# Pairwise Comparisons with Flexible Time-Dynamics

**Lucas Maystre**, Victor Kristof, Matthias Grossglauser

KDD Research Track 2 — August 6th, 2019

# Pairwise comparison data

Association football example:

| Team 1 | Team 2 | Score |
|---|---|---|
| France | Portugal | 2-5 |
| Luxembourg | Greece | 0-3 |
| ... | ... | ... |
| Turkey | Slovakia | 1-1 |
| Bulgaria | Kosovo | 0-1 |

Questions we might want to ask:

"How can we quantify the **skill** of France?"

"How likely is South Korea to **beat** Germany?"

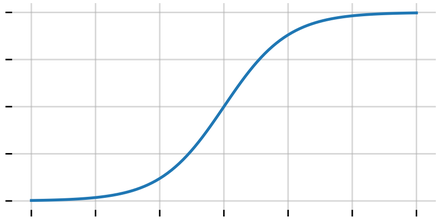
→ we need **pairwise comparison** models.

# Latent skill model

[Zermelo, 1928]
[Thurstone, 1927]

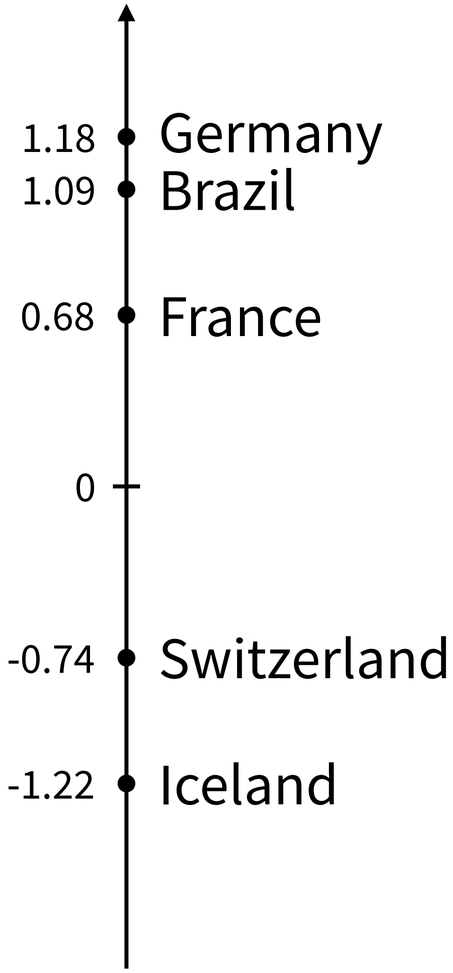
$$s_i \sim \mathcal{N}(0, \sigma^2)$$

latent skill of *i*



$$p(i \succ j) = \frac{1}{1 + \exp[-(s_i - s_j)]}$$

"*i* wins over *j*"




Given data $\mathcal{D}$, the posterior distribution is

$$p(\boldsymbol{s} \mid \mathcal{D}) \propto p(\boldsymbol{s}) \prod_{(i,j) \in \mathcal{D}} p(i \succ j)$$

# Actual setting

Data come with a **timestamp**.

| Date | Team 1 | Team 2 | Score |
|---|---|---|---|
| 1923-09-15 | France | Portugal | 2-5 |
| 1923-09-16 | Luxembourg | Greece | 0-3 |
| ... | ... | ... | ... |
| 2018-06-21 | Turkey | Slovakia | 1-1 |
| 2018-06-21 | Bulgaria | Kosovo | 0-1 |

Questions we might want to ask:

"How strong was France in **1972**? In **2018**?"

"How likely is South Korea to beat Germany **today**?"

→ we need **dynamic** models.

