GRADE: Machine-Learning Support for Graduate Admissions

Austin Waters and Risto Miikkulainen

■ *This article describes GRADE, a statistical machine-learning system developed to support the work of the graduate admissions committee at the University of Texas at Austin Department of Computer Science (UTCS). In recent years, the number of applications to the UTCS Ph.D. program has become too large to manage with a traditional review process. GRADE uses historical admissions data to predict how likely the committee is to admit each new applicant. It reports each prediction as a score similar to those used by human reviewers, and accompanies each by an explanation of what applicant features most influenced its prediction. GRADE makes the review process more efficient by enabling reviewers to spend most of their time on applicants near the decision boundary and by focusing their attention on parts of each applicant's file that matter the most. An evaluation over two seasons of Ph.D. admissions indicates that the system leads to dramatic time savings, reducing the total time spent on reviews by at least 74 percent.*

Graduate programs in fields such as computer science have received increasing interest in recent years. While the number of applicants to such programs has grown two- to threefold (figure 1), the number of faculty available to review applications has remained constant or grown very slowly over time. The result is that admissions committees face a prohibitively large workload, making it difficult to review applications thoroughly.

This article describes a system developed to support the work of the graduate admissions committees in the Department of Computer Science at the University of Texas at Austin (UTCS). The system, named GRADE (graduate admissions evaluator), uses statistical machine learning to estimate the quality of new applicants based on past admissions decisions. GRADE does not determine who is admitted or rejected from the graduate program. Rather, its purpose is to inform the admissions committee and make the process of reviewing files more efficient. The heart of GRADE is a probabilistic classifier that predicts how likely the committee is to admit each applicant based on the information provided in his or her application file. For each new applicant, the system estimates this probability, expresses it as a numerical score similar to those used by human reviewers, and generates human-readable information explaining what factors most influenced its prediction.

While every application is still looked at by a human reviewer, GRADE makes the review process much more efficient. This is for two reasons. First, GRADE reduces the total number of full application reviews the committee must perform. Using the system's predictions, reviewers can quickly identify a large number of weak candidates who will likely be rejected and a smaller number of exceptionally strong candi-

 ISSN 0738-4602

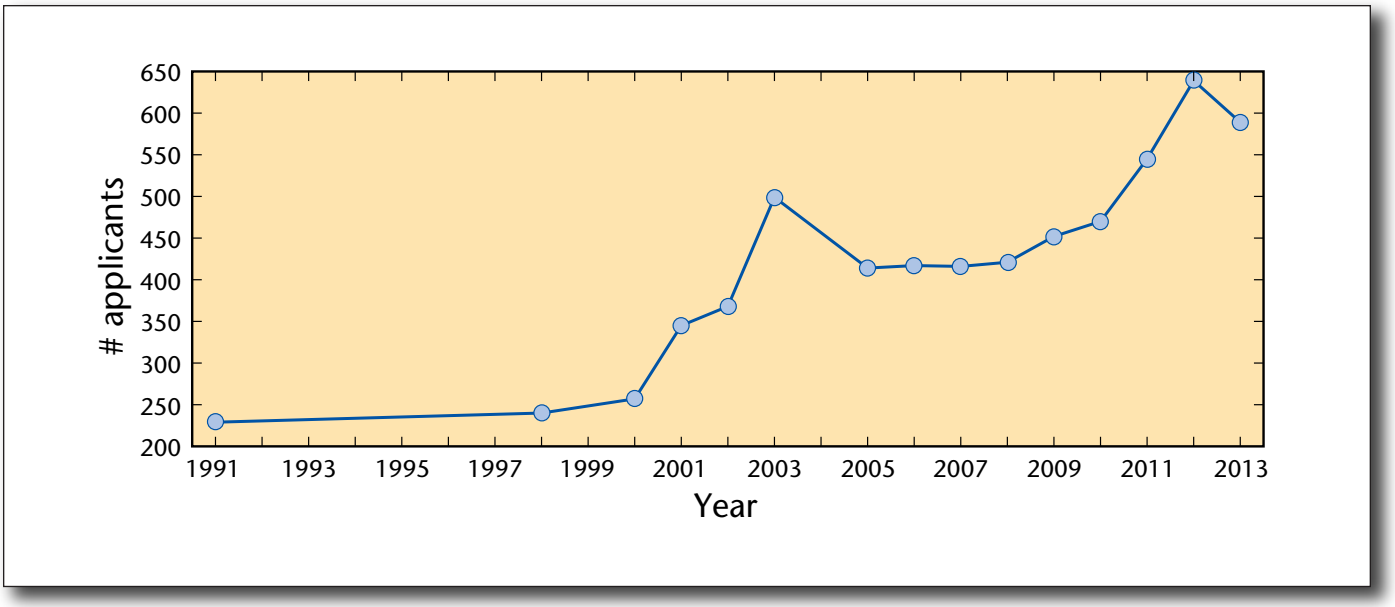

Figure 1. Number of Applicants to the UTCS Ph.D. Program over Time.

Applicant pools have grown significantly in recent years, putting a strain on admissions committees, who have finite resources. (Data not available for some years.)

dates who will likely be admitted. These decisions can be validated quickly, leaving the committee with more time to carefully evaluate the remaining, borderline applicants. Second, GRADE makes it easier to review an individual applicant's file. The system generates an initial ranking for the applicant pool, which makes it possible to review applicants in descending order of quality and provides an appropriate context for each new review. Its explanations also provide a starting point for the human reviewer to identify strong and weak attributes of each application, reducing the review time further. The system was first tested alongside the regular review process in the 2012 admissions season and fully integrated into the graduate admissions process in 2013. Compared to previous years, GRADE reduced the number of full reviews required per applicant by 71 percent and, by a conservative estimate, cut the total time spent reviewing files by at least 74 percent.

As a machine-learning problem, GRADE frames graduate admissions as probabilistic binary classification. For training data, the system reads an internal departmental database of past admissions decisions, which contains an anonymized version of each applicant's file and a binary label indicating whether or not the person was admitted to the graduate program. Each student's file is represented as a high-dimensional feature vector that encodes the institutions previously attended, GPAs, test scores, letters of recommendation, area of research interest, and preferred faculty advisor. Given the historical data, the goal is to predict the probability that the admissions committee will admit each new applicant to the program.

Internally, GRADE is implemented with an L_1-regularized logistic regression model. The regularization acts as a feature-selection mechanism in practice, producing a sparse model that uses only a small subset of highly predictive features. Upon inspection, the model focuses on much of the same information that human committee members use when reviewing applications. In particular, GRADE prefers applicants with high GPAs and test scores, backgrounds from reputable institutions, and recommendation letters that support the applicant's potential as a researcher. The feature weights learned by the classifier are discussed with other results in the Evaluation section.

