Breiman, L. (1996). Bias, variance, and arcing classifiers. Unpublished manuscript.

Buntine, W. , and Weigend, A. (1991). "Bayesian back-propagation". *Complex Systems*, **5**, 603-643.

Friedman, J. (1996). On bias, variance, 0/1-loss, and the curse of dimensionality. Unpublished manuscript.

Geman, S. et al. (1992). "Neural networks and the bias/variance dilemma". *Neural Computation,* **4**, 1-58.

Kohavi, R. and Wolpert, D. (1996). "Bias plus variance for zero-one loss functions". Submitted.

Kong E. B. and Dietterich, T. G. (1995). "Error-correcting output coding corrects bias and variance", Proceedings of the 13th international conference on Machine Learning, 314-321, Morgan Kauffman.

Perrone, M. (1993). Improving regression estimation: averaging methods for variance reduction with extensions to general convex measure optimization. Ph.D. thesis, Brown University Physics Dept.

Tibshirani, R. (1996). Bias, variance, and prediction error for classification rules. Unpublished manuscript.

Wolpert, D. (1994). "Filter Likelihoods and Exhaustive Learning". In *Computational Learning Theory and Natural Learning Systems II*, S. J. Hanson et al. (Eds), 29-50, MIT Press.

Wolpert, D. (1995). "The relationship between PAC, the statistical physics framework, the Bayesian framework, and the VC framework". In *The Mathematics of Generalization*, D. Wolpert (Ed.), Addison-Wesley.

Wolpert, D. (1996). "The lack of a priori distinctions between learning algorithms", and "The existence of a priori distinctions between learning algorithms". *Neural Computation*. In press.

Wolpert, D., and Kohavi, R. (1996). "The mathematics of the bias-plus-variance decomposition for zero-one loss functions". In preparation.

the case here, it arises due to the fact that $P(y_H \mid y_F, h, f, q) = P(y_H \mid h, q)$. Similarly, in [Wolpert 1995a] it is shown that because $P(h \mid d, f) = P(h \mid d)$, $E(C \mid d)$ is a non-Euclidean inner product between the (H-indexed) vector $P(h \mid d)$ and the (F-indexed) vector $P(f \mid d)$.

2. The latter point follows from the following identities:

For the optimal algorithm, whose guessing is governed by a delta function in **Y**, the variance is given by

$$\Sigma_d P(d \mid m, q) \ [E(Y_H{}^2 \mid d, q) \ - \ E^2(Y_H \mid q, m)]$$
$$=$$
$$\Sigma_d P(d \mid m, q) \ [E^2(Y_F \mid d, q) \ - \ E^2(Y_F \mid q, m)].$$

In addition, the covariance is given by

$$\Sigma_{d,y_F,y_H} P(y_H \mid d, q) \, P(y_F \mid d, q) \, P(d \mid m, q) \times [y_H - E(Y_H \mid q, m)] \times [y_F - E(Y_F \mid q, m)]$$
$$=$$
$$\Sigma_d P(d \mid m, q) \times [E(Y_F \mid d, q) - E(Y_F \mid q)] \times [E(Y_F \mid d, q) - E(Y_F \mid q)].$$

3. Note that $E(C \mid f, H = E(H \mid f, m, q), q)$ for quadratic loss is $\Sigma_{y_H,y_F} L(y_H, y_F) \, f(q, y_F) \int dh \, P(h \mid d) \, P(d \mid f, m) \, h(q, y_H)$. However this just equals $E(C \mid f, m, q)$, rather than (as for logarithmic scoring) $E(C \mid f, m, q)$ minus a variance term. This difference between the two cases reflects the fact that whereas expected error for loss functions is linear in h in general, expected error for scoring rules is not.

Acknowledgments. I would like to thank Ronny Kohavi, Tom Dietterich and Rob Schapire for getting me interested in the problem of bias-plus-variance for non-quadratic loss functions, and Ronny Kohavi and David Rosen for helpful comments on the manuscript. This work was supported in part by the Santa Fe Institute and in part by TXN Inc.

References

Bernardo J. and Smith, A. (1994). *Bayesian Theory*. Wiley and Sons.

Breiman, L. (1994). Bagging Predictors.TR 421, Berkeley Department of Statistics.

involve things like model mis-specification in the learning algorithm.)

2) Investigate the real-world manifestations of the "bias-variance" trade-off for the logarithmic and quadratic scoring definitions of bias and variance used here.

3) See if there are alternative definitions of bias and variance for logarithmic and quadratic scoring that meet our desiderata. More generally, investigate how unique bias-plus-variance decompositions are.

4 Investigate what aspects of the relationship between C and the other random variables (like Y_H and Y_F) are necessary for there to be a bias-plus-variance decomposition for E(C | f, m, q) that meets conditions (i) through (iv) and (a) through (c). Investigate how those aspects change as one modifies z and/or the conditions one wishes the decomposition to meet.

5) The EBF provides several natural z-dependent ways to determine how "close" one learning algorithm is to another. For example, one could define the distance between learning algorithms A and B, Δ(A, B), as the mutual information between the distributions P(c | z, A) and P(c | z, B). One could then explore the relationship between Δ(A, B) on the one hand, and how similar the terms in the associated bias-plus-variance decompositions are, on the other.

6) Investigate the "data work" for other z's besides {m, q} and/or other C's besides the quadratic loss function. In particular, for what C's will all our desiderata be met and yet the intrinsic noise be given exactly by the error associated with the Bayes-optimal learning algorithm?

7) Instead of going from a C(F, H, Q) to definitions for the associated bias, variance, etc., do things in reverse. I.e., investigate the conditions under which one can "back out" to find an associated C(F, H, Q), given arbitrary definitions of bias, intrinsic noise, variance and covariance (arbitrary within some class of "reasonable" such definitions).

FOOTNOTES

1. Note that if L(., .) is a symmetric function of its arguments, this expression for E(C | f, h, q) is a non-Euclidean inner product between the (**Y**-indexed) vectors f(q, .) and h(q, .). Such inner products generically arise in response to conditional independencies among the random variables. In

just before Eq. (13). In addition, define

$$E(\,F^2(.,y)\,|\,z) \equiv \Sigma_q \int df\, P(f, q\,|\,z) \times f^2(q, y)\,,$$
$$E(\,H^2(.,y)\,|\,z) \equiv \Sigma_q \int dh\, P(h, q\,|\,z) \times h^2(q, y)\,,$$

and

$$E(\,F(.,y)\,H(.,y)\,|\,z) \equiv \Sigma_q \int df\, dh\, P(h, f, q\,|\,z) \times f(q, y)\, h(q, y)\,.$$

Then we can write

$$(14)\ E(C\,|\,z) = \sigma_{qs;z} + \text{bias}_{qs;z} + \text{variance}_{qs;z} - 2\text{cov}_{qs;z},$$

where

$$\sigma_{qs;z} \equiv \Sigma_y\ \{\, E(\,F^2(.,y)\,|\,z) - [\, E(\,F(.,y)\,|\,z)\,]^2\,\},$$
$$\text{bias}_{qs;z} \equiv \Sigma_y\, [P(Y_H = y\,|\,z) - P(Y_F = y\,|\,z)]^2,$$
$$\text{variance}_{qs;z} \equiv \Sigma_y\, \{\, E(\,H^2(.,y)\,|\,z) - [\, E(\,H(.,y)\,|\,z\,]^2\,\},$$

and

$$\text{cov}_{qs;z} \equiv \Sigma_y\, \{\, E(\,F(.,y)\,H(.,y)\,|\,z) - E(\,F(.,y)\,|\,z) \times E(H(.,y)\,|\,z)\,\}.$$

As usual, these decompositions meet essentially all of desiderata (a) through (c).

X FUTURE WORK

Future work consists of the following:

1) Investigate the real-world manifestations of the Bayesian correction to bias-plus-variance for quadratic loss. (For example, it seems plausible that whereas the bias-variance trade-off involves things like the number of parameters involved in the learning algorithm, the covariance term may

h to equal f always, regardless of d. Accordingly (see desideratum (a)), the intrinsic noise term must equal zero.

To determine the bias term for quadratic scoring, employ the same trick used for logarithmic scoring to write bias as $E(C \mid f = Opt(E(F \mid z)), h = Opt(E(H \mid z)), q)$. Again as with logarithmic scoring, for any **X**-condition distribution over **Y**, u, $Opt(u) = u$. Accordingly, for quadratic scoring, our bias term is $E(C \mid f = E(F \mid z), h = E(H \mid z), q)$. For z = {f, m, q}, this reduces to $E(C \mid f, h = E(H \mid f, m, q), q) = \Sigma_y [f(q, y) - E(H(q, y) \mid f, m, q)]^2$:

$$\text{bias}_{qs;f,m,q} = \Sigma_y [P(Y_F = y \mid f, m, q) - P(Y_H = y \mid f, m, q)]^2.$$

As with logarithmic scoring, we then set variance to be the difference between $E(C \mid f, m, q)$ and the sum of the intrinsic noise and bias terms. So for quadratic scoring,

$$E(C \mid f, m, q) = \sigma_{qs;f,m,q} + \text{bias}_{qs;f,m,q} + \text{variance}_{qs;f,m,q} ,$$

where

$$\sigma_{qs;f,m,q} \equiv 0,$$
$$\text{bias}_{qs;f,m,q} \equiv \Sigma_y [P(Y_H = y \mid f, m, q) - P(Y_F = y \mid f, m, q)]^2, \text{ and}$$
$$\text{variance}_{qs;f,m,q} \equiv \Sigma_y \int dh\, P(h \mid f, m, q) [h(q, y)]^2 - \Sigma_y [\int dh\, P(h \mid f, m, q)\, h(q, y)]^2 .$$

As usual, these are in agreement with desiderata (a) through (c). Interestingly, $\text{bias}_{qs;f,m,q}$ is the same as the bias^2 for zero-one loss for z = {f, m, q} (see [Kohavi and Wolpert 1196, Wolpert and Kohavi 1996).

ii) The arbitrary z decomposition

To present the general-z case, recall the definitions of $E(F(., y) \mid z)$ and $E(H(., y) \mid z)$ made

$$= \Sigma_q \int df\, dh \quad \Sigma_d P(q \mid z) P(f \mid q, z) P(d \mid f, q, z) P(h \mid d, q, z) \; f(q, y) \ln [h(q, y)].$$

Then simple algebra verifies the following:

(13) $E(C \mid z) = \sigma_{ls;z} + \text{bias}_{ls;z} + \text{variance}_{ls;z} + \text{cov}_{ls;z}$,

where

$$\sigma_{ls;z} \equiv -\Sigma_y E(F(., y) \mid z) \ln[E(F(., y) \mid z)],$$

$$\text{bias}_{ls;z} \equiv -\Sigma_y E(F(., y) \mid z) \ln \left[\frac{E(H(., y) \mid z)}{E(F(., y) \mid z)} \right],$$

$$\text{variance}_{ls;z} \equiv -\Sigma_y E(F(., y) \mid z) \{ E(\ln[H(., y)] \mid z) - \ln[E(H(., y) \mid z)] \},$$

and

$$\text{cov}_{ls;z} \equiv -\Sigma_y E(F(., y) \ln[H(., y)] \mid z) - E(F(., y) \mid z) \times E(\ln[H(., y)] \mid z).$$

Note that in Eq. (13) we add the covariance term rather than subtract it (as in Eq. (2)). Intuitively, this reflects the fact that -ln(.) is a monotonically *decreasing* function of its argument, in contrast to $(.)^2$. However even with the sign "backward" the covariance term as it occurs in Eq. (13) still means that if the learning algorithm tracks the posterior - if when f(q, y) rises so does h(q, y) - then the expected cost is smaller than it would be otherwise.

