

A Review of Bessen and Hunt's Analysis of Software Patents

Robert W. Hahn and Scott Wallsten*

AEI-Brookings Joint Center
1150 17th St., NW
Washington, DC 20036

November 2003

* Robert Hahn is the executive director of the AEI-Brookings Joint Center for Regulatory Studies and a resident scholar at the American Enterprise Institute. Scott Wallsten is a fellow at the Joint Center and a resident scholar at the American Enterprise Institute. We would like to thank Anne Layne-Farrar and Bernard Reddy for helpful comments and Irina Danilkina for research assistance. The authors gratefully acknowledge the support of Microsoft. The opinions in this paper reflect those of the authors and do not necessarily reflect those of the institutions with which they are affiliated.

© 2003 by the authors. All rights reserved.

Executive Summary

This note analyzes a paper that has been used to support opposition to the European Proposed Directive on software patentability. The paper, by James Bessen and Robert Hunt analyzes software patents and research and development (R&D). They conclude that software patents have become “cheaper” than other patents and that software patents substitute for R&D spending. The authors argue that firms primarily use software patents strategically.

The data, analytical methods, and empirical results, however, do not support these conclusions. In this note we explain in detail the problems in the paper and how those problems prevent one from drawing any firm conclusions. Our critique is based on a concern that this unpublished manuscript will unduly influence policy decisions and is not intended to diminish the importance of the questions the authors address. We therefore conclude the note with recommendations for future research.

Here are some of the problems we identify:

Data

- Bessen and Hunt (“B&H”) use an extremely broad definition of “software patent,” so it is not clear that their findings pertain to software, *per se*.
- B&H appear to match firms to R&D and financial data only if those firms existed in 1989. Thus, they exclude a large number of firms potentially important to the study.
- The authors use patent grants rather than applications, contrary to nearly all research. The application-grant delay means that they capture little information about firms’ patenting decisions after 1996, although that period is crucial for the analysis.

Are software patents “cheaper” than other patents?

- The regressions show a positive correlation between software and total patenting in the 1990s (and the reverse in the 1980s), not necessarily that software patents became “cheaper.”
- B&H do not evaluate the benefits of software patents. Their estimates suggest that software patents might have greater social benefits than other patents.
- The authors weight observations using a transformation of the dependent variable. This unusual approach is likely to yield inconsistent coefficient estimates.
- The authors’ model relies on the unrealistic assumption that firm output is related to R&D but not to patents.
- The authors’ model is based on a firm’s *stock* of patents, but the empirical analysis relies on the annual *flow* of patent grants.

R&D and software patents: complements or substitutes?

- Patents, R&D, and sales affect each other, but the authors do not account for the endogenous nature of these relationships.
- The authors’ empirical specification makes it impossible to determine whether any coefficient estimate implies that R&D and patents are complements or substitutes.
- One empirical result in the paper reveals a statistically significant positive correlation between patents and R&D.

A Review of Bessen and Hunt's Analysis of Software Patents

Robert W. Hahn and Scott Wallsten

1. Introduction

Patents are intended to encourage research by granting an innovator a temporary monopoly on the returns to innovation, and can also contribute to knowledge spillovers and technology transfer by making certain details of innovations public. Firms patent for many reasons, however, including strategic ones. Under some conditions patent protection can stimulate innovation and under other conditions suppress or distort it (Gallini 2002, Jaffe 2000). The empirical literature tends to find a positive correlation between patents and R&D (e.g., Arora, et al. 2003, Hall, et al. 1986, Jaffe and Trajtenberg 2003).

Changes in patent policies can affect the extent to which patents encourage innovation and which innovations firms choose to patent. Such policy changes are often controversial as they affect not only research behavior, but also firms' value and thus the returns to their shareholders (Lerner 1994). It is therefore especially important when formulating patent policy to understand the potential effects of any changes.

Software patents have been particularly controversial. Proponents argue that software patents encourage innovation in a segment of the economy that has become increasingly important over the past decade. Opponents argue that firms use software patents mostly for strategic reasons, rather than to protect true innovation. In the current European debate regarding software patents, some economists have been vocal in opposing a Proposed Directive that some say would clarify and others say would ease applications for software patents in Europe.¹

As part of their argument, the economists opposed to the Proposed Directive cite findings in an unpublished paper by James Bessen and Robert Hunt ("B&H"). The

¹ See, in particular, "An Open Letter to the European Parliament Concerning the Proposed Directive on the Patentability of Computer-Implemented Inventions" (Andersen, et al. 2003). Note that while the letter has 14 signatories it is generally referred to as the "12 economists' letter."

paper, “An Empirical Look at Software Patents,”² asks the following two questions regarding software patents:

1. How “costly” are software patents for firms to obtain relative to other patents; and
2. What is the effect of software patents on R&D?

The authors conclude that software patents are “cheap” relative to non-software patents and that they substitute for R&D spending. Together, these two conclusions suggest to B&H that rather than stimulating innovation, software patents are primarily used by firms strategically to build “thickets” for anticompetitive reasons.

