Patents and Innovation in Software

Tom Augspurger and Caleb Floyd

March 12, 2013

This Project

- Investigate the state of intellectual property protection.
  - Economic and social implications
- Search for theories that would allow us to analyze IP protection in the software industry.
  - Find commonalities in competing theories.
  - Search for existing empirical results.
  - Investigate the open source phenomenon.

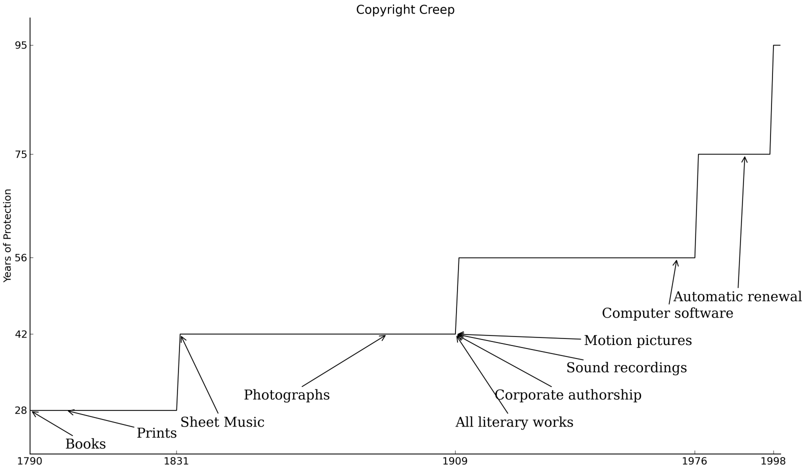
Where are we?

- Some theories are amenable; some aren't.
- One example where the open source phenomenon fits.
- One example where it doesn't.

Outline

- Copyright Law and Patents.
- Boldrin and Levine (2008).
- Acemoglu and Akcigit (2012).

The Copyright Creep



Ethics for the Information Age (2ND Edition) Micheal J. Quinn

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Copyright Term Extension Act

- CTEA of 1998
  - Created prior to 1978: 95 year protection.
  - Created after 1978: lifetime of the author plus 70 years.
  - Challenged on grounds of:
    - The Copyright Clause – "limited Times"
    - The First Amendment
    - The public trust doctrine
  - Upheld in *Eldred v. Ashcroft* by SCOTUS (January 15th, 2003).

Diamond v. Diehr (1981)

- Prior to 1981 software was effectively not patentable.
- Mathematical formulas in the abstract are not eligible for patent protection.
- However, a physical machine or process which makes use of a mathematical algorithm is different from an invention which claims the algorithm in the abstract.
- Software is effectively patentable.

Diamond v. Diehr (1981)

- Prior to 1981 software was effectively not patentable.
- Mathematical formulas in the abstract are not eligible for patent protection.
- However, a physical machine or process which makes use of a mathematical algorithm is different from an invention which claims the algorithm in the abstract.
- Software is effectively patentable.

Diamond v. Diehr (1981)

- Prior to 1981 software was effectively not patentable.
- Mathematical formulas in the abstract are not eligible for patent protection.
- However, a physical machine or process which makes use of a mathematical algorithm is different from an invention which claims the algorithm in the abstract.
- Software is effectively patentable.

Diamond v. Diehr (1981)

- Prior to 1981 software was effectively not patentable.
- Mathematical formulas in the abstract are not eligible for patent protection.
- However, a physical machine or process which makes use of a mathematical algorithm is different from an invention which claims the algorithm in the abstract.
- Software is effectively patentable.

Amazon One-Click Patent

A method and system for placing an order to purchase an item via the Internet. The order is placed by a purchaser at a client system and received by a server system. The server system receives purchaser information including identification of the purchaser, payment information, and shipment information from the client system. The server system then assigns a client identifier to the client system and associates the assigned client identifier with the received purchaser information. The server system sends to the client system the assigned client identifier and an HTML document identifying the item and including an order button. The client system receives and stores the assigned client identifier and receives and displays the HTML document. In response to the selection of the order button, the client system sends to the server system a request to purchase the identified item. The server system receives the request and combines the purchaser information associated with the client identifier of the client system to generate an order to purchase the item in accordance with the billing and shipment information whereby the purchaser effects the ordering of the product by selection of the order button.

U.S. Patent Sep. 28, 1999 Sheet 9 of 11 **5,960,411**

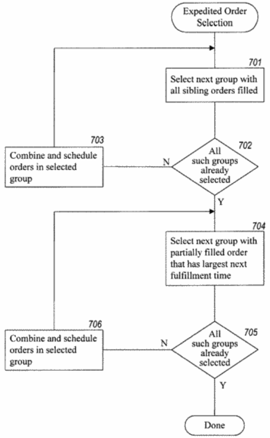
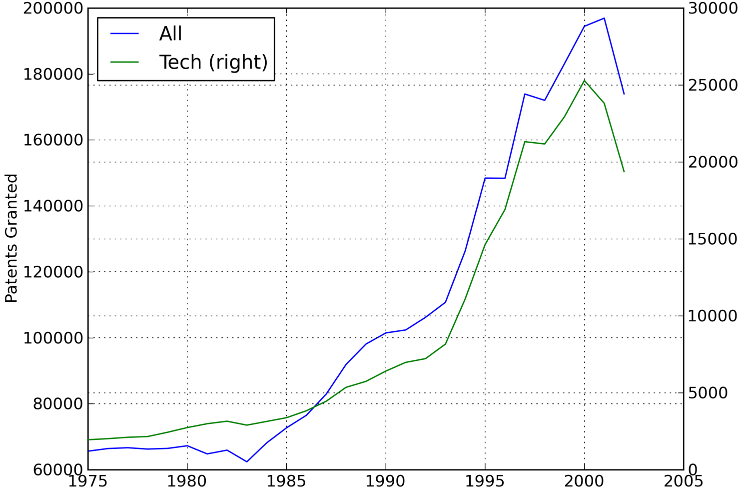


Fig. 7

Some Pictures

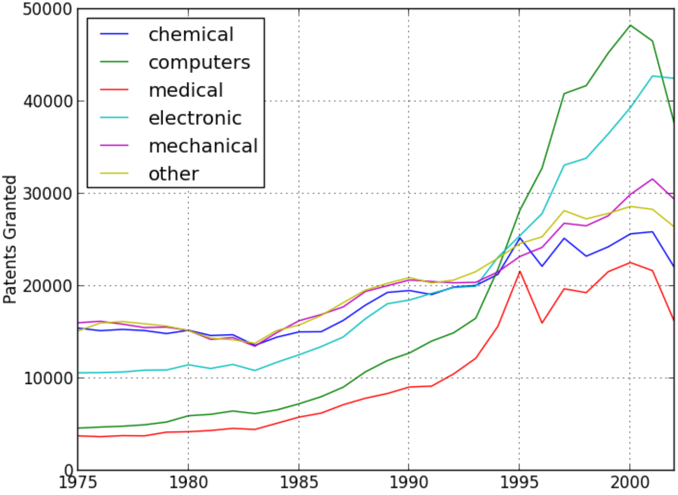
- NBER Patent Data File. Hall, Jaffe, Trajtenberg (2001).
- Patents *granted* from 1963 - 1999. (Since updated).
- Date, industry, citations, and company.

