 as the nearest neighbor to x_1, then the expected cost is

$$d(x_1, x_2) = \frac{1}{100} \sum_{i=1}^{100} R(f_i(x_2)|f_i(x_1)) \tag{2}$$

which is the optimal distance measure in this case.

It is easy to extend this argument to the case where we have a distribution over tasks, not necessarily discrete. In that case, the average becomes an expectation:

$$d(x_1, x_2) = E_f[R(f_i(x_2)|f_i(x_1))] \tag{3}$$

This is the "canonical distortion measure" as Baxter defined it. It is canonical because it is uniquely determined by the distribution over tasks and the loss function. Using a particular distance measure is equivalent to assuming a particular distribution over tasks and loss function. If $R(a|b) = 1 - \delta(a - b)$ then (3) is simply the probability that x_1 and x_2 are in a different class.

Let's consider a further generalization of (3). Often the true classification corresponding to a measurement x is ambiguous. For example, the gender of a person cannot be computed from their height, even with a complete understanding of biology and an infinite amount of training data. In this case, instead of a classifier function $f(x)$, we have a probability distribution over classes: $f(c|x)$. This is the true fraction of times that a measurement x would have class c. In the previous case, $f(c|x)$ was $\delta(c - f(x))$. If the classifier picks x_2 as the nearest neighbor to x_1, then the expected cost is

$$d(x_1, x_2) = E_f[\sum_{a,b} R(a|b) f(a|x_2) f(b|x_1)] \tag{4}$$

In this formula, we sum over all the labels a that x_2 might have and all of the true classifications b of x_1. For example, if we had

$$f(a|x_2) = \begin{cases} .9 & \text{male} \\ .1 & \text{female} \end{cases} \quad f(b|x_1) = \begin{cases} .8 & \text{male} \\ .2 & \text{female} \end{cases} \tag{5}$$

then the probability of error is $(.9)(.2) + (.1)(.8) = .26$ for that f. Equation (4) is the definition of "canonical distance measure" used in this paper. When $R(a|b) = 1 - \delta(a - b)$ it retains the interpretation as the probability that x_1 and x_2 are in a different class.

An interesting thing to note about this distance measure is that it is generally not a metric in the formal sense: it may not satisfy the triangle inequality, it may not be symmetric, and it may not be self-minimal. Since the CDM is provably optimal, this implies that the metric axioms have no special status for 1-nearest-neighbor classification. (Similarity measured by humans has also been shown to not have metric properties, and it is plausible that a CDM-like argument is behind it.)

As an example of a measure which is not self-minimal, consider the case (5). For task f, the measurement x_1 should be closer to x_2 than to itself, because the probability of error would otherwise be $(.8)(.2) + (.2)(.8) = .32$. Intuitively, if x_1 is near a class boundary, we don't want to match it with another point near the boundary, whose label is noisy. Instead we should favor points whose true classification is more certain. More discussion of this phenonenon can be found in Yianilos (1995).

Baxter's definition (3), by the way, is always self-minimal for a reasonable loss function.

The CDM can also be motivated by a cross-validation argument. Instead of being given 100 known classifier functions, suppose you are given a huge amount of labeled data for the 100 tasks. One approach is to estimate the fraction $f_i(c|x)$ from the data on the ith task, then apply the CDM formula (4). Another approach is cross-validation: pick the distance measure which has lowest misclassification cost on the training data. The cost is that incurred by reclassifying each datum based on its nearest neighbor. But that cost is exactly (4), using empirical counts to estimate $f_i(c|x)$. Therefore the two methods are identical and the CDM is the unique distance measure which minimizes cross-validated cost.

2 Estimating the CDM for one task

The CDM formula (4) and its cross-validation interpretation is perfectly valid even when there is only one task. This suggests that we might be able to learn a distance measure in the process of solving a particular task.