IX BIAS PLUS VARIANCE FOR QUADRATIC SCORING

i) The {f, m, q}-conditioned decomposition

In quadratic scoring $C(f, h, q) = \Sigma_y [f(q, y) - h(q, y)]^2$. This is not to be confused with the "quadratic score function", which has $C(y_F, h, q) = 1 - \Sigma_y [h(q, y) - \delta(y, y_F)]^2$. This score function can be used when one only has a finite test set, which is not the case for quadratic scoring. (See [Bernardo and Smith 1994].) Analysis of bias plus variance decompositions for the quadratc score function is the subject of future work.

For quadratic scoring, for z = {f, m, q}, the lower bound on an algorithm's error is zero: guess

would be the average difference between h and the average h,

$$-\int dh\, P(h \mid f, m, q)\, \Sigma_y\, E(H \mid f, m, q) \ln[h(q, y)].$$

(Cf. the formula at the beginning of this section giving $E(C \mid f, m, q)$ for logarithmic scoring.)

This can be rewritten as

$$-\Sigma_y \{ \Sigma_{d'} P(d' \mid f, m) \int dh\, P(h \mid d')\, h(q, y) \} \Sigma_d P(d \mid f, m) \int dh\, P(h \mid d) \ln[h(q, y)]$$

However consider the case where $P(h \mid d) = \delta(h - f)$ for all d for which $P(d \mid f, m) \neq 0$. With this alternative definition of variance, in such a situation we would have the variance equalling $-\Sigma_y f(q, y) \ln[f(q, y)] = \sigma_{ls;f,m,q}$, not zero. (Indeed, just having $P(h \mid d) = \delta(h - h')$ for some h' for all d for which $P(d \mid f, m) \neq 0$ suffices to violate our desiderata, since this will in general *not* result in zero "variance".) Moreover, in this scenario, the variance would also equal $E(C \mid f, m, q)$. So for this scenario, bias = $E(C \mid f, m, q)$ - variance - $\sigma_{ls;f,m,q}$ would equal $-\sigma_{ls;f,m,q}$, not zero. This violates our desideratum concerning bias.

Yet another possible formulation of the variance would be (in analogy to the formula we presented above for logarithmic scoring intrinsic noise) the Shannon entropy of the average H, $E(H \mid z)$. But again, this formulation of variance would violate our desiderata. In particular, for this definition of variance, having h be independent of d would not result in zero variance.

iii) Corrections to the decomposition for when z ≠ {f, m, q}

Finally, just as there is an additive Bayesian correction to the {f, m, q}-conditioned quadratic loss bias-plus-variance formula, there is also one for the logarithmic scoring formula. As useful shorthand, write

$$\begin{aligned}
E(F(., y) \mid z) &\equiv \Sigma_q \int df\, P(f, q \mid z) \times f(q, y) \\
&= \Sigma_q \int df\, P(q \mid z)\, P(f \mid q, z)\, f(q, y), \\
E(H(., y) \mid z) &\equiv \Sigma_q \int dh\, P(h, q \mid z) \times h(q, y), \\
&= \Sigma_q \int dh\, \Sigma_d \int df\, P(q \mid z)\, P(f \mid q, z)\, P(d \mid f, q, z)\, P(h \mid d, q, z)\, h(q, y), \\
E(\ln[H(., y)] \mid z) &\equiv \Sigma_q \int dh\, P(h, q \mid z) \times \ln[h(q, y)] \\
&= \Sigma_q \int dh\, \Sigma_d \int df\, P(q \mid z)\, P(f \mid q, z)\, P(d \mid f, q, z)\, P(h \mid d, q, z) \ln[h(q, y)],
\end{aligned}$$

and

$$E(F(., y) \ln[H(., y)] \mid z) \equiv \Sigma_q \int df\, dh\, P(h, f, q \mid z) \times f(q, y) \ln[h(q, y)]$$

(12) $E(C \mid f, m, q) \; = \; \sigma_{ls;f,m,q} + bias_{ls;f,m,q} + variance_{ls;f,m,q}$.

It is straightforward to establish that $variance_{ls;f,m,q}$ meets the requirements in desideratum (c). First, consider the case where $P(h \mid d) = \delta(h - h')$ for some h' (this delta function is the Dirac delta function). In this case the term inside the curly brackets in Eq. (11) just equals $\ln[h'(q, y)] - \ln[h'(q, y)] = 0$. So $variance_{ll}$ does equal zero when the guess h is independent of the training set d. (In fact, it equals zero even if h is not-single-valued, the precise case (c) refers to.) Next, since the log is a concave function, we know that the term inside the curly brackets is never greater than zero. Since $f(q, y) \geq 0$ or all q and y, this means that $variance_{ls;f,m,q} \geq 0$ always.

Finally, we can examine the $P(h \mid d)$ that make $variance_{ll}$ large. Any h is an $|\mathbf{X}|$-fold cartesian product of vectors living on $|\mathbf{Y}|$-dimensional unit simplices. Accordingly, for any d, $P(h \mid d)$ is probability density function in a Euclidean space. To simplify matters further, assume that $P(h \mid d)$ is deterministic, so it specifies a single unique distribution h for each d, indicated by h_d. Then the term inside the curly brackets in Eq. (11) equals

$$\Sigma_d \; P(d \mid f, m) \; \ln[h_d(q, y)] - \ln[\, \Sigma_d \, P(d \mid f, m) \, h_d(q, y) \,].$$

This is the difference between an average of a function and the function evaluated at the average. Since the function in question is concave, this difference grows if the points going into the average are far apart. I.e., to have large $variance_{ls;f,m,q}$, the $h_d(q, y)$ should differ markedly as d varies. This establishes the final part of desideratum (c).

ii) Alternative {f, m, q}-conditioned decompositions

The approach taken here to deriving a bias-plus-variance formula for logarithmic scoring is not "perfect". For example, the formula for $variance_{ls;f,m,q}$ is not identical to the formula for $\sigma_{ls;f,m,q}$ under the interchange of F with H (as is the case for the variance and intrinsic noise terms for quadratic loss). In addition, $variance_{ls;f,m,q}$ can be made infinite by having $h_d(q, y) = 0$ for one d and one y, assuming both $f(q, y)$ and $P(d \mid f)$ are nowhere zero. Although not surprising given that we're interested in logarithmic scoring, this is not necessarily "desirable" behavior in a variance-like quantity.

Other approaches tend to have even more major problems however. For example, as an alternative to the approach taken here, one could imagine trying to define a "variance" first, and then define bias by requiring that the bias plus the variance plus the noise gives the expected error. It is not clear how to follow this approach however. In particular, one natural definition of variance

$$E(C \mid f = Opt(E(F \mid z)),\ h = Opt(E(H \mid z)),\ q,\ m).$$

Unlike the original expression for bias^2 for quadratic loss, this new expression can be evaluated even for logarithmic scoring. The *Opt*(.) function is different for logarithmic scoring and quadratic loss. For logarithmic scoring, by Jensen's inequality, $Opt(u) = u$. (Scoring rules obeying this property are sometimes said to be "proper" scoring rules.) Accordingly, our bias term can be written as $E(C \mid f = E(F \mid z), h = E(H \mid z), q)$. For $z = \{f, m, q\}$, this reduces to $E(C \mid f, h = E(H \mid f, m, q), q)$.

As an aside, note that for quadratic loss, this same expression would instead be identified with noise + bias. This different way of interpreting the same expression reflects the difference between measuring cost by comparing y_H and y_F versus doing it by comparing h and f.

Writing it for logarithmic scoring, the bias term for logarithmic scoring is

$$-\Sigma_y f(q, y)\ \ln \{ \Sigma_d P(d \mid f, m) \int dh\, P(h \mid d)\, h(q, y) \} .$$[3]

One difficulty with this expression is that its minimal value over all learning algorithms is greater than zero. To take care of that I will subtract from this expression the additive constant of its minimal value. That minimal value is given by the learning algorithm that always guesses h = f, independent of the training set. Accordingly, our final measure for the "bias" for logarithmic scoring is the Kullback-Leibler distance between the distribution f(q, .) and the average h(q, .). With some abuse of notation, this can be written as follows:

$$(10)\quad bias_{ls;f,m,q} \equiv -\Sigma_y\ f(q, y)\ \ln [E(H(q, y) \mid f, m) / f(q, y)] =$$
$$-\Sigma_y\ f(q, y)\ \ln \left[\frac{\Sigma_d P(d \mid f, m) \int dh\, P(h \mid d)\, h(q, y)}{f(q, y)} \right] .$$

This definition of $bias_{ls;f,m,q}$ meets desideratum (b).

Given these definitions, the "variance" for logarithmic scoring and conditioning on f, m, and q, $variance_{ls;f,m,q}$, is fixed, and given by

$$(11)\ variance_{ls;f,m,q} \equiv -\Sigma_y\ f(q, y)\ \ \{ E(\ln[H(q, y)] \mid f, m) - \ln[E(H(q, y) \mid f, m)] \}$$
$$= -\Sigma_y\ f(q, y)\ \ \{ \Sigma_d\ P(d \mid f, m)\ \int dh\, P(h \mid d) \ln[h(q, y)]$$
$$-\ \ln[\Sigma_d P(d \mid f, m) \int dh\, P(h \mid d)\, h(q, y)]\ \}.$$

Combining, for logarithmic scoring,

ditions (i) through (iv).

One natural way to measure intrinsic noise for logarithmic scoring is as the Shannon entropy of f,

9) $\sigma_{ls;f,m,q} \equiv -\Sigma_y \; f(q, y) \ln[f(q, y)]$.

Note that this definition meets all three parts of desideratum (a). It is also the "expected error of f at guessing itself", in close analogy to the intrinsic noise term for quadratic loss (see condition (i)).

To form an expression for logarithmic scoring that is analogous to the bias^2 term for quadratic loss, we cannot directly start with the terms involved in quadratic loss because they need not be properly defined. (E.g., $E(Y_F \mid z)$ is not defined for categorical spaces **Y**, even though logarithmic scoring is perfectly well defined for that case.) To circumvent this difficulty, first note that for quadratic loss, expected **Y** values are the modes of optimal hypotheses (for quadratic loss, minimizing expected loss means taking a mean **Y** value, in general). In addition, squares of differences between **Y** values are expected costs. Keeping this in mind, the bias^2 term for quadratic loss can be rewritten as the following sum:

$$\Sigma_{y,y'} L(y, y') \;\; \delta(y, E(Y_F \mid f, q, m)) \;\; \delta(y, E(Y_H \mid f, q, m)).$$

This is the expected quadratic loss between two **X**-conditioned distributions over **Y** given by the two delta functions. The first of those distributions is what the optimal hypothesis would be if the target were given by $E(F \mid z)$ (for $z = \{f, m, q\}$, $E(F \mid z) = f$). More formally, the first of the two distributions is $\delta(y, E(Y_F \mid z)) = \text{argmin}_h \, E(C \mid f = E(F \mid z), m, q, h)$. I will indicate this by defining $Opt(u) \equiv \text{argmin}_h \, E(C \mid f = u, m, q, h)$, where u is any distribution over Y. So this first of our two distributions is $Opt(E(F \mid z))$.