Related Work

The literature contains numerous studies that perform statistical analyses on past admissions decisions, for example by Bruggink and Gambhir (1996); however, very little work has been done on systems that can predict admissions decisions and assist in the review process. One exception is by Moore (1998), who used decision tree induction to model admissions to an MBA program. In that work, predictions were made available to reviewers but otherwise had no role in the admissions process. Additionally, that model was evaluated on a much smaller data set than GRADE and did not incorporate features for schools' reputations, applicants' letters of recommendation, and others. In this sense, GRADE is a unique effort to incorporate machine learning into an admissions process.

Graduate Admissions

To understand the impact of the prediction system, we first give a high-level overview of the graduate admissions process.

UTCS, like many other departments, accepts applications for its graduate programs exclusively through an online system. Students fill out a series of forms with their educational history, test scores, research interests, and other information. Upon submitting the online application, a student's information is stored in a departmental database. References listed in the application are emailed and asked to submit letters of recommendation through the same online system.

When the time window for accepting applications has closed, faculty members use an internal web-based system to review the pool of applicants. After reading each file, a reviewer submits a real-valued score in the range 0–5 to rate the quality of the applicant and enters a text comment explaining his or her score to other reviewers. The time required for each full review varies with the reviewer's style and experience, the quality and content of the application, and the stage of the review process, but a typical full review takes 10–30 minutes. The committee typically performs multiple review passes over the pool and then admits or rejects each applicant based on the scores and comments of the reviewers who looked at his or her file. Although the primary criterion for this decision is quality, it is modulated to a significant degree by the current research opportunities in the department (that is, the number of new students the faculty request to be admitted in each research area).

Historically, the admissions committee has run a uniform review process, assigning a minimum of two reviewers to every file. Some candidates, particularly those that the committee was "on the fence" about, received up to five reviews. This process originates from the early 1990s, when the department received approximately 200 Ph.D. applications a year. Incoming applicant pools have grown almost threefold since that time and are expected to continue to grow in the future (figure 1). Thus, a uniform approach is no longer feasible: with the current applicant pool sizes, it would require about 1400 reviews and an estimated 700 hours of faculty time. This volume of applicants presents a significant challenge to the admissions committee, which has a strong incentive to admit the best possible students but also has a limited amount of time and resources.

In 2013, UTCS introduced a new, more efficient review process using GRADE to scale admissions to large applicant pools without sacrificing the quality of reviews. Instead of performing multiple full reviews on every file, GRADE focuses the committee's attention to the files of borderline applicants, where it is needed most. The GRADE system and the new review process are described in detail in the following section.

Method

GRADE consists of five main components. First, the system reads applicants files' from the departmental database and performs preprocessing to standardize the data. Second, the files are encoded as high-dimensional feature vectors. Third, a logistic regression classifier is trained on the feature-encoded historical data. Fourth, the classifier predicts the probability that each new applicant will be admitted and generates information to report to the admissions committee. Fifth, this information is used to decide which files should be reviewed fully and which can be checked quickly to verify the model's predictions. These steps are described in detail in the following subsections.

Preprocessing

Most applicant data is stored within the UTCS database in structured formats that are straightforward for the GRADE system to interpret. However, two key pieces of information — namely, the names of universities previously attended and grade point averages — are stored in unstructured string formats. These fields must be preprocessed to make their values interpretable by the statistical model. Rather than editing the data manually, automated techniques that require little or no human intervention are used.

The first preprocessing task disambiguates the names of universities listed in applicants' educational histories. The challenge is that applicants use a wide variety of strings to refer to each institution. For example, some people might say they attended The University of Texas at Austin, while others would refer to the school as University of Texas, Austin or UT Austin. In addition to such standard variations, a number of universities have undergone official name changes and have different "old" and "new" names. For example, in the recent past University of Missouri, Rolla was renamed to Missouri University of Science and Technology. Applicants may refer to such schools by the old or new names, and both names should be recognized as the same institution.

To disambiguate institutions, GRADE uses a web search engine to map each name to the domain name of the school's website (for example, UT-Austin to utexas.edu), and then uses the domain name as the identifier in all subsequent modeling steps. To locate a school's web address, the system searches for the institution name with the Bing Web Search API and looks for an educational domain name (*.edu, *.ac.in, and others) in the search results. If none is found but there is a Wikipedia article in the results, the system looks for the school's web address in the article's content. Failing that, the institution name is mapped to a special identifier for an unknown school. This approach works well and identifies all but a small handful of institutions in the applicant database.

A second preprocessing step is necessary to disambiguate grade point averages in some historical data.

In years past, GPAs were reported in unvalidated text fields in the online application. While many applicants explicitly stated the grading scale of their institution (for example, "3.8 / 4.0," "14.4 / 15"), others reported only a single, unscaled number ("14.4," "75"). In order to compare the grades of all applicants, GRADE must determine the GPA scales for this latter group. For each unscaled GPA encountered, the system first looks for another student from the same institution who reported the school's grading scale. Approximately 64 percent of ambiguous GPAs can be resolved in this manner. In the remaining cases, the system assumes the scale is the one that provides the closest upper bound to the reported GPA. For example, "14.4" is assumed to mean "14.4 / 15" rather than "14.4 / 100." Note that this preprocessing step does not need to be performed on new and recent data because the online application has been updated to have separate input fields for GPA and GPA scale.

Feature Encoding

GRADE encodes each application file as a high-dimensional feature vector. The overall approach is to generate a large number of features that represent a wide variety of information about each file. Although some features may turn out to be poor predictors for whether or not an applicant is admitted, there is little harm in including them in the representation because the L_1-regularized logistic regression classifier tends to assign zero weight to such features. This is discussed further in the Classifier subsection.

Applicant data comes in the following three forms: numerical data, namely, test scores and GPAs; categorical data taking on a discrete set of values, such as the student's undergraduate institution, research area, and preferred faculty advisor; and text data, namely, letters of recommendation and the applicant's statement of purpose. Each type of data requires a separate type of feature encoding. Each encoding scheme is detailed in turn in the following section. Except where otherwise indicated, all data are encoded as binary vectors. For each piece of information, an extra bit is reserved at the end of the feature vector to denote when the value was not reported or not applicable. The applicant's file as a whole is represented by concatenating the feature encodings for each constituent part.

For the remainder of the article, we refer to the data in an applicant's file as his or her attributes. This is to distinguish the data itself from the features that encode that information in the classifier. In general, a single attribute may be represented by more than one feature.