# This work: kickscore

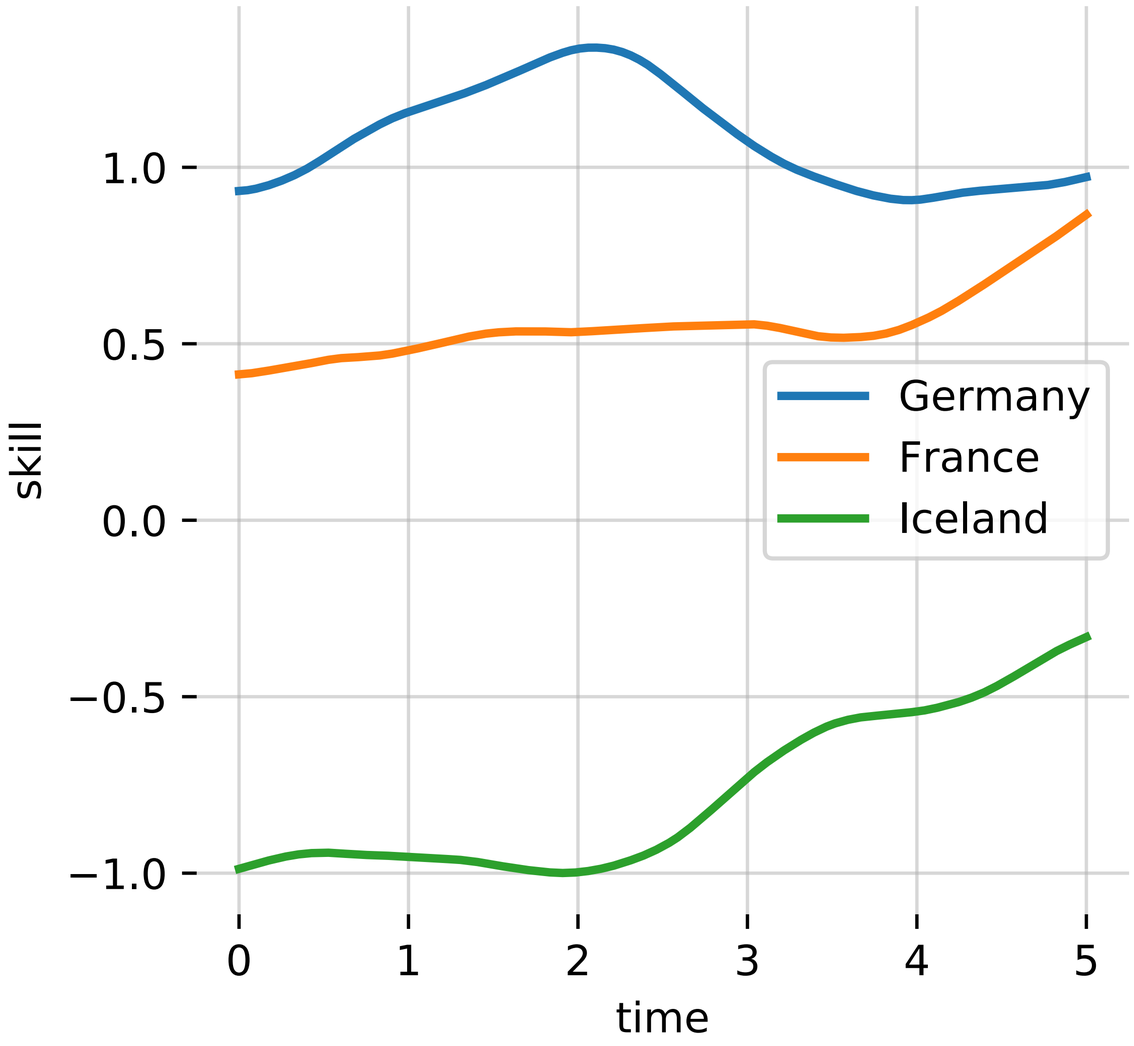
Skill becomes a (latent) **stochastic process**