For $z = \{f, m, q\}$, $P(y_F \mid f = E(H \mid z), q, m) = f(q, y_F)$ evaluated for $f = \int dh \; h \; P(h \mid f, m, q) \; h$. I.e., it equals the vector $\int dh \; h \; P(h \mid f, m, q)$ evaluated for indices q and y_F. By the properties of vector spaces, this can be rewritten as $\int dh \; h(q, y_F) \; P(h \mid f, m, q)$. However this is just $P(y_H \mid f, q, m)$ evaluated for $y_H = y_F$. Accordingly, $E(Y_H \mid f, q, m) = E(Y_F \mid f = E(H \mid z), q, m)$. So the second of the two distributions in our expression for bias^2 is what the optimal hypothesis would be if the target were given by $E(H \mid z)$: $\text{argmin}_h \, E(C \mid f = E(H \mid z), m, q, h)$. We can indicate this second optimal hypothesis by $Opt(E(H \mid z))$.

Combining, we can now rewrite the bias^2 for quadratic loss as

η, and as mentioned above η does not have the formal equal footing with f that h does. So rather than a proper covariance between η and f, we instead have here an unconventional "covariance-like" term. And this term does not disappear even for fixed f. As a sort of substitute, the covariance-like term instead is uniquely determined by the values of the noise, bias, and variance, if f is fixed.

VIII BIAS PLUS VARIANCE FOR LOGARITHMIC SCORING

i) The {f, m, q}-conditioned decomposition

To begin the analysis of bias-plus-variance for logarithmic scoring, consider the case where z = {f, m, q}. The logarithmic scoring rule is given by

$$E(C \mid f, h, q) = -\Sigma_y \; f(q, y) \ln[h(q, y)],$$

so

$$E(C \mid f, m, q) = -\Sigma_y \; f(q, y) \; \Sigma_d \; P(d \mid f, m) \int dh \, P(h \mid d) \ln[h(q, y)].$$

Unlike quadratic loss, logarithmic scoring is not symmetric under $f \leftrightarrow h$.

This scoring rule (sometimes instead referred to as the "log loss function", and proportional to the value of the "logarithmic score function" for an infinite test set) can be appropriate when the output of the learning algorithm h is meant to be a guess for the entire target distribution f [Bernardo and Smith, 1994]. This is especially true when **Y** is a categorical rather than a numeric space. To understand that, consider the case where you guess h and have some "test set" T generated from f that you wish to use to score h. How to do that? One obvious way is to score h as the log-likelihood of T given h. If we now average this over all T generated from f, we get logarithmic scoring. Note that logarithmic scoring can be used even when there is no metric structure on **Y** (as there must be for quadratic loss to be used).

In creating an analogy of the bias-plus-variance formula for cases where C is not given by quadratic loss, one would like to meet conditions (i) through (iv) and (a) through (c) presented above. However often there is no such thing as $E(Y_F \mid f, q)$ when logarithmic scoring is used (i.e., often **Y** is not a metric space when logarithmic scoring is used). So we cannot measure intrinsic noise relative to $E(Y_F \mid f, q)$, as in the quadratic loss bias-plus-variance formula. This means that meeting conditions (i) through (iv) will be impossible; the best we can do is come up with a formula whose terms meet desiderata (a) through (c), and which have analogous behavior in some sense to the con-

$$2 \ \{ \ E[\ R(\sim y^*, \eta) \ | \ f, m, q] \times f(q, \sim y^*)$$
$$-$$
$$R(\sim y^*, E[\eta \ | \ f, m, q]) \times f(q, \sim y^*) \ \} \ .$$

(As an aside, note that for the zero-one R(., .), R(~y*, E[η | f, m, q]) = 0, since by definition ~y* is not the Y value that maximizes E[η | f, m, q].)

Again following along with Friedman, have E[η | f, m, q] fixed while increasing variability. Next presume that that increase in 'variability' increases E[R(~y*, η) | z], as Friedman does. (E[R(~y*, η) | z] gives the amount of "spill" of the learning algorithm's guess η_1 into ~y*, the output label other than the one corresponding to the algorithm's average η_1.) Doing all this will always decrease the contribution to the expected error arising from the covariance term. As mentioned above, it will also always increase the variance term's contribution to the expected error. So long as f(q, ~y*) > 1/2, the former phenomenon will dominate the latter, and overall, expected error will decrease. This condition on f(q, ~y*) is exactly the one that corresponds to Friedman's 'peculiar behavior'; it means that the target is weighted towards the opposite Y value from the one given by the expected guess of the learning algorithm.

So whether the peculiar behavior holds when one increases variability depends on whether the associated increase in variance manages to offset the associated increase in covariance (this latter increase resulting in a decrease in contribution to expected error). As in so much else having to do with the bias-variance decomposition, we have a classical trade-off between two competing phenomena both arising in response to the same underlying modification to the learning problem. There is nothing 'peculiar' in this, but in fact only classical bias-variance phenomenology.

Nonetheless, it should be noted that for zero-one R, the covariance just equals 2 f(q, y*) times the variance. Moreover, as was pointed out above, for zero-one R and z = {f, m, q}, noise + bias = 1 - f(q, y*). So if noise and bias are held constant, it is impossible to change the variance while "everything else is held constant" - the covariance will necessarily change as well, assuming the bias and noise do not.

This behavior should not be too surprising. Since the zero-one loss can only take on two values, one might expect that it is not possible for all four of noise, bias, variance, and covariance to be independent. The reason for the precise dependency among those terms encountered here can be traced to two factors: choosing z to equal {f, m, q}, and performing the analysis in terms of η rather than h. As discussed in previous sections, for decompositions involving h (rather than η), the covariance term relates variability in h to variability in f, and therefore must vanish for the fixed f induced by having z equal {f, m, q}. However in this section we are doing the analysis in terms of

$$- \; S_{y \neq y^*} \left(R(y, E[\eta \mid z]), \;\; E[\, F(Q, y) \mid z \,] \right) \;\; \} \,.$$

For many R the expression on the first line of Eq. (8) is non-negative. For example, due to the definition of y*, this is the case for the zero-one R. In addition, that expression does not involve f directly, equals zero when the learning algorithm's guesses never changes, etc. Accordingly, that expression (often) meets the usual desiderata for a variance, and will here be identified as a variance.

Note that if everything else is kept constant while this variance is increased, then expected error also increases - there is none of the 'peculiar behavior' Friedman found if one identifies variance with this expression from Eq. (8). For the zero-one R, this variance term is the probability that the y maximizing η_y is not the one that maximizes the expected η_y.

The remaining terms in Eq. (8) collectively act like (the negative of) a covariance. Indeed, for the case Friedman considers where r = 2, if we define ~y* as the element of Y that differs from y*, we can write (the negative of) those terms as

$$2 \; \{ \; E[\, R(\sim y^*, \eta) \times F(Q, \sim y^*) \mid z] \; - \; R(\sim y^*, E[\eta \mid z]) \times E[\, F(Q, \sim y^*) \mid z] \; \} \,.$$

Note the formal parallel between this expression for the "remaining terms in Eq. (8)" and the functional forms of the covariance terms in the bias-variance decompositions for other costs that were presented above.

Of course, the parallel isn't exact. In particular, this expression isn't exactly a covariance - a covariance would have an $E[R \mid z] \times E[f \mid z]$ type term rather than an $R(.., E[\, .. \mid z]) \times E[f \mid z]$ term. The presence of the R(., .) functional is also somewhat peculiar. Indeed, the simple fact that the covariance term is non-zero for z = {f, m, q} is unusual (see the quadratic loss decomposition for example). Ultimately, all these effects follow from the fact that the decomposition considered here does not treat targets and hypotheses the same way; targets are represented by f's, whereas hypotheses are represented by η's rather than h's. (Recall that h is determined by the maximal component of η.) If instead hypotheses were represented by h's, then the zero-one loss decomposition would involve a proper covariance, without any R(., .) functional [Kohavi and Wolpert, 1996].

It is the covariance term, not the variance term, which results in the possibility that an increase in 'variability' reduces expected generalization error. To see this, along with Friedman, take z to equal {f, m, q}. Then our covariance term becomes

$$C = \Sigma_{y \neq y^*} R(y, \eta) \;+\; \Sigma_{y \neq y^*} f(q, y)$$
$$- \; S_{y \neq y^*} (R(y, \eta_y), \; f(q, y)) \,,$$

if we use the shorthand

$$S_{\{i\}} (g(i), h(i)) \equiv \Sigma_i \; g(i) \; \Sigma_i \; h(i) \;+\; \Sigma_i \; g(i)\, h(i) \,.$$

Now we must determine what noise + bias is. Since η and f don't live in the same space, not all of the desiderata listed previously are well-defined. Nonetheless, we can satisfy their spirit by taking noise + bias to be $E[\, C(., \; E[\eta \mid z], q) \mid z)$, i.e., by taking it to be the expected value of C when one guesses using the expected output of the learning algorithm. In particular, this definition makes sense in light of desiderata (iii).

Writing it out, by using the multilinearity of S(., .) we see that for this definition noise + bias is given by

$$\Sigma_{y \neq y^*} R(y, E[\eta \mid z]) \;+\; \Sigma_{y \neq y^*} E[F(Q, y) \mid z\,]$$
$$- \;\; S_{y \neq y^*} (R(y, E[\eta \mid z]), \; E[F(., y) \mid z]) \,.$$

(As a point of notation, "$E[F(Q, y) \mid z\,]$" means $\int df \, \Sigma_q \, f(q, y) \, P(f, q \mid z)$; it is the y component of the z-conditioned average target for average q.)

For the zero-one loss R(., .) Friedman considers, $R(y, E[\eta \mid z]) = 0$ for all $y \neq y^*$. Therefore noise + bias reduces to $\Sigma_{y \neq y^*} E[F(Q, y) \mid z\,]$. For his z, {f, m, q}, this is just $1 - f(q, y^*)$. So for Friedman's $r = 2$ scenario, if the class corresponding to the learning algorithm's average η_1 is the same as the target's average class, noise + bias is $\min_y f(q, y)$. Otherwise it is $\max_y f(q, y)$. If we identify $\min_y f(q, y)$ with the noise term, this means that the bias either equals zero, or it equals $| f(q, y = 1) - f(q, y = 2) |$, depending on whether the class corresponding to the learning algorithm's average η_1 is the same as the target's average class.