If there is only one task, then the CDM follows immediately from the density $p(c|x)$ for that task. So the method is to estimate $p(c|x)$ and then compute the CDM. Of course, $p(c|x)$ could be used as a classifier, but here it is used only to derive a metric for nearest-neighbor classification. This provides robustness to modeling assumptions and estimator quality. For two classes and a 0/1 loss function, the CDM would be

$$d(x_1, x_2) = p(c = 1|x_1)p(c = 2|x_2) + p(c = 2|x_1)p(c = 1|x_2) \tag{6}$$

To better understand this procedure, consider what happens if we estimate $p(c|\mathbf{x})$ by assuming each class-conditional density $p(\mathbf{x}|c)$ is Gaussian with variance $\mathbf{V}$:

$$\begin{aligned} p(\mathbf{x}|c) &= \mathcal{N}(\mathbf{x}; \mathbf{m}_c, \mathbf{V}) && (7) \\ p(c = 1|\mathbf{x}) &= \frac{p(\mathbf{x}|c = 1)p(c = 1)}{p(\mathbf{x})} && (8) \\ &= \frac{1}{1 + \exp(-\mathbf{a}^{\mathrm{T}}\mathbf{x} + b)} && (9) \\ \mathbf{a} &= \mathbf{V}^{-1}(\mathbf{m}_1 - \mathbf{m}_2) && (10) \end{aligned}$$

The class probability only changes with movement along $\mathbf{a}$, so the resulting CDM will only measure distance along $\mathbf{a}$. The vector $\mathbf{a}$ is the "Fisher feature" projection for this task, demonstrating that Fisher features are a special case of the CDM.

A variation on this method is to estimate $p(c|x)$ locally to derive a local CDM. The local CDM is then used to classify x_1. The advantage of this method is that a simpler density estimate can be used. For example, the assumption that $p(x|c)$ is Gaussian will be more valid locally than globally. Using Gaussian densities locally results in a local Fisher feature projection, a technique used by Short & Fukunaga (1981) and Hastie & Tibshirani (1996) (each with a different motivation).

Another variation is to estimate $p(c|x)$ one feature at a time. If the measurement vector is $\mathbf{x} = [x_1, x_2]$, then we estimate $p(c|x_1)$ and $p(c|x_2)$ rather than $p(c|\mathbf{x})$. This significantly simplifies the modeling task: if measurements are discrete than we might simply count the fraction of times that a point with $x_1 = z$ has class c to get $p(c|x_1 = z)$. From these estimates we get a CDM along each feature dimension. These distance measures are all in the same units, the units of the loss function, so any of the usual distance combination rules can be used to get a distance on the whole space. This approach is related to the "value distance measure" used by Stanfill & Waltz (1986).

3 Estimating a task distribution

Another way to learn a distance measure is to estimate the task distribution. The technique, pioneered by Yianilos (1995), is to treat each class in each task as an exchangeable draw from a common class distribution. This is typical, for example, in face recognition, where each face class is a random sample from the distribution of faces, regardless of the task it is part of. An exception would be if the faces where somehow chosen to make the task difficult or easy. Under the exchangeability assumption, estimating the task distribution reduces to estimating the common class distribution, which can be done from one task as long as there are enough classes present.

One approach is to model the class prior as Gaussian. That is, we sample from a Gaussian to get class parameters, then we sample from each class to get data. If the classes are themselves Gaussian, then this is a standard hierarchical model used in statistics, which can be estimated via empirical Bayes techniques. The CDM then follows from the Gaussian class prior and corresponds to Mahalanobis distance. One crude method is to estimate the class parameters by maximum likelihood and then, assuming these are the true parameters, fit a Gaussian prior. This is clearly suboptimal for finite data but often works. Interestingly, deriving a CDM in this way is equivalent to the technique of "Fisher features" (linear discriminant analysis), as shown in appendix A.

However, we are not limited to Gaussian priors. Much more flexibility is afforded by using a mixture of Gaussians. That is, we sample from a mixture of Gaussians to get each class parameter. This is the model used by Yianilos (1995). In practice, he uses the crude method of assuming the class estimates are true and fitting a mixture of Gaussians to them. The resulting CDM is a mixture of different metrics, corresponding to the different Gaussians in the prior. Instead of one global Mahalanobis distance, we have a combination of Mahalanobis distance measures, with the combination weights varying spatially. Details are given by Yianilos (1995).