$$s_i(t) \sim \mathrm{GP}[0, \underline{k(t, t')}]$$

covariance function,
defines **time dynamics**

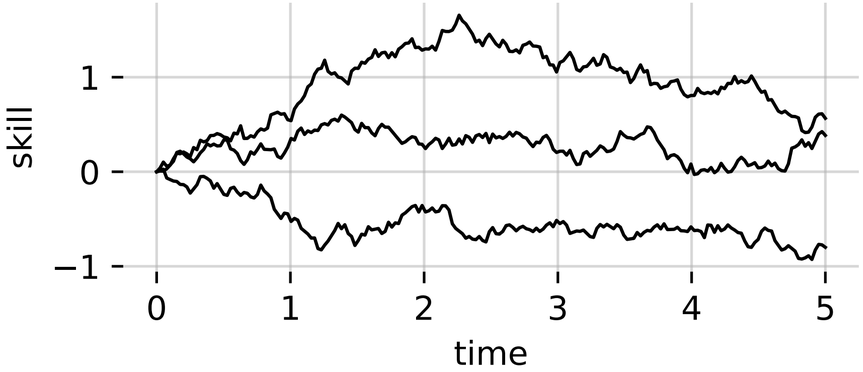
Obs. model is **conditionally parametric**:

$$p(i \succ j \mid t) = \frac{1}{1 + \exp\{-[s_i(t) - s_j(t)]\}}$$

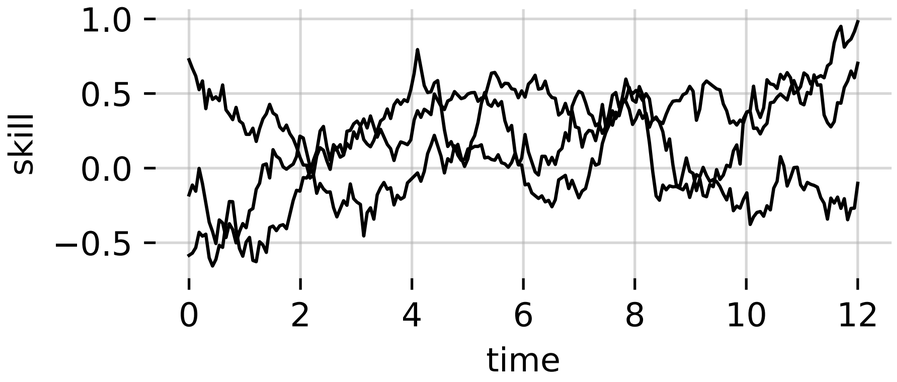


# Covariance functions

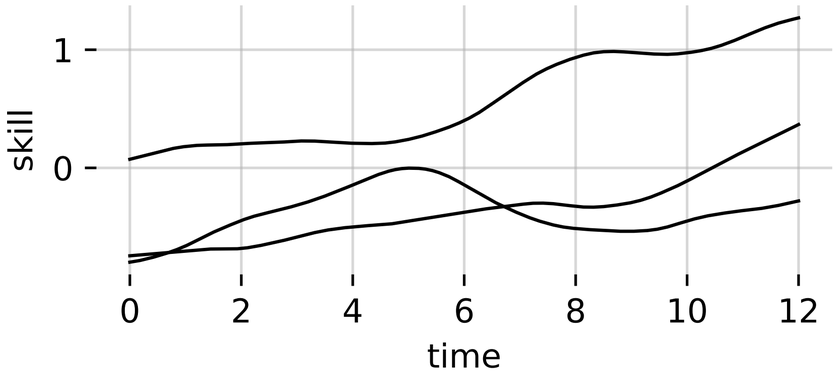
Brownian motion



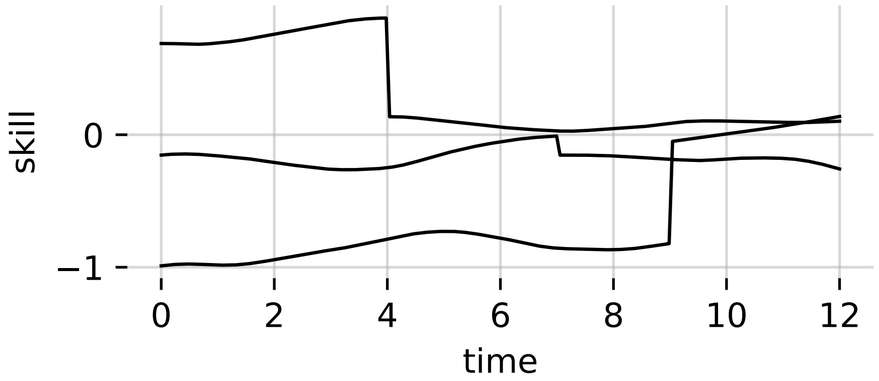

Mean-reverting, stationary dynamics




Smooth dynamics




Discontinuities



# Outline

1

**Inference algorithm**

2

**Experimental evaluation**

# Model inference

$$s_i(t) \sim \mathrm{GP}[0, k(t, t')]$$

$$\boldsymbol{s}_i = \begin{bmatrix} s_i(t_1) & \cdots & s_i(t_N) \end{bmatrix}$$

$$\approx \mathcal{N}[s_i(t) \mid \tilde{\mu}_{it}, \tilde{\sigma}_{it}] \cdot \mathcal{N}[s_j(t) \mid \tilde{\mu}_{jt}, \tilde{\sigma}_{jt}]$$

$$p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M \mid \mathcal{D}) \propto \prod_{i=1}^{M} p(\boldsymbol{s}_i) \prod_{(i,j,t) \in \mathcal{D}} p(i \succ j \mid t)$$