Numerical Attributes

The numerical attributes in the application file include scores from the GRE General test (in the quantitative, verbal, and analytical sections), the GRE Computer Science subject test score, the student's GPA at his or her last undergraduate institution, and, when applicable, the GPA at the last graduate institution.[1] All test scores and GPAs are converted to percentile ranks within the training set.

Instead of using the raw numerical values as features, the values are quantized coarsely and encoded as binary strings. The undergraduate and graduate GPAs are represented with the following indicator features:

Undergraduate GPA percentile < {20, 40, 60, 80},
≥ {20, 40, 60, 80}
Graduate GPA percentile < {20, 40, 60, 80},
≥ {20, 40, 60, 80}.

Note that these features are not mutually exclusive; for example, a 42nd percentile undergraduate GPA would be encoded as (0011 1100 0). Likewise, the GRE scores are encoded as binary strings as follows:

GRE quantitative percentile < {70, 80, 90},
≥ {70, 80, 90}
GRE subject percentile < 80, ≥ 80
GRE writing percentile < 50, ≥ 50
GRE verbal percentile < 80, ≥ 80

Compared to using the raw numerical value, this encoding has the benefit of introducing nonlinearities into the feature representation, enabling the classifier to model complex trends in the underlying value. For example, in practice, the classifier learns that higher GRE quantitative scores make an applicant more likely to be admitted, but that progressively higher scores have lower marginal benefit. In other words, it matters more that the score is "good enough" than that it is the absolute highest among all applicants.

Categorical Attributes

Categorical attributes are encoded in two different ways depending on the number of possible values the attribute can take. Those with a small number of K possible values (roughly, $K < 20$) are encoded as sparse 1 of K binary strings (a K-length string with a single 1 denoting the selected value). Attributes of this type include the student's highest degree attained (bachelor's, master's, Ph.D.), residency status, and research area of interest (11 possible subfields).

The remaining categorical attributes take on a larger number of possible values. This group includes the applicant's undergraduate and/or graduate institutions (714 unique values in 2009–2012) and the preferred faculty advisor (44 unique values). For these data, the 1 of K representation is impractical, as most features would be used too infrequently in the training set to make any reliable inference about their effect. Instead, we generate a single numerical feature for each containing the historical log odds of being admitted given the attribute value. The log odds representation is appropriate because the logistic regression classifier used by GRADE (see the Classifier subsection) can be thought of as a linear model that predicts the log odds of admission. In practice, the log odds features are among the most important fea-

tures in the model and are significant improvements over 1 of K representations for the past institutions and preferred faculty advisors.

Log odds are estimated in a Bayesian manner with a simple beta-binomial model. For an attribute a taking on values $v \in V$, let $n^+_{a=v}$ and $n^-_{a=v}$ be the number of applicants with $a = v$ that have historically been admitted and rejected, respectively, and let $p_{a=v}$ denote the (unobserved) true proportion of applicants admitted. Assuming a prior $p_{a=v} \sim \text{Beta}(\alpha_a, \beta_a)$, the proportion has posterior

$$p_{a=v} \mid n^+_{a=v}, n^-_{a=v} \sim \text{Beta}(\alpha_a + n^+_{a=v}, \beta_a + n^-_{a=v}).$$

$$\mathbb{E}[p_{a=v}] = \frac{\alpha_a + n^+_{a=v}}{\alpha_a + n^+_{a=v} + \beta_a + n^-_{a=v}},$$

$$\mathbb{E}\left[\log \frac{p_{a=v}}{1 - p_{a=v}}\right] = \psi(\alpha_a + n^+_{a=v}) - \psi(\beta_a + n^-_{a=v}),$$

where $\psi(\cdot)$ denotes the digamma function, the derivative of the log of the gamma function. Note that for attribute values that do not appear in the historical data, the estimate (3) reduces to the log odds of the prior, while for values used frequently, the estimate approaches the empirical log odds. (For large arguments, $\psi(x) \approx \log x$.) The parameters α_a, and β_a are set to induce a weak prior with the mean set to the admission probability given any rare value. For example, for the undergraduate and graduate institutions, α_a, and β_a are set so that the prior mean is the empirical proportion of applicants admitted from institutions with two or fewer total applicants.

School Reputation

For the undergraduate and graduate institutions, the log odds features (described above) serve as a data-driven estimate of each school's reputation or quality. These features are very useful in practice: unlike the human committee members that GRADE aspires to model, the system has no external knowledge with which to judge the quality of institutions.

To represent school quality more explicitly, the system is also provided with reputation features that describe the applicant's last undergraduate and (when applicable) graduate institution. Reputation is represented as a binary string of length three, with separate bits encoding whether the school is considered elite, good, or other. These categories were created by surveying UTCS faculty who were familiar with the universities in various parts of the world. The elite and good categories are defined by explicit lists; any school not in the first two categories is considered other.

Combination Features

In addition to the features listed above, we generate combination features that allow interaction effects between attributes to be modeled. For two attributes x and y that are encoded as m- and n-length binary strings, respectively, the encoding of combination $x \times y$ is an $(m \cdot n)$-length binary string, where each bit corresponds to a pair of bits from x and y. The classifier is provided with the following combination features:

GRE Quantitative × GRE Verbal × Research Area
GRE Quantitative × GRE Verbal × Undergraduate Reputation
Undergraduate Reputation × Research Area
Undergraduate GPA × Undergraduate Reputation

Text

Each applicant's file contains two forms of text data: a statement of purpose, in which the student describes his or her background and research interests, and three or more letters of recommendation. All documents are submitted through the online application system in PDF format.

To generate features for the letters of recommendation, we first extract the embedded text from each PDF file, stem the words with the Porter stemmer (Van Rijsbergen, Robertson, and Porter 1980), and apply simple stop-word filtering. The letters are then combined into a single bag of words per applicant. Finally, latent semantic analysis (LSA) (Deerwester et al. 1990) is applied to project each set of letters into a 50-dimensional feature vector.

We ran a number of experiments using the same procedure to generate features for the applicants' statements of purpose. However, in practice, these features were not discriminative and did not improve the quality of GRADE's predictions. As a result, we omit the statements of purpose and use only the letters of recommendation.

Information Not Represented

It is important to note that some valuable information in an applicant's file is not represented in the current feature encoding. Most importantly, there are currently no features describing the applicant's publications or any fellowships or awards that he or she may have been received. In addition, the system has no features representing the identities, titles, or scholarly reputations of the recommendation letter authors. Finally, there are no features for the student's undergraduate area of study. (Although most applicants have a scientific background in computer science or a related field, this is not uniformly true.) We would like to use this information in the system but currently have no reliable, automated method of extracting it from the application files. Instead, reviewers are informed that this information is missing from the model and are instructed to pay particular attention to these factors when reviewing.