Growth in Patent Applications



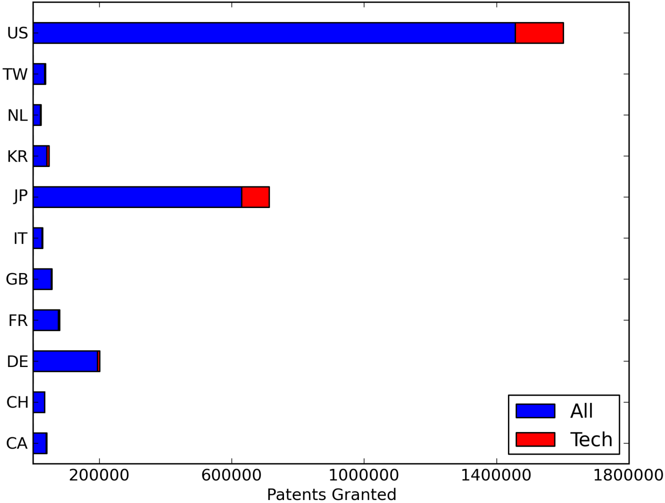
Hall, et al (2001). "The NBER Patent Citation Data File".

By Industry



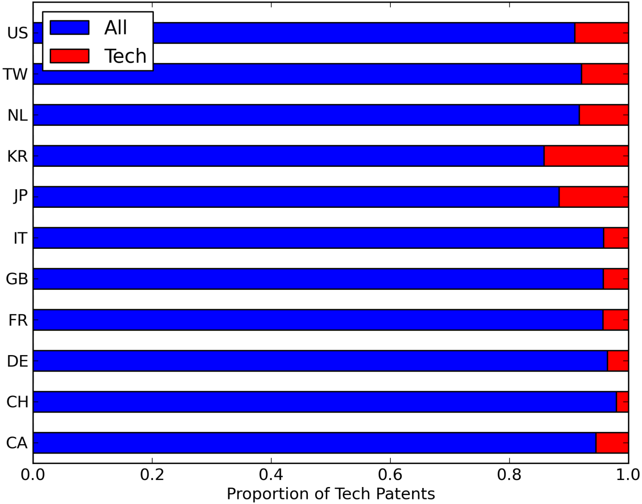
Hall, et al (2001). “The NBER Patent Citation Data File”.

Patents by Country



Hall, et al (2001). "The NBER Patent Citation Data File".

Patents by Country



Hall, et al (2001). "The NBER Patent Citation Data File".

Most Cited Patent

- February 2, 1988: Patent No. 4,723,129.
- Bubble jet recording method and apparatus in which a heating element generates bubbles in a liquid flow path to project droplets.

Most Cited Patent

- February 2, 1988: Patent No. 4,723,129.
- Bubble jet recording method and apparatus in which a heating element generates bubbles in a liquid flow path to project droplets.

Most Cited Patent

- February 2, 1988: Patent No. 4,723,129.
- Bubble jet recording method and apparatus in which a heating element generates bubbles in a liquid flow path to project droplets.
- Canon Ink Jet printers.

Why does open source coexist?

- Control over product performance.
- Hobbyists and enthusiasts.
- Display of skill or resume padding.
  - Lerner and Tirole (2002)
  - Hann et. al (2004)
- Competitive rents (Boldrin & Levine).
  - Which model version fits?
  - What can we say about the implications?

Why does open source coexist?

- Control over product performance.
- Hobbyists and enthusiasts.
- Display of skill or resume padding.
  - Lerner and Tirole (2002)
  - Hann et. al (2004)
- Competitive rents (Boldrin & Levine).
  - Which model version fits?
  - What can we say about the implications?

Why does open source coexist?

- Control over product performance.
- Hobbyists and enthusiasts.
- Display of skill or resume padding.
  - Lerner and Tirole (2002)
  - Hann et. al (2004)
- Competitive rents (Boldrin & Levine).
  - Which model version fits?
  - What can we say about the implications?

Why does open source coexist?

- Control over product performance.
- Hobbyists and enthusiasts.
- Display of skill or resume padding.
  - Lerner and Tirole (2002)
  - Hann et. al (2004)
- Competitive rents (Boldrin & Levine).
  - Which model version fits?
  - What can we say about the implications?

Why does open source coexist?

- Control over product performance.
- Hobbyists and enthusiasts.
- Display of skill or resume padding.
  - Lerner and Tirole (2002)
  - Hann et. al (2004)
- Competitive rents (Boldrin & Levine).
  - Which model version fits?
  - What can we say about the implications?

Why does open source coexist?

- Control over product performance.
- Hobbyists and enthusiasts.
- Display of skill or resume padding.
  - Lerner and Tirole (2002)
  - Hann et. al (2004)
- Competitive rents (Boldrin & Levine).
  - Which model version fits?
  - What can we say about the implications?

The evidence (and the common sense of anyone involved with OS software) shows that the source of competitive rents is the complementary sale of expertise.

...only small rents can be obtained through the sale of copies. [Purchasers] also have a demand for services, ranging from support and consulting to customization. They naturally prefer to hire the creators of the programs who in the process of writing the software have developed specialized expertise that is not easily matched by imitators.

- Boldrin & Levine (2009)

Boldrin & Levine

Boldrin & Levine: alternate notation

Table : Alternate Notation

BL		New
δ	$\longrightarrow$	β
β	$\longrightarrow$	λ
ζ	$\longrightarrow$	$1-\delta$

Boldrin & Levine: General Model Revisited

- Distinguish between productive input and consumption good: $\{k, c\}$.
- $c_t = F(k_t^c, l_t^c)$, $x_t = G(k_t^k, l_t^k)$.
- Agent solves $\sum_{t=0}^{\infty} \beta^t [u(c_t) - wL_t]$:
  - λk_t units available tomorrow: $k_{t+1} = \lambda k_t + x_t$.
  - $\lambda > 1$ gives us the 24/7 case.

Boldrin & Levine: General Model Revisited

- Distinguish between productive input and consumption good: $\{k, c\}$.
- $c_t = F(k_t^c, l_t^c)$, $x_t = G(k_t^k, l_t^k)$.
- Agent solves $\sum_{t=0}^{\infty} \beta^t [u(c_t) - wL_t]$:
  - λk_t units available tomorrow: $k_{t+1} = \lambda k_t + x_t$.
  - $\lambda > 1$ gives us the 24/7 case.

Boldrin & Levine: General Model Revisited

- Distinguish between productive input and consumption good: $\{k, c\}$.
- $c_t = F(k_t^c, l_t^c)$, $x_t = G(k_t^k, l_t^k)$.
- Agent solves $\sum_{t=0}^{\infty} \beta^t [u(c_t) - wL_t]$:
  - λk_t units available tomorrow: $k_{t+1} = \lambda k_t + x_t$.
  - $\lambda > 1$ gives us the 24/7 case.

Boldrin & Levine: General Model Revisited

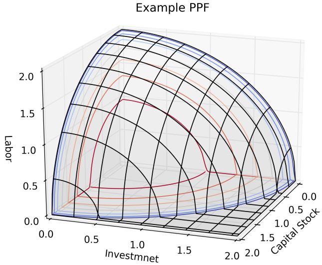
- Distinguish between productive input and consumption good: $\{k, c\}$.
- $c_t = F(k_t^c, l_t^c)$, $x_t = G(k_t^k, l_t^k)$.
- Agent solves $\sum_{t=0}^{\infty} \beta^t [u(c_t) - wL_t]$:
  - λk_t units available tomorrow: $k_{t+1} = \lambda k_t + x_t$.
  - $\lambda > 1$ gives us the 24/7 case.