The CDM formula simplifies as follows. Let $p(\theta)$ be the class prior and let $p(x|\theta)$ be the data distribution within a class. (Note that we are not assuming that the classes come from a common parametric family; θ can perfectly well select among different families.) If $R(a|b) = 1 - \delta(a - b)$, then we want to compute the probability that two points (x_1, x_2) might be from the same class. Let $p(\text{same})$ be the prior probability of this event, which depends on how (x_1, x_2) are sampled and on the class sizes. For example, if points are drawn randomly from two equal-size classes then $p(\text{same}) = 1/2$. The posterior probability of the event is therefore

$$
\begin{aligned}
p(\text{same}|x_1, x_2) &= \frac{p(x_1, x_2|\text{same})p(\text{same})}{p(x_1, x_2|\text{same})p(\text{same}) + p(x_1, x_2|\text{different})p(\text{different})} && (11) \\
&= \frac{1}{1 + \frac{p(x_1, x_2|\text{different})}{p(x_1, x_2|\text{same})}\frac{p(\text{different})}{p(\text{same})}} && (12)
\end{aligned}
$$

Since distance measures are invariant to monotonic transformations, the CDM is equivalent to an evidence ratio:

$$d(x_1, x_2) = \frac{p(x_1, x_2|\text{different})}{p(x_1, x_2|\text{same})} \tag{13}$$

$$\text{where } p(x_1, x_2|\text{different}) = \left(\int_\theta p(\theta)p(x_1|\theta)\right)\left(\int_\theta p(\theta)p(x_2|\theta)\right) \tag{14}$$

$$p(x_1, x_2|\text{same}) = \int_\theta p(\theta)p(x_1|\theta)p(x_2|\theta) \tag{15}$$

Yianilos (1995) used essentially this formula, except he assumed $p(x_1, x_2|\text{different})$ was constant and ignored it. Unfortunately, under his rule, points which have high marginal probability, i.e. are more typical, have their distance artifically decreased. This means that classes which are more typical under the prior will be favored in the classification process.

Equation (13) was used as a distance measure by El-Yaniv et al. (1997), where it was estimated using compression lengths. Note that the paper has a typographical error in the left side of equation (2) which omitted the $p(x_1, x_2|\text{different})$ term (that term doesn't get much respect, apparently). The class prior was not learned but fixed by the choice of compression algorithm. Also, the measure was not used for classification but for clustering, so the argument for using the CDM given here does not apply.

The evidence ratio (13) and its connection to information theory is discussed by Minka (1999).

A Fisher discriminant analysis as a prior on Gaussians

Linear discriminant analysis (LDA) was originally devised as a fast but crude way to estimate a decision boundary. Nowadays, we have more accurate methods for this, but LDA lives on as a simple technique for dimensionality reduction. After extracting Fisher features, many classification techniques, such as nearest neighbor, can be used in the reduced feature space.

Our generative model is that first a mean $\mathbf{m}$ is sampled from the distribution

$$p(\mathbf{m}) = \mathcal{N}(0, \mathbf{V}_0) \tag{16}$$

and then a data set $D = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ is sampled from the Gaussian distribution with parameters $(\mathbf{m}, \mathbf{V})$:

$$p(D|\mathbf{m}) = \prod_i p(\mathbf{x}_i|\mathbf{m}) \tag{17}$$

$$p(\mathbf{x}|\mathbf{m}) = N(\mathbf{m}, \mathbf{V}) \tag{18}$$

The CDM follows from the evidence ratio

$$R = \frac{p(D)}{\prod_i p(\mathbf{x}_i)} \tag{19}$$

$$\text{where } p(\mathbf{x}) = \int_{\mathbf{m}} p(\mathbf{x}|\mathbf{m})p(\mathbf{m}) \quad (20)$$

$$p(D) = \int_{\mathbf{m}} p(D|\mathbf{m})p(\mathbf{m}) \quad (21)$$

(In practice, D will be the two points that we want to measure the distance between.) The denominator of the ratio is

$$p(\mathbf{x}) = \mathcal{N}(0, \mathbf{V} + \mathbf{V}_0) \quad (22)$$

and the numerator is

$$p(D|\mathbf{m}) = N(m; \bar{x}, \mathbf{V}/N)\frac{|2\pi\mathbf{V}/N|^{1/2}}{|2\pi\mathbf{V}|^{N/2}}\exp(-\frac{1}{2}\mathrm{tr}(\mathbf{S}\mathbf{V}^{-1})) \quad (23)$$

$$p(D) = \mathcal{N}(\bar{x}; 0, \mathbf{V}/N + \mathbf{V}_0)\frac{|2\pi\mathbf{V}/N|^{1/2}}{|2\pi\mathbf{V}|^{N/2}}\exp(-\frac{1}{2}\mathrm{tr}(\mathbf{S}\mathbf{V}^{-1})) \quad (24)$$