Classifier

GRADE's main task is to use historical data to infer how likely new applicants are to be admitted to the

graduate program. This problem can be framed as one of probabilistic binary classification. The historical admissions decisions take the form of labeled training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector encoding of the applicant's file, and $y_i \in \{+1, -1\}$ indicates whether the applicant was admitted or rejected. Given the past decisions and new applicants $\{\mathbf{x}_i^{\text{new}}\}_{i=1}^{N_{\text{test}}}$, the goal is to predict $p(y_i = +1 | \mathbf{x}_i^{\text{new}})$, the probability that the admissions committee will accept each applicant $\mathbf{x}_i^{\text{new}}$.

GRADE models the problem using L_1-regularized logistic regression. Under this model, the estimated probability of an applicant being admitted takes the parametric form

$$p(y_i = +1 \mid \mathbf{x}_i) = \frac{1}{1 + \exp\{-\mathbf{w}^\top \mathbf{x}_i\}},$$

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^d} \hat{\mathcal{L}}(\mathbf{X}, \mathbf{y})$$

$$\hat{\mathcal{L}}(\mathbf{X}, \mathbf{y}) = \sum_{i=1}^{N} \ell(\mathbf{x}_i, y_i, \mathbf{w}) + \lambda \| \mathbf{w} \|_1 .$$

The first term of the objective (6) measures how well the probability estimates of the classifier fit the admissions decisions in the training data. Here,

$$S(a_{ij}) = p(y_i = +1 \mid f(\mathbf{a}_i)) - \sum_{k=1}^{N_{\text{test}}} \frac{p(y_i = +1 \mid f(\mathbf{a}_i^{jk}))}{N_{\text{test}}}.$$

is log loss, or prediction error, on the ith training example. The second term of (6) is the L_1 norm of the weight vector times a scalar parameter $\lambda \geq 0$. This regularization term serves to pressure the classifier away from "complex" models and toward ones that use small or zero weights for some features. The λ parameter controls the trade-off between the fit of the model to the training data and the complexity of the resulting weight vector. In GRADE, the value of λ is selected using 10-fold cross-validation.

In practice, the L_1 regularization in logistic regression acts as a robust feature-selection mechanism. In general, the learned weight vector is sparse (that is, some feature weights w_j^* are set to zero) with larger λ values leading to greater sparsity (Koh, Kim, and Boyd 2007). Zero weights correspond to features in the input with low predictive power, so that in effect the model uses only a subset of the most discriminative features. This feature-selection mechanism works well even when the number of nonpredictive features in the input is very large. This is because the sample complexity of the learner — the number of training examples required to learn a near-optimal logistic regression classifier — grows only logarithmically in the number of nonpredictive features (Ng 2004). In other words, there is little harm in giving extra features to the L_1-regularized logistic regression that turn out not to be of use. In contrast, other common classification methods have sample complexities that grow at least linearly in the number of nonpredictive features, and are expected to perform poorly in such settings.

L_1 regularization has a significant impact in GRADE: the trained classifier places nonzero weight on only 58 of the 427 dimensions of the feature space. Such sparsity has two important benefits to our system. First, it reduces the classifier's tendency to overfit the training data and improves the quality of predictions on new applicants. Second, because the learned model is parsimonious, it is easier to interpret. One can quickly examine the nonzero elements of the weight vector to see what information the classifier uses to make its predictions. This is important as it makes it possible to check that the model is reasonable before using it to score a set of applicants. Interpreting the model is also interesting in its own right, giving insight into how committee members evaluate applicants.

In addition to logistic regression, we experimented with a number of other learning methods, including multilayer perceptrons and support vector machines (SVMs) with both linear and nonlinear kernels. Multilayer perceptrons performed worse than logistic regression and tended to overfit the training set, even with a small number of hidden units and the use of early stopping. SVM performed nearly as well as logistic regression. However, SVM's feature weights are in general dense, making the model difficult to interpret. Overall, we chose to use L_1-regularized logistic regression due to its ability to produce sparse, interpretable models that perform well in this domain. It is possible that variants of the above methods that incorporate more sophisticated regularization, such as sparse SVM (Tan, Wang, and Tsang 2010), may perform better.

Model Output

After training the logistic regression classifier on historical data, GRADE is run on new applicants. First, the system predicts the probability that each new applicant will be admitted. Second, these quantities are mapped to an estimated reviewer score for each file. Then, by performing sensitivity analysis on the classifier, the system identifies the attributes in each file that might stand out as being particularly strong or weak to a human reviewer.

Score

To estimate reviewer score, the probability output by the classifier is mapped to a numerical score between 0 and 5. Human reviewers use scores to measure a candidate's quality in an absolute sense, not merely relative to the current set of applicants; for example, a score of 5 is exceptionally rare and is only given to

Percentile	100	96.8	78.8	49.5	28.4	0
Score	5.0	4.5	4.0	3.5	3.0	1.0

Table 1. Scoring Guidelines

the very best applicants the department has ever received. For this reason, GRADE computes score as a function of the candidate's percentile rank within a global pool consisting of all past and present applicants. The percentile ranks are converted to scores using the guidelines in table 1.

The system uses linear interpolation to achieve scores of finer granularity so that, for example, a 98.4th percentile applicant is given a score of 4.75. These guidelines are derived from historical data. For consistency, they are included in the instructions given to human reviewers.

Strongest/Weakest Attributes

Human committee members typically accompany their numerical scores with a text comment that notes any particularly strong or weak parts of the applicant's file. For example, a reviewer might note, "GRE scores are very impressive" or "letters of recommendation are weak." These comments serve to justify the reviewer's score and point out salient pieces of information to other reviewers. In GRADE, a variant of sensitivity analysis (Saltelli, Chan, and Scott 2000) is used to automatically generate a description of each applicant's strengths and weaknesses. This information is reported in a text comment that appears with the numerical score.

For a given applicant, a strong attribute is defined as one that makes him or her significantly more likely to be admitted when compared to the average value held by others. More formally, attribute strength is quantified as follows. Let $\mathbf{a} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{N\text{test}}\}$ denote the attributes for each new applicant. Let $\mathbf{a}^{ik}_i$ denote the attributes of the ith applicant with the jth entry replaced with the corresponding value from applicant k. Finally, let f denote the feature function that maps attributes into feature vectors, so that $f(\mathbf{a}_i) = x_i^{\text{new}}$. Then, the strength of attribute a_{ij} is measured as

$$\ell(\mathbf{x}_i, y_i, \mathbf{w}) = \log(1 + e^{-y_i \mathbf{w}^\top \mathbf{x}_i})$$

The first term is the admission probability assigned by the classifier to the applicant's original file, while the sum in the second term is the classifier's average response when a_{ij} is replaced with values of other applicants.