Unfortunately, B&H’s paper contains too many flaws to answer the questions of why firms patent software and how those patents affect R&D. We will show that there are serious problems with the data, analytical methods, and the analysis performed by B&H. Unpublished manuscripts typically do not warrant such scrutiny, but this paper is playing a large role in the debate over software patents, and a detailed analysis of it is therefore important.

Our review follows the presentation by B&H. Section 2 reviews key problems with the data, beginning with the definition of a software patent. Section 3 discusses the analysis and methods. Finally, section 4 discusses our conclusions and suggests areas for research.

2. Data

In this section we will identify key flaws in the ways B&H compiled their dataset. Here is a brief summary of our key points:

² James Bessen and Robert M. Hunt, “An Empirical Look At Software Patents,” (Research on Innovation) available from <http://www.researchoninnovation.org/swpat.pdf>; accessed October 28 2003.

- B&H use an extremely broad definition of “software patent,” provide no analytical justification of their definition, and ignore the IPC classification system, which other authors have used to construct more rigorous definitions of “software patents.”
- The authors match firms to their R&D and financial data only if those firms existed and were publicly-traded in 1989, assuming they used the NBER match file. As a result, their analysis excludes all firms that went public in the 1990s.
- B&H use patent grants rather than applications, contrary to nearly all literature. Grants follow applications by about two years, on average. This lag means that the analysis captures almost no information regarding patenting behavior after 1996, which is the period of greatest interest.
- The authors find that manufacturing firms receive most of the software patents, as well as most patents overall, but do not mention that they actually hold a *smaller* share of software patents than their share of total patents would suggest.

2.1 The definition of “software patent” is too broad

The term “software patent” is ambiguous, and Bessen and Hunt recognize that ambiguity. To address this problem, they define software patents as those that

- use the word “software” in the specification, or
- use the words “computer” and “program” in the specification, and
- do not use the words “semiconductor,” “chip,” “circuit,” “circuitry” or “bus” in the title. (B&H, p. 5)

This algorithm yields 134,690 software patents granted between 1976 and 1999 (B&H, p. 5).

B&H’s algorithm includes an extremely broad range of patents that may or may not actually be software. This algorithm includes, for example, patents entitled “genetic control of flowering” and “frozen food product.”³ Rather than explain why they believe their method to be correct, they justify it by comparing their results to two other sources. First, they compare their list to those of a consultant, Gregory Aharonian, who appears to use his own judgment to identify software patents (B&H, footnote 8). B&H claim their software patent time-trend is similar to Aharonian’s (though the trend comparison does not appear to be based on a formal test).

³ U.S. patent numbers 6,265,637 and 6,096,867, respectively.

Next, they compare their list to Allison and Tiller (2003), who evaluated a random sample of 1,000 patents granted from 1996-1998. Allison and Tiller classified 9.2% of their random sample as software patents.⁴ B&H, who classified 12.2% of all patents as software during that time period, contend that these figures are similar (B&H, pp. 4-5). This difference implies that B&H would classify about 11,000 more patents as software in their sample than would Allison and Tiller from 1996 to 1998 alone. If we assume that Allison and Tiller would consistently identify only about three-quarters as many software patents as would B&H, then Allison and Tiller would have identified nearly 34,000 fewer software patents than B&H between 1976 and 1999.

One alternative to the B&H approach would be the extensive International Patent Classification (IPC) scheme, reported by the U.S. Patent Office alongside U.S. patent classifications.⁵ The authors dismiss the official U.S. system, contending that it “does not classify inventions as software or something else,” and do not address the IPC system (B&H, p. 4). Indeed, the IPC system is far too detailed to have a single, simple, category called “software.” The patent classification system, however, allows researchers to identify software by aggregating subclasses of patents, as, for example, Graham and Mowery (2003) do in a highly relevant survey paper (“Intellectual Property Protection in the U.S. Software Industry”).⁶ Even if the authors believe the IPC would not yield an accurate count of software patents, they should consider explaining why their methodology is better and comparing their estimates to those obtained through this more conventional approach.

⁴ Allison and Tiller (2003) do not describe their method of identifying software patents; instead, they discuss their methodology for creating an “Internet patent data set.” Their method for identifying software patents is described in Allison and Lemley (2000), who actually identify only 7.6 percent of their 1000 patents as software. The change to 9.2% was based on “better knowledge of the drafting of software claims” and documented in “private communication from John Allison” (B&H, footnote 6).

⁵ See <http://www.uspto.gov/go/classification/uspcindex/indexa.htm> and http://www.wipo.int/classifications/fulltext/new_ipc/index.htm.

⁶ Graham and Mowery’s (2003) method may not be precisely correct for an analysis like B&H’s. Graham and Mowery base their definition on patenting by leading producers of personal computer software, potentially eliminating important categories that may contain software patents. The point here, though, is that rigorous categories exist for identifying software patents, and B&H do not use them. The subclasses that Graham and Mowery (2003) use to identify software patents include those under the headings “Electric Digital Data Processing;” “Recognition of Data;” “Presentation of Data;” “Record Carriers, Handling Record Carriers;” and “Electric Communication Technique.”