Continuing with our more general analysis, the difference between $E(C \mid z)$ and noise + bias is

8) $$\Sigma_{y \neq y^*} E[\, R(y, \eta) \mid z] \;-\; R(y, E[\, \eta \mid z])$$
$$-$$
$$\{ \;\; E[\, S_{y \neq y^*} (R(y, \eta), \; F(Q, y)) \;\mid\; z\,]$$

$$C = C(f, \eta, q) = \Sigma_y [1 - R(y, \eta)] \; f(q, y)$$
$$= 1 - \Sigma_y R(y, \eta) \; f(q, y) \,.$$

It is required that $R(y, \eta)$ be unchanged if one relabels both y and the components of η simultaneously and in the same manner. As an example of an R, for the precise case Friedman considers, there are two possible values of y, and $R(y, \eta) = 1$ for $y = \text{argmax}_i(\eta_i)$, 0 for all other y.

Note that when expressed this way the cost is given by a dot product between two vectors indexed by Y values, namely R and f. Moreover, for many R (e.g., the zero-one R), that dot product is between two probability distributions over y. Note also that whereas h and f are on an equal footing in the EBF (cf. section 2, and in particular points (10) through (13)), the same is not true for η and f. Indeed, whereas h and f arise in a symmetric manner in the zero-one loss cost ($C(f, h, q) = \Sigma_{y_H, y_F} [1 - \delta(y_H, y_F)] \, h(q, y) f(q, y)$), the same is manifestly not true for the variables η and f.

For fixed η and q respectively, for both the (Y-indexed) vector $R(y, \eta)$ and the vector $f(q, y)$ there are only r - 1 free components, since in both cases the components must sum to 1. Accordingly, as in Friedman's analysis, here it makes sense to reduce the number of free variables to 2 r - 2 by expressing one of the R components in terms of the others and similarly for one of the f components.

A priori, it is not clear which such component should be re-expressed this way. Here I will choose to re-express the component

$$y^*(z) \equiv \text{argmax}_y \, R(y, E[\eta \mid z])$$

this way for both R and f. (For Friedman's R(., .), this $y^*(z)$ is equivalent to the y maximizing $E([\eta_y \mid z])$. So in the example above of the Friedman effect, $y^*(z)$ = class 1.) Accordingly, from now on I will replace $R(y^*(z), \eta)$ with $1 - \Sigma_{y \neq y^*(z)} R(y, \eta_y)$ and I will replace $f(q, y^*(z))$ with $1 - \Sigma_{y \neq y^*(z)} f(q, y)$ wherever those terms appear. (From now on, when the context makes z clear I will write "y*" rather than "y*(z)".)

Having made these replacements, we can write

$$C = \Sigma_{y \neq y^*} R(y, \eta) + \Sigma_{y \neq y^*} f(q, y)$$
$$- \{ \Sigma_{y \neq y^*} R(y, \eta) \times \Sigma_{y \neq y^*} f(q, y) \} - \Sigma_{y \neq y^*} R(y, \eta) \, f(q, y) \,,$$

which we can rewrite as

lar, note that whether the Friedman effect obtains - whether the “variability” term increases or diminishes expected error - depends directly not only on the learning algorithm and the distribution over training sets, but also on the target (the average η_1 must result in a guessed class label that differs from the optimal prediction as determined by the target for the Friedman effect to hold). This direct dependence of the Friedman effect on the target is an important clue. Recall in particular that our desiderata preclude identifying a term in the decomposition of expected error as a”variance” if it has such a direct dependence. However such a dependence of the ‘variability’ term could be allowed if the ‘variability’ that is being increased is not equated identically with a variance, but rather with a variance *combined with another quantity*.

Given the general form of the bias-variance decomposition, the obvious choice for that other quantity is a term reflecting a covariance. Indeed, by having that covariance involve varying over y_F values as well as training sets, we can readily imagine getting the direct dependence on the target that arises in the Friedman effect.

In addition, viewing the variability as involving a covariance term along with a variance term could potentially explain away the peculiarity of the Friedman effect’s having error shrink as variability increases. This would be the case for example if holding the covariance part of the variability fixed while increasing the variance part always *did* increase expected generalization error. The idea would be that the way that ‘variability’ is increased in the Friedman effect involves changes to the covariance term as well as the variance term, and it is the changes to the covariance term that are ultimately responsible for the reduction in expected generalization error. Increasing the variance term, by itself, can only increase expected error, exactly as in the conventional bias-plus-variance decomposition.

As it turns out, this hypothesized behavior is exactly what lies behind the Friedman effect. This can best be seen by using a different decomposition from Friedman’s. In exploring that alternative decomposition, we will see that there is nothing inherently unusual about how variance is related to generalization error for zero-one loss; in this alternative decomposition, the decrease in generalization error associated with increasing ‘variability’ is due to the covariance term rather than the variance term. Common intuition is salvaged. This alternative to Friedman’s analysis has the additional major advantage that it is valid even if the assumptions that Friedman’s analysis requires do not hold. Moreover this alternative decomposition is additive rather than multiplicative, holds for all algorithms, and in general avoids the other oddities of Friedman’s analysis.

Unfortunately, to clarify all this, it is easiest to work with somewhat generalized notation. (The reader’s forbearance is requested.) The reason for this is that since it can only take on two values, the zero-one loss can “hide” a lot of phenomena via “accidental cancellation” of terms and the like. To have all such phenomena manifest we need the generalized notation.

First write the cost in terms of an abstraction of the expected zero-one loss:

that even if the Gaussian in question is fairly peaked about a value well within [0, 1), that part of the dependence of the integral on certain quantities that arises due to the presence of those quantities in the limits on the integration may be significant, in comparison to the other ways that that integral depends on those quantities.) In addition, as Friedman defines it, bias can be negative. Moreover, his bias does not reduce to zero for the Bayes classifier (and in fact the approximations invoked in his analysis are singular for that classifier.) And perhaps most importantly, his definition of variance depends directly on the underlying target distribution f, rather than indirectly through the f-induced distribution over training sets P(d | f, m, q). This last means that the "variance" term does not simply reflect how sensitive the learning algorithm is to (target-governed) variability in the training set - the variance also changes even if one makes a change to the target that has no effect on the induced probability distribution over training sets.

These difficulties notwithstanding, Friedman's analysis is a major contribution. Perhaps the most important aspect of Friedman's analysis is its drawing attention to the following phenomenon: Consider the case where we're interested in expected zero-one loss conditioned on a fixed target, test set question, and training set size. Presume further that we're doing binary classification (r = 2), so the class label probabilities guessed by the learning algorithm reduce to a single real number η_1 giving the guessed probability of class 1 (i.e., $\eta = (\eta_1, 1 - \eta_1)$). Examine the case where, for the test set point at hand, the average (over training sets generated from f) value of η_1 is greater than 1/2. So the guess corresponding to that average η_1 is class 1. However let's say that the truly optimal prediction (as determined by the target) is class 2. Now modify the scenario so that the variability (over training sets) of the guess η_1 grows, while its average stays the same (i.e., the width of the distribution over η_1 grows). Assume that this extra variability results in having $\eta_1 <$ 1/2 for more training sets. So for more training sets, the η_1 produced by the learning algorithm corresponds to the (correct) guess of class 2. Therefore increasing the 'variability' of η_1 while keeping the average the same has reduced overall generalization error (!). Note that this effect only arises when the average of η_1 results in a non-optimal prediction, i.e., only when that average is "wrong". (This point will be returned to shortly.)

Now view this 'variability' as, intuitively, akin to a variance. (This definition differs only a little from the formal definition of variance Friedman advocates.) Similarly, view the 'average' of η_1 as giving a bias. Then we have the peculiar result that increasing variance while keeping bias fixed can reduce overall expected generalization error. I will refer to such behavior as the "Friedman effect". (See also [Breiman 1996], in particular the discussion in the first appendix.)

'Variability' can be identified with the width of the distribution over η_1, and in that sense can indeed be taken to be a 'variance'. The question is whether it makes sense to view it as a 'variance' in the restricted desiderata-based sense appropriate to "bias-variance decompositions". In particu-

quadratic loss.

VII AN ALTERNATIVE ANALYSIS OF THE FRIEDMAN EFFECT

Kong and Dietterich recently [1995] raised the issue of what the appropriate bias-plus-variance decomposition is for zero-one (misclassification) loss, $L(y_H, y_F) = 1 - \delta(y_H, y_F)$. They raised the issue in the context of the classical conditioning event: the target, the training set size, and the test set question. The decomposition they suggested (i.e., their suggested definitions of "bias" and "variance" for zero-one loss) has several shortcomings. Not least of these is that decomposition's allowing negative variance. Several subsequent papers [Kohavi and Wolpert 1996, Wolpert and Kohavi 1996, Tibshirani 1996, Breiman 1996] have offered alternative decompositions, with different strengths and weaknesses, as discussed in [Kohavi and Wolpert 1996, Wolpert and Kohavi 1996]. Recently Friedman contributed another zero-one loss decomposition [1996] to the discussion.

Friedman's decomposition only applies to learning algorithms that perform their classification by first predicting the probabilities η_y of all the possible output classes and then picking the class $\text{argmax}_i[\eta_i]$. In other words, he considers cases where h is single-valued, but the value $h(q) \in Y$ is determined by finding the maximum over the components of a Euclidean vector random variable η, dependent on q and d, whose components always sum to 1 and are all non-negative. Intuitively, the η_y are the probabilities of the various possible Y_F values for q, as guessed by the learning algorithm in response to the training set.

The restriction of Friedman's analysis to such algorithms is not lacking in consequence. For example, it rules out perhaps the simplest possible learning algorithm, one that is of great interest in the computational learning community: from a set of candidate hypothesis input-output functions, pick that one which best fits the training set. This restriction makes Friedman's analysis less general than the other zero-one loss decompositions that have been suggested.

There are several other peculiar aspects to Friedman's decomposition. Oddly, it is multiplicative in (its suggested definitions of) bias and variance rather than additive. This begs the question of whether anything is gained by labelling those terms "bias" and "variance", or if it invites confusion to not given those terms new names.

More importantly, it assumes that as one varies training sets d while keeping f, m, and q constant, the induced distribution over η is Gaussian. It then further assumes that the truncation on integrals over such a Gaussian distribution imposed by the limits on the range of each of the η_y (each $\eta_y \in [0, 1]$) are irrelevant, so that erf functions can be replaced by integrals from $-\infty$ to $+\infty$. (Note

i) General properties of convex and/or concave loss functions

There are a number of other special properties of quadratic loss besides Eq.'s (1) through (6). For example, for quadratic loss, for any f, $E(C \mid f, m, q, \text{algorithm A}) \le E(C \mid f, m, q, \text{algorithm B})$ so long as A's guess is the average of B's (formally, so long as we have $P(y_H \mid d, q, A) = \delta(y_H, E(Y_H \mid d, q, B)) = \delta(y_H, \Sigma_y\, y\, h(q, y)\, P(h \mid d, B))$. So *without any concerns for priors*, one can always construct an algorithm that is assuredly superior to an algorithm with a stochastic nature: simply guess the stochastic algorithm's average. (This is a result of Jensen's inequality; see [Wolpert 1995b, Perrone 1993].) This is true whether the stochasticity is due to non-single-valued h or (as with backprop with a random initial weight) due to the learning algorithm's being non-deterministic.