$$\min_{\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}} \mathrm{KL}(q \| p)$$

$$\approx q(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) \doteq \prod_i \mathcal{N}(\boldsymbol{s}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$q(\boldsymbol{s}_i)/p(\boldsymbol{s}_i)$$

Alternative viewpoint:

$$q(\boldsymbol{s}_i) \propto p(\boldsymbol{s}_i) \overline{\prod_{t \in \mathcal{D}} \mathcal{N}[s_i(t) \mid \tilde{\mu}_{it}, \tilde{\sigma}_{it}]}$$

# Inference algorithm

Converges in a few **linear time** iterations

**for each observation:**
approximate $p(i \succ j \mid t)$ by

$$\mathcal{N}[s_i(t) \mid \tilde{\mu}_{it}, \tilde{\sigma}_{it}] \times \mathcal{N}[s_j(t) \mid \tilde{\mu}_{jt}, \tilde{\sigma}_{jt}]$$

**for each item:**
recompute skill posterior

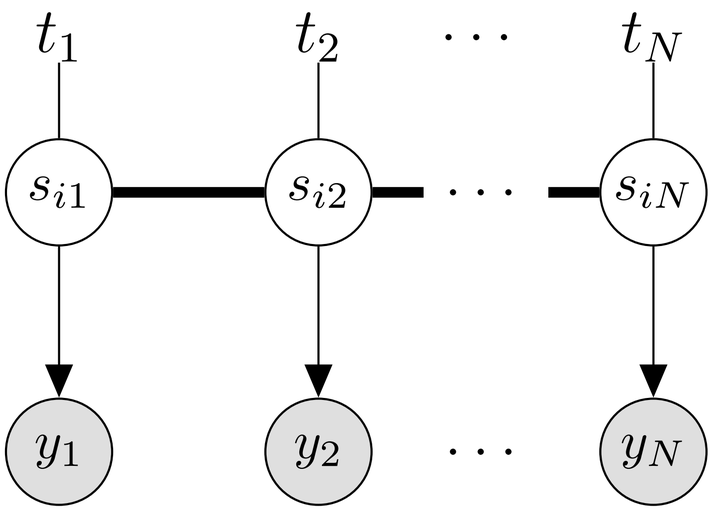
$$q(\boldsymbol{s}_i) \propto p(\boldsymbol{s}_i) \prod_{t \in \mathcal{D}} \mathcal{N}[s_i(t) \mid \tilde{\mu}_{it}, \tilde{\sigma}_{it}]$$