2.2 The NBER data file match does not include firms that went public after 1989

B&H match their list of software patents to the NBER Patent Citations Data File.⁷ The NBER dataset represents an enormous effort to allow researchers to exploit the vast amount of information contained in patents. The data identifies the patentholder (i.e., the individual, firm, or other entity that holds the patent) and, when the firm is publicly-traded, allows the firm to be matched to its financial data as provided by Compustat.

B&H do not explain clearly how they match firms to the NBER dataset. The lack of clarity in the paper on this matter is unfortunate because the NBER data has particular quirks that can affect an analysis, as Hall, et al. (2003) discuss. In particular, NBER's Compustat match is based on the 1989 universe of firms, which excludes all firms that went public in the 1990s—many of which were high-tech, and therefore of great interest in the context of software patenting.⁸ The paper has four relevant sentences regarding this match:

The NBER data set links patent numbers to the assignees (usually the firm that first owned the patent). We use that match to obtain financial data from Compustat on publicly held firms that are also patent holders. In addition, the largest 25 public firms in the software publishing industry (SIC 7372) were added. These firms obtain few software patents, so only one was included in the NBER match file (B&H, p. 5).

Note that these sentences—regarding a crucial component of the data—do not explain precisely how B&H added the Compustat financial data. There are at least two possibilities. One is that they worked with the Compustat data directly, matching as many of the firms as possible to the data. In this case, the authors would need to discuss how they dealt with firm births and deaths throughout the analysis. The paper contains no such discussion. A second possibility is that the authors used the Compustat match provided by the NBER datafile, which is the more likely possibility for two reasons. First, matching each firm manually would be a monumental undertaking. Indeed, Hall, et al. (2003) note that “this was one of the most difficult and time-consuming tasks of the entire [NBER] data construction project.” It is unlikely that B&H replicated and updated

⁷ See Hall, et al. (2003) for a detailed description of the NBER data file, or <http://www.nber.org/patents/> for the data itself.

this massive, multi-year effort without commenting on having done so. Moreover, B&H note that they used the “NBER match file,” which is the name of the NBER computer file that matches the firms to Compustat, further suggesting that they used the NBER match rather than their own (B&H, p. 5).

In principle, there is nothing wrong with using the NBER match. Indeed, the efforts by Hall, et al. (2003) to create that match should yield great insights as researchers exploit this rich resource. For the purposes of B&H’s paper, however, that method yields serious problems due to its being tied to the 1989 universe of publicly traded companies. In other words, this approach means that the paper excludes a large share of the very firms that are important to study. It may be partly for this reason that their final dataset contains only 42% of the “software” patents that they had previously identified (B&H, p. 5).

2.3 Using patent grants rather than patent applications is not appropriate

B&H use patent *grants* in a year rather than patent *applications* as their measure of firm patents. This choice is problematic, given that the paper is interested in the effects of policies on firms’ patenting behavior. The firm chooses when to apply for a patent, but has little control over the grant date, which is affected by events such as delays in the patent office. While there is a good deal of variability in the time from patent application to grant, the delay averages about two years (Hall, et al. 2003).⁹ It is for these reasons that applications are the more commonly-used metric in empirical analyses of innovation.¹⁰ By relying on the grant date, the analysis is less likely to capture information about the firm’s patenting decisions.

Patents contain the application date (as does the NBER data), so the authors could, in principle, use that information rather than the grant date. B&H “use patents granted as opposed to patent applications under the assumption that corporate intellectual

⁸ Between 1989 and 1999 (inclusive) there were 5,105 IPOs. For the data, see http://bear.cba.ufl.edu/ritter/publ_papers/IPOALL.xls.

⁹ The delay time can also indicate patent quality. Johnson and Popp (2003) find that more significant innovations take longer to go through the approval process (because significant innovations are often more complicated).

¹⁰ The patent-R&D literature generally finds that *applications* vary with R&D (Griliches 1989, e.g., Griliches 1986, Hall, et al. 1986, Jaffe and Trajtenberg 2003).

property departments anticipate appropriability needs and control the patenting process to meet those needs” (B&H, p. 11). The authors’ choice, though, in addition to being inconsistent with the literature, is especially problematic for their analysis.

Their paper aims to determine whether patenting behavior and its relationship to R&D has changed over time. As described in more detail below, they do this by interacting time period dummies with their patent variable. But the delay between application and grant means that, for example, when they claim to identify an effect of patents from 1996-1999, they are actually estimating an effect of patent applications from approximately two years earlier. In other words, a patent granted in year x is the ultimate result of a decision by the firm to submit an application in year $x - t$, where t averages approximately two years.

Perhaps most importantly, their choice means that they capture very little information regarding firms’ patenting decisions after 1996, when the USPTO “eliminated any remaining uncertainty over the patentability of computer programs” (B&H, p. 1). This date potentially represents an important policy shift, and it is pivotal in their analysis of whether firms’ patenting behavior changed in response to software patent policy changes. But grants (unlike applications) immediately following the 1996 policy shift are a response to firm patenting decisions prior to the new policy, not in response to it. As a result, the data say nothing regarding whether firm behavior changed in response to the new software policy adopted by the patent office.