Suppose all eigenvalues of $\mathbf{V}_0$ are either zero or infinity. Define the joint diagonalizer $\mathbf{R}$ to satisfy

$$\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{V} \quad (25)$$

$$\mathbf{R}\mathbf{E}\mathbf{R}^{\mathrm{T}} = \mathbf{V}_0 \quad (26)$$

where $\mathbf{E}$ is diagonal with elements zero or infinity. We can simplify matters by changing variables to

$$\tilde{\mathbf{x}} = \mathbf{R}\mathbf{x} \quad (27)$$

$$\tilde{\bar{\mathbf{x}}} = \mathbf{R}\bar{\mathbf{x}} \quad (28)$$

$$\tilde{\mathbf{S}} = \mathbf{R}\mathbf{S}\mathbf{R}^{\mathrm{T}} \quad (29)$$

The Jacobian of this transformation is $|\mathbf{V}|^{N/2}$. This gives

$$p(\tilde{\mathbf{x}}) = \mathcal{N}(0, \mathbf{I} + \mathbf{E}) \quad (30)$$

$$p(\tilde{D}) = \mathcal{N}(\tilde{\bar{\mathbf{x}}}; 0, \mathbf{I}/N + \mathbf{E})\frac{\exp(-\frac{1}{2}\mathrm{tr}(\tilde{\mathbf{S}}))}{(2\pi)^{(N-1)d/2}N^{d/2}} \quad (31)$$

The diagonal matrix $(\mathbf{I} + \mathbf{E})^{-1}$ will only have elements which are zero or one:

$$(\mathbf{I} + \mathbf{E})^{-1} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (32)$$

$$(\mathbf{I}/N + \mathbf{E})^{-1} = N\begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (33)$$

so

$$\prod_i p(\tilde{\mathbf{x}}_i) = \frac{1}{|2\pi(\mathbf{I}+\mathbf{E})|^{N/2}} \exp(-\frac{1}{2}\tilde{\mathbf{X}}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tilde{\mathbf{X}}^{\mathrm{T}}) \tag{34}$$

$$\mathrm{tr}(\tilde{\mathbf{S}}) = \tilde{\mathbf{X}}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tilde{\mathbf{X}} - N\bar{\tilde{\mathbf{x}}}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\tilde{\mathbf{x}}} + \tilde{\mathbf{X}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tilde{\mathbf{X}} - N\bar{\tilde{\mathbf{x}}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\tilde{\mathbf{x}}} \tag{35}$$

$$\frac{p(\tilde{D})}{\prod_i p(\tilde{\mathbf{x}}_i)} = |\mathbf{I}+\mathbf{E}|^{(N-1)/2} \exp(-\frac{1}{2}\left(\tilde{\mathbf{X}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tilde{\mathbf{X}} - N\bar{\tilde{\mathbf{x}}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\tilde{\mathbf{x}}}\right)) \tag{36}$$

The exponent is the scatter along the dimensions spanned by $\mathbf{V}_0$, which is what we wanted to show.

References

Baxter, J. (1995). *Learning internal representations.* Doctoral dissertation, The Flinders University of South Australia.
`ftp://archive.cis.ohio-state.edu/pub/neuroprose/Thesis/baxter.thesis.ps.Z`.

Baxter, J. (1997). The canonical distortion measure for vector quantization and approximation. *ICML*. `http://wwwsyseng.anu.edu.au/~jon/papers/icml97.ps.gz`.

El-Yaniv, R., Fine, S., & Tishby, N. (1997). Agnostic classification of Markovian sequences. *NIPS* (pp. 465–471). MIT Press.
`http://www.cs.huji.ac.il/labs/learning/Papers/MLT_list.html`.

Hastie, T., & Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Pattern Analysis and Machine Intelligence*, *18*, 607–616.
`http://www-stat.stanford.edu/~hastie/Papers/dann.ps`.

Minka, T. P. (1999). Bayesian inference, entropy, and the multinomial distribution.
`http://vismod.www.media.mit.edu/~tpminka/papers/multinomial.html`.

Short, R. D., & Fukunaga, K. (1981). The optimal distance measure for nearest neighbor classification. *IEEE Trans Info Theory*, *27*, 622–627.

Stanfill, C., & Waltz, D. L. (1986). Toward memory-based reasoning. *Communications of the ACM*, *29*, 1213–1228.

Yianilos, P. N. (1995). Metric learning via normal mixtures.
`http://www.neci.nj.nec.com/homepages/pny/papers/mlnm/`.