Boldrin & Levine: General Model Revisited

- Distinguish between productive input and consumption good: $\{k, c\}$.
- $c_t = F(k_t^c, l_t^c)$, $x_t = G(k_t^k, l_t^k)$.
- Agent solves $\sum_{t=0}^{\infty} \beta^t [u(c_t) - wL_t]$:
  - λk_t units available tomorrow: $k_{t+1} = \lambda k_t + x_t$.
  - $\lambda > 1$ gives us the 24/7 case.

Boldrin & Levine: General Model Revisited

- Given $\{k_t, x_t, L_t\}$, the solution $c_t = T(k_t, x_t, L_t)$ traces a production possibility frontier



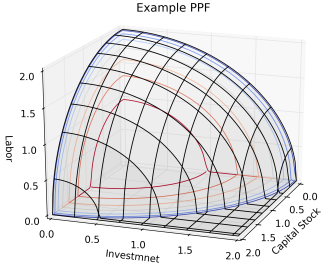
- L_t solves $\max\limits_{L_t} u[T(k_t, x_t, L_t)] - wL_t$
- The problem restated:

$$\nu(k_0) = \max_{\{k_t\}_{t=1}^{\infty}} \sum_{t=0}^{\infty} \beta^t V(k_t, k_{t+1} - \lambda k_t)$$

$$s.t. \qquad \lambda k_t + \overline{x}(k_t) \geq k_{t+1} \geq \lambda k_t$$

- As before, $q_0 = \nu'(k_0) > 0$ yields positive competitive rents

Boldrin & Levine: General Model Revisited



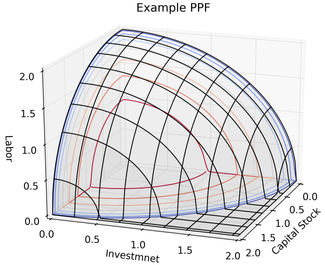
- Given $\{k_t, x_t, L_t\}$, the solution $c_t = T(k_t, x_t, L_t)$ traces a production possibility frontier
  - L_t solves $\max_{L_t} u[T(k_t, x_t, L_t)] - wL_t$
  - The problem restated:

$$\nu(k_0) = \max_{\{k_t\}_{t=1}^{\infty}} \sum_{t=0}^{\infty} \beta^t V(k_t, k_{t+1} - \lambda k_t)$$

$$s.t. \qquad \lambda k_t + \overline{x}(k_t) \geq k_{t+1} \geq \lambda k_t$$

  - As before, $q_0 = \nu'(k_0) > 0$ yields positive competitive rents

Boldrin & Levine: General Model Revisited



- Given $\{k_t, x_t, L_t\}$, the solution $c_t = T(k_t, x_t, L_t)$ traces a production possibility frontier
  - L_t solves $\max_{L_t} u[T(k_t, x_t, L_t)] - wL_t$
  - The problem restated:

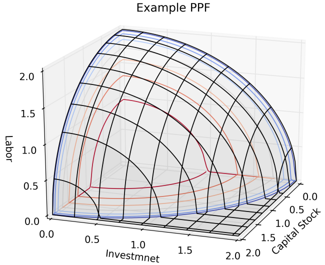
$$\nu(k_0) = \max_{\{k_t\}_{t=1}^{\infty}} \sum_{t=0}^{\infty} \beta^t V(k_t, k_{t+1} - \lambda k_t)$$

$$s.t. \qquad \lambda k_t + \overline{x}(k_t) \geq k_{t+1} \geq \lambda k_t$$

- As before, $q_0 = \nu'(k_0) > 0$ yields positive competitive rents

Boldrin & Levine: General Model Revisited

- Given $\{k_t, x_t, L_t\}$, the solution $c_t = T(k_t, x_t, L_t)$ traces a production possibility frontier



  - L_t solves $\max_{L_t} u[T(k_t, x_t, L_t)] - wL_t$
  - The problem restated:

$$\nu(k_0) = \max_{\{k_t\}_{t=1}^{\infty}} \sum_{t=0}^{\infty} \beta^t V(k_t, k_{t+1} - \lambda k_t)$$

$$s.t. \qquad \lambda k_t + \overline{x}(k_t) \geq k_{t+1} \geq \lambda k_t$$

  - As before, $q_0 = \nu'(k_0) > 0$ yields positive competitive rents

Special Case

Pills and Reverse Engineering is offered as a special case of the general model.

- This special case fits their idea of competitive rents in open source.
- c_t = services, h_t = expertise

Open Source Innovation and Selling Expertise

- Investment: $x_t = G(L_t)$
  (labor is chosen according to $L_t = g(x_t)$)
- Consumption (services): $c_t = f(h_t)$
- Law of motion:
  $$h_{t+1} = x_t + (1 - \delta)h_t$$
- Innovator starts with h_0
  - Instantly, others can begin accumulating productive capacity (expertise in the the software)

$$x_0^* = G(L_0^*)$$

Open Source Innovation and Selling Expertise

- Investment: $x_t = G(L_t)$
  (labor is chosen according to $L_t = g(x_t)$)
- Consumption (services): $c_t = f(h_t)$
- Law of motion:
  $$h_{t+1} = x_t + (1 - \delta)h_t$$
- Innovator starts with h_0
  - Instantly, others can begin accumulating productive capacity (expertise in the the software)

$$x_0^* = G(L_0^*)$$

Open Source Innovation and Selling Expertise

- Investment: $x_t = G(L_t)$
  (labor is chosen according to $L_t = g(x_t)$)
- Consumption (services): $c_t = f(h_t)$
- Law of motion:
  $$h_{t+1} = x_t + (1 - \delta)h_t$$
- Innovator starts with h_0
  - Instantly, others can begin accumulating productive capacity (expertise in the the software)
  $$x_0^* = G(L_0^*)$$

Open Source Innovation and Selling Expertise

- Investment: $x_t = G(L_t)$
  (labor is chosen according to $L_t = g(x_t)$)
- Consumption (services): $c_t = f(h_t)$
- Law of motion:
  $$h_{t+1} = x_t + (1 - \delta)h_t$$
- Innovator starts with h_0
  - Instantly, others can begin accumulating productive capacity (expertise in the the software)

$$x_0^* = G(L_0^*)$$

Open Source Innovation and Selling Expertise

- Investment: $x_t = G(L_t)$
  (labor is chosen according to $L_t = g(x_t)$)
- Consumption (services): $c_t = f(h_t)$
- Law of motion:
  $$h_{t+1} = x_t + (1 - \delta)h_t$$
- Innovator starts with h_0
  - Instantly, others can begin accumulating productive capacity (expertise in the the software)

$$x_0^* = G(L_0^*)$$

Open source innovation and selling expertise

- Consumer utility same as the general case
- Planners Problem:
  $$\nu(h_t) = \max_{x_t \geq 0}\{u(c_t) - wg(x_t) + \beta\nu(h_{t+1})\}$$
- First order condition:
  $$wg'(x_t) = \beta\nu'(h_{t+1})$$
- This can be decentralized with prices $\{p_t, q_t\}$ for services and capital
  - $p_t = u'(c_t)$
  - $q_t = \nu'(h_t) = u'(c_t)f'(h_t) + \beta(1-\delta)\nu'(h_{t+1})$