Now the EBF is symmetric under $h \leftrightarrow f$. Accordingly, this kind of result can immediately be turned around. In such a form it says, loosely speaking, that a prior that is the "average" of another prior assuredly results in lower expected cost, *regardless of the learning algorithm*. In this particular sense, for quadratic loss, one can place an algorithm-independent ordering over some priors. (Of course, one can also order them in an algorithm-dependent manner if one wishes, for example by looking at the expected generalization error of the Bayes-optimal learning algorithm for the prior in question.)

The exact opposite behavior holds for loss functions that are concave rather than convex. For such functions, guessing randomly is assuredly superior to guessing the average, regardless of the target. (There is a caveat to this: one cannot have a loss function that is both concave everywhere across an infinite **Y** and nowhere negative, so formally, this statement only holds if we know that the y_F and y_H are both in a region of concave loss.)

ii) General properties of metric loss functions

Finally, there are other special properties that some loss functions possess but that quadratic loss does not. For example, if the loss can be written as a function L(., .) that is a metric (e.g., absolute value loss, zero-one loss), then for any f,

$$7) \qquad \left| E(C \mid f, h_1, m, q) - E(C \mid f, h_2, m, q) \right| \le \Sigma_{y,y'} L(y, y')\, h_1(q, y)\, h_2(q, y').$$

So for such loss functions, you can bound how much replacing h_1 by h_2 can improve / hurt generalization by looking only at h_1 and h_2. That bound holds without any concern for the prior over f. It is simply the expected loss between h_1 and h_2.

Unfortunately, quadratic loss is not a metric, and therefore one cannot employ this bound for

Moreover, Eq. (4) holds even when z is not a subset of {f, m, q}. So for example, $E(C \mid h, q)$ can be expressed directly in terms of bias-plus-variance by using Eq. (4). However there is no simply way to express the same quantity in terms of Eq. (1).

Indeed, Eq. (4) even allows us to write the purely Bayesian quantity $E(C \mid d)$ in bias-plus-variance terms, something we cannot do using Eq. (1):

6) $E(C \mid d) = \sigma_d^2 + (\text{bias}_d)^2 + \text{variance}_d - 2\text{cov}_d$,

where

$$\sigma_d^2 \equiv E(Y_F^2 \mid d) - [E(Y_F \mid d)]^2,$$
$$\text{bias}_d \equiv E(Y_F \mid d) - E(Y_H \mid d),$$
$$\text{variance}_d \equiv E(Y_H^2 \mid d) - [E(Y_H \mid d)]^2, \text{ and}$$
$$\text{cov}_d \equiv \Sigma_{y_F, y_H} P(y_H, y_F \mid d) \times [y_H - E(Y_H \mid d)] \times [y_F - E(Y_F \mid d)].$$

By using this formula one doesn't even have to go to a "mid-way point" between Bayesian analysis and conventional bias-plus-variance analysis to relate the two. Rather Eq. (6) directly provides a fully Bayesian bias-plus-variance decomposition. So for example, as long as one is aware that different variables are being averaged over than are for the conventional bias-variance decomposition (namely q and f rather than d), Eq. (6) allows us to directly say that for quadratic loss the Bayes-optimal learning algorithm has zero bias.

None of this means that one "should not" use the appropriate average of Eq. (1) (in those cases where there is such an average) rather than Eq. (4) and its corollaries. There are scenarios in which that average provides a helpful perspective on the learning problem that Eq. (4) does not. For example, if **Y** contains very few values (e.g., two) then in many scenarios $\text{bias}_{f,m}$, which equals $\Sigma_q P(q \mid f, m) [E(Y_H \mid f, m, q) - E(Y_F \mid f, q)]$, is close to zero, regardless of the learning algorithm. In the same scenarios though, the associated "q-average of Eq. (1)" term, $\Sigma_q P(q \mid f, m) \text{bias}_{f,m,q}^2 = \Sigma_q P(q \mid f, m) [E(Y_H \mid f, m, q) - E(Y_F \mid f, q)]^2$, is often far from zero. In such cases the bias term in Eq. (5) is not very informative whereas the associated q-average term is.

In the end, which bias-plus-variance decomposition one uses, just like the choice of what variables z represents, depends on what one is trying to understand about one's learning problem.

VI OTHER CHARACTERISTICS ASSOCIATED WITH THE LOSS

for the one presented in the previous subsection, there is not a bias-variance dilemma. Rather there is a bias-variance-covariance dilemma.

It is only for a rather specialized kind of analysis, where both q and f are fixed, that one can ignore the covariance term. For almost all other analyses - in particular the popular "generalization error" analyses - the huge body of lore explaining various aspects of supervised learning in terms of a "bias-variance dilemma" is less than the whole story.

ii) Averaging over a random variable vs. having it be in z

Before leaving the topic of Eq.'s (4, 5), it should be pointed out that one could just as easily use Eq. (1) and write $E(C \mid f, m) = \Sigma_q P(q \mid f, m) [\sigma_{f,m,q}^2 + \text{bias}_{f,m,q}^2 + \text{variance}_{f,m,q}]$ rather than use Eq. (5). Under commonly made assumptions (see [Wolpert 1994]) $P(q \mid f, m)$ just equals $P(q)$, which is often called the "sampling distribution". In such cases, this q-average of Eq. (1) is straightforward to evaluate.

It is instructive to compare such a q-average to the expansion in Eq. (5). First, such a q-average is often less informative than Eq. (5). As an example, consider the simple case where f is single-valued (i.e., a single-valued function from **X** to **Y**), h is single-valued, and the same h is guessed for all training sets sampled from f. Then $\Sigma_q P(q \mid f, m) \sigma_{f,m,q}^2 = \Sigma_q P(q \mid f, m) \text{variance}_{f,m,q} = 0$; the q-averaged intrinsic noise and variance terms provide no useful information, and the expected error is given solely by the bias term. On the other hand, $\sigma_{f,m}^2$ tells us the amount that f(q) varies about its average (over q) value, and $\text{variance}_{f,m}$ is given by a similar term. In addition $\text{bias}_{f,m}$ tells us the difference between those q-averages of f and of h. And finally, the covariance term tells us how much our h "tracks" f as one moves across **X**. So all the terms in Eq. (5) provide helpful information, whereas the q-average of Eq. (1) reduces to the tautology "$\Sigma_q P(q \mid f, m) E(C \mid f, m, q) = \Sigma_q P(q \mid f, m) E(C \mid f, m, q)$".

In addition to this difficulty, in certain respects the individual terms in the q-average of Eq. 1 do not meet all of our desiderata. In particular, desideratum (b) is not met in general, if one tries to identify "bias" as $\Sigma_q P(q \mid f, m) \text{bias}_{f,m,q}^2$ (note that here, the 'z' referred to in desideratum (b) is {f, m}). On the other hand, the terms in Eq. (5) do meet all of our desiderata, including the third part of (a). (The best possible algorithm if one is given f and m is the algorithm that always guess $h(q, y) = \delta(y, E(Y_F \mid f))$, regardless of the data.)

There are also "aesthetic" advantages to using Eq. (4) and its corollaries (like Eq. (5)) rather than averages of Eq. (1). For example, Eq. (4) doesn't treat z = {f, m, q} as special in any sense; all z's are treated equally. This contrasts with formulas based on Eq. (1), in which one writes $E(C \mid f, m)$ as a q-average of $E(C \mid f, m, q)$, $E(C \mid m, q)$ as an f-average of $E(C \mid f, m, q)$, etc.

(as opposed to applied) communities; it seems fair to say that it is far more commonly investigated than is E(C | f, m, q) in both the machine learning and neural net literatures.

To address generalization error, note that for any set of random variables Z, taking (set) values z,

4) $E(C \mid z) = \sigma_z^2 + (\text{bias}_z)^2 + \text{variance}_z - 2\text{cov}_z$,

where

$\sigma_z^2 \equiv E(Y_F^2 \mid z) - [E(Y_F \mid z)]^2$,
$\text{bias}_z \equiv E(Y_F \mid z) - E(Y_H \mid z)$,
$\text{variance}_z \equiv E(Y_H^2 \mid z) - [E(Y_H \mid z)]^2$, and
$\text{cov}_z \equiv \Sigma_{y_F, y_H} P(y_H, y_F \mid z) \times [y_H - E(Y_H \mid z)] \times [y_F - E(Y_F \mid z)]$.

The terms in this formula can (usually) be interpreted just as the terms in the conventional (z = {f, m, q}) decomposition can. So for example variance_z measures the variability of the learning algorithm's guess as one varies over those random variables not specified in z.

Eq. (2) is a special case of this formula where z = {m, q}, and Eq. (1) is a special case where z = {f, m, q} (and consequently the covariance term vanishes). However both of these equations have z contain q, when (by Eq. (4)) we could just as easily have z not contain q. By doing that we get a correction to the bias-plus-variance formula for "generalization error". (Note that this correction is in no sense a "Bayesian" correction.) To be more precise, the following is an immediate corollary of Eq. (4):

5) $E(C \mid f, m) = \sigma_{f,m}^2 + (\text{bias}_{f,m})^2 + \text{variance}_{f,m} - 2\text{cov}_{f,m}$,

(with definitions of the terms given in Eq. (4)).

Note that corrections similar to that of Eq. (5) hold for E(C | f, d) and E(C | m). In all three of these cases, σ_z^2, bias_z, and variance_z play the same role as do $\sigma_{f,m,q}^2$, $\text{bias}_{f,m,q}$, and $\text{variance}_{f,m,q}$ in Eq. (1). The only difference is that different quantities are averaged over. (So for example, in E(C | f, d), variance reflects variability in the learning algorithm's guess as one varies q.) In particular, most of our desiderata for these quantities are met.

In addition though, for all three of these cases, $P(y_H, y_F \mid z) \neq P(y_H \mid z)\, P(y_F \mid z)$ in general, and therefore the covariance correction term is non-zero, in general. So for these three cases, as well as

$\sigma_{m,q}{}^2$. So the data-worth is the difference between the expected cost of this algorithm and that of the Bayes-optimal algorithm.

Note the nice property that when the variance (as one varies training sets d) of the Bayes-optimal algorithm is large, so is data-worth. So, reasonably enough, when the Bayes-optimal algorithm's variance is large, there is a large potential gain in paying attention to the data. Conversely, if the variance for the Bayes-optimal algorithm is small, then not much can be gained by using it rather the optimal data-independent learning algorithm.

As it must, $E(C \mid m, q)$ reduces to the expression in Eq. (1) for $E(C \mid f = f^*, m, q)$ for the prior $P(f) = \delta(f - f^*)$. The special case of Eq. (2) where there is no noise, and the learning algorithm always guesses the same single-valued input-output function for the same training set, is given in [Wolpert, 1995a].

One can argue that $E(C \mid m, q)$ is usually of more direct interest than $E(C \mid f, m, q)$, since one can rarely specify the target in the real world but must instead be content to characterize it with a probability distribution. Insofar as this is true, by Eq. (2) there is not a "bias-variance" trade-off, as is conventionally stated. Rather there is a "bias-variance-covariance" trade-off.