For each applicant, GRADE computes the strengths of all attributes and outputs a string listing of the top two strongest and weakest attributes along with their effect strengths.

New Admissions Process

The new review process, deployed in the 2013 Ph.D. admission season, uses GRADE to focus the work of the admissions committee. The system was first trained with the data from the past four years (2009–2012, 1467 applications in total) and was then used to evaluate each file in the new pool of 588 applications. As described above, GRADE predicted the probability that the committee would admit the applicant and estimated the numerical score that would be given by human reviewers. In addition, the system generated a human-readable list of perceived strengths and weaknesses of each file. This information was uploaded to the online review system where it was visible to the reviewers.

The files were then assigned to the reviewers based on GRADE's predictions. For each research area, the applicants were ordered according to the model's score. Those with score above 4.4 (27 applications) were identified as likely admits, those between 4.4 and 3.83 (148) as possible admits, and those at or below 3.83 (413) as likely rejects. Each of the nine committee members thus received a list of approximately 3 top applications, 16 midrange applications, and 46 bottom applications.

Importantly, committee members were only required to perform a full review on the midrange applications. For the top- and bottom-ranked files, the reviewers were instructed to simply check the score of the statistical model. A check included evaluating possibly useful information not available to the statistical model, including awards, publications, unusual vita items, reputations of letter writers, unusual home institutions, and letters whose text could not be automatically extracted. Such a check could be done quickly in most cases, and if the reviewer agreed with the model's score, he or she did not need to enter a score. However, in cases where the reviewer disagreed with the model, he or she was asked to do a full review and enter a score. In some cases, such as when the reviewer was evaluating his or her own student or an applicant outside of the reviewer's expertise, the reviewer asked other faculty to do a review instead, or in addition.

The applications were then resorted using the human scores if there were any, and the model's score if not. An initial admit/reject decision boundary was identified for each area, based on the need for new students in each area (as determined by surveying the faculty). A second round of reviews was assigned for applications that were close to the decision boundary, as well as for applications that were somewhat further away from it where a reviewer disagreed significantly with the model (that is, more than 0.2 points in score). A total of 103 applications were thus assigned to the second round, or an average of about 11 per reviewer.

Based on the results of the second round, the applications were once again resorted and the deci-

sion boundaries for each area identified. Finally, in a meeting of the entire committee, these orderings and boundaries were discussed and slightly adjusted to make the final decisions.

Evaluation

In 2012, the newly developed GRADE system was tested alongside the regular Ph.D. application review process. Committee members did not see the system's output when performing their reviews, and the final decisions were made based only on human scores. However, in the first pass over the applicant pool, all files were only given one full review instead of two as in previous years. If both GRADE and the first reviewer agreed that the applicant should be rejected, the file was not reviewed further. In all other cases, the review process proceeded in the usual manner. Although the primary goal was to evaluate how reliable GRADE's predictions were, this process reduced the number of required human reviews by 24 percent. Indeed, although UTCS received 17 more Ph.D. applicants in 2012 than 2011, fewer total reviews were actually required.

The remainder of this section focuses on the role of GRADE in the 2013 season, when the system was fully deployed into the review process. GRADE is evaluated in two ways: by measuring the quality of its predictions, and by estimating the amount of faculty time it saved.

Classification Performance

Of 588 Ph.D. applicants in 2013, the admissions committee admitted 92 (15.6 percent) and rejected 546 (84.4 percent). GRADE predicted the correct admissions decision with 87.1 percent accuracy. Note that this result alone is not very meaningful: due to the imbalance of the classes in the test set, one could achieve 84.4 percent accuracy just by rejecting all applicants. Rather than looking at accuracy, the system is best understood by its precision-recall characteristics.

Figure 2a shows the precision-recall curve for GRADE's 2013 predictions. In terms of graduate admissions, precision and recall measure the system's ability to identify applicants who will be admitted. As the top left corner of figure 2a shows, GRADE is very good at finding a small number of high-quality applicants that the committee will admit. However, precision quickly drops at larger recalls. This reflects that there are many mid-scoring applicants for which the committee's decision is difficult to predict.

In practice, GRADE is much better at identifying applicants that the committee will reject. Figure 2b shows the classifier's true negative rate (specificity) versus false negative rate; these are the fractions of applicants that the system would correctly and incorrectly reject at varying decision thresholds. The data indicate that GRADE can identify a large proportion of the applicants who will be rejected while maintaining a very low false negative rate. As detailed in the following section, the system gives many of these applicants scores that are sufficiently low that they need only be checked once by a human reviewer.

Time Savings

As we previously mentioned, integrating GRADE into the admissions process saves reviewers time in two ways. Primarily, GRADE allows the committee to quickly identify many applicants who will likely be rejected, as well as a smaller number of applicants who will likely be admitted. Secondarily, GRADE makes the reviewing itself more efficient by ordering the files and by providing information about the strong and weak attributes of each applicant.

Figure 3 shows the number of reviews performed in 2011 and 2013 broken down by mean reviewer score. In 2011, the committee (without GRADE) required 1125 total reviews to evaluate 545 files, or an average of 2.06 per applicant. Using GRADE in 2013, only 362 full reviews were required to evaluate 588 files. (In addition, 150 extra reviews were performed by committee members who did not follow the chair's instructions; they were unnecessary and did not affect the outcome.) Altogether, only 0.616 full reviews were needed per applicant, constituting a 71 percent reduction from 2011. The primary reason for this drop is that a majority of the files (362) could be evaluated with only a "check"; that is, both GRADE and a human reviewer strongly agreed that the applicant should be admitted or rejected. Note, however, that the committee still performed many full reviews on the mid-to-high-scoring echelons of the applicant pool.

Committee members reported that the time required to perform a full review ranged within 10–30 minutes, with an average of about 20 minutes per file. This time is estimated to be approximately 25 percent lower than in previous years due to the information and initial ranking provided by GRADE. Finally, the time required to check GRADE's prediction on a file was measured to be 1/5 of the time of a full review for reviewers well familiar with the model's outputs. From these numbers, the entire review process in 2013 was estimated to require 145 hours of reviewer time, while the traditional review process would have needed about 549 hours — a reduction of 74 percent.

Score Agreement

Although GRADE is not trained to predict reviewer scores directly, the utility of the system does depend on how well its scores match those given by actual reviewers. Figure 4 shows the human and model scores given to applicants in the 2013 admissions cycle. In cases where the reviewer agreed with the model during a "check," the reviewer's score is considered to be equal to the model's.