Open source innovation and selling expertise

- Consumer utility same as the general case
- Planners Problem:
$$\nu(h_t) = \max_{x_t \geq 0} \{u(c_t) - wg(x_t) + \beta\nu(h_{t+1})\}$$
- First order condition:
$$wg'(x_t) = \beta\nu'(h_{t+1})$$
- This can be decentralized with prices $\{p_t, q_t\}$ for services and capital
  - $p_t = u'(c_t)$
  - $q_t = \nu'(h_t) = u'(c_t)f'(h_t) + \beta(1-\delta)\nu'(h_{t+1})$

Open source innovation and selling expertise

- Consumer utility same as the general case
- Planners Problem:
  $$\nu(h_t) = \max_{x_t \geq 0}\{u(c_t) - wg(x_t) + \beta\nu(h_{t+1})\}$$
- First order condition:
  $$wg'(x_t) = \beta\nu'(h_{t+1})$$
- This can be decentralized with prices $\{p_t, q_t\}$ for services and capital
  - $p_t = u'(c_t)$
  - $q_t = \nu'(h_t) = u'(c_t)f'(h_t) + \beta(1-\delta)\nu'(h_{t+1})$

Open source innovation and selling expertise

- Consumer utility same as the general case
- Planners Problem:

$$\nu(h_t) = \max_{x_t \geq 0}\{u(c_t) - wg(x_t) + \beta\nu(h_{t+1})\}$$

- First order condition:

$$wg'(x_t) = \beta\nu'(h_{t+1})$$

- This can be decentralized with prices $\{p_t, q_t\}$ for services and capital
  - $p_t = u'(c_t)$
  - $q_t = \nu'(h_t) = u'(c_t)f'(h_t) + \beta(1-\delta)\nu'(h_{t+1})$

Open source innovation and selling expertise

- Consumer utility same as the general case
- Planners Problem:
$$\nu(h_t) = \max_{x_t \geq 0}\{u(c_t) - wg(x_t) + \beta\nu(h_{t+1})\}$$
- First order condition:
$$wg'(x_t) = \beta\nu'(h_{t+1})$$
- This can be decentralized with prices $\{p_t, q_t\}$ for services and capital
  - $p_t = u'(c_t)$
  - $q_t = \nu'(h_t) = u'(c_t)f'(h_t) + \beta(1-\delta)\nu'(h_{t+1})$

Open source innovation and selling expertise

- Consumer utility same as the general case
- Planners Problem:
$$\nu(h_t) = \max_{x_t \geq 0}\{u(c_t) - wg(x_t) + \beta\nu(h_{t+1})\}$$
- First order condition:
$$wg'(x_t) = \beta\nu'(h_{t+1})$$
- This can be decentralized with prices $\{p_t, q_t\}$ for services and capital
  - $p_t = u'(c_t)$
  - $q_t = \nu'(h_t) = u'(c_t)f'(h_t) + \beta(1-\delta)\nu'(h_{t+1})$

Open source innovation and selling expertise

- Rearranging:

$$q_0 = \sum_{t=0}^{\infty} (\beta(1-\delta))^t u'(c_t) f'(c_t)$$

- The open source innovation is viable as long as $q_0 k_0 > C$
- Perhaps more elucidating:

$$q_0 = \underbrace{u'(c_0) f'(h_0)}_{\text{first mover advantage}} + \underbrace{(1-\delta) w g'(x_0)}_{\text{cost of imitation}}$$

Open source innovation and selling expertise

- Rearranging:

$$q_0 = \sum_{t=0}^{\infty} (\beta(1-\delta))^t u'(c_t) f'(c_t)$$

- The open source innovation is viable as long as $q_0 k_0 > C$
- Perhaps more elucidating:

$$q_0 = \underbrace{u'(c_0)f'(h_0)}_{\text{first mover advantage}} + \underbrace{(1-\delta)wg'(x_0)}_{\text{cost of imitation}}$$

Open source innovation and selling expertise

- Rearranging:

$$q_0 = \sum_{t=0}^{\infty} (\beta(1-\delta))^t u'(c_t) f'(c_t)$$

- The open source innovation is viable as long as $q_0 k_0 > C$
- Perhaps more elucidating:

$$q_0 = \underbrace{u'(c_0) f'(h_0)}_{\text{first mover advantage}} + \underbrace{(1-\delta) w g'(x_0)}_{\text{cost of imitation}}$$

Conclusion

- Competitive rents can explain the open source phenomenon.
- Innovation occurs without the excludability of ideas.
- Particularly fatal to the notion that monopolies must exist for innovation.

Conclusion

- Competitive rents can explain the open source phenomenon.
- Innovation occurs without the excludability of ideas.
- Particularly fatal to the notion that monopolies must exist for innovation.

Conclusion

- Competitive rents can explain the open source phenomenon.
- Innovation occurs without the excludability of ideas.
- Particularly fatal to the notion that monopolies must exist for innovation.

Intro

- Daron Acemoglu and Ufuk Akcigit (2012) - Intellectual Property Rights Policy, Competition And Innovation.
- Optimal *state-dependent* Intellectual Property Rights policy in a dynamic environment.
- Standard tradeoff between monopoly distortions and motivation.
- Novel motivation for leaders.

Intro

- Daron Acemoglu and Ufuk Akcigit (2012) - Intellectual Property Rights Policy, Competition And Innovation.
- Optimal *state-dependent* Intellectual Property Rights policy in a dynamic environment.
- Standard tradeoff between monopoly distortions and motivation.
- Novel motivation for leaders.

Intro

- Daron Acemoglu and Ufuk Akcigit (2012) - Intellectual Property Rights Policy, Competition And Innovation.
- Optimal *state-dependent* Intellectual Property Rights policy in a dynamic environment.
- Standard tradeoff between monopoly distortions and motivation.
- Novel motivation for leaders.

Intro

- Daron Acemoglu and Ufuk Akcigit (2012) - Intellectual Property Rights Policy, Competition And Innovation.
- Optimal *state-dependent* Intellectual Property Rights policy in a dynamic environment.
- Standard tradeoff between monopoly distortions and motivation.
- Novel motivation for leaders.

Previous Research

- *Static* tradeoff between R&D incentive and monopoly distortions. Mixed conclusions.
- Mechanism design approach. Menu of patents and fees.
- Step-by-step innovation (Aghion, Harris, and Vickers 1997) — Higher growth from stiffer competition.

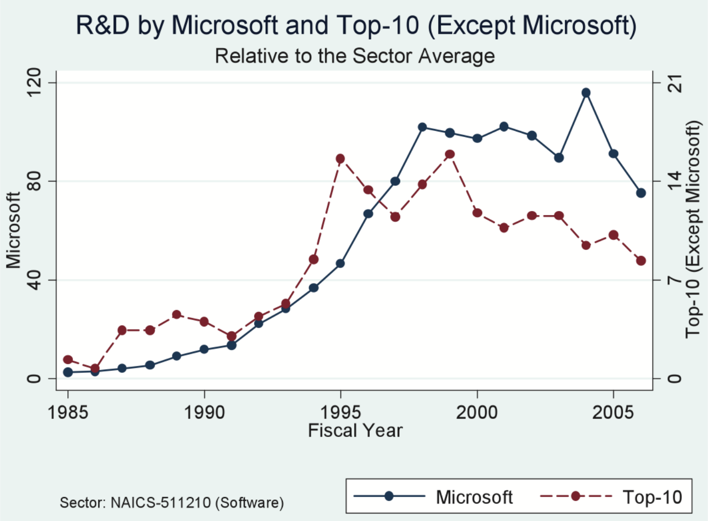
Previous Research

- *Static* tradeoff between R&D incentive and monopoly distortions. Mixed conclusions.
- Mechanism design approach. Menu of patents and fees.
- Step-by-step innovation (Aghion, Harris, and Vickers 1997) — Higher growth from stiffer competition.