More generally, one can argue that one should *always* analyze the distribution $P(C \mid z)$ where z is chosen to directly reflect the statistical scenario with which one is confronted. (See the discussion of the "honesty principle" at the end of [Wolpert 1995a]). In the real world, this usually entails having $z = \{d\}$. In toy experiments with a fixed target, it usually means having $z = \{f, m\}$. For other kinds of experiments it means other kinds of z's.

The only justification for not setting z this way is calculational intractability of the resultant analysis and/or difficulty in determining the distributions needed to perform that resultant analysis. (This latter issue is why Bayesian analysis does not automatically meet our needs in the real world - with such analysis there is often the issue of how to determine P(f), the prior.) However at the level of abstraction of this paper, neither difficulty arises. So for example the 'applicability' of the covariance terms introduced in this paper is determined solely by the statistical scenario with which one is confronted.

V OTHER CORRECTIONS TO QUADRATIC LOSS BIAS-PLUS VARAIANCE

i) The general quadratic loss bias-plus-variance-plus-covariance decomposition

Often in supervised learning one is interested in "generalization error", the average error between f and h over all q. For fixed f and m, the expectation of this error is $E(C \mid f, m)$. This quantity is ubiquitous in computational learning theory (COLT) as well as several other popular theoretical approaches to supervised learning [Wolpert 1995a]. Nor is interest in it restricted to the theoretical

This is intuitively reasonable. Indeed, the importance of such "tracking" between the learning algorithm P(h | d) and the posterior P(f | d) is to be expected, given that E(C | m, q) can also be written as a non-Euclidean inner product between P(f | d) and P(h | d). (This is true for any loss function - see [Wolpert 1995a].)

ii) Discussion

The terms $\text{bias}_{m,q}$, $\text{variance}_{m,q}$, and $\sigma_{m,q}$ play the same roles as $\text{bias}_{f,m,q}$, variance, and $\sigma_{f,m,q}$ do in Eq. (1). The major difference is that here they involve averages over f according to P(f), since the target f is not fixed. In particular, desiderata (b) and (c) are obeyed exactly by $\text{bias}_{m,q}$ and $\text{variance}_{m,q}$. Similarly the first part of desideratum (a) is obeyed exactly, if the reference to "f" there is taken to mean all f for which P(f) is non-zero, and if the delta functions referred to in (a) are implicitly restricted to be identical (at q) for all such f. In addition $\sigma_{m,q}{}^2$ is independent of the learning algorithm, in agreement with the second part of desideratum (a).

However now that we have the covariance term, the third part of desideratum (a) is no longer obeyed. Indeed, by using $P(d, y_F \mid m, q) = P(y_F \mid d, q)\, P(d \mid m, q)$ we can rewrite $\sigma_{m,q}{}^2$ as the sum of the expected cost of the best possible (Bayes-optimal) learning algorithm for quadratic loss, that is the loss of the learning algorithm that obeys $P(y_H \mid d, q) = \delta(y_H, E(Y_F \mid d, q))$, plus another term. (Here and throughout, any expression of the form "δ(., .)" indicates the Kronecker delta function.) That term is called the "data-worth" of the problem, since as explained below, it sets how much of an improvement in error can be had from paying attention to the data.

$$3)\quad \sigma_{m,q}{}^2 = \Sigma_{d,y_F} P(d, y_F \mid m, q)\ [y_F - E(Y_F \mid d, q)]^2 \qquad \text{(the Bayes-optimal algorithm's cost)}$$
$$+$$
$$\Sigma_d\, P(d \mid m, q)\ \left([E(Y_F \mid d, q)]^2 - [E(Y_F \mid m, q)]^2 \right) \quad \text{(the data-worth)}$$

One might wonder why all of this does not also apply to the conventional bias-variance decomposition for z = {f, m, q}. The reason is that for f-conditioned probabilities, the best possible algorithm doesn't guess $E(Y_F \mid d, q)$ but rather $E(Y_F \mid f, q)$. This is why this decomposition doesn't also apply to $\sigma_{f,m,q}{}^2$.

Note that for the Bayes-optimal learning algorithm, the data-worth is exactly $\text{cov}_{m,q}$. This is to be expected, since for that learning algorithm $\text{bias}_{m,q}$ equals 0 and $\text{variance}_{m,q}$ equals $\text{cov}_{m,q}$.[2]

To see why the data-worth measures how much paying attention to the data can help you to guess f, note that the expected cost of the best possible *data-independent* learning algorithm equals

- the algorithm that always "by luck" guesses $y_H = E(Y_F \mid f, q)$, independent of d. (Indeed, Breiman's bagging scheme [Breiman 1994] is usually justified as a way to try to estimate that "lucky" algorithm.) In addition, in this "mid-way" approach, rather than fix d as in the Bayesian approach, one averages over d, as in bias-plus-variance. In this way one maintains the illustrative power of the bias-plus-variance formula.

The result is the following "Bayesian correction" to the quadratic loss bias-plus-variance formula [Wolpert 1995a]:

2) $E(C \mid m, q) = \sigma_{m,q}^2 + (\text{bias}_{m,q})^2 + \text{variance}_{m,q} - 2\text{cov}_{m,q}$,

where

$$\sigma_{m,q}^2 \equiv E(Y_F^2 \mid q) - [E(Y_F \mid q)]^2,$$
$$\text{bias}_{m,q} \equiv E(Y_F \mid q) - E(Y_H \mid m, q),$$
$$\text{variance}_{m,q} \equiv E(Y_H^2 \mid m, q) - [E(Y_H \mid m, q)]^2, \text{ and}$$
$$\text{cov}_{m,q} \equiv \Sigma_{y_F, y_H} P(y_H, y_F \mid m, q) \times [y_H - E(Y_H \mid m, q)] \times [y_F - E(Y_F \mid q)].$$

In this equation, the terms $E(Y_F \mid q)$, $E(Y_F^2 \mid q)$, $E(Y_H \mid q, m)$ and $E(Y_H^2 \mid q, m)$ are given by the formulas just before Eq. (1), provided one adds an outer integral $\int df\, P(f)$, to average out f. To evaluate the covariance term, use $P(y_H, y_F \mid m, q) = \int dh\, df\, \Sigma_d P(y_H, y_F, h, d, f, \mid m, q)$. Then use the simple identity

$$P(y_H, y_F, h, d, f, \mid m, q) = f(q, y_F)\, h(q, y_H)\, P(h \mid d)\, P(d \mid f, q, m)\, P(f).$$

Formally, the reason that the covariance term exists in Eq. (2) when there was none in Eq. (1) is that y_H and y_F are conditionally independent if one is given f and q (as in Eq. (1)), but not only given q (as in Eq. (2)). To illustrate the latter point, note that knowing y_F , for example, tells you something about f you don't already know (assuming f is not fixed, as it is in Eq. (1)). This in turn tells you something about d, and therefore something about h and y_H. In this way y_H and y_F are statistically coupled if f is not fixed.

Intuitively, the covariance term simply says that one would like the learning algorithm's guess to "track" the (posterior) most likely targets, as one varies training sets. Without such tracking, simply having low $\text{bias}_{m,q}$ and low $\text{variance}_{m,q}$ does not imply good generalization.

(e.g., from a finite data set). If there were discontinuities and the target were near such a discontinuity, the resultant estimates would often be poor. More generally, it would be difficult to ascribe the usual intuitive meanings to the terms in the decomposition if they were discontinuous functions of the target;

Desiderata (a) through (e) are somewhat more general than conditions (i) through (iv), in that (for example) they are meaningful even if **Y** is a non-numeric space, so that expressions like "$E(Y_F \mid z)$" are not defined. Accordingly, I will rely on them more than on conditions (i) through (iv) in the extensions of the bias-plus-variance formula presented below.

In the final analysis though, both the conditions (i) through (iv) and the desiderata (a) through (e) are not God-given principles that any bias-plus-variance decomposition must obey. Rather they are useful aspects of the decomposition that facilitate that decompostion's "intuitive and easy" interpretation. There is nothing that precludes one's using slight variants of these conditions, or perhaps even replacing them altogether.

The next two sections show how to generalize the bias-plus-variance formula to other conditioning events besides z = {f, m, q} while still obeying (almost all of) (i) through (iv) and (a) through (c). First in section 4 the generalization to z = {m, q} is presented. Then in section 5, the generalization to arbitrary z is explored.

IV THE MID-WAY POINT BETWEEN BAYESIAN ANALYSIS AND QUADRATIC LOSS BIAS-PLUS-VARIANCE

i) Bias-plus-variance for when one averages over targets - the covariance correction

It is important to realize that illustrative as it is, the bias-plus-variance formula "examines the wrong quantity". In the real world, it is almost never $E(C \mid f, m)$ that is *directly* of interest, but rather $E(C \mid d)$. (We know d, and therefore can fix its value in the conditioning event. We do not know f.) Analyzing $E(C \mid d)$ is the purview of Bayesian analysis [Buntine and Weigend 1991, Bernardo and Smith 1994]. Generically, it says that for quadratic loss, one should guess the posterior average y [Wolpert 1995a].

As conventionally discussed, $E(C \mid d)$ does not bear any connection to the bias-plus-variance formula. However there is a "mid-way" point between Bayesian analysis and the kind of analysis that results in the bias-plus-variance formula.

In this middle approach, rather than fix f as in bias-plus-variance, one averages over it, as in the Bayesian approach. In this way one circumvents the annoying fact that there need not be a bias-variance trade-off, in that there exists an algorithm with both zero $\text{bias}_{f,m,q}$ and zero $\text{variance}_{f,m,q}$

algorithm, and the variance term to reflect the learning algorithm alone. In particular, for the usual intuitive characterization of variance to hold, we want it to reflect how sensitive the algorithm is to changes in the data set.

Note that although σ_z^2 appears to be identical to variance_z if one simply replaces Y_F with Y_H, the two quantities have different kinds of relations with the other random variables. For example, for z = {f, m, q}, variance_z depends on the target as well as the learning algorithm, whereas σ_z^2 only depends on the target.

So expected quadratic loss reflects noise in the target, plus the difference between the target and the average guess, plus the variability in the guessing. In particular, we have the following properties, which can be viewed as desiderata for our three terms:

a) If f is a delta function in **Y** for q (i.e., if at the point **X** = q, f is a single-valued function from **X** to **Y**), the intrinsic noise term (i) equals 0. In addition, the intrinsic noise term is independent of the learning algorithm. Finally, the intrinsic noise term is a lower bound on the error - for no learning algorithm can E(C | z) be lower than the intrinsic noise term. (In fact, for the decomposition in (1), the intrinsic noise term is the greatest upper bound on that error);

b) If the average hypothesis-determined guess equals the average target-determined "guess"**,** then $\text{bias}_z = 0$. It is large if the difference between those averages is large;

c) Variance is non-negative, equals 0 if the guessed h is always the same single-valued function (independent of d), and is large when the guessed h varies greatly in response to changes in d.

d) The variance does not depend on z directly, but only indirectly through the induced distribution P(h | z). I.e., for any z, the associated variance is set by the h-dependence of P(h | z). Now P(h | z) = Σ_d P(h | d, z) P(d | z). Moreover, it is often the case that P(h | d, z) = P(h | d). (E.g., this holds for z = {f, m q}.) In such a case, if one knows the algorithm (i.e., if one know P(h | d)), then the h-dependence of P(h | z) is set by the d-dependence of P(d | z). This means that changes to z that do not affect the induced distribution over d do not affect the variance. As before, this is needed so that the variance term reflects only how sensitive the algorithm is to changes in the training set.

e) All the terms in the decomposition are continuous functions of the target. This is particularly desirable when one wishes to estimate those terms from limited information concerning the target

pense of increased variance.