Using **EP** [Minka, 2001] or **CVI** [Khan et al., 2017]

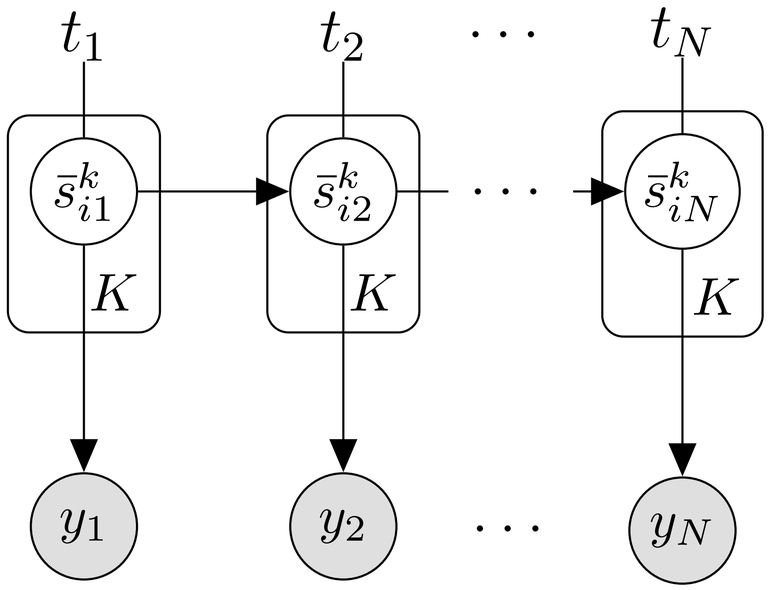
Using **SSM** reformulation [Hartikainen & Särkka., 2010]

# SSM reformulation

[O'Hagan, 1978]
[Hartikainen & Särkkä, 2010]




$s_i(t)$ is a Gaussian process




$\bar{s}_i(t)$ is a **Gauss-Markov** process

# Experimental evaluation

| Dataset | *N* | Timespan |
|---|---|---|
| ATP tennis | 618,934 | 1991-2017 |
| NBA basketball | 67,642 | 1946-2018 |
| FIFA football | 19,158 | 1908-2018 |
| Chess | 7,169,202 | 1475-2017 |
| StarCraft WoL | 61,657 | — |
| StarCraft HotS | 28,582 | — |

Measure **predictive performance**, compare against:

- Static
$$s_i(t) \equiv s_i$$
- Elo rating system
$$s_i \leftarrow s_i + \lambda \frac{\partial}{\partial s_i} \log p(i \succ j)$$
- Trueskill
$$s_i(t + \Delta t) = s_i(t) + \mathcal{N}[0, \sigma^2 \Delta t]$$

# Predictive log-loss

| Dataset | Constant | Elo | TrueSkill | kickscore | Covariance function |
|---|---|---|---|---|---|
| ATP tennis | 0.581 | 0.563 | 0.563 | **0.552** | Affine + Wiener |
| NBA basketball | 0.692 | 0.634 | 0.634 | **0.630** | Constant + Matérn 1/2 |
| World football | 0.929 | 0.950 | 0.937 | **0.926** | Constant + Matérn 1/2 |
| ChessBase small | 1.030 | 1.035 | 1.030 | **1.026** | Constant + Wiener |