2.4 The discussion of the distribution of software patents by industry is incomplete

B&H first summarize their data by tabulating the types of firms that receive software patents. The authors find that manufacturing firms receive 69% of software patents, while software publishers receive only 6% (B&H, table 4, p. 33).¹¹ In addition, holders of software patents tend to be larger and more established firms. B&H conclude that “[t]his pattern conflicts with the simple view that all firms are equally likely to obtain software patents to protect individual software inventions” (B&H, p. 8). But it is not immediately obvious why one would hold this view. Indeed, a more plausible view

¹¹ Graham and Mowery (2003) reach a qualitatively similar finding: electronic systems and component

might be that the share of all software patents a firm holds would match its share of all software R&D expenditures. The authors present no evidence to suggest that the distribution of software R&D by industry differs dramatically from the distribution of software patents.

Moreover, the authors' own tabulation shows that while manufacturing firms are responsible for 69% of software patents, they are also responsible for 85% of all patents (B&H, table 4, p. 33). In other words, while manufacturing firms do hold the bulk of software patents, they are actually responsible for a *smaller* share of software patents than their overall patent holdings might suggest.

3. Analysis and Methods

In this section we review B&H's methods of analysis. Here is a brief summary of our key points:

firms have far more software patents than do software publishers.

Are software patents “cheaper” than other patents?

- The regressions do not necessarily reveal much about the “cost” of patents. Taken at face value they demonstrate that increased software and total patents were positively correlated in the 1990s and negatively correlated in the 1980s.
- The underlying model makes the unrealistic assumption that firm output is a function of R&D, but not of patents.
- The empirical analysis does not directly follow from the model. The model is based on a firm’s *stock* of patents, but the empirical analysis uses the annual *flow* of patent grants.
- B&H do not properly incorporate patent benefits. The authors’ numbers show that citations—a reasonable measure of a patent’s value—to software patents far exceed citations to non-software patents, suggesting that software patents may have greater social benefits than other patents.
- The authors weight observations using the dependent variable, which is likely to yield inconsistent coefficient estimates.
- B&H discard firms with no patents, throwing away potentially valuable information regarding patenting decisions.

Are R&D and software patents complements or substitutes?

- Patents, R&D, and sales affect each other, but the authors do not account for these endogenous relationships, making it difficult to interpret coefficients.
- B&H use the ratio of R&D to sales as the dependent variable rather than the level of R&D, and the *share* of patents that are software rather than the *number* of software patents as the relevant independent variable. As a result, it is not possible to determine whether any coefficient estimate implies that R&D and patents are complements or substitutes.
- B&H eliminate firms where R&D is greater than one-half sales. While these firms may be outliers, the paper contains no discussion of this choice, which effectively truncates the dependent variable and thus potentially biases the results.
- One of the authors’ results in the “cost” analysis reveals a statistically significant positive correlation between patents and R&D, suggesting that they may be complements.

3.1 Are software patents and other patents equally costly?

To estimate the relative cost of patents over time, the authors derive the following equation:

$$\ln n_{it} = a_i + m_t + b \ln v_{it} + \sum_j d_T I(j = T) s_{i,t-1} + e_{it} \quad (\text{B\&H, equation 5, p. 13}),$$

where n_{it} is the number of patents firm i received in year t , s_{it} is the share of the firm's patents that are software patents, v is a vector of control variables,¹² and T represents four different groups of years: 1981-85, 1986-90, 1991-95, and 1996-99.¹³ The regression also controls for time and industry fixed effects.

Estimating this equation, however, reveals little information. First, its underlying model is built on unrealistic assumptions. Second, estimating the equation can reveal the correlation between the share of software patents firms have and their total number of patents, and how this correlation has changed over time. Interpreting these results, however, is difficult and the authors' interpretation is not necessarily correct. Finally, the estimation strategy itself has problems. We discuss these issues below.

3.1.1 *The model relies on unrealistic economic assumptions*

The authors begin their model by assuming that the firm's output depends only on R&D, but not on patents (B&H, equation 1, p. 9). In reality, both R&D *and* patents can affect sales, and probably in different ways.¹⁴ If B&H more realistically made output a function of R&D and patents, the rest of their theoretical model and the empirical analysis would no longer easily follow.¹⁵

In addition, the model is based on the "size of the firm's patent portfolio" (B&H, p. 9). That is, the model takes into account the firm's *stock* of patents. The equations are estimated, however, using the number of patents granted in a given year—i.e., a *flow* of patents. The empirical analysis makes no attempt to account for the contribution of the

¹² Including cash flow, market value, lagged R&D, capital per employee, share of patents with U.S. inventors, patent citations, and self-citations.

¹³ The authors do not explain why they chose these groups of years. The 1981 divide may be related to the 1981 *Diamond v. Diehr* Supreme Court decision, which made software patenting easier, and the 1996 divide may be related to the USPTO's 1996 examination guidelines. No explanation is provided for the groupings separated by 1985/86 and 1990/91.

¹⁴ Moreover, the firm's own patents and other firms' patents can affect sales.