In addition, the terms in the bias-plus-variance formula all involve (functions of) expectation values of the fundamental random variables described in the previous section - no new random variables are involved. This means that the bias-plus-variance formula is particularly intuitive and easy to interpret, as the discussion in the next subsection illustrates.

ii) Desiderata obeyed by the terms in the quadratic loss bias-plus-variance formula

To facilitate the generalization of the bias-plus-variance formula, define $z \equiv \{f, m, q\}$, the set of values of random variables we're conditioning on in Eq. 1. Then intuitively, in Eq. 1, for the point q,

i) σ_z^2 measures the intrinsic error due to the target f, independent of the learning algorithm. Here it is given by $E(C \mid h, z) / 2$ for $h = f$, i.e., it equals half the expected loss of f at "guessing itself" at the point q;

ii) The bias measures the difference between the average Y_H and the average Y_F (where Y_F is formed by sampling f and Y_H is formed by sampling h's created from d's that are in turn created from f);

iii) Alternatively, $\sigma_{f,m,q}^2$ plus the squared bias measures the expected loss between Y_F and the average Y_H, $E((Y_F - [E(Y_H \mid z)])^2 \mid z)$;

iv) The variance measures the "variability" of the guessed y_H about the average y_H as one varies over training sets (generated according to the given fixed value of z). If the learning algorithm always guesses the same h for the same d, and that h is always a single-valued function from **X** to **Y**, then the variance is given directly by the variability of the learning algorithm's guess as d is varied.

v) It is worth pointing one special property for when $z = \{f, m, q\}$. For that case the variance does not depend on f directly, but only indirectly through the induced distribution over training sets, $P(d \mid f, m, q)$. So for example consider having the target be changed in such a way that the resultant distribution $P(d \mid f, m, q)$ over training sets does not change. (For instance, this would be the case if there were negligibly small probability that the q at hand exists in the training set, perhaps because $n >> m$.) Then the variance term does not change. This is desirable because we wish the intrinsic noise term to reflect the target alone, the bias to reflect the target's relation with the learning

$$E(Y_H^2 \mid f, q, m) = \int dh\, \Sigma_d P(d \mid f, q, m) P(h \mid d) \Sigma_y y^2 h(q, y),$$

where for succinctness the m-conditioning in the expectation values is not indicated if the expression is independent of m. These are, in order, the average **Y** and $\mathbf{Y}^2$ values of the target (at q), and of the average of the hypotheses made in response to training sets generated from the target (again, evaluated at q). Note that these averages need not exist in **Y**, in general. For example, this is almost always the case if **Y** is binary.

Now write $C = (Y_H - Y_F)^2$. Then simple algebra (use the conditional independence of Y_H and Y_F) verifies the following formula:

1) $E(C \mid f, m, q) = \sigma_{f,m,q}{}^2 + (\text{bias}_{f,m,q})^2 + \text{variance}_{f,m,q}$,

where

$\sigma_{f,m,q}{}^2 \equiv E(Y_F{}^2 \mid f, q) - [E(Y_F \mid f, q)]^2$,

$\text{bias}_{f,m,q} \equiv E(Y_F \mid f, q) - E(Y_H \mid f, q, m)$,

$\text{variance}_{f,m,q} \equiv E(Y_H{}^2 \mid f, q, m) - [E(Y_H \mid f, q, m)]^2$.

The subscript {f,m,q} indicates the conditioning event for the expected error, and will become important below. When the conditioning event is clear, or not important, $\text{bias}_{f,m,q}$ may be referred to simply as "the bias", and similarly for the variance and the noise.

The bias-variance formula in [Geman et al. 1992] is a special case of Eq. (1), where the learning algorithm always guesses the same h given the same training set d (something which is not the case for backpropagation with a random initial weight, for example). In addition, in [Geman et al. 1992] the hypothesis h that the learning algorithm guesses is always a single-valued mapping from **X** to **Y**.

Note that essentially no assumptions are made in deriving Eq. (1). Any likelihood is allowed, any learning algorithm, and relationship between q and f and/or d, etc. This will be true for all of the analysis in this paper.

In addition to such generality, the utility of the bias-plus-variance formula lies in the fact that very often there is a "bias-variance" trade-off. For example, it may be that a modification to a learning algorithm improves its bias for the target at hand. (This often happens when more free parameters are incorporated into the learning algorithm's model, for example.) But this is often at the ex-

The sets **X** and **Y**, of sizes n and r:	The input and output space, respectively.
The set d, of m **X**-**Y** pairs:	The training set.
The **X**-conditioned distribution over **Y**, f:	The target, used to generate test sets.
The **X**-conditioned distribution over **Y**, h:	The hypothesis, used to guess for test sets.
The real number c:	The cost.
The **X**-value q:	The test set point.
The **Y**-value y_F:	The sample of the target f at point q.
The **Y**-value y_H:	The sample of the hypothesis h at point q.
$P(h \mid d)$:	The learning algorithm.
$P(f \mid d)$:	The posterior.
$P(d \mid f)$:	The likelihood.
$P(f)$:	The prior.

If $c = L(y_F, y_H)$, $L(.,.)$ is the "loss function". Otherwise c is given by a "scoring rule".

Table 1: Summary of the terms in the EBF.

III BIAS PLUS VARIANCE FOR QUADRATIC LOSS

i) The bias-plus-variance formula

This section reviews the conventional bias-plus-variance formula for quadratic los, with a fixed targets and averages over training sets. Write

$$E(Y_F \mid f, q) = \Sigma_y \, y \, f(q, y),$$

$$E(Y_F{}^2 \mid f, q) = \Sigma_y \, y^2 \, f(q, y),$$

$$E(Y_H \mid f, q, m) = \int dh \, \Sigma_d \, P(d \mid f, q, m) \, P(h \mid d) \, \Sigma_y \, y \, h(q, y), \text{ and}$$

As an example, for "logarithmic scoring", $P(c \mid f, h, q) = \delta\{ c - \Sigma_y f(q, y) \ln[h(q, y)] \}$. This cost is (the logarithm of the geometric mean of) the probability one would assign to an infinite data set generated according to the target f, if one had assumed (erroneously) that it was actually generated according to the hypothesis h.

15) The "generalization error function" used in much of supervised learning is given by $c' \equiv E(C \mid f, h, d)$. It is the average over all q of the cost c, for a given target f, hypothesis h, and training set d.

v) Miscellaneous

16) Note the implicit rule of probability theory that any random variable not conditioned on is marginalized over. So for example (using the conditional independencies in conventional supervised learning), expected cost given the target, training set size, and test set point, is given by

$$E(C \mid f, m, q) = \int dh\, \Sigma_d\, E(C \mid h, d, f, q)\, P(h \mid d, f, q, m)\, P(d \mid f, q, m)$$

$$= \int dh\, \Sigma_d\, E(C \mid f, h, q)\, P(h \mid d)\, P(d \mid f, q, m)$$

$$= \int dh\, E(C \mid f, h, q)\, \{\Sigma_d\, P(h \mid d)\, P(d_Y \mid f, d_X)\, P(d_X \mid f, m, q)\}.$$

(I do not equate $P(d_X \mid f, q, m)$ with $P(d_X \mid m)$ - as is conventionally (though implicitly) done in most theoretical supervised learning - because in general the test set point q may be coupled to d_X and even f. See [Wolpert 1995a].)

some y. Such a distribution is a single-valued function from **X** to **Y**. As an example, if one is using a neural net as one's regression through the training set, usually the (neural net) h is single-valued. On the other hand, when one is performing probabilistic classification (as in softmax), h isn't single-valued.

9) Any (!) learning algorithm (aka "generalizer") is given by $P(h \mid d)$, although writing down a learning algorithm's $P(h \mid d)$ explicitly is often quite difficult. A learning algorithm is "deterministic" if the same d always gives the same h. Backprop with a random initial weight is not deterministic. Nearest neighbor is.

10) The learning algorithm only sees the training set d, and in particular does not directly see the target. So $P(h \mid f, d) = P(h \mid d)$, which means that $P(h, f \mid d) = P(h \mid d) \times P(f \mid d)$, and therefore $P(f \mid h, d) = P(h, f \mid d) / P(h \mid d) = P(f \mid d)$.

11) By definition of f, in supervised learning, Y_F and Y_H are conditionally independent given f and q: $P(y_F, y_H \mid f, q) = P(y_F \mid y_H, f, q)\, P(y_H \mid f, q) = P(y_F \mid f, q)\, P(y_H \mid f, q)$.

12) Similarly, Y_F and Y_H are conditionally independent given d and q.
Proof: $P(y_F, y_H \mid d, q) = P(y_F \mid d, q)\, P(y_H \mid d, q, y_F) = P(y_F \mid d, q) \int dh\, P(y_H \mid h, d, q, y_F)\, P(h \mid d, q, y_F) = P(y_F \mid d, q) \int dh\, P(y_H \mid h, q)\, P(h \mid d) = P(y_F \mid d, q) \int dh\, P(y_H \mid h, d, q)\, P(h \mid d, q) = P(y_F \mid d, q)\, P(y_H \mid d, q)$. QED.

iv) The cost and "generalization error"

13) Given values of F, H, and a test set point $q \in X$, the associated "cost" or "error" is indicated by the random variable C. Often C is a "loss function", and can be expressed in terms of a mapping L taking $Y \times Y$ to a real number. Formally, in these cases the probability that C takes on the value c, conditioned on given values h, f and q, is $P(c \mid f, h, q) = \Sigma_{y_H, y_F} P(c \mid y_H, y_F)\, P(y_H, y_F \mid f, h, q) = \Sigma_{y_H, y_F} \delta\{c - L(y_H, y_F)\}.h(q, y_H)\, f(q, y_F)$[1]. As an example, quadratic loss has $L(y_H, y_F) = (y_H - y_F)^2$, so $E(C \mid f, h, q) = \Sigma_{y_H, y_F} f(q, y_F)\, h(q, y_H)\, (y_H - y_F)^2$.

14) Generically, when the distribution of c given f, h and q cannot be reduced in this way to a loss function from $Y \times Y$ to **R**, it will be referred to as a "scoring rule". Scoring rules are often appropriate when you're trying to guess a distribution over Y, and loss functions are usually appropriate when you are trying to guess a particular value in Y.

set, y_F and y_H associated samples of the outputs of the two neural nets for that element (the sampling of y_F including the effects of the superimposed noise), and c the resultant "cost" (e.g., c could be $(y_F - y_H)^2$).

ii) Training sets and targets

4) m is the number of elements in the (ordered) training set d. $\{d_{\mathbf{X}}(i), d_{\mathbf{Y}}(i)\}$ is the set of m input and output values in d. m' is the number of distinct values in $d_{\mathbf{X}}$.