kickscore **outperforms baselines** on all datasets

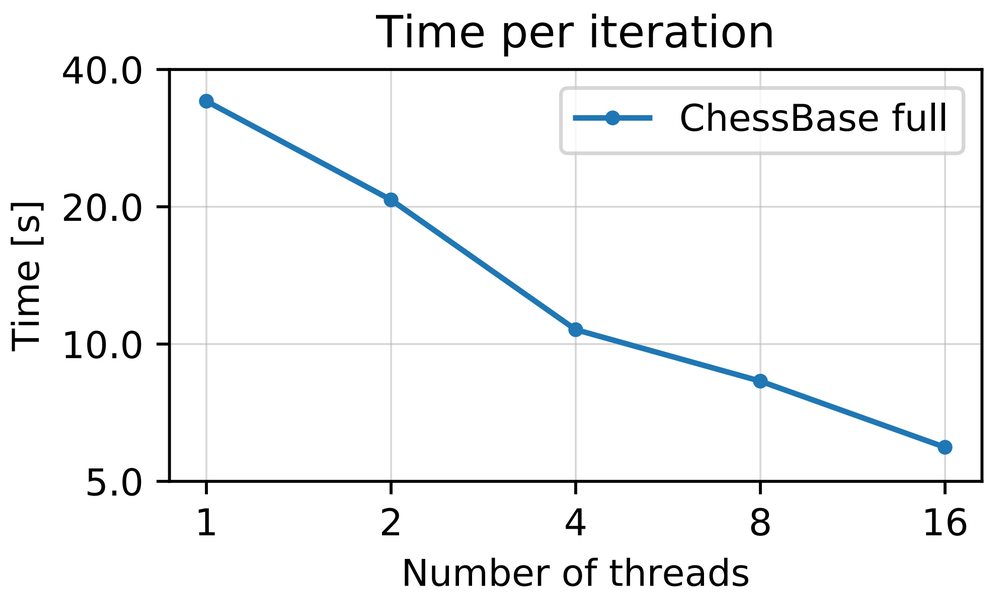
Best dynamics **vary** across datasets

# Experimental evaluation

Multi-threaded implementation in Go

→ **linear speed-up** with # threads

→ **scales to 7M** observations seamlessly




Impact of variational approximation

$$p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M \mid \mathcal{D}) \propto \prod_{i=1}^{M} p(\boldsymbol{s}_i) \prod_{(i,j,t)\in\mathcal{D}} p(i \succ j \mid t)$$

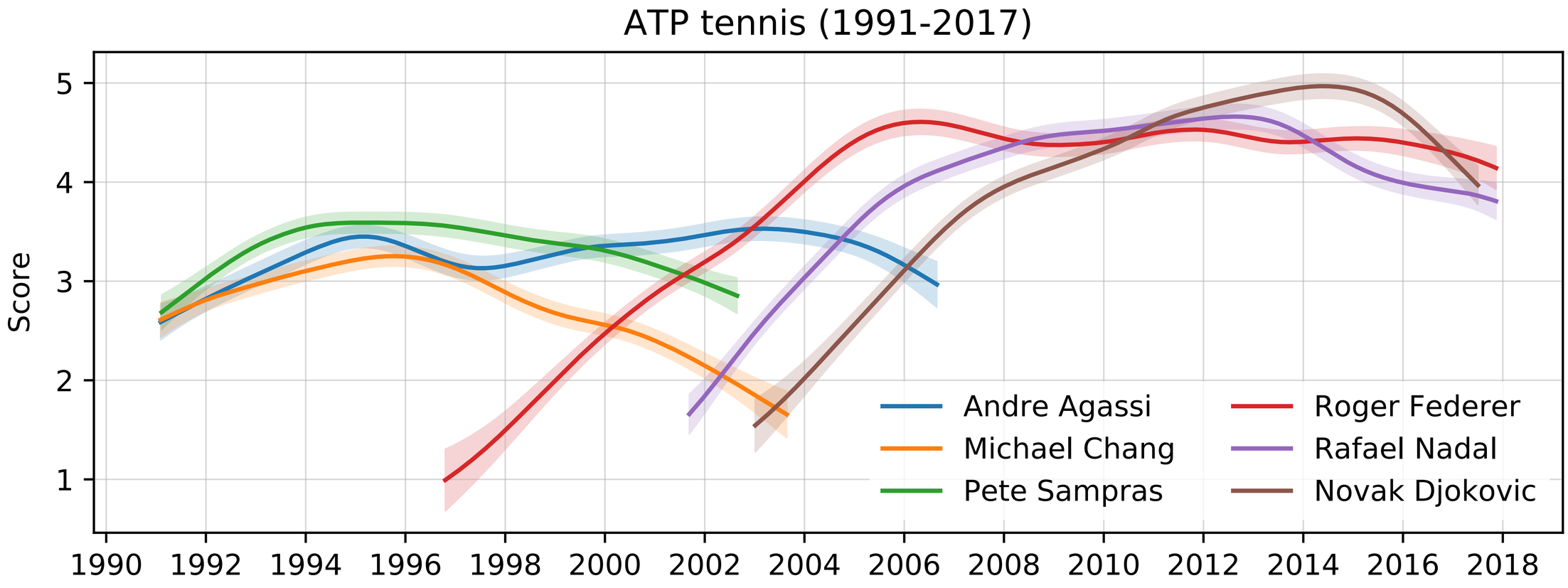
$$\approx q(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) \doteq \prod_i \mathcal{N}(\boldsymbol{s}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