¹⁵ Equation (1) currently reads: $V(n, R) = m A(n, N) Q(R)$, where n is the size of the firm's patent portfolio, N is the size of other firms' patent portfolio, A is the "appropriability function," Q is output, and R is R&D (B&H, p. 9). Note that Q depends only on R&D. Equation (1) should more accurately read $V(n, R) = m A(n, N) Q(R, n, N)$ so that output depends on R&D and patents. While this change would better approximate reality, the rest of the model would no longer flow easily from this key equation.

firm's existing patents to firm profitability. In other words, the empirical analysis, which relies on patent flow, does not follow from the theoretical model, which relies on patent stock.

The explanation of the model in the text does not precisely follow the mathematical derivation. The textual explanation is, itself, highly problematic. The coefficients of interest in the equation reproduced above are the d 's. The authors' basic assumption is reasonable: a firm has an optimal mix of patents in its patent portfolio, and the portfolio is determined in part by the cost of patenting. According to the authors, if "software patents are equivalent to other patents in cost and marginal value [then] the composition of the portfolio has no effect on optimal portfolio size" (B&H, p. 11). In this case, the empirical analysis would result in $d=0$. The rest of the explanation in the text, though, is less plausible. As B&H explain it, if software patents are more costly than other patents then "the greater use of software patents at a higher cost reduces the remaining budget available for the acquisition of non-software patents" (B&H, p. 11). In this case, the analysis would result in $d<0$: if the share of software patents increases, the total patent portfolio declines. Likewise, if software patents are less costly than other patents, then firms would have extra funds in their patenting "budget" and would be able to apply for additional patents. In this case, the analysis would yield $d>0$: if the share of software patents increases, the total patent portfolio increases.

The authors' explanation of their model is economically questionable. Unless firms are severely capital constrained—which is unlikely given that, as B&H note, these firms tend to be large—they are unlikely to have a patenting budget, *per se*. Instead, a rational firm will apply for a patent if it believes the expected value of that patent to the firm exceeds the cost of applying for it. In other words, the opportunity cost of applying for a patent is unlikely to be another patent for which they could not apply due to budget constraints.

While firms are unlikely to have patenting budget constraints, it is not obvious that the d coefficients actually represent relative patent costs unless one assumes that such a budget constraint exists. Indeed, one could imagine several possibilities that would change the relationship between the share of software patents and the total number

of patents over time that have little to do with patenting costs. For example, as technology evolves, the relationships between software and other patents may change.

3.1.2 The analysis does not properly account for patent value

Another problem with the analysis is that the cost of a patent to a firm is not, by itself, the relevant policy issue. A firm will decide to apply for a patent based on the firm's own interests: it will apply if the expected benefits to the firm exceed the costs to the firm. But the policy question is whether the total benefits of granting the patent (the total value of the patent to society) exceed the total costs.

How to measure patent value is hotly-debated. Still, scholars generally agree that citations (excluding self-citations) are a reasonable indicator of a patent's social value (e.g., Trajtenberg 1990). And B&H note that in their sample software patents received, on average, 9.7 citations each, while other patents received only 4.6 citations. The share of self-citations was 12% for software patents and 13% for other patents (B&H, table 2, p. 32). In other words, even if software patents were less costly than other patents, they patents seemed to be more valuable to society.¹⁶

3.1.3 The weighting scheme is econometrically incorrect

When estimating their equation, the authors weight observations by the number of patents the firm receives.¹⁷ In other words, they weight observations by the dependent variable. It is true, as the authors state, that the software share of patents will vary more with smaller numbers of patents across observations, but variation in an explanatory variable is not typically a cause for concern. Indeed, the variance should help make the coefficient estimates more precise. If, however, the authors are worried about heteroscedasticity then weighted least squares might be appropriate, but the weights should be proportional to the variance of ϵ , not the variance of s .

¹⁶ These averages complement Graham and Mowery (2003), who find that software patent quality, as measured by citations, has remained fairly constant from the 1980s through late 1990s. B&H include patent citations as a control variable in their regressions, but our point is that rather than holding patent value constant, the analysis should evaluate whether software patents have become more or less valuable.

¹⁷ In one case, the weight is actually a transformation of the number of patents: $1/(1/n_{it} + 1/n_{i,t-5})$.

As a technical matter, using n as a weight in weighted least squares is likely to produce inconsistent estimates. The dependent variable $\ln n_{it}$, and thus n_{it} , depends on the error term ε_{it} . Therefore, weighting the right-hand-side variables by n will induce correlation between the explanatory variables and the error term. That specification violates one of the classical assumptions of the least squares model, yielding inconsistent OLS estimates.¹⁸ As Greene (2003) states, the “weighted least squares estimator ... is consistent ... as long as the weights are uncorrelated with the disturbances.”¹⁹ If the weight is the dependent variable (or a transformation of it), then it is correlated with the disturbance and the estimator is no longer consistent.