5) Targets f are always assumed to be of the form of **X**-conditioned distributions over **Y**, indicated by the real-valued function $f(x \in \mathbf{X}, y \in \mathbf{Y})$ (i.e., $P(y_F \mid f, q) = f(q, y_F)$). Equivalently, where S_r is defined as the r-dimensional unit simplex, targets can be viewed as mappings $f: \mathbf{X} \to S_r$. Note that any such target is a finite set of real numbers indexed by an **X**-value and a **Y** value.

Any restrictions on f are imposed by the full joint distribution P(f, h, d, c), and in particular by its marginalization, P(f). Note that any output noise process is automatically reflected in $P(y_F \mid f, q)$. Note also that the definition $P(y_F \mid f, q) = f(q, y_F)$ only directly refers to the generation of test set elements; in general, training set elements can be generated from targets in a different manner.

6) The "likelihood" is P(d | f). It says how d was generated from f. As an example, the conventional IID likelihood is $P(d \mid f) = \Pi^{m}_{i=1} \ \pi(d_{\mathbf{X}}(i)) \times f(d_{\mathbf{X}}(i), d_{\mathbf{Y}}(i))$ (where $\pi(x)$ is the "sampling distribution"). In other words, under this likelihood d is created by repeatedly and independently choosing an input value $d_{\mathbf{X}}(i)$ by sampling $\pi(x)$, and then choosing an associated output value by sampling $f(d_{\mathbf{X}}(i), .)$, the same distribution used to generate test set outputs.

None of the results in this paper depend on the choice of the likelihood.

7) The "posterior" usually means P(f | d), and the "prior" usually means P(f).

iii) The learning algorithm

8) Hypotheses h are always assumed to be of the form of **X**-conditioned distributions over **Y**, indicated by the real-valued function $h(x \in \mathbf{X}, y \in \mathbf{Y})$ (i.e., $P(y_H \mid h, q) = h(q, y_H)$). Equivalently, where S_r is defined as the r-dimensional unit simplex, hypotheses can be viewed as mappings $h: \mathbf{X} \to S_r$. Note that any such hypothesis is a finite set of real numbers.

Any restrictions on h are imposed by the full joint distribution P(f, h, d, c).

Here and throughout, a "single-valued" distribution is one that, for a given x, is a delta function about

1. Readers unsure of any aspects of this synopsis, and in particular unsure of any of the formal basis of the EBF or justifications for any of its assumptions, are directed to the detailed exposition of the EBF in appendix A of [Wolpert 1996; paper 1].

i) Overview

1) The input and output spaces are **X** and **Y**, respectively. For simplicity, they are taken to be finite. (This imposes no restrictions on the real-world utility of the results in this paper, since in the real world data is always analyzed on a finite digital computer, and is in based on the output of instruments having a finite number of possible readings.)

The two spaces contain n and r elements respectively. A generic element of **X** is indicated by ‘x’, and a generic element of **Y** is indicated by ‘y’. Sometimes (e.g., when requiring a Bayes-optimal algorithm to guess an expected **Y** value) it will implicitly be assumed that **Y** is a large set of real numbers that are very close to one another, so that there is no significant difference between the element in **Y** closest to some real number ψ and that number itself.

2) Random variables are indicated using capital letters. Associated instantiations of a random variable are indicated using lower case letters. Note though that some quantities (e.g., the space **X**) are neither random variables nor instantiations of random variables, and therefore their written case carries no significance.

Only rarely will it be necessary to refer to a random variable rather than an instantiation of it. In particular, whenever possible, the argument of a probability distribution will be taken to indicate the associated random variable. (E.g., whenever possible, “P(a)” will be written rather than “$P_A(a)$”.)

In accord with standard statistics notation, “E(A | b)” will be used to mean the expectation value of A given B = b, i.e., to mean $\int da\, a\, P(a \mid b)$. (Sums replace integrals if appropriate.)

3) The primary random variables are the hypothesis **X**-**Y** relationship output by the learning algorithm (indicated by H), the target (i.e., “true”) **X**-**Y** relationship (F), the training set (D), and the real world cost (C).

These variables are related to one another through other random variables representing the (test set) input space value (Q), and the associated target and hypothesis **Y**-values, Y_F and Y_H respectively (with instantiations y_F and y_H respectively).

This completes the list of random variables. Formal definitions of them appear below.

As an example of the relationship between these random variables and supervised learning, f, a particular instantiation of a target, could refer to a “teacher” neural net together with superimposed noise. This noise-corrupted neural net generates the training set d. The hypothesis h on the other hand could be the neural net made by one’s “student” algorithm after training on d. Then q would be an input element of the test

distribution over the output space rather than a single guessed output value. In those scenarios "scoring rules" are usually a more appropriate form of measuring generalization performance than are loss functions. This paper ends by presenting extensions of the fixed-target version of the bias-plus-variance formula to the logarithmic and quadratic scoring rules, and then presents the associated additive corrections to those formulas. ("Scoring rules" as explored in this paper are similar to what are called "score functions" in the statistics literature [Bernardo and Smith 1994].)

All of the correction terms presented in this paper are a covariance, between the learning algorithm and the posterior distribution over targets. Accordingly, in the (very common) contexts in which they apply, there is not a "bias-variance trade-off", or a "bias-variance dilemma", as one often hears. Rather there is a bias-variance-*covariance* trade-off.

Section 2 of this paper presents the formalism that will be used in the rest of the paper. Section 3 uses this formalism to recapitulate the traditional bias-plus-variance formula. Certain desiderata that the terms in the bias-plus-variance decomposition should meet are also presented there. Sections 4 and 5 then present the corrections to this formula appropriate for quadratic loss. Section 6 discusses some other loss-function-specific aspects of supervised learning.

Recently Friedman drew attention to an important aspect of expected error for zero-one loss []. His analysis appeared to indicate that under certain circumstances, when (what he identified as) the variance increased it could result in *decreased* generalization error. In Section 7, it is shown how to perform a bias-variance decomposition for Friedman's scenario where common-sense characteristics of bias and variance (like error being an increasing function of each) are preserved. This discussion serves as a useful illustration of the utility of covariance terms, since it is the presence of that term that explains Friedman's apparently counter-intuitive results.

Section 8 then begins presenting the extensions of the bias-plus-variance formula for scoring-rule-based rather than loss function-based error. That section investigates logarithmic scoring. Section 9 investigates quadratic scoring. Finally, section 10 discusses future work.

II NOMENCLATURE

This paper use the Extended Bayesian Formalism [Wolpert 1996, Wolpert 1994a, Wolpert 1992]. In the current context, the EBF is just conventional probability theory, applied to the case where one has a different random variable for the hypothesis output by the learning algorithm and for the target relationship. It is this crucial extension that separates the EBF from conventional Bayesian analysis, and that allows the EBF (unlike conventional Bayesian analysis) to subsume all other major mathematical treatments of supervised learning like computational learning theory, sampling theory statistics, etc. (See [Wolpert 1994a].)

This section presents a synopsis of the EBF. A quick reference of this synopsis can be found in Table

I INTRODUCTION

The bias-plus-variance formula [Geman et al. 1992] is an extremely powerful tool for analyzing supervised learning scenarios that have quadratic loss functions, fixed targets, and averages over training sets. Indeed there is little doubt that it is the most frequently cited formula in the statistical literature for analyzing such scenarios. Despite this breadth of utility in such scenarios however, the bias-plus-variance formula has never previously been extended to other learning scenarios.

In this paper an additive correction to the formula is presented, appropriate for learning scenarios where the target is not fixed. The associated formula for expected loss constitutes a "mid-way point" between Bayesian analysis and conventional bias-plus-variance analysis, in that both targets and training sets are averaged over.

After presenting this correction other correction terms are presented, appropriate for when other sets of random variables are averaged over. In particular, the correction term to the bias-plus-variance formula for when the test set point is not fixed - as it is not in almost all of computational learning theory as well as most other investigations of "generalization error" - is presented. (The conventional bias-plus-variance formula has the test point fixed.) In addition, it is shown how to cast conventional Bayesian analysis (where the training is fixed but targets are averaged over) directly in terms of bias-plus-variance. All of this serves to emphasize that the conventional bias-plus-variance decomposition is only a very specialized case of a much more general and important phenomenon.

Next is a brief discussion of some other loss-function-specific properties of supervised learning. In particular, it is shown how with quadratic loss there is a scheme that assuredly, independent of the target, improves the performance of any learning algorithm with a random component. On the other hand, using the same scheme for concave loss functions results in assured degradation of performance. It is also shown that, without any concern for the target, one can bound the change in zero-one loss generalization error associated with making some guess h_1 rather than a different guess h_2. (This is not possible for quadratic loss.)

All of these extensions to the conventional version of the bias-plus-variance formula use the same quadratic loss function occurring in the conventional formula itself. That loss function is often appropriate when the output spaces is numeric. Kong and Dietterich recently proposed an extension of the conventional (fixed target, training set-averaged) formula to the zero-one loss function [Kong and Dietterich 1995]. (See also the forthcoming papers [Wolpert and Kohavi 1996, Kohavi and Wolpert 1996].) That loss function is often appropriate when one's output space is categorical rather than numeric.

For such categorical output spaces sometimes one's algorithm produces a guessed probability

ON BIAS PLUS VARIANCE

by

David H. Wolpert

TXN Inc., and The Santa Fe Institute

Currently at IBM Almaden Research Center, N5Na/D3, 650 Harry Rd., San Jose, CA 95120

dhw@almaden.ibm.com

SFI TR 95-08-074

Abstract: This paper presents several additive "corrections" to the conventional quadratic loss bias-plus-variance formula. One of these corrections is appropriate when both the target is not fixed (as in Bayesian analysis) and also training sets are averaged over (as in the conventional bias-plus-variance formula). Another additive correction casts conventional fixed-training-set Bayesian analysis directly in terms of bias-plus-variance. Another correction is appropriate for measuring full generalization error over a test set rather than (as with conventional bias-plus-variance) error at a single point. Yet another correction can help explain the recent counter-intuitive bias-variance decomposition of Friedman for zero-one loss. After presenting these corrections this paper then discusses some other loss-function-specific aspects of supervised learning. In particular, there is a discussion of the fact that if the loss function is a metric (e.g., zero-one loss), then there is bound on the change in generalization error accompanying changing the algorithm's guess from h_1 to h_2 that depends only on h_1 and h_2 and not on the target. This paper ends by presenting versions of the bias-plus-variance formula appropriate for logarithmic and quadratic scoring, and then all the additive corrections appropriate to those formulas. All the correction terms presented in this paper are a covariance, between the learning algorithm and the posterior distribution over targets. Accordingly, in the (very common) contexts in which those terms apply, there is not a "bias-variance trade-off", or a "bias-variance dilemma", as one often hears. Rather there is a bias-variance-covariance trade-off.