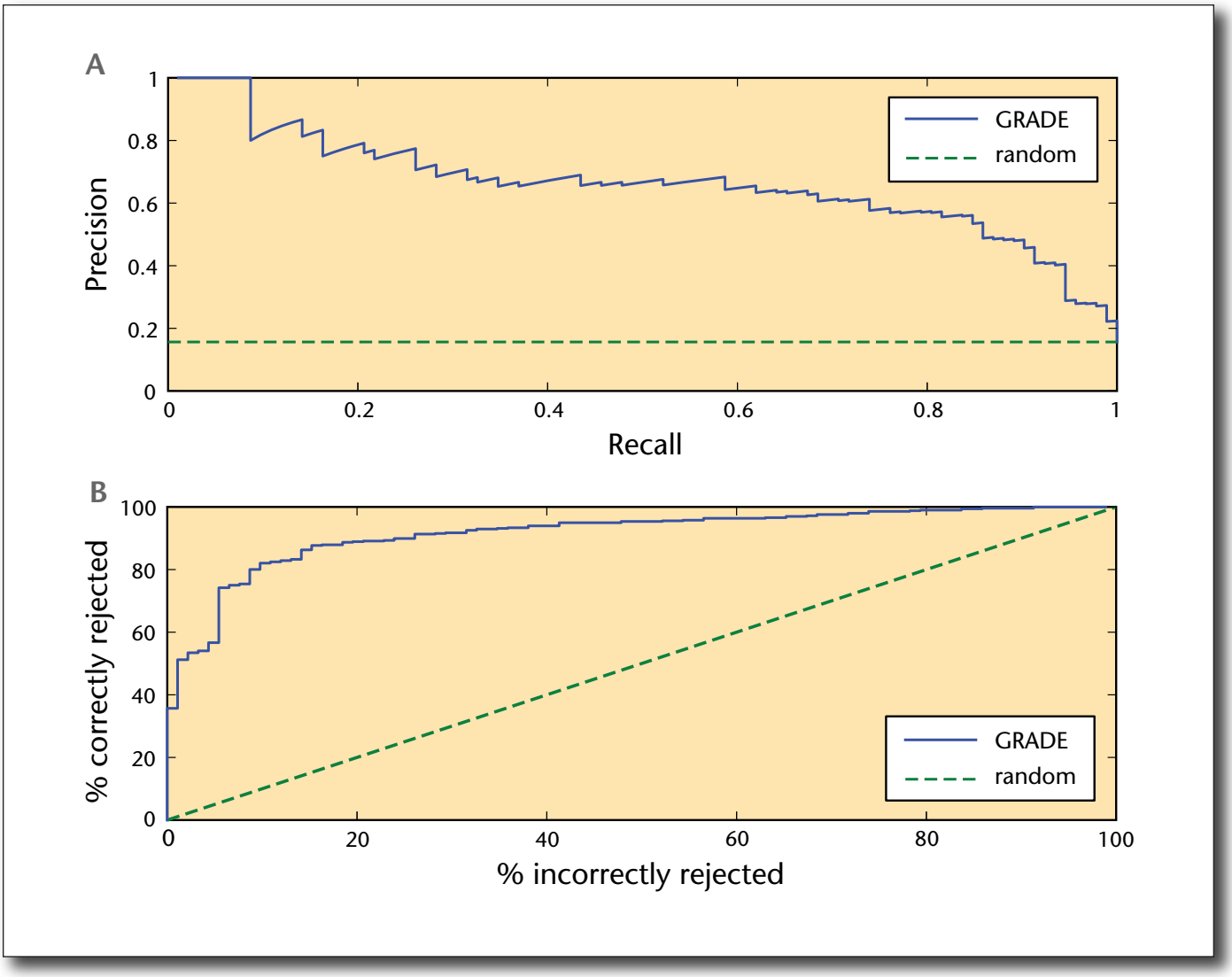

Figure 2. Classification Performance of GRADE in 2013.

In (a), precision and recall measure the system's ability to identify applicants who were admitted, while (b) shows how well it identifies rejected applicants. The latter shows the fraction of applicants who would be correctly and incorrectly rejected at different decision boundaries (that is, the true negative rate versus (1 – recall)). In each plot, "random" shows the performance of a classifier that admits a randomly selected subset of applicants. GRADE identifies ≈ 10 percent of admitted applicants with perfect precision and ≈ 50 percent of rejected applicants with nearly no false rejections.

The results show that reviewers generally agree with GRADE's score a majority of the time. GRADE's score was within 0.2 of the human score on 40 percent of files, while humans agreed with each other at this level 50.3 percent of the time. In cases of disagreement, the system tends to underestimate the applicant's score. The largest deviations occur on files that GRADE considers to be of very low quality (with scores between 1 and 2) that reviewers rate as midrange (with scores between 2 and 4). These cases may be due to the model lacking information that is available to human reviewers: for example, a student may have publications that are not represented in the system.

Learned Feature Weights

The feature weights learned by the logistic regression classifier indicate what information GRADE uses to make its predictions. Out of 427 dimensions of the feature space, the classifier assigned zero weight to all but 58. Upon inspection, the features used by the classifier correspond to information that human reviewers also use. The following are some of the most predictive features in the model:

Undergraduate GPA. Higher values are better, and a B+ average or lower makes a student very unlikely to be admitted. (Graduate GPA, however, receives zero weight.)