3.1.4 Eliminating firms with no patents discards valuable information

B&H delete any observations where the number of patents, n_{it} , is zero “since neither $\ln n$ nor s is meaningful when $n = 0$ ” (B&H, p. 11). B&H have a computational problem—this is characteristic of analyses that use log transformations. Deleting observations, however, is not appropriate for at least two reasons. First, firms that choose not to patent or only rarely patent potentially provide valuable information. Deleting those firms biases the results even more sharply towards firms that patent heavily. It is possible that this truncation decision deletes only a small number of firms, but it may also eliminate a large number of firms—the authors provide no information on this point. Second, truncating the dependent variable can bias the coefficient estimates. In summary, this approach inappropriately discards important information from the analysis and may bias the results, but the authors do not provide any discussion of its implications.

Because the loss of information is problematic, researchers have developed many techniques to deal with the $n = 0$ problem that do not involve discarding observations. Some researchers, for example, add a small constant to the number—for example, $\ln(n + 1)$. Other techniques more explicitly handle this type of data. A common, and more rigorous, approach is to estimate a Poisson regression, which is appropriate for nonnegative integer count data such as patents (Hausman, et al. 1984). Either of these

¹⁸ That is, that the explanatory variables must be uncorrelated with the error term. Greene (2003), assumption A3, p. 42.

¹⁹ Greene (2003), p. 226.

approaches would be preferable to discarding observations and the valuable information they contain regarding firm patenting decisions.

3.2 Are software patents and R&D complements or substitutes?

This section of the B&H paper asks whether patents induce additional software R&D or substitute for it. The authors conclude that software patents substitute for R&D. However, as we will show, their analysis actually tells us little about that relationship.

To evaluate whether software patents and R&D are substitutes or complements, the authors estimate the following equation:

$$\Delta \frac{R_{it}}{Q_{it}} = a_t + \sum_j f_t I(j=t) \Delta s_{it} + \sum_j g_j \Delta \ln p_t^j + e_{it} \quad (\text{B\&H, equation 8, p. 17}),$$

where R_{it} is R&D spending; Q_{it} is sales; I is an indicator function that divides the sample into three time periods: 1985-89, 1990-94, and 1995-99; s_{it} is the share of patents classified as software patents; and p includes industry-level price indices.²⁰ All the variables are expressed as change over five years (i.e., the Δ 's). The coefficients of interest are those on the *share of patents*, s , interacted with the time periods (i.e., the f 's). The authors conclude that software patents and R&D were complements in late 1980s ($f > 0$) but became substitutes by the second half of the 1990s ($f < 0$). As we will see, however, the equation does not actually justify these conclusions.

We first discuss a major problem with their general approach. We then discuss flaws with the equation itself. These flaws make it impossible to draw firm conclusions from the results.

²⁰ These include price indices for capital, labor, materials, energy, purchased services, and IT capital. They also estimate the equation including “deflated” lag log sales, the standard deviation of the firm’s annual stock price growth, and change in long-term debt divided by capital, and whether the firm “entered the Compustat database within 5 years” (B&H, pp. 20-21 and note to table 6, p. 35).

3.2.1 Patents and R&D are simultaneously determined, but the analysis does not treat them that way

The authors treat the share of software patents as exogenous to R&D. But R&D can affect the decision to patent, and patents can affect the decision to do R&D. For example, Arora, et al. (2003), build a model that explicitly recognizes that R&D and patenting efforts are linked. While Arora, et al. do not look at software patents in particular, they find that patenting in general stimulates R&D across almost all manufacturing industries.

B&H could, in principle, analyze their data in a manner similar to Arora, et al. (2003). Their first section is, after all, an attempt to model the firm's decision to patent, and the second tries to look at the effects of that decision. A more rigorous version of this same paper would try to estimate those equations simultaneously, but the authors do not appear to acknowledge that patents and R&D are interrelated.

3.2.2 Choice of relevant independent variables flawed

B&H investigate the relationship between a firm's share of software patents and its R&D to sales ratio. This choice of variables makes it difficult to interpret their results. Consider the use of the *share* of patents that are software patents as the relevant independent variable rather than the *level* of software patenting. The share could increase if, for example, the number of software patents remained constant while the total number of patents decreased (or if the number of software patents increase at a faster rate than the number of total patents). Note that the use of total R&D spending—rather than software R&D spending—as the dependent variable exacerbates the problem. If total patenting decreased and software patenting remained constant (so that the share of software patents increased), while non-software (and thus total) R&D spending decreased, the analysis would find that ($f < 0$), and B&H would conclude that software patents and R&D were substitutes. But it would not be obvious in that case that the change in R&D spending had anything to do with software patenting. In other words, given the equation, one cannot conclude whether a coefficient estimate actually means that R&D and software patents are complements or substitutes.