predictive accuracy is **preserved**

Extensions of the model

- handle **intransitive data**
- accommodate other **observations likelihoods** (Poisson, Gaussian, ...)

# ATP tennis


ATP tennis (1991-2017)
Score
5
4
3
2
1
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
Andre Agassi
Michael Chang
Pete Sampras
Roger Federer
Rafael Nadal
Novak Djokovic

# Online resources

lucasmaystre/kickscore: Pairwi

GitHub, Inc. [US] | https://github.com/lucasmaystre/kickscore

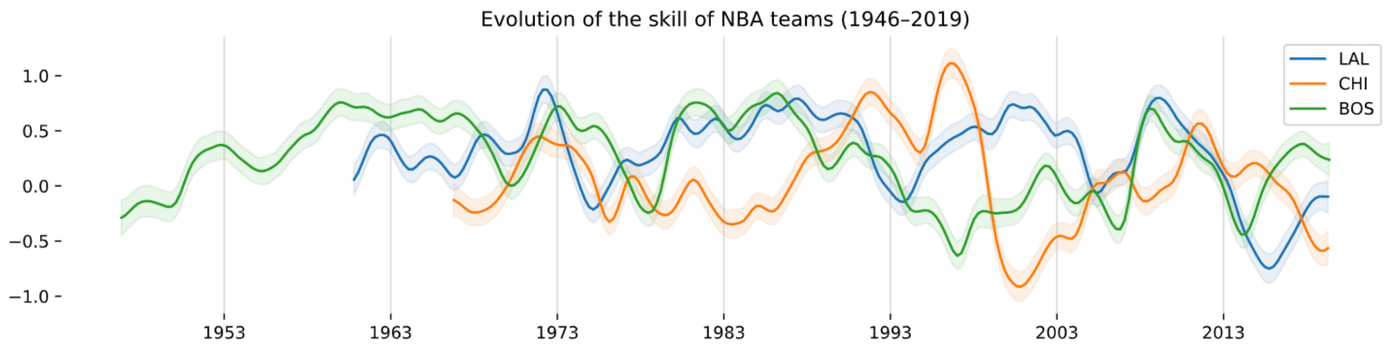
README.rst

## kickscore

build passing | codecov 86%