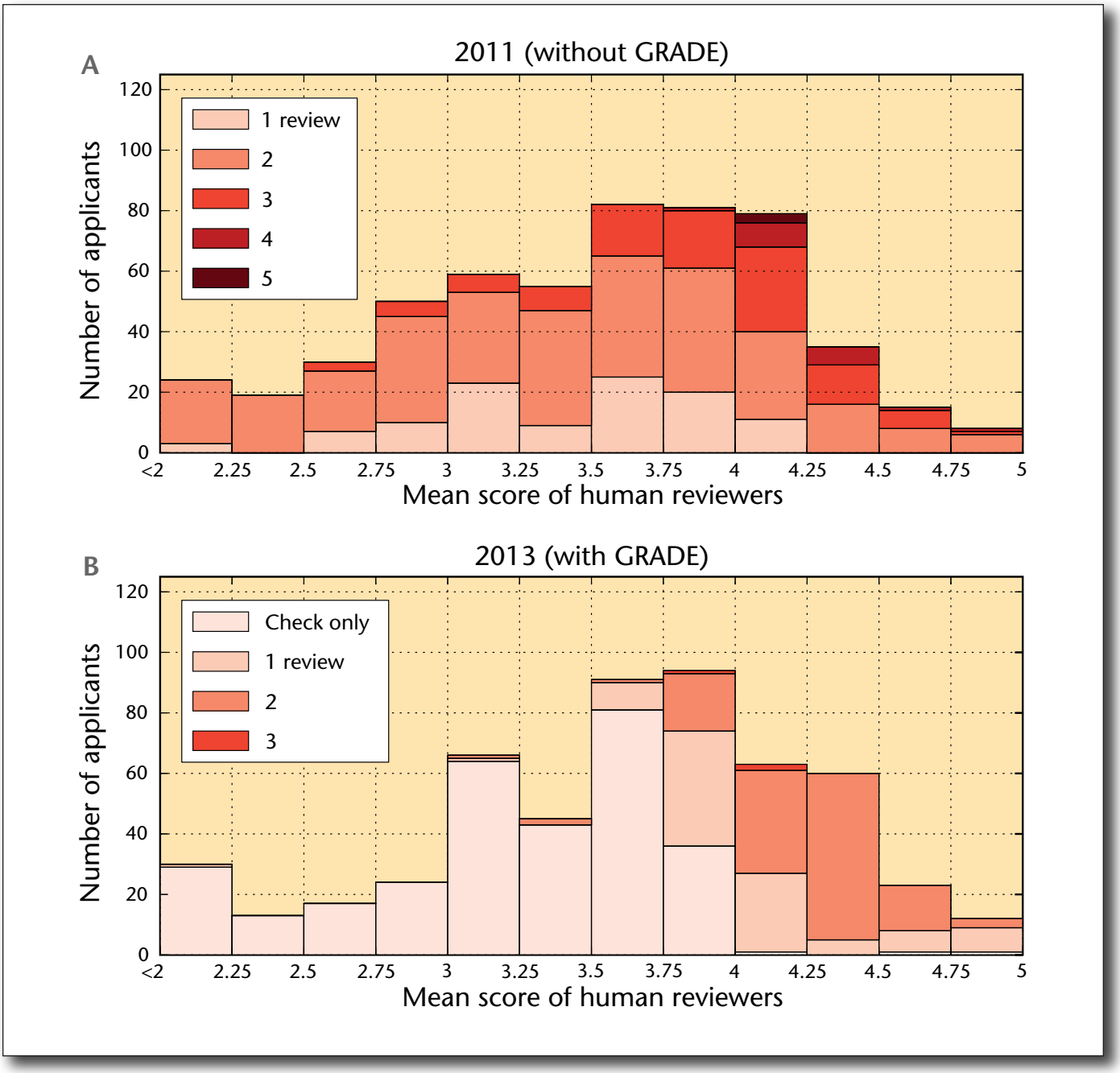

Figure 3. Number and Type of Reviews Needed to Evaluate New Ph.D. Applicants.

(a) with the uniform process of 2011, and (b) assisted by GRADE in 2013. In (b), note the large number of low-scoring applicants whose files were only "checked." These are students who were rejected after the first human reviewer agreed with GRADE that the applicant was not competitive. As a result of the system, fewer reviews were required, and the committee had more time to consider mid-to-high-scoring applicants.

Institutions previously attended. The log odds of the applicant's last school is one of the model's highest-weighted features. Students with very high undergraduate GPAs at elite universities also receive a boost from the model.

GRE quantitative score. Higher scores are better; < 80 percentile makes a student very unlikely to be admitted.

GRE verbal score. A high verbal score (≥ 80 percentile) is better, but only if the applicant also has a high quantitative score. (Otherwise, the verbal score appears to be irrelevant.)

Research area. Students applying to study machine learning or artificial intelligence/robotics are less likely to be admitted. This is because UTCS receives many well-qualified applicants in these areas.

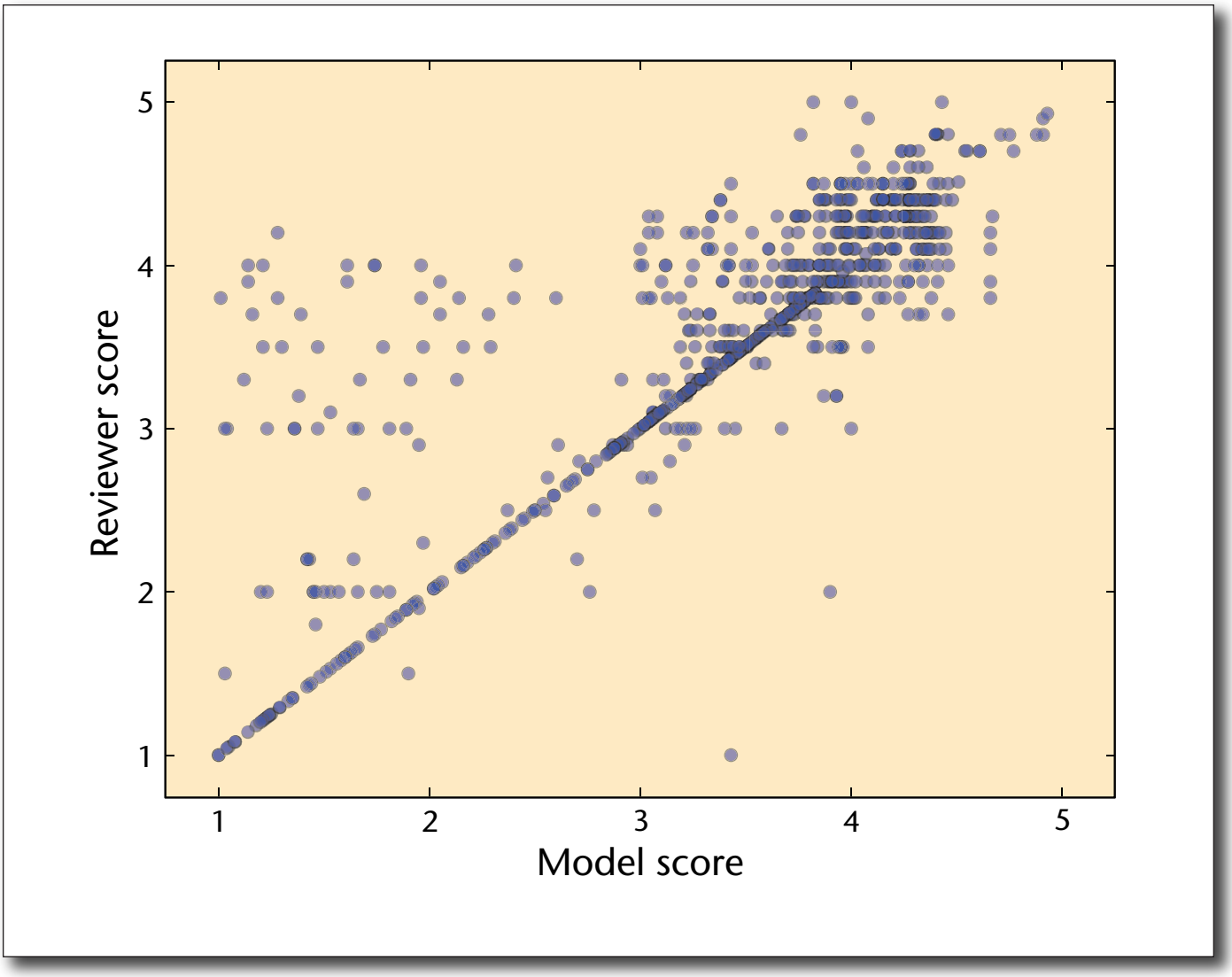

Figure 4. Scores Given to 2013 Ph.D. Applicants by Human Reviewers Versus Those Predicted by GRADE.