3.2.3 Questionable decisions regarding outliers and weights

The authors weight the observations in the sample by the number of patents the firm receives. The procedure may not be statistically questionable, as in the section above, as it is not a transformation of the dependent variable.²¹ However, in addition to this weighting, the authors chose to exclude any observations as outliers where R&D exceeds one-half of sales. This decision to exclude some observations is not discussed anywhere in the text, but is simply noted in one sentence at the bottom of Table 6.²²

It may be necessary to deal with outliers. Unfortunately, B&H do not provide any information on how those outliers affect the analysis, why it is necessary to delete them, or how many firms are dropped from the analysis. Relatively few firms are likely to have R&D greater than one-half sales. According to the National Science Foundation, the average R&D to sales ratio across all industries tends to be between 4 and 5 percent.²³ But the U.S. Department of Commerce Technology Administration (2001) finds that the most R&D-intensive industries spend 15 to 40 percent of sales on R&D.²⁴ These documents provide only means and no variance, so it is difficult to know what share of firms would spend more than 50 percent of sales on R&D. Nonetheless, given these figures, it is possible that the number—even if small—is nontrivial. Moreover, eliminating outliers here means truncating the dependent variable. This kind of truncation can bias the coefficient estimates, but the authors do not discuss this potential source of bias.

²¹ If, however, current R&D affects patents then statistical problems similar to those discussed in the weighting section may, in fact, arise here, too.

²² It is worth noting that Arora, et al. (2003) note in their sample the outlier problem involves observations “with unrealistically high levels of patents per million dollars of R&D investment.” In other words, Arora, et al. believe that outliers tend to involve firms that have large numbers of patents, not firms that do too much R&D. The samples and data used by Arora, et al. and B&H differ, so it is not possible to say whether the outliers would be the same. Given the potential effects of B&H’s weight and outlier-trimming choices on the analysis, though, it would be worth more discussion.

²³ See <http://www.nsf.gov/sbe/srs/nsf00301/expendit.htm#intensity>.

²⁴ These industries include “non-diagnostic biologic product manufacturers” (39%), “in-vitro diagnostic substance manuf.” (~27%), “semiconductor indus. machine manuf” (~15%), and “software publishers” (~15%). See http://www.technology.gov/reports/CorpR&D_Inv/1996-2000_Figures2.xls.

3.2.4 Other results in the paper suggest that R&D and patents may be complements

The authors conclude from their analysis that software patents and R&D have become substitutes, although, as discussed above, the results do not warrant such a conclusion. It is worth noting that some of the authors' own findings contradict this conclusion. In particular, their patent equation uses R&D spending as a control variable to, they say, "proxy for changes that might not be captured in the [firm] profit measure" (B&H, p. 13). Though they do not discuss the coefficient estimates on this variable, their Table 5 reveals that lagged R&D is positively and significantly correlated with the number of patents the firm receives. In other words, the authors' analysis hints that R&D and patents may be complements even in their own data.

We do not claim that simply including R&D as an "independent" variable on the right-hand side of the patent equation is a rigorous test or the right way to conduct the analysis. It merely demonstrates that R&D and patents exhibit a statistically significant positive correlation even within B&H's own data.

4. Conclusion

Software patents are controversial. Proponents argue that patents are as important for protecting intellectual property rights and inducing innovation in software as they are in other technologies. Opponents counter that software patents are often too easy to obtain, that firms use them to build strategic "patent thickets," and that they may actually substitute for innovative activity. As the role of technology and software in the economy has steadily increased, research informing this debate is sorely needed.

Bessen and Hunt address these issues by analyzing how the cost of software patents to firms relative to other patents has changed over time and the relationship between software patents and R&D. They conclude that software patents have become "cheaper" than other patents and that software patents substitute for R&D. As discussed in detail in this note, however, problems with their data, methods of analysis, and the analysis itself do not justify their conclusions. The flaws of the B&H paper, of course, do not prove that the opposite results are true, nor can we rule out the possibility that they are correct. The bottom line is that this paper does not add much to this important debate.

The research on the interactions between patents and R&D is large and complex. Basing patent policy on any single paper would be a mistake. Basing it on a paper that has so many problems would be an even bigger mistake. Yet the flaws of the B&H paper do not diminish the importance of the policy question regarding software and strategic patenting, in general. Hall and Ziedonis (2001), for example, explore patenting in the U.S. semiconductor industry and reach a nuanced conclusion: stronger U.S. patent protection was correlated with strategic patent portfolio races among some firms, but also stimulated entry by specialized firms “and contributed to vertical disintegration in this industry.” This result is consistent with Arora, et al. (2001), who find that stronger patent protection helped smaller firms in India by giving them an incentive to innovate (the potential profits through licensing) even when they did not have the capability to commercialize their innovation themselves.

Hall and Ziedonis (2001) also note, though, that even when firms engage in strategic patenting, it is not possible to determine whether that is good or bad in a welfare sense. Such strategic behavior could be an implicit tax on research, they write, or it could facilitate technology transfer and knowledge spillovers. And, for the purpose of this particular policy debate, they do not explore the role of software patents, *per se*.

Future research should investigate innovation and software patents in a more rigorous manner if we hope to uncover the true relationship between them. Arora, et al. (2003) and Graham and Mowery (2003) provide useful models for such an exploration. Arora, et al. develop a model that explicitly recognizes that R&D and patenting affect each other, and find that patent protection seems to stimulate R&D across all industries. They do not, however, look explicitly at software patents. Graham and Mowery, meanwhile, develop a rigorous definition of “software patents” using the USPTO classification. They also use estimates of software R&D found in annual reports of IBM and Microsoft.