Previous Research

- *Static* tradeoff between R&D incentive and monopoly distortions. Mixed conclusions.
- Mechanism design approach. Menu of patents and fees.
- Step-by-step innovation (Aghion, Harris, and Vickers 1997) — Higher growth from stiffer competition.

Motivation


R&D by Microsoft and Top-10 (Except Microsoft)
Relative to the Sector Average
Microsoft
Top-10 (Except Microsoft)
0
40
80
120
0
7
14
21
1985
1990
1995
2000
2005
Fiscal Year
Sector: NAICS-511210 (Software)
Microsoft
Top-10

Preview

- Consumers eating final good. Supplying Labor.
- Final goods producer aggregates.
- Intermediate producers hiring labor and innovating.
- State-dependent patent policy will motivate monopolists.
- Describe equilibrium vaguely.
- Numerical example to compare optimal policies.

Preview

- Consumers eating final good. Supplying Labor.
- Final goods producer aggregates.
- Intermediate producers hiring labor and innovating.
- State-dependent patent policy will motivate monopolists.
- Describe equilibrium vaguely.
- Numerical example to compare optimal policies.

Preview

- Consumers eating final good. Supplying Labor.
- Final goods producer aggregates.
- Intermediate producers hiring labor and innovating.
- State-dependent patent policy will motivate monopolists.
- Describe equilibrium vaguely.
- Numerical example to compare optimal policies.

Preview

- Consumers eating final good. Supplying Labor.
- Final goods producer aggregates.
- Intermediate producers hiring labor and innovating.
- State-dependent patent policy will motivate monopolists.
- Describe equilibrium vaguely.
- Numerical example to compare optimal policies.

Preview

- Consumers eating final good. Supplying Labor.
- Final goods producer aggregates.
- Intermediate producers hiring labor and innovating.
- State-dependent patent policy will motivate monopolists.
- Describe equilibrium vaguely.
- Numerical example to compare optimal policies.

Preview

- Consumers eating final good. Supplying Labor.
- Final goods producer aggregates.
- Intermediate producers hiring labor and innovating.
- State-dependent patent policy will motivate monopolists.
- Describe equilibrium vaguely.
- Numerical example to compare optimal policies.

Consumers' Preferences

- Single final good. Continuum of 1 individuals.

$$\mathbb{E}_t \int_t^{\infty} exp(-\rho(s-t)) \ln C(s) ds$$

where ρ is the discount factor.

- Supply 1 unit of labor.
- Own balanced portfolio of intermediate goods producers.

Consumers' Preferences

- Single final good. Continuum of 1 individuals.

$$\mathbb{E}_t \int_t^\infty exp(-\rho(s-t))\ln C(s)ds$$

where ρ is the discount factor.

- Supply 1 unit of labor.
- Own balanced portfolio of intermediate goods producers.

Consumers' Preferences

- Single final good. Continuum of 1 individuals.

$$\mathbb{E}_t \int_t^{\infty} exp(-\rho(s-t)) \ln C(s) ds$$

where ρ is the discount factor.

- Supply 1 unit of labor.
- Own balanced portfolio of intermediate goods producers.

Technology-Final Good

- Output of final good: $Y(t) = C(t)$.
- Production of $Y(t)$:

$$\ln Y(t) = \int_0^1 \ln y(j,t)dj$$

where $y(j,t)$ is the quantity of intermediate good j used.
- Perfect substitutes between intermediate varieties.

Technology-Final Good

- Output of final good: $Y(t) = C(t)$.
- Production of $Y(t)$:

$$\ln Y(t) = \int_0^1 \ln y(j,t) dj$$

where $y(j,t)$ is the quantity of intermediate good j used.
- Perfect substitutes between intermediate varieties.

Technology-Final Good

- Output of final good: $Y(t) = C(t)$.
- Production of $Y(t)$:

$$\ln Y(t) = \int_0^1 \ln y(j,t)dj$$

where $y(j,t)$ is the quantity of intermediate good j used.
- Perfect substitutes between intermediate varieties.

Technology-Intermediate Good

- Each industry $j \in [0,1]$ has two firms. Firms denoted by i (leader) and $-i$ (follower).
- Output:

$$y(j,t) = q_i(j,t) l_i(j,t)$$

where q_i is a technology level and l_i is labor used.

- Limit pricing:

$$p(j,t) = \frac{w(t)}{q_{-i}(j,t)}$$

- Cobb-Douglas production of final good implies:

$$y(j,t) = \frac{q_{-i}(j,t)}{w(t)} Y(t)$$

Technology-Intermediate Good

- Each industry $j \in [0,1]$ has two firms. Firms denoted by i (leader) and $-i$ (follower).
- Output:

$$y(j,t) = q_i(j,t) l_i(j,t)$$

where q_i is a technology level and l_i is labor used.

- Limit pricing:

$$p(j,t) = \frac{w(t)}{q_{-i}(j,t)}$$

- Cobb-Douglas production of final good implies:

$$y(j,t) = \frac{q_{-i}(j,t)}{w(t)} Y(t)$$

Technology-Intermediate Good

- Each industry $j \in [0,1]$ has two firms. Firms denoted by i (leader) and $-i$ (follower).
- Output:

$$y(j,t) = q_i(j,t) l_i(j,t)$$

  where q_i is a technology level and l_i is labor used.
- Limit pricing:

$$p(j,t) = \frac{w(t)}{q_{-i}(j,t)}$$

- Cobb-Douglas production of final good implies:

$$y(j,t) = \frac{q_{-i}(j,t)}{w(t)} Y(t)$$

Technology-Intermediate Good

- Each industry $j \in [0, 1]$ has two firms. Firms denoted by i (leader) and $-i$ (follower).
- Output:

$$y(j,t) = q_i(j,t) l_i(j,t)$$

where q_i is a technology level and l_i is labor used.
- Limit pricing:

$$p(j,t) = \frac{w(t)}{q_{-i}(j,t)}$$

- Cobb-Douglas production of final good implies:

$$y(j,t) = \frac{q_{-i}(j,t)}{w(t)} Y(t)$$

Technology-Innovation

- Poisson Innovation:

$$x_i(j,t) = F(h_i(j,t))$$

where $h_i(j,t)$ is the number of workers in R&D. Also define $G(x_i(j,t)) \equiv F^{-1}(x_i(j,t))$ (R&D employment).

- Leader innovation: technology $\uparrow$ by factor $\lambda > 1$.
- Follower innovation: quick catch-up (not patent infringing).
- Technology ladder: $q_i(j,t) = \lambda^{n_{ij}(t)}$, with $n_{ij}(t)$ giving the rung.
- Technology gap: $n_j(t) = n_{ij}(t) - n_{-ij}(t)$

Technology-Innovation

- Poisson Innovation:

$$x_i(j,t) = F(h_i(j,t))$$

  where $h_i(j,t)$ is the number of workers in R&D. Also define $G(x_i(j,t)) \equiv F^{-1}(x_i(j,t))$ (R&D employment).
- Leader innovation: technology $\uparrow$ by factor $\lambda > 1$.
- Follower innovation: quick catch-up (not patent infringing).
- Technology ladder: $q_i(j,t) = \lambda^{n_{ij}(t)}$, with $n_{ij}(t)$ giving the rung.
- Technology gap: $n_j(t) = n_{ij}(t) - n_{-ij}(t)$

Technology-Innovation

- Poisson Innovation:

$$x_i(j,t) = F(h_i(j,t))$$

  where $h_i(j,t)$ is the number of workers in R&D. Also define $G(x_i(j,t)) \equiv F^{-1}(x_i(j,t))$ (R&D employment).
- Leader innovation: technology $\uparrow$ by factor $\lambda > 1$.
- Follower innovation: quick catch-up (not patent infringing).
- Technology ladder: $q_i(j,t) = \lambda^{n_{ij}(t)}$, with $n_{ij}(t)$ giving the rung.
- Technology gap: $n_j(t) = n_{ij}(t) - n_{-ij}(t)$

Technology-Innovation

- Poisson Innovation:

$$x_i(j,t) = F(h_i(j,t))$$

  where $h_i(j,t)$ is the number of workers in R&D. Also define $G(x_i(j,t)) \equiv F^{-1}(x_i(j,t))$ (R&D employment).
- Leader innovation: technology $\uparrow$ by factor $\lambda > 1$.
- Follower innovation: quick catch-up (not patent infringing).
- Technology ladder: $q_i(j,t) = \lambda^{n_{ij}(t)}$, with $n_{ij}(t)$ giving the rung.
- Technology gap: $n_j(t) = n_{ij}(t) - n_{-ij}(t)$

Technology-Innovation

- Poisson Innovation:

$$x_i(j,t) = F(h_i(j,t))$$

  where $h_i(j,t)$ is the number of workers in R&D. Also define $G(x_i(j,t)) \equiv F^{-1}(x_i(j,t))$ (R&D employment).
- Leader innovation: technology $\uparrow$ by factor $\lambda > 1$.
- Follower innovation: quick catch-up (not patent infringing).
- Technology ladder: $q_i(j,t) = \lambda^{n_{ij}(t)}$, with $n_{ij}(t)$ giving the rung.
- Technology gap: $n_j(t) = n_{ij}(t) - n_{-ij}(t)$

Patent Policy

- Patents expire at Poisson rate: $\eta_{n_j}(t)$.
- Law of motion for technology gap in industry j:

$$\eta_j(t+\Delta t) = \begin{cases} \eta_j(t)+1 & \text{prob } x_i(j,t)\Delta t + o(\Delta t) \\ 0 & \text{prob } x_{-i}(j,t)\Delta t + \eta_{n_{j(t)}}\Delta t + o(\Delta t) \\ \eta_j(t) & \text{with the remainder} \end{cases}$$

- Write the patent policy as $\boldsymbol{\eta} : \mathbb{N} \to \mathbb{R}$.

Patent Policy

- Patents expire at Poisson rate: $\eta_{n_j}(t)$.
- Law of motion for technology gap in industry j:

$$\eta_j(t+\Delta t) = \begin{cases} \eta_j(t)+1 & \text{prob } x_i(j,t)\Delta t + o(\Delta t) \\ 0 & \text{prob } x_{-i}(j,t)\Delta t + \eta_{n_{j(t)}}\Delta t + o(\Delta t) \\ \eta_j(t) & \text{with the remainder} \end{cases}$$

- Write the patent policy as $\boldsymbol{\eta} : \mathbb{N} \to \mathbb{R}$.

Patent Policy

- Patents expire at Poisson rate: $\eta_{n_j}(t)$.
- Law of motion for technology gap in industry j:

$$\eta_j(t+\Delta t) = \begin{cases} \eta_j(t)+1 & \text{prob } x_i(j,t)\Delta t + o(\Delta t) \\ 0 & \text{prob } x_{-i}(j,t)\Delta t + \eta_{n_{j(t)}}\Delta t + o(\Delta t) \\ \eta_j(t) & \text{with the remainder} \end{cases}$$

- Write the patent policy as $\boldsymbol{\eta} : \mathbb{N} \to \mathbb{R}$.

Equilibrium

- $\boldsymbol{\mu}(t) \equiv \mu_n(t)_{n=0}^{\infty}$ is a distribution of *industries* over *technology gaps.*
- Loosely define an ALLOCATION as a sequence of decisions for leaders and followers, sequence of wage rates, and a sequence of distributions over gaps.
- Loosely define an EQUILIBRIUM as a sequence of decisions, wages, and output such that markets clear, firms' expected profits are maximized, and R&D policies are best responses.

Equilibrium

- $\boldsymbol{\mu}(t) \equiv \mu_n(t)_{n=0}^{\infty}$ is a distribution of *industries* over *technology gaps*.
- Loosely define an ALLOCATION as a sequence of decisions for leaders and followers, sequence of wage rates, and a sequence of distributions over gaps.
- Loosely define an EQUILIBRIUM as a sequence of decisions, wages, and output such that markets clear, firms' expected profits are maximized, and R&D policies are best responses.

Equilibrium

- $\boldsymbol{\mu}(t) \equiv \mu_n(t)_{n=0}^{\infty}$ is a distribution of *industries* over *technology gaps.*
- Loosely define an ALLOCATION as a sequence of decisions for leaders and followers, sequence of wage rates, and a sequence of distributions over gaps.
- Loosely define an EQUILIBRIUM as a sequence of decisions, wages, and output such that markets clear, firms' expected profits are maximized, and R&D policies are best responses.

Labor Market

- Three sources of demand: Production of intermediaries, and R&D by each firm.
- Combine the demand for intermediates: $y(j,t) = \frac{q_{-i}(j,t)}{w(t)} Y(t)$ with the production function to get:

$$l_n(t) = \frac{\lambda^{-n} Y(t)}{w(t)}$$

and so

$$1 \geq \sum_{n=0}^{\infty} \mu_n(t) \Big[\frac{1}{\omega(t)\lambda^n} + G(x_n(t)) + G(x_{-n}(t)) \Big]$$

where $\omega(t)$ is labor's share of income.

Labor Market

- Three sources of demand: Production of intermediaries, and R&D by each firm.
- Combine the demand for intermediates: $y(j,t) = \frac{q_{-i}(j,t)}{w(t)} Y(t)$ with the production function to get:

$$l_n(t) = \frac{\lambda^{-n} Y(t)}{w(t)}$$

and so

$$1 \geq \sum_{n=0}^{\infty} \mu_n(t) \left[\frac{1}{\omega(t)\lambda^n} + G(x_n(t)) + G(x_{-n}(t)) \right]$$

where $\omega(t)$ is labor's share of income.

Firm's Value Function

- Net present value when leading by n:

$$V_n(t) = \mathbb{E}_t \int_t^{\infty} exp(-r(s-t))[\Pi(s) - w(s)G(\hat{x}(s))]\, ds$$

- "Normalized"value function ($v_n(t) = V_n(t)/Y(t)$):

$$pv_n = \max_{x_n \geq 0} \; (1-\lambda^{-n}) - \omega^* G(x_n) + x_n[v_{n+1} - v_n] + [x^*_{-n} + \eta_n][v_0 - v_n]$$

- Instantaneous operating profits: $(1 - \lambda^{-n})$.
- R&D costs: $\omega^*(t)G(x_n(t))$.

Firm's Value Function

- Net present value when leading by n:

$$V_n(t) = \mathbb{E}_t \int_t^{\infty} exp(-r(s-t))[\Pi(s) - w(s)G(\hat{x}(s))]\, ds$$

- "Normalized" value function ($v_n(t) = V_n(t)/Y(t)$):

$$pv_n = \max_{x_n \geq 0} \; (1-\lambda^{-n}) - \omega^* G(x_n) + x_n[v_{n+1} - v_n] + [x^*_{-n} + \eta_n][v_0 - v_n]$$

- Instantaneous operating profits: $(1-\lambda^{-n})$.
- R&D costs: $\omega^*(t)G(x_n(t))$.

Firm's Value Function

- Net present value when leading by n:

$$V_n(t) = \mathbb{E}_t \int_t^{\infty} exp(-r(s-t))[\Pi(s) - w(s)G(\hat{x}(s))]\, ds$$

- "Normalized" value function ($v_n(t) = V_n(t)/Y(t)$):

$$pv_n = \max_{x_n \geq 0} \; (1-\lambda^{-n}) - \omega^* G(x_n) + x_n[v_{n+1} - v_n] + [x^*_{-n} + \eta_n][v_0 - v_n]$$

- Instantaneous operating profits: $(1-\lambda^{-n})$.
- R&D costs: $\omega^*(t)G(x_n(t))$.

Firm's Value Function

- Net present value when leading by n:

$$V_n(t) = \mathbb{E}_t \int_t^\infty exp(-r(s-t))[\Pi(s) - w(s)G(\hat{x}(s))]\, ds$$

- "Normalized" value function ($v_n(t) = V_n(t)/Y(t)$):

$$pv_n = \max_{x_n \geq 0} \ (1-\lambda^{-n}) - \omega^* G(x_n) + x_n[v_{n+1} - v_n] + [x^*_{-n} + \eta_n][v_0 - v_n]$$

- Instantaneous operating profits: $(1 - \lambda^{-n})$.
- R&D costs: $\omega^*(t)G(x_n(t))$.

Firm's Value Function

- Tied firm's value function:

$$\rho v_0 = \max_{x_0 \geq 0} -\omega^* G(x_0) + x_0[v_1 - v_0] + x_0^*[v_1 - v_0]$$

- Follower's value function:

$$\rho v_{-n} = \max_{x_{-n} \geq 0} -\omega^* G(x_{-n}) + [x_{-n} + \eta_n][v_0 - v_{-n}] + x_n^*[v_{-n-1} - v_{-n}])$$

Firm's Value Function

- Tied firm's value function:

$$\rho v_0 = \max_{x_0 \geq 0} -\omega^* G(x_0) + x_0[v_1 - v_0] + x_0^*[v_1 - v_0]$$

- Follower's value function:

$$\rho v_{-n} = \max_{x_{-n} \geq 0} -\omega^* G(x_{-n}) + [x_{-n} + \eta_n][v_0 - v_{-n}] + x_n^*[v_{-n-1} - v_{-n}])$$

Optimal R&D

$$x_n^* = max\{G'^{-1}\left(\frac{[v_{n+1} - v_n]}{\omega^*}\right), 0\} \tag{1}$$

$$x_{-n}^* = max\{G'^{-1}\left(\frac{[v_0 - v_{-n}]}{\omega^*}\right), 0\} \tag{2}$$

$$x_0^* = max\{G'^{-1}\left(\frac{[v_1 - v_0]}{\omega^*}\right), 0\} \tag{3}$$

Say patent protection weakens for $n+1$:

- Disincentive Effect: $\downarrow v_{n+1} \Rightarrow \downarrow v_{n+1} - v_n \Rightarrow \downarrow x_n^*$.
- Incentive Effect: $\downarrow v_{n+1} \Rightarrow \uparrow v_{n+2} - v_{n+1} \Rightarrow \uparrow x_{n+1}^*$.
- Composition Effect: x_n is decreasing in n. More firms in close competition. Higher aggregate R&D.
- Level Effect: Reduce prices and markups. Less duplicative R&D.

Optimal R&D

$$x_n^* = max\{G'^{-1}\left(\frac{[v_{n+1} - v_n]}{\omega^*}\right), 0\} \tag{1}$$

$$x_{-n}^* = max\{G'^{-1}\left(\frac{[v_0 - v_{-n}]}{\omega^*}\right), 0\} \tag{2}$$

$$x_0^* = max\{G'^{-1}\left(\frac{[v_1 - v_0]}{\omega^*}\right), 0\} \tag{3}$$

Say patent protection weakens for $n + 1$:

- Disincentive Effect: $\downarrow v_{n+1} \Rightarrow \downarrow v_{n+1} - v_n \Rightarrow \downarrow x_n^*$.
- Incentive Effect: $\downarrow v_{n+1} \Rightarrow \uparrow v_{n+2} - v_{n+1} \Rightarrow \uparrow x_{n+1}^*$.
- Composition Effect: x_n is decreasing in n. More firms in close competition. Higher aggregate R&D.
- Level Effect: Reduce prices and markups. Less duplicative R&D.

Optimal R&D

$$x_n^* = max\{G'^{-1}\left(\frac{[v_{n+1} - v_n]}{\omega^*}\right), 0\} \tag{1}$$

$$x_{-n}^* = max\{G'^{-1}\left(\frac{[v_0 - v_{-n}]}{\omega^*}\right), 0\} \tag{2}$$

$$x_0^* = max\{G'^{-1}\left(\frac{[v_1 - v_0]}{\omega^*}\right), 0\} \tag{3}$$

Say patent protection weakens for $n+1$:

- Disincentive Effect: $\downarrow v_{n+1} \Rightarrow \downarrow v_{n+1} - v_n \Rightarrow \downarrow x_n^*$.
- Incentive Effect: $\downarrow v_{n+1} \Rightarrow \uparrow v_{n+2} - v_{n+1} \Rightarrow \uparrow x_{n+1}^*$.
- Composition Effect: x_n is decreasing in n. More firms in close competition. Higher aggregate R&D.
- Level Effect: Reduce prices and markups. Less duplicative R&D.

Optimal R&D

$$x_n^* = max\{G'^{-1}\left(\frac{[v_{n+1} - v_n]}{\omega^*}\right), 0\} \tag{1}$$

$$x_{-n}^* = max\{G'^{-1}\left(\frac{[v_0 - v_{-n}]}{\omega^*}\right), 0\} \tag{2}$$

$$x_0^* = max\{G'^{-1}\left(\frac{[v_1 - v_0]}{\omega^*}\right), 0\} \tag{3}$$

Say patent protection weakens for $n + 1$:

- Disincentive Effect: $\downarrow v_{n+1} \Rightarrow \downarrow v_{n+1} - v_n \Rightarrow \downarrow x_n^*$.
- Incentive Effect: $\downarrow v_{n+1} \Rightarrow \uparrow v_{n+2} - v_{n+1} \Rightarrow \uparrow x_{n+1}^*$.
- Composition Effect: x_n is decreasing in n. More firms in close competition. Higher aggregate R&D.
- Level Effect: Reduce prices and markups. Less duplicative R&D.

Optimal R&D

$$x_n^* = max\{G'^{-1}\left(\frac{[v_{n+1} - v_n]}{\omega^*}\right), 0\} \tag{1}$$

$$x_{-n}^* = max\{G'^{-1}\left(\frac{[v_0 - v_{-n}]}{\omega^*}\right), 0\} \tag{2}$$

$$x_0^* = max\{G'^{-1}\left(\frac{[v_1 - v_0]}{\omega^*}\right), 0\} \tag{3}$$

Say patent protection weakens for $n + 1$:

- Disincentive Effect: $\downarrow v_{n+1} \Rightarrow \downarrow v_{n+1} - v_n \Rightarrow \downarrow x_n^*$.
- Incentive Effect: $\downarrow v_{n+1} \Rightarrow \uparrow v_{n+2} - v_{n+1} \Rightarrow \uparrow x_{n+1}^*$.
- Composition Effect: x_n is decreasing in n. More firms in close competition. Higher aggregate R&D.
- Level Effect: Reduce prices and markups. Less duplicative R&D.

Uniform Policy

Problem is identical for all followers.
Given some assumptions (positive R&D, non-zero profits) ...

- $v_{-1} \leq v_0$; $\{v_n\}_{n=0}^{\infty}$ is bounded and strictly increasing.
- $x_0^* > x_1^*, x_0^* \geq x_{-1}^*$, and $x_{n+1}^* \leq x_n^*$ $\forall n \in \mathbb{N}$.
- Tied firms innovate the most.
- Past that innovation is decreasing in the gap.

Uniform Policy

Problem is identical for all followers.
Given some assumptions (positive R&D, non-zero profits) . . .

- $v_{-1} \leq v_0$; $\{v_n\}_{n=0}^{\infty}$ is bounded and strictly increasing.
- $x_0^* > x_1^*, x_0^* \geq x_{-1}^*$, and $x_{n+1}^* \leq x_n^*$ $\forall n \in \mathbb{N}$.
- Tied firms innovate the most.
- Past that innovation is decreasing in the gap.

Optimal Uniform IPR

- Turns out that if $\eta_n = \eta \; \forall n$, $\eta^* = 0$ (Patents never expire).
- Positive composition effect (more firms in close competition) overwhelmed by negative disincentive effect.

Optimal Uniform IPR

- Turns out that if $\eta_n = \eta \ \forall n$, $\eta^* = 0$ (Patents never expire).
- Positive composition effect (more firms in close competition) overwhelmed by negative disincentive effect.

Optimal State-Dependent IPR

- Growth rate increases from 1.86% to 2.04%.
- Optimal patent length is increasing in technology gap.

Optimal State-Dependent IPR

- Growth rate increases from 1.86% to 2.04%.
- Optimal patent length is increasing in technology gap.

Comparing Optimal Policies

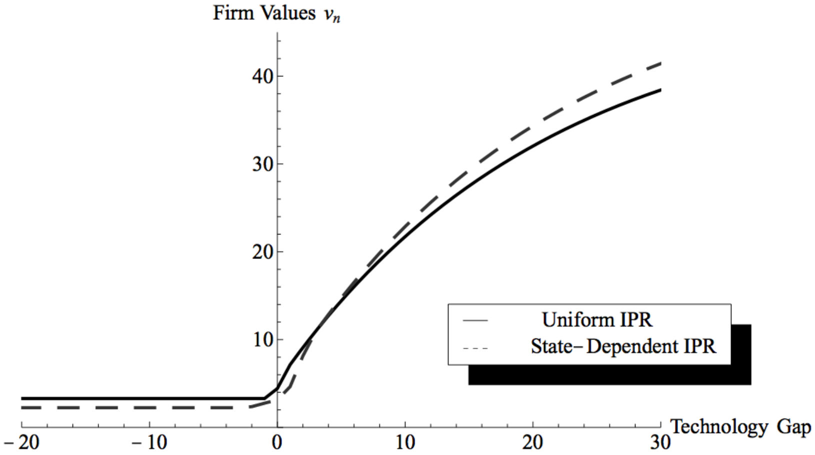


FIGURE 2. Value functions.

Comparing Optimal Policies

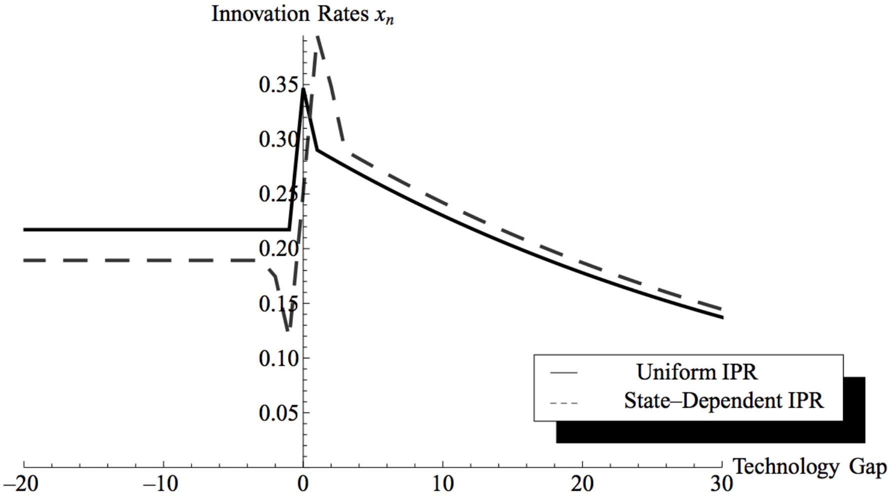


FIGURE 3. R&D efforts.

Comparing Optimal Policies

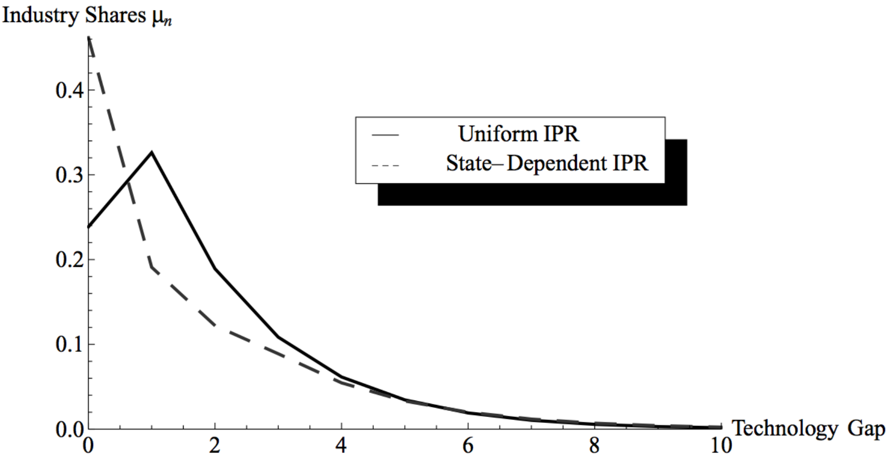


FIGURE 4. Industry shares.

Summary

- State-Dependent patent policy to motive all producers to innovate.
- Found that stronger protection should be given to those further ahead.

Summary

- State-Dependent patent policy to motive all producers to innovate.
- Found that stronger protection should be given to those further ahead.

Comparison

- Ask different questions: Are patents necessary for innovation? vs. What the optimal patent policy in this framework?
- Uniform vs. state-dependent.
- Patent policy recommendations:
  - Boldrin-Levine: Not necessary (Acemoglu agrees in his book)
  - Acemoglu-Akcigit: Optimal policy is infinite if uniform, increasing otherwise.

Conclusion

- In the open source spirit, fork away!
  - https://github.com/TomAugspurger/software
  - https://github.com/CalebFloyd/software