Entries on the diagonal indicate cases where the reviewer agreed with the model's assessment during a check. The reviewer and model scores are often close. GRADE sometimes underestimates the scores of applicants it considers to be of very low quality (with a predicted score of 1–2). However, in many of these cases, the system still correctly predicts that these applicants will be rejected.

Letters of recommendation (LSA features). Experiments indicate that letters containing terms such as "best," "award," "research," "Ph.D.," and others are predictive of admission, while letters containing "good," "class," "programming," "technology," and others. are predictive of rejection. In part, this pattern reflects the faculty's preference for students with strong research potential over technical ability. In contrast, as mentioned above, the statement of purpose was deemed nonpredictive by the model.

Another interesting finding is that the applicant's gender, ethnicity, and national origin receive zero weight when provided as features to the model. This result indicates that UTCS admissions decisions are based on academic merit.

Discussion

Results from the 2013 admissions season demonstrate that GRADE can predict which students will be admitted to or rejected from the UTCS Ph.D. program with reasonable accuracy and that the admission probabilities can be used to estimate of the scores of human reviewers. Most importantly, GRADE's predictions can be used to make the review process much more efficient, while still allowing human reviewers to be in charge of all admissions decisions.

Although GRADE works well as it is, its performance is likely to improve over time. One reason is that the amount of labeled data available to train the system will grow every year as the admissions com-

mittee evaluates new pools of Ph.D. applicants. Future versions of the system may also be able to utilize information on applicants' publications, awards, and fellowships, which would likely improve the quality of predictions. Finally, other gains may be had by using more sophisticated techniques in some modeling steps, for example by using probabilistic topic models instead of LSA to analyze recommendation letters.

As GRADE improves, its role in the review process may be expanded commensurately; for example, future committees should be able to assign a larger proportion of files to be checked instead of given full reviews. However, because the review process is inherently noisy, neither GRADE nor any other system will ever be able to perfectly predict the decisions of a human committee. Thus, GRADE's role supporting human reviewers is appropriate for the foreseeable future.

Development and Maintenance

GRADE was developed by Austin Waters with technical assistance from UTCS staff over the 2011–2012 academic year. Financial support was provided in the form of a two-semester graduate research assistantship. The system can be maintained and extended by a UTCS graduate student with a part-time appointment. Operationally, the system requires minimal human interaction to run, and can be used by UTCS staff as part of their regular duties.

GRADE has minimal software and hardware dependencies. It is implemented in Python with the pandas and scikit-learn packages, which are open source and freely available. The system runs on ordinary desktop hardware, requiring only a single CPU and ≈ 200 MB of memory and disk space.

Conclusion

This article describes GRADE, a system that uses statistical machine learning to scale graduate admissions to large applicant pools where a traditional review process would be infeasible. GRADE allows reviewers to identify very high- and low-scoring applicants quickly and reduces the amount of time required to review each application. While all admission decisions are ultimately made by human reviewers, GRADE reduces the total time spent reviewing files by at least 74 percent compared to a traditional review process and makes it possible to complete reviews with limited resources without sacrificing quality.

Acknowledgements

Thanks to John Chambers and Patti Spencer for their technical assistance, Bruce Porter for financial support, and Peter Stone, Raymond Mooney, and the admissions committees of 2012 and 2013 for their patience and valued feedback in the development of this work. This research was supported in part by the Department of Computer Science at the University of Texas at Austin, and in part by NSF under grant IIS-0915038.

Notes

1. Some international applicants to UTCS also submit TOEFL scores. However, because the paper- and Internet-based tests have different grading scales, and TOEFL is not required of many applicants, it was left out of the model altogether.

References

Bruggink, T. H., and Gambhir, V. 1996. Statistical Models for College Admission and Enrollment: A Case Study for a Selective Liberal Arts College. *Research in Higher Education* 37(2): 221–240. dx.doi.org/10.1007/BF01730116

Deerwester, S.; Dumais, S.; Furnas, G.; Landauer, T.; and Harshman, R. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41(6):391–407. dx.doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

Koh, K.; Kim, S.; and Boyd, S. 2007. An Interior-Point Method for Large-Scale L1-Regularized Logistic Regression. *Journal of Machine Learning Research* 8(8):1519–1555.

Moore, J. S. 1998. An Expert System Approach to Graduate School Admission Decisions and Academic Performance Prediction. *Omega* 26(5): 659–670. dx.doi.org/10.1016/S0305-0483(98)00008-5

Ng, A. Y. 2004. Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance. In *Proceedings of the 21st International Conference on Machine Learning*. Princeton, NJ: International Machine Learning Society, Inc.

Saltelli, A.; Chan, K.; Scott, E. 2000. *Sensitivity Analysis: Gauging the Worth of Scientific Models*. New York: Wiley.

Tan, M.; Wang, L.; and Tsang, I. 2010. Learning Sparse SVM for Feature Selection on Very High Dimensional Datasets. In *Proceedings of the 27th International Conference on Machine Learning*. Princeton, NJ: International Machine Learning Society, Inc.

Van Rijsbergen, C.; Robertson, S.; and Porter, M. 1980. New Models in Probabilistic Information Retrieval. British Library Research and Development Report 5587. London: British Library.

Austin Waters is a Ph.D. candidate in the Department of Computer Science at the University of Texas at Austin. He received an M.S. in Computer Science from the University of Texas at Austin in 2008. His research is in statistical machine learning, with an emphasis on topic models, Bayesian nonparametrics, and methods for approximate inference.

Risto Miikkulainen is a professor of computer sciences at the University of Texas at Austin. He received an M.S. in engineering from the Helsinki University of Technology, Finland, in 1986, and a Ph.D. in computer science from UCLA in 1990. His current research focuses on methods and applications of neuroevolution, as well as models of natural language processing and self-organization of the visual cortex; he is an author of more than 250 articles in these research areas. He is currently on the Board of Governors of the Neural Network Society and an associate editor of *IEEE Transactions on Computational Intelligence, AI in Games,* and *IEEE Transactions on Autonomous Mental Development*.