Together, these papers provide a possible research roadmap. First, it would be possible to more rigorously estimate the relationship between R&D and software patents using the Arora, et al. (2003) method, dividing their measure of patents into software and non-software using a definition of software patents along the lines described in Graham and Mowery (2003). Second, because a small number of firms are responsible for a large

share of patents (software and others), it may be possible to extend the Graham and Mowery approach, culling data from annual reports of these large firms to get a better view of their R&D breakdowns. Such an approach would lose the econometric rigor of the Arora, et al. method, but could potentially allow for more institutional details about the particular firms involved. Research along those lines has the potential to shed real light on the relationship between software patents and innovation and contribute to important policy questions.

References

- Allison, John R., and Mark A. Lemley. "Who's Patenting What? An Empirical Exploration of Patent Prosecution." *Vanderbilt Law Review* 53, no. 6 (2000): 2099-174.
- Allison, John R., and Emerson H. Tiller. *An Empirical Analysis of Internet Business Method Patents* 2003 [cited October 2003]. Available from http://www7.nationalacademies.org/step/Tiller_et_al_paper.doc.
- Andersen, Birgitte, Paul A. David, Davis. Lee N., Giovanni Dosi, David Encaoua, Dominique Foray, Alfonso Gambardella, Aldo Geuna, Bronwyn H. Hall, Dietmar Harhoff, Bengt-Åke Lundvall, Peter Holmes, Luc Soete, and W. Edward Steinmueller. *An Open Letter to the European Parliament Concerning the Proposed Directive on the Patentability of Computer-Implemented Inventions* 2003 [cited November 4, 2003]. Available from <http://www.researchineurope.org/policy/patentdirltr.pdf>.
- Arora, Ashish, Marco Ceccagnoli, and Wesley Cohen. "R&D and the Patent Premium." In *NBER Working Paper Series*. Cambridge, MA, 2003.
- Arora, Ashish, Andrea Fosfuri, and Alfonso Gambardella. *Markets for Technology: The Economics of Innovation and Corporate Strategy*. Cambridge, MA: MIT Press, 2001.
- Gallini, Nancy. "The Economics of Patents: Lessons from Recent U.S. Patent Reform." *Journal of Economic Perspectives* 16, no. 2 (2002): 131-54.
- Graham, Stuart J. H., and David C. Mowery. "Intellectual Property Protection in the U.S. Software Industry." In *Patents in the Knowledge-Based Economy*, edited by Wesley M. Cohen and Stephen A. Merrill, 219-58. Washington, DC: National Research Council, 2003.
- Greene, William H. *Econometric Analysis*. 5th ed. Upper Saddle River, NJ: Prentice Hall, 2003.
- Griliches, Zvi. "Patents: Recent Trends and Puzzles." *Brookings Papers on Economic Activity* Special Issue (1989): 291-319.
- . "Productivity, R&D, and the Basic Research at the Firm Level in the 1970's." *American Economic Review* 76, no. 1 (1986): 141-54.
- Hall, Bronwyn, Zvi Griliches, and Jerry Hausman. "Patents and R&D: Is There a Lag?" *International Economic Review* 27, no. 2 (1986): 265-83.

- Hall, Bronwyn, Adam Jaffe, and Manuel Trajtenberg. "The NBER Patent Citations Data File: Lessons, Insights, and Methodological Tools." In *Patents, Citations, and Innovations: A Window on the the Knowledge Economy*, edited by Adam Jaffe and Manuel Trajtenberg. Cambridge, MA: MIT Press, 2003.
- Hall, Bronwyn, and Rosemarie Ham Ziedonis. "The Patent Paradox Revisited: An Empirical Study of Patenting in the U.S. Semiconductor Industry, 1979-1995." *Rand Journal of Economics* 32, no. 1 (2001): 101-28.
- Hausman, Jerry, Bronwyn Hall, and Zvi Griliches. "Econometric Models for Handling Count Data with an Application to the Patents-R&D Relationship." *Econometrica* 52 (1984): 908-38.
- Jaffe, Adam. "The U.S. Patent System in Transition: Policy Innovation and the Innovation Process." *Research Policy* 29 (2000): 531-57.
- Jaffe, Adam, and Manuel Trajtenberg. "Introduction." In *Patents, Citations, and Innovations*, edited by Adam Jaffe and Manuel Trajtenberg. Cambridge, MA: MIT Press, 2003.
- Johnson, Daniel, and David Popp. "Forced out of the Closet: The Impact of the American Inventors Protection Act on the Timing of Patent Disclosure." *Rand Journal of Economics* 34, no. 1 (2003): 96-112.
- Lerner, Josh. "The Importance of Patent Scope: An Empirical Analysis." *Rand Journal of Economics* 25, no. 2 (1994): 319-33.
- Technology Administration. "U.S. Corporate R&D Investment, 1994 - 2000 Final Estimates." Washington, DC: Department of Commerce, 2001.
- Trajtenberg, Manuel. "A Penny for Your Quotes: Patent Citations and the Value of Innovations." *Rand Journal of Economics* 21 (1990): 172-87.