TÓPICOS EM CIÊNCIA DE DADOS PARA O ESPORTE

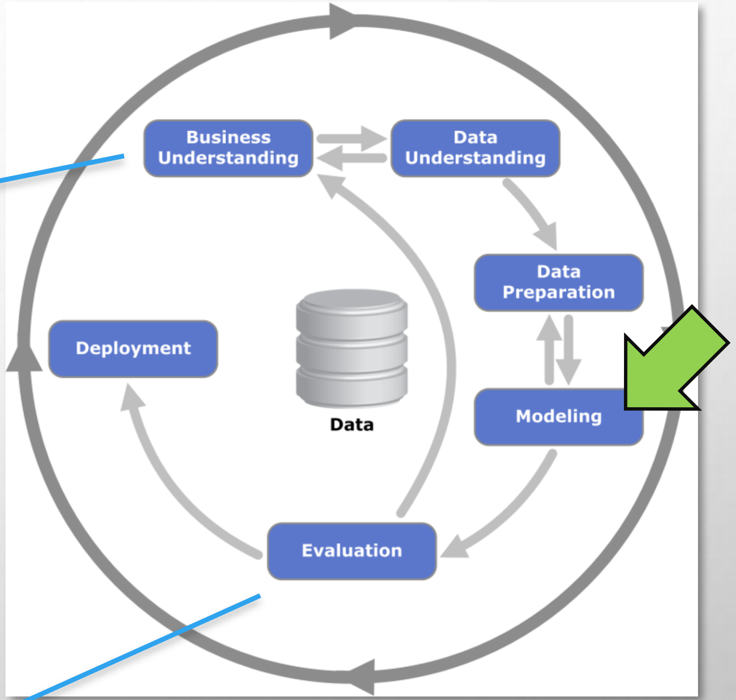
ALGORITMO TRUESKILL

DIEGO RODRIGUES DSC

INFNET

CRONOGRAMA

DIA	NÚMERO	ÁREA	AULA	TRABALHOS
30/1/2024	1	Intro	Introdução a Disciplina e Organização do Ambiente	
1/2/2024	2	Dados	Coleta de Dados e Sensoriamento	
6/2/2024	3	Estatística	Variáveis Aleatórias	Grupos
8/2/2024	4		Análise Exploratória	
15/2/2024	5		Estatísticas para Ranqueamento	
20/02/2024	6		Ranqueamento Estatístico : ELO	
22/02/2024	7		Ranqueamento Estatístico : Glicko	
27/2/2024	8		Ranqueamento Estatístico : TrueSkill	
29/2/2024	9		Ranqueamento Estatístico : XELO	Base de Dados
5/3/2024	10	ML	Modelos de Aprendizado de Máquina	
7/3/2024	11		Machine Learning: Classificação	
12/3/2024	12		Machine Learning: Regressão	
14/3/2024	13		Machine Learning: Agrupamento	Pesquisa
19/3/2024	14		Machine Learning: Visão Computacional	
21/3/2024	15	Esportes	Aplicações & Artigos: Esportes Independentes	Modelo
26/3/2024	16		Aplicações & Artigos: Esportes de Objeto	
28/3/2024	17		Aplicações & Artigos: Esportes de Combate	
2/4/2024	18		Aplicações & Artigos : Betting	
4/4/2024	19	Workshop	Workshop	
9/4/2024	20		Apresentações de Trabalhos I	Apresentação
11/4/2024	21		Apresentações de Trabalhos II	



CRISP-DM

AGENDA

- PARTE 1 : TEORIA
  - CONTEXTO
  - ALGORITMO TRUESKILL
- PARTE 2 : PRÁTICA
  - PROGRAMA PYTHON → RANQUEAMENTO ESTATÍSTICO DO BRASILEIRÃO (ELO VS TRUESKILL)