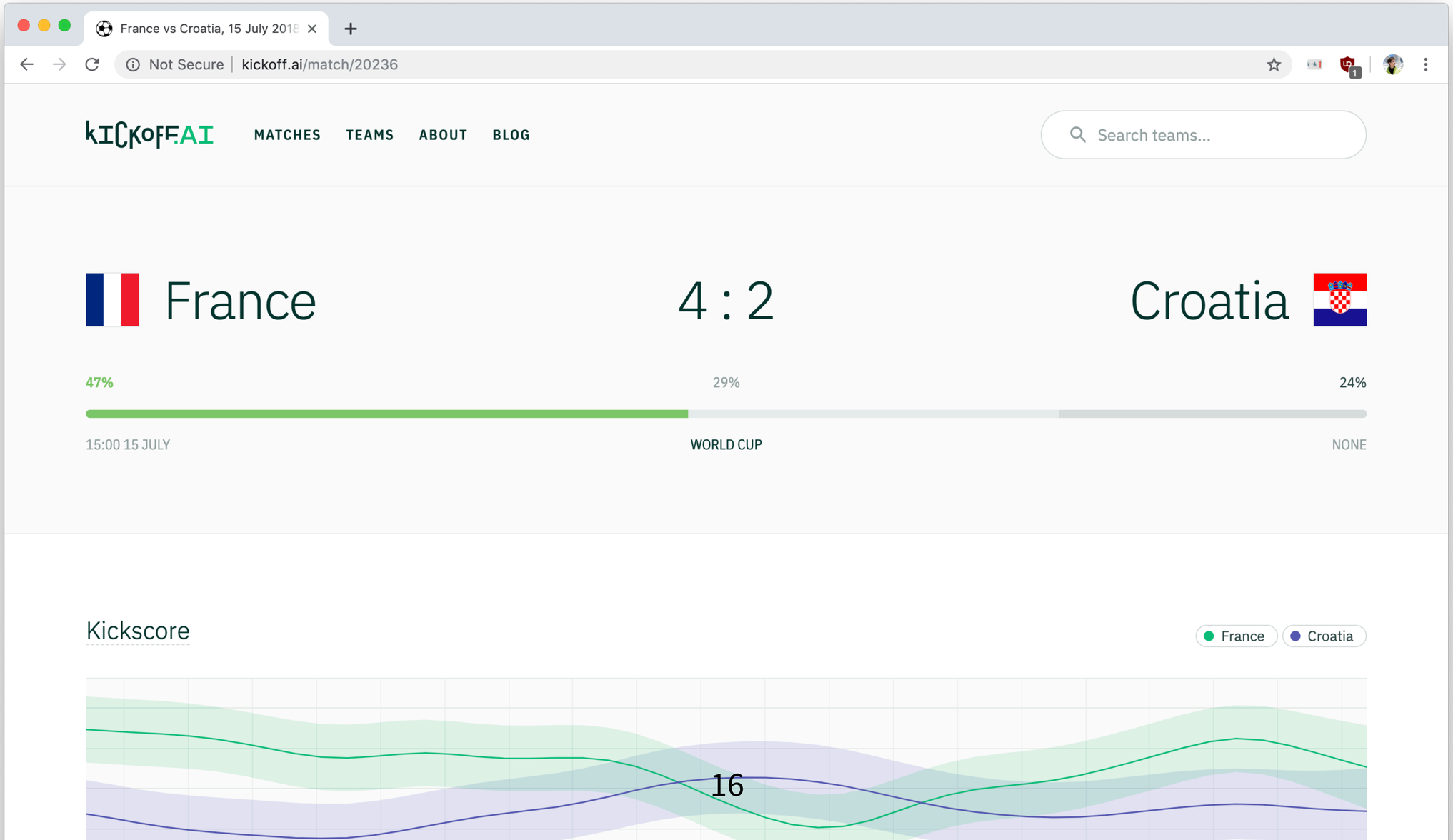
`kickscore` is the dynamic skill rating system powering Kickoff.ai.

In short, `kickscore` can be used to understand & visualize the skill of players (or teams) competing in pairwise matches, and to predict outcomes of future matches. It extends the Elo rating system and TrueSkill.



# Online resources

https://kickoff.ai



# Conclusion

Key contributions:

→ Pairwise comparison model with **flexible time-dynamics**

→ **Linear-time** inference algorithm

Try it out:

```
pip install kickscore
```