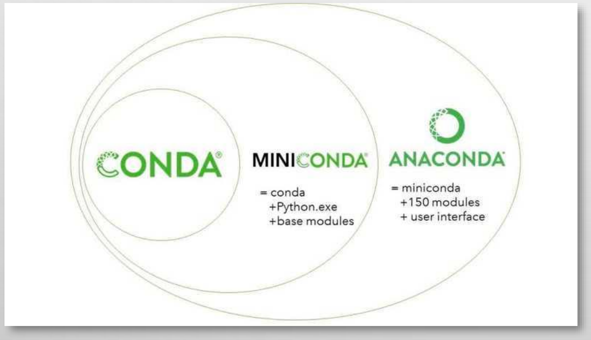
SETUP INICIAL DO AMBIENTE PYTHON

4. Variáveis Aleatórias

5. Visualização

6. Estimação e Inferência

1. Editor de Código



2. Gestor de Ambiente

3. Ambiente Python do Projeto

3. Notebook Dinâmico

CONTEXTO

FACTOR GRAPHS

Frank R. Kschischang

498 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 47, NO. 2, FEBRUARY 2001

Factor Graphs and the Sum-Product Algorithm

Frank R. Kschischang, *Senior Member, IEEE*, Brendan J. Frey, *Member, IEEE*, and Hans-Andrea Loeliger, *Member, IEEE*

***Abstract*—Algorithms that must deal with complicated global functions of many variables often exploit the manner in which the given functions factor as a product of “local” functions, each of which depends on a subset of the variables. Such a factorization can be visualized with a bipartite graph that we call a *factor graph*. In this tutorial paper, we present a generic message-passing algorithm, the sum-product algorithm, that operates in a factor graph. Following a single, simple computational rule, the sum-product algorithm computes—either exactly or approximately—various marginal functions derived from the global function. A wide variety of algorithms developed in artificial intelligence, signal processing, and digital communications can be derived as specific instances of the sum-product algorithm, including the forward/backward algorithm, the Viterbi algorithm, the iterative “turbo” decoding algorithm, Pearl’s belief propagation algorithm for Bayesian networks, the Kalman filter, and certain fast Fourier transform (FFT) algorithms.**

***Index Terms*—Belief propagation, factor graphs, fast Fourier transform, forward/backward algorithm, graphical models, iterative decoding, Kalman filtering, marginalization, sum-product algorithm, Tanner graphs, Viterbi algorithm.**

The aim of this tutorial paper is to introduce factor graphs and to describe a generic message-passing algorithm, called the *sum-product algorithm*, which operates in a factor graph and attempts to compute various marginal functions associated with the global function. The basic ideas are very simple; yet, as we will show, a surprisingly wide variety of algorithms developed in the artificial intelligence, signal processing, and digital communications communities may be derived as specific instances of the sum-product algorithm, operating in an appropriately chosen factor graph.

Genealogically, factor graphs are a straightforward generalization of the “Tanner graphs” of Wiberg *et al.* [31], [32]. Tanner [29] introduced bipartite graphs to describe families of codes which are generalizations of the low-density parity-check (LDPC) codes of Gallager [11], and also described the sum-product algorithm in this setting. In Tanner’s original formulation, all variables are codeword symbols and hence “visible”; Wiberg *et al.*, introduced “hidden” (latent) state variables and also suggested applications beyond coding. Factor

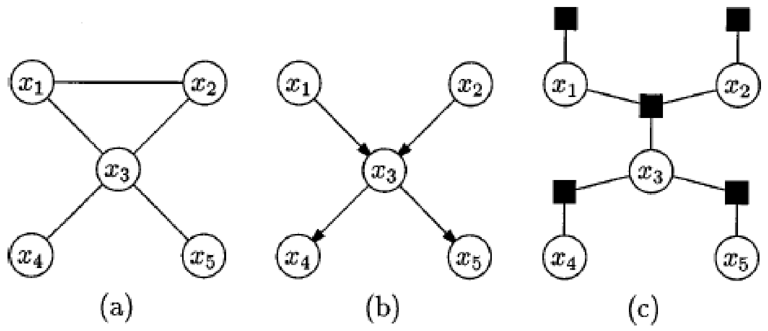


Fig. 24. Graphical probability models. (a) A Markov random field. (b) A Bayesian network. (c) A factor graph.

MESSAGE PASSING ALGORITHM

Frank R. Kschischang

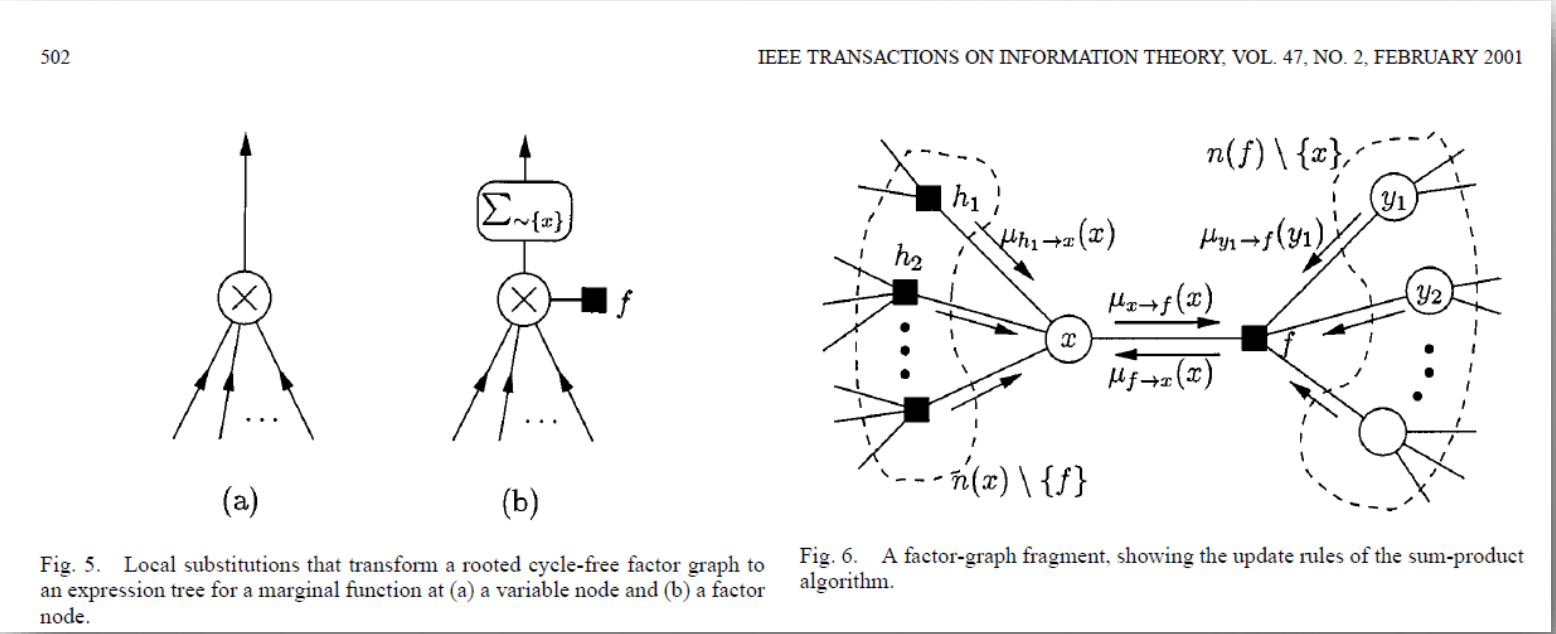


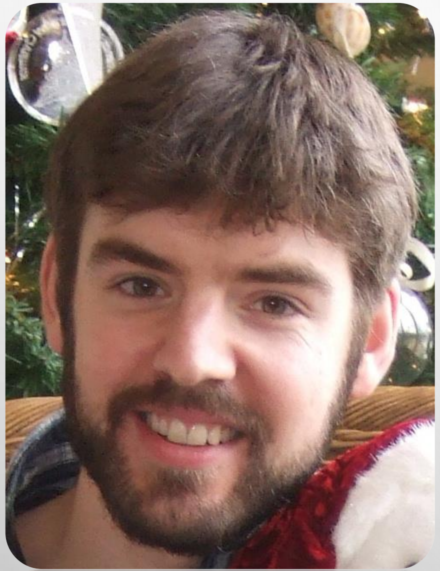
Fig. 5. Local substitutions that transform a rooted cycle-free factor graph to an expression tree for a marginal function at (a) a variable node and (b) a factor node.

Fig. 6. A factor-graph fragment, showing the update rules of the sum-product algorithm.

ALGORITMO TRUESKILL

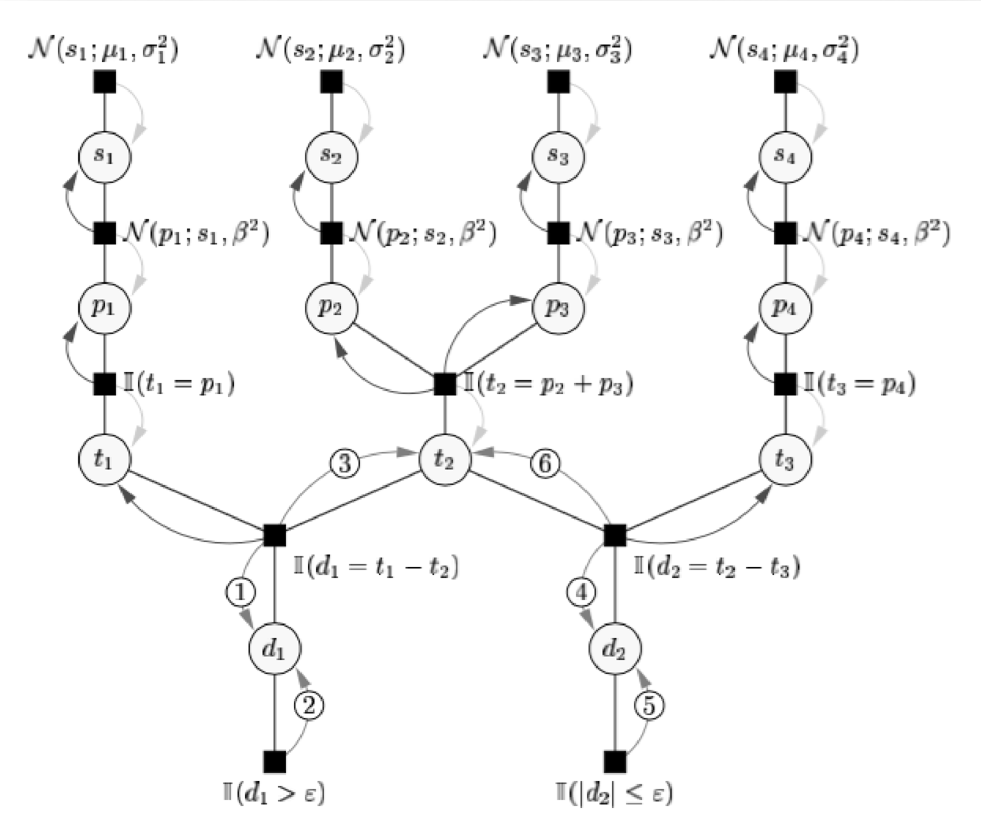
TRUESKILL

$$p(\mu|\mathbf{y}, A) = \frac{P(\mathbf{y}|\mu, A)\, p(\mu)}{P(\mathbf{y}|A)}$$

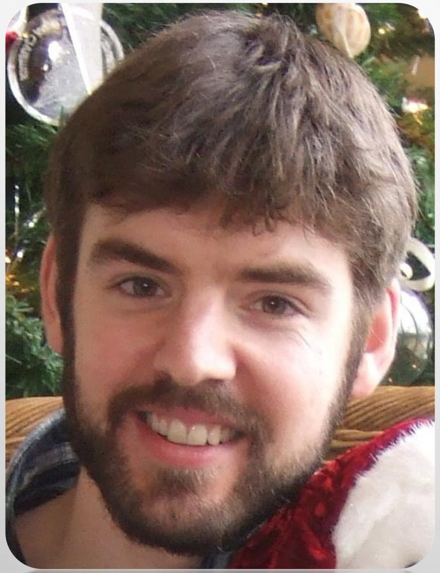


Tom Minka

- ESTENDE O GLICKO PARA O CASO DE TIMES COM TAMANHO DESBALANCEADO. A HABILIDADE DO TIME É A SOMA DAS HABILIDADES INDIVIDUAIS DOS JOGADORES.
- ATUALIZA AS HABILIDADES UTILIZANDO UM ALGORITMO DE PROPAGAÇÃO DE EXPECTATIVA, UTILIZANDO UM GRÁFICO DE FATORES.



PARÂMETROS DO MODELO



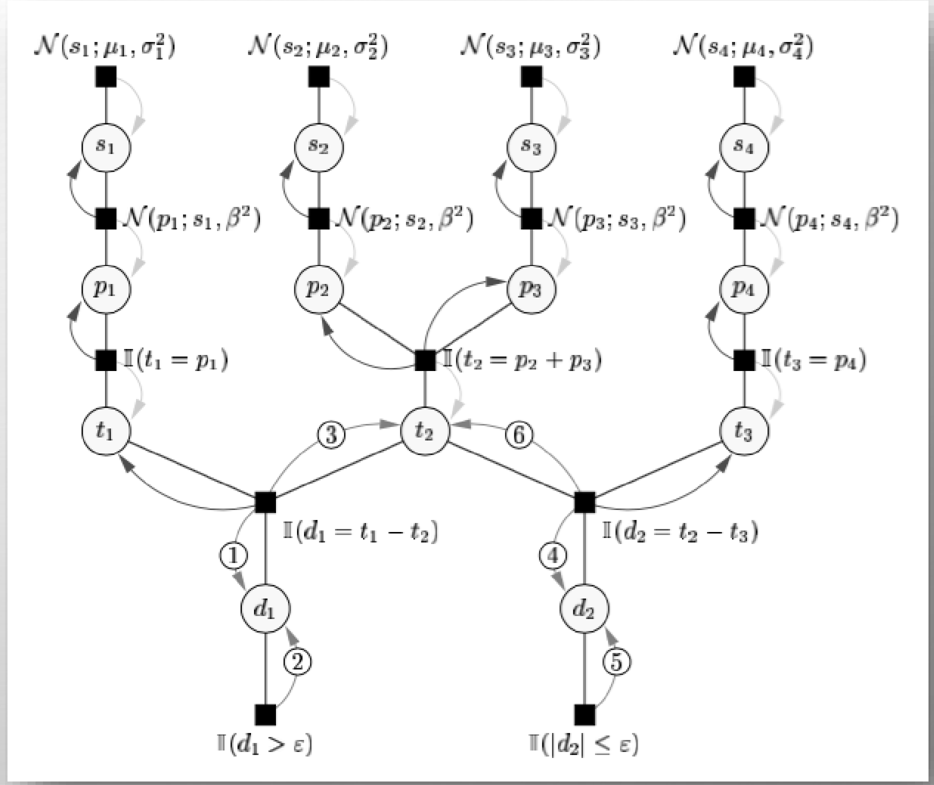
Tom Minka

M= M INICIAL

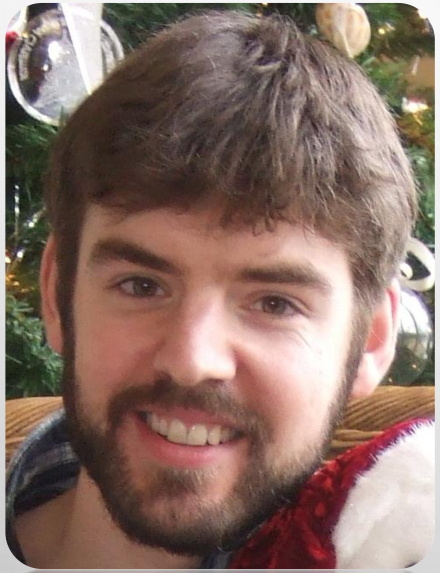
Σ= M/3

BETA= Σ/2

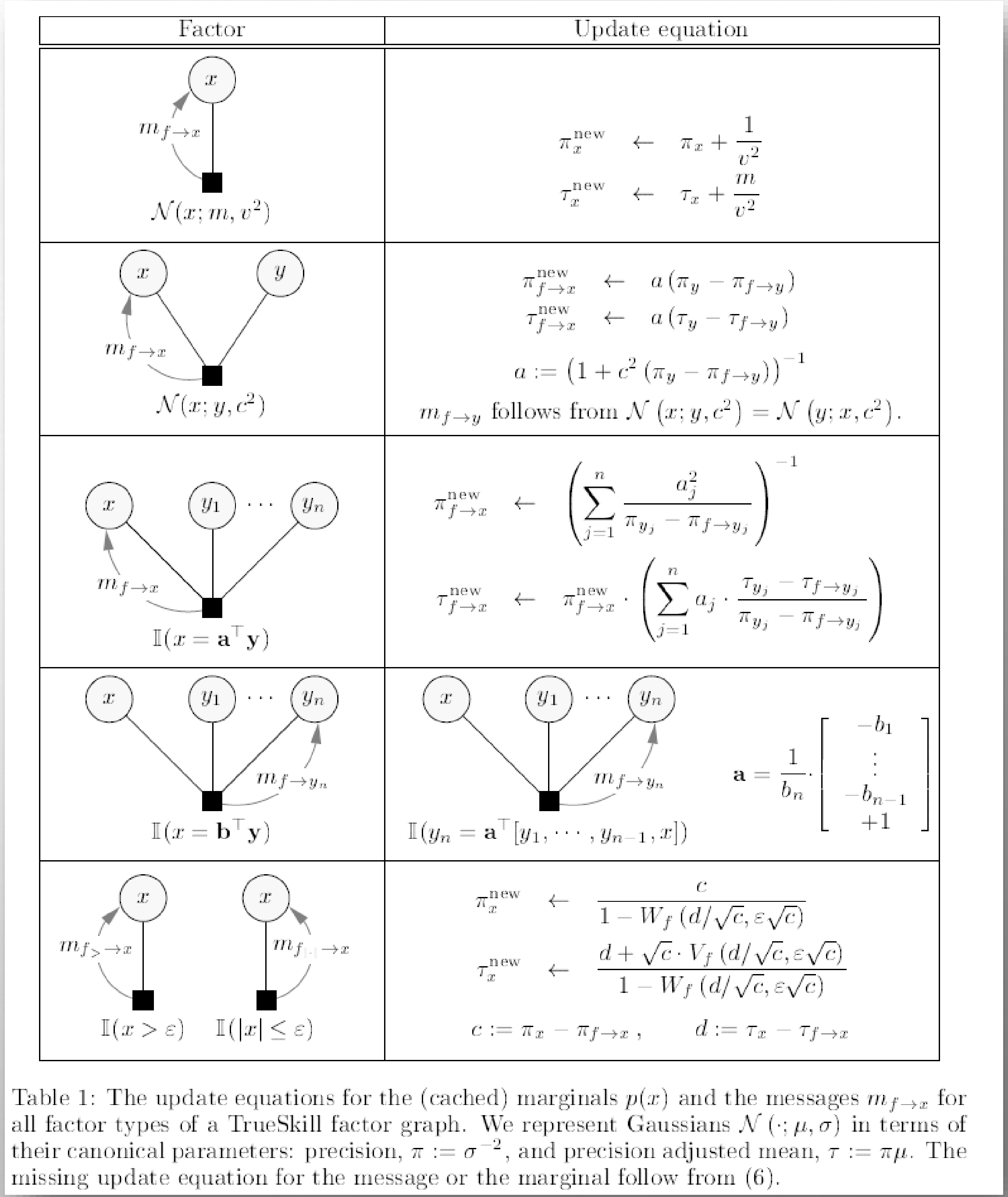
TAU= Σ/100



ATUALIZAÇÃO DOS PARÂMETROS



Tom Minka



Factor	Update equation
$\mathcal{N}(x; m, v^2)$	$\pi_x^{\text{new}} \leftarrow \pi_x + \frac{1}{v^2}$ $\tau_x^{\text{new}} \leftarrow \tau_x + \frac{m}{v^2}$
$\mathcal{N}(x; y, c^2)$	$\pi_{f\to x}^{\text{new}} \leftarrow a\left(\pi_y - \pi_{f\to y}\right)$ $\tau_{f\to x}^{\text{new}} \leftarrow a\left(\tau_y - \tau_{f\to y}\right)$ $a := \left(1 + c^2\left(\pi_y - \pi_{f\to y}\right)\right)^{-1}$ $m_{f\to y}$ follows from $\mathcal{N}\left(x; y, c^2\right) = \mathcal{N}\left(y; x, c^2\right)$.
$\mathbb{I}(x = \mathbf{a}^\top \mathbf{y})$	$\pi_{f\to x}^{\text{new}} \leftarrow \left(\sum_{j=1}^{n} \frac{a_j^2}{\pi_{y_j} - \pi_{f\to y_j}}\right)^{-1}$ $\tau_{f\to x}^{\text{new}} \leftarrow \pi_{f\to x}^{\text{new}} \cdot \left(\sum_{j=1}^{n} a_j \cdot \frac{\tau_{y_j} - \tau_{f\to y_j}}{\pi_{y_j} - \pi_{f\to y_j}}\right)$
$\mathbb{I}(x = \mathbf{b}^\top \mathbf{y})$	$\mathbb{I}(y_n = \mathbf{a}^\top[y_1, \cdots, y_{n-1}, x])$ $\mathbf{a} = \frac{1}{b_n} \cdot \begin{bmatrix} -b_1 \\ \vdots \\ -b_{n-1} \\ +1 \end{bmatrix}$
$\mathbb{I}(x > \varepsilon)$ $\mathbb{I}(\|x\| \leq \varepsilon)$	$\pi_x^{\text{new}} \leftarrow \frac{c}{1 - W_f\left(d/\sqrt{c}, \varepsilon\sqrt{c}\right)}$ $\tau_x^{\text{new}} \leftarrow \frac{d + \sqrt{c} \cdot V_f\left(d/\sqrt{c}, \varepsilon\sqrt{c}\right)}{1 - W_f\left(d/\sqrt{c}, \varepsilon\sqrt{c}\right)}$ $c := \pi_x - \pi_{f\to x}, \quad d := \tau_x - \tau_{f\to x}$

Table 1: The update equations for the (cached) marginals $p(x)$ and the messages $m_{f\to x}$ for all factor types of a TrueSkill factor graph. We represent Gaussians $\mathcal{N}(\cdot; \mu, \sigma)$ in terms of their canonical parameters: precision, $\pi := \sigma^{-2}$, and precision adjusted mean, $\tau := \pi\mu$. The missing update equation for the message or the marginal follow from (6).

PAPER

CALCULANDO O RATING PARA UM DRILL DE FUTEBOL AMERICANO DESBALANCEADO

ESTIMATIVA DO VENCEDOR

$$P(\text{ataque}) = \Phi\left(\frac{\mu_{\text{qb}} + \mu_{\text{wr}} - \mu_{\text{db}}}{\sqrt{3\beta^2 + \sigma_{\text{qb}}^2 + \sigma_{\text{wr}}^2 + \sigma_{\text{db}}^2}}\right)$$

RATING: LIMITE INFERIOR DO INTERVALO DE CONFIANÇA PARA M.

$$h_i = \mu_i - 3\sigma_i$$

$$\hat{h}_i = \frac{h_i}{h^*}$$

RATING AGREGADO – MÉDIA PONDERADA DOS RATINGS, POR REPETIÇÃO.

$$r_i = \frac{\sum_j n_{ij}\hat{h}_{ij}}{\sum_j n_{ij}}$$

DESAFIO: RODAR O BOOTELO NA MÁQUINA III

PRÓXIMA AULA
LEITURA: INTRODUÇÃO A MACHINE